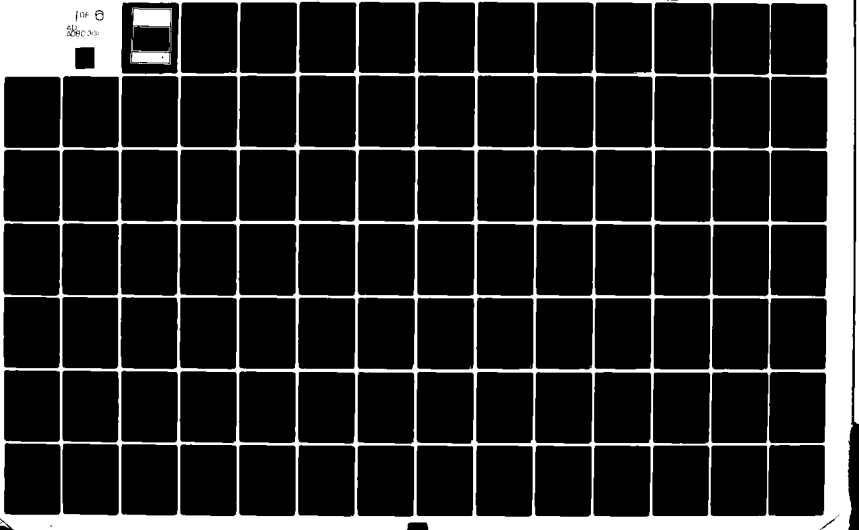
AD-A080 301 ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT--ETC F/G 9/5
AVIONICS RELIABILITY, ITS TECHNIQUES AND RELATED DISCIPLINES.(U)
OCT 79 M C JACOBSEN

UNCLASSIFIED AGARD-CP-261 NL



AGARD-CP-261

AGARD-CP-261

AGARD

ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT

7 RUE ANCELLE 92200 NEUILLY SUR SEINE FRANCE

AD A080301

AGARD CONFERENCE PROCEEDINGS No. 261

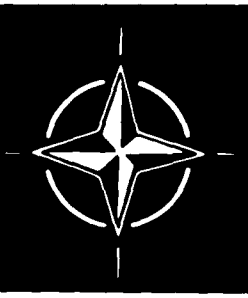
Avionics Reliability, its Techniques and Related Disciplines

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

FEB
A

DDC FILE COPY

NORTH ATLANTIC TREATY ORGANIZATION



DISTRIBUTION AND AVAILABILITY
ON BACK COVER

80 1 30 053

14 AGARD-CP-261

NORTH ATLANTIC TREATY ORGANIZATION

ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT

(ORGANISATION DU TRAITE DE L'ATLANTIQUE NORD)

AGARD Conference Proceedings No.261

AVIONICS RELIABILITY, ITS TECHNIQUES AND RELATED DISCIPLINES

Edited by

C.Jacobsen
AEG-Telefunken
Elisabethenstrasse 3
D-7900 Ulm
Fed. Rep. of Germany

Copies of papers presented and discussions held at a Meeting of the Avionics Panel held in Ankara, Turkey, 9–13 April 1979.

THE MISSION OF AGARD

The mission of AGARD is to bring together the leading personalities of the NATO nations in the fields of science and technology relating to aerospace for the following purposes:

- Exchanging of scientific and technical information;
- Continuously stimulating advances in the aerospace sciences relevant to strengthening the common defence posture;
- Improving the co-operation among member nations in aerospace research and development;
- Providing scientific and technical advice and assistance to the North Atlantic Military Committee in the field of aerospace research and development;
- Rendering scientific and technical assistance, as requested, to other NATO bodies and to member nations in connection with research and development problems in the aerospace field;
- Providing assistance to member nations for the purpose of increasing their scientific and technical potential;
- Recommending effective ways for the member nations to use their research and development capabilities for the common benefit of the NATO community.

The highest authority within AGARD is the National Delegates Board consisting of officially appointed senior representatives from each member nation. The mission of AGARD is carried out through the Panels which are composed of experts appointed by the National Delegates, the Consultant and Exchange Programme and the Aerospace Applications Studies Programme. The results of AGARD work are reported to the member nations and the NATO Authorities through the AGARD series of publications of which this is one.

Participation in AGARD activities is by invitation only and is normally limited to citizens of the NATO nations.

The content of this publication has been reproduced
directly from material supplied by AGARD or the authors

Published October 1979

Copyright © AGARD 1979
All Rights Reserved

ISBN 92-835-0254-X

Printed by Technical Editing and Reproduction Ltd
Harford House, 7–9 Charlotte St, London, W1P 1HD

THEME

The demand for higher avionics reliability and better maintainability is dictated by the requirement of flight safety in terms of tolerable hazard rates, mission reliability in terms of increased mission success probability, equipment availability in terms of reduced mean time to repair, and reduction of avionic support cost by savings in maintenance manpower, spares, test equipment, training and technical data.

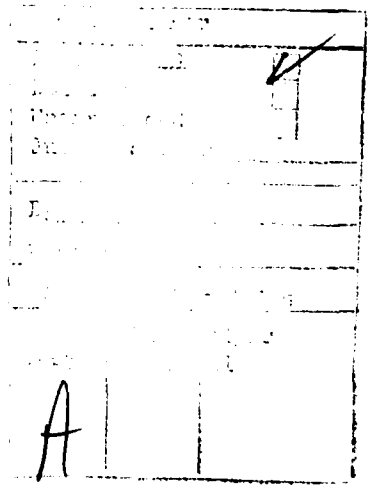
There is a definite need for better adaptation of reliability related considerations to common engineering practices. The improvement of component and avionic system reliability must be accompanied by a commensurate emphasis on the optimal allocation of reliability according to the system complexity, the establishment of cost-effective specifications, qualitative and quantitative reliability analysis and testing to ensure achievement of specifications. The objectives must be clearly formulated in terms of responsibilities, procedures, methods and standards.

Regardless of the level of effort, the early influence of quality assurance including reliability engineering during design and development is essential in achieving avionics operational effectiveness determined by the two system parameters availability and capability.

These objectives require a steady growth in the appreciation and application of reliability and maintainability techniques and methods.

The meeting provided a state of the art review of topics related to reliability and logistics in avionic systems.

Papers presented should be of considerable interest to design and logistic engineers as well as to the operators.



PROGRAM AND MEETING OFFICIALS

PROGRAM CHAIRMAN: Mr Manfred Jacobson
AEG-Telefunken
Elisabethenstrasse 3
D-7900 Ulm
Fed. Rep. of Germany

MEMBERS

Mr W.F.Ball
Associate Head, Avionics Div.
Code 4041
Naval Weapons Center
China Lake, Ca 93555
USA

Mr M.Garnier
Service Technique des Transmissions
de l'Armée de l'Air (STTA)
129, Rue de la Convention
F-75015 Paris
France

Mr J.Naresky
Rome Air Development Center
Griffiss Air Force Base
New York 13441
USA

Mr F.S.Stringer
MOD (PE)
Royal Aircraft Establishment
Farnborough, Hants
UK

Dr Helmut Gross
Messerschmitt-Bölkow-Blohm
UBF, Systemunterstützung FE 07
Postfach 801160
D-8000 München 80
Fed. Rep. of Germany

AVIONICS PANEL

CHAIRMAN: Ir. H.A.T.Timmers
NLR
The Netherlands

DEPUTY CHAIRMAN: Dr M.Vogel
DFVLR, e.v.
Fed. Rep. of Germany

HOST COORDINATOR

Mr Yusuf Kasokoğlu
(Turkish Aircraft Industries)
Atatürk Bulvari 227
Ankara, Turkey

PANEL EXECUTIVE

Lt Col. John B.Catiller
AGARD/NATO

CONTENTS:

	Page
THEME	iii
PROGRAM AND MEETING OFFICIALS	iv
TECHNICAL EVALUATION REPORT	ix
	Reference
SESSION I – GENERAL CONCEPTS	
AN ANALYSIS OF THE EVOLUTION OF THE RELIABILITY AND MAINTAINABILITY DISCIPLINES by M.B.Kline, J.Di Pasquale, T.A.Hamilton and R.L.Masten	1
DIFFICULTIES IN PREDICTING AVIONICS RELIABILITY by J.E.Green	2
RELIABILITY GROWTH MODELS by W.M.Woods	3
A SIMULATION PROGRAM FOR THE DETERMINATION OF SYSTEM RELIABILITY OF COMPLEX AVIONIC SYSTEMS by C.Krause and H.Limbrunner	4
MICRO-ELECTRONIC SYSTEMS RELIABILITY PREDICTION by P.D.T.O'Connor	5
MARKOVIAN AVAILABILITY MODEL FOR A NETWORK OF COMMUNICATING COMPUTERS by T.L.Regulinski	6
ESTIMATION RAPIDE DES TROIS PARAMETRES D'UNE LOI DE WEIBULL par R.Attuly et C.Bertin	7
SESSION II – RELIABILITY/AVAILABILITY REQUIREMENTS, TESTING AND DEMONSTRATION	
RELIABILITY IMPROVEMENT WARRANTY – AN OVERVIEW by H.S.Balaban	8
LES CLAUSES DE FIABILITE DANS LES CONTRATS par J.P.Plantard	9
ETUDE DE LA CROISSANCE DE LA FIABILITE D'UN EQUIPEMENT ELECTRONIQUE SOUMIS A DES CLAUSES DE FIABILITE par J.C.Chabin	10
AMELIORATIONS DE FIABILITE DUES A L'APPLICATION DES CLAUSES DE FIABILITE OPERATIONNELLE par J.Laurensou	11
PAPER 12 – CANCELLED	12
PRACTICAL CONSIDERATIONS AND EXPERIENCE OF RELIABILITY DEMONSTRATIONS* by J.G.Milner	13
PRODUCTION RELIABILITY ASSURANCE (PRA)–TESTING by A.Weihe	14

* Not available at time of printing.

Reference

METHODES UTILISEES POUR CONNAITRE LA FIABILITE D'UN RADAR D'AVION D'ARMES
par J.C.Charlot 15

A FAULT TOLERANT ARCHITECTURE APPROACH TO AVIONICS RELIABILITY IMPROVEMENT
by D.C.Fraser and J.J.Deyst 16

TRENDS IN RELIABILITY MODELING TECHNOLOGY FOR FAULT TOLERANT SYSTEMS
by S.J.Bavuso 17

NON-ELECTRONIC ASPECTS OF AVIONIC SYSTEM RELIABILITY
by C.V.Kenmir, R.G.Hilton and H.H.Dixon 18

SESSION III – RELIABILITY AND MAINTAINABILITY PRACTICES AND EFFECTS IN AVIONICS DESIGN, DEVELOPMENT AND PRODUCTION

IMPACTS OF TECHNOLOGIES SELECTED ON THE RELIABILITY AND OPERATIONAL AVAILABILITY OF EQUIPMENTS. COST CONSIDERATIONS
by J.M.Girard and M.Giraud 19

A NEW APPROACH TO MAINTAINABILITY PREDICTION
by J.J.Naresky 20

RELIABILITY GROWTH THROUGH ENVIRONMENTAL STIMULATION
by L.J.Phaller 21

THE A-7 HEAD-UP DISPLAY RELIABILITY PROGRAMME
by K.W.Boardman 22

MILITARY ADAPTION OF A COMMERCIAL VOR/ILS AIRBORNE RADIO WITH A RELIABILITY IMPROVEMENT WARRANTY
by E.I.Feder and D.L.Niemoller 23

EMULATION APPLIED TO RELIABILITY ANALYSIS OF RECONFIGURABLE, HIGHLY RELIABLE, FAULT-TOLERANT COMPUTING SYSTEMS FOR AVIONICS
by G.E.Migneault 24

RELIABILITY ASSURANCE FOR LARGE SCALE INTEGRATED CIRCUITS
by R.A.McDonald 25

RELIABILITY OF HIGH-BRIGHTNESS CRT'S FOR AIRBORNE DISPLAYS
by J.P.Galves and J.Brun 26

RELIABILITY INVESTIGATIONS ON AN AUTOMATIC TEST SYSTEM
by H-H.Molter 27

APPLICATION OF THE LOGNORMAL DISTRIBUTION TO CORRECTIVE MAINTENANCE DOWNTIMES
by M.B.Kline and R.Almog 28

RELIABILITY MANAGEMENT OF THE AVIONIC SYSTEM OF A MILITARY STRIKE AIRCRAFT
by A.P.White and J.D.Pavier 29

SESSION IV – SOFTWARE RELIABILITY

INTRODUCTION TO SOFTWARE RELIABILITY – A KEY ISSUE OF COMPUTING SYSTEMS RELIABILITY
by G.Heiner 30

SOFTWARE RELIABILITY – UNDERSTANDING AND IMPROVING IT
by L.Mackie 31

Reference

FORMAL METHODS FOR ACHIEVING RELIABLE SOFTWARE
by J.Goldberg 32

QUANTITATIVE ASSESSMENTS OF SOFTWARE RELIABILITY
by J-C.Rault, G.Memmi and S.Pimont 33

AN ANALYSIS OF SOFTWARE RELIABILITY PREDICTION MODELS
by A.N.Sukert 34

ANALYTICAL SOFTWARE VERIFICATION
by W.Ehrenberger and P.Puhr-Westerheide 35

SOFTWARE QUALITY AND ITS ASSURANCE
by P.Weigel 36

SOFTWARE DEVELOPMENT FOR TORNADO – A CASE HISTORY FROM THE RELIABILITY AND MAINTAINABILITY ASPECT
by D.J.Harris 37

SESSION V – AVIONICS LOGISTICS SUPPORT ASPECTS

INTEGRATED LOGISTICS SUPPORT ADDS ANOTHER DIMENSION TO MATRIX MANAGEMENT
by R.M.Drake 38

"MEK" – A NEW PROCEDURE FOR DEVELOPMENT OF MAINTENANCE POLICIES
by K.Lewandowski 39

THE IMPORTANCE OF INTEGRATED LOGISTICS SUPPORT CONSIDERATIONS DURING DESIGN
by R.C.Rassa 40

THE INTEGRATED MANAGEMENT OF RELIABILITY AND MAINTAINABILITY IN PROCUREMENT
by S.E.Shapcott and K.A.P.Brown 41

THE EFFECTS OF DEFECT DATA ANALYSIS ON LIFE CYCLE COST*
by H.J.Moser 42

RELIABILITY AND SUPPORT DATA FOR STATISTICAL EVALUATION
by A.Andrews 43

COMPUTER SIMULATION MODEL OF THE LOGISTIC SUPPORT SYSTEM FOR ELECTRICAL ENGINEERING TEST EQUIPMENT 44
by C.J.P.Haynes

THE RELIABILITY IMPROVEMENT WARRANTY AND ITS APPLICATION TO THE F-16 MULTINATIONAL FIGHTER PROGRAM
by G.T.Harrison, Jr 45

APPENDIX A – LIST OF PARTICIPANTS A

* Not available at time of printing.

TECHNICAL EVALUATION REPORT

by

M.C.Jacobsen
AEG-Telefunken
7900 Ulm-Donau
FRG

EXECUTIVE SUMMARY

In the following, the significant results from the Technical Evaluation are set forth.

Conclusions

The most conspicuous technology gap that we have facing us today is our inability to specify, predict and measure, quantitatively, software reliability and maintainability. Until we can do that, we will never be able to solve the problem of quantitatively specifying, predicting and measuring overall system reliability, i.e. hardware and software combined.

Another area that needs further highlighting in NATO is the application and use of Reliability Improvement Warranties which could become key to improvements in Reliability, Maintainability and Logistic Support Concepts.

Future proposed avionics should be very critically examined for unessential complexity, especially in view of the imbalance between NATO and WP Forces and the vital importance of high availability of provisioned aircraft. Avionics reliability and thus availability levels continue to be limited by the extent of performance demands which result in complexity.

NATO-wide procedures of acceptable methods for reliability analysis/prediction of advanced electronic components and systems have to be developed. MIL-HBK-217 requires major alterations in the micro-electronics area.

Increasing emphasis should be placed on extensive test programs during development, supported by efficient test/analysis/fix activities to obtain reliability growth.

Maintenance policies and logistic support concepts need to be improved for more effective and economical support. The necessity of application of modern logistic engineering technologies, in particular in the European NATO nations, cannot be overemphasized.

Recommendations

- NATO or AGARD to support a Conference/Symposium dedicated to Software Reliability/Maintainability Methods.
- NATO to investigate possibilities for a more general application of the Reliability Improvement Warranty Concept.
- NATO-wide implementation of new Maintenance Policies and Logistic Support Strategies to improve Life Cycle Cost.

TECHNICAL EVALUATION REPORT

1. Introduction

The AVP Symposium on Avionics Reliability, its Techniques and Related Disciplines was held at the P.T.T. Training Center in Ankara, Turkey, from 9 through 13 April 1979. The Program Committee consisted of Mr M.Jacobsen Program Chairman - (FRG), Mr J.Naresky (USA), Dr H.Gross (FRG), Mr F.S.Stringer (UK), Mr W.Ball (USA), Mr R.Voles (UK), and Mr J.Garnier (F) as Session Chairmen.

The objective of the symposium was two-fold, to review the status of hardware and software reliability and maintainability technology for the improvement of technical performance, and to explore concepts to reduce operating and support cost during the avionic equipment life cycle.

PRECEDING PAGE BLANK-

In order to achieve this goal, the symposium was organized in five sessions in which 44 formal papers were presented by the authors, covering

- General concepts
- Reliability/Availability Requirements, Testing and Demonstration
- Reliability and Maintainability Practices and Effects in Avionics Design, Development and Production
- Software Reliability
- Avionics Logistic Support Aspects

This report represents an attempt by the Program Committee to provide an overview of the entire symposium and to draw conclusions and derive recommendations from the presentations and discussions.

2. Symposium Theme

The symposium originated from a proposal of the Danish Air Force. Because of its general interest to the NATO countries, the subject had been adopted by the Avionics Panel of AGARD.

Weapon systems developed, manufactured and purchased for the armed forces have reached a volume and complexity beyond easy comprehension. This results in the massive problem of maintaining equipment in operating condition. Reliability, Maintainability and Logistic Engineering Technologies become key requirements.

The demand for higher avionic reliability and better maintainability results from the necessity of system effectiveness, operational readiness and affordability.

There is an ever increasing need for better adaptation of reliability and maintainability related considerations to common engineering practices in all phases of a program.

The improvement of component and avionic system reliability must be emphasized by the optimal allocation of reliability according to the system complexity, the establishment of cost-effective specifications, qualitative and quantitative reliability analysis and testing to ensure achievement of specifications. The objectives must be clearly formulated in terms of responsibilities, procedures, methods and standards.

Regardless of the level of efforts, the early influence of reliability and maintainability engineering during design and development is essential in achieving avionics operational effectiveness determined by the two system parameters availability and capability. Because of the high degree of digitalization in modern avionic equipment, all software related matters require our full attention.

The continually increasing operation and maintenance cost during the life cycle of avionic systems is of great concern to the NATO Governments. New concepts and methods are necessary to improve the present situation by providing better visibility and control over life cycle cost.

These objectives require a steady growth in the appreciation and application of reliability, maintainability and logistic engineering techniques and methods.

Therefore, it was opportune to provide a state of the art review of topics related to reliability, maintainability, quality assurance and logistics support in avionic systems.

3. Technical Evaluation

Session I, "General Concepts", covered primarily, in 7 papers, besides some historical views of the development of reliability and maintainability disciplines, the subject of reliability prediction and reliability growth.

The paper by Dr Kline (1) on "An Analysis of the Evolution of the Reliability and Maintainability Disciplines" presented a new and interesting classification structure for the R and M disciplines that should be useful in organizing future symposia classifying papers and preparing bibliographies.

Mr Green's paper (2) delivered by Mr Milner on the "Difficulties in Predicting Avionics Reliability" presented some new information that could be used in increasing the accuracy of reliability predictions for avionics equipment.

A follow-on paper describing the applications of these factors and the results in indicating the degree of effectiveness in increasing reliability prediction accuracy would be very worthwhile. RADC is currently pursuing an R & D program to update and revise the avionics environmental factors in MIL-HDBK-217. Mr Green's comments and recommendations will be included as one of the inputs to that program.

Dr Wood's paper (3) on "Reliability Growth Models" presents two interesting reliability growth models. What needs to be done now is to test them on a specific equipment/system to see just how accurate they are. Dr Wood neglected to reference what can be considered to be some important work on reliability growth models done several years ago. They

looked at six of the most commonly used growth models and tested them for accuracy against growth data from a group of equipment/systems. As a result, they were able to develop criteria which enable a designer to determine which growth model best fits his specific equipment design.

Mr Krause's paper (4) on "A Simulation Program for the Determination of System Reliability of Complex Avionics Systems" described a computer program for rapidly simulating system reliability under various equipment/subsystem configurations. It should be useful in determining optimum system configurations for a given reliability. If one were able to add maintainability and cost data, its usefulness would be greatly enhanced by enabling one to determine the optimum system configuration at minimum life cycle costs. It would also be interesting to check the model against actual system data to determine its accuracy.

Mr O'Connor's paper (5) on "Microelectronic Systems Reliability Prediction" suggested revision of some of the microcircuit models in MIL-HDBK-217. These models are currently undergoing revision at RADC, the US organization responsible for updating MIL-HDBK-217, and Mr O'Connor's recommendations will be a valuable input. To support his revised models, more actual data will still be required.

Dr Regulinski's paper (6) on "Markovian Availability Model for a Network of Communicating Computers" described how one might use Markov process to analyze a computer network in terms of determining those elements that are contributing most to system unavailability. A follow-on example of how the analysis will work in real life is desirable.

Mr Attuly's paper (7) on "Rapid Estimation of the Parameters of the Weibull Distribution" was a worthwhile paper, but highly theoretical. The Weibull model which generalizes that of Poisson is largely utilised in reliability theory and, more generally, for studying the random failure of a system due to a measurable phenomenon. A new method was proposed to resolve the delicate estimation problem.

In Session II on "Reliability/Availability Requirements, Testing and Demonstratic a total of 8 papers were presented.

The three papers by Mr H.S.Balaban (8), Mr J.P.Plantard (9) and Mr J.C.Chabin (10) dealt with the introduction of reliability clauses into contracts, one of the most promising concepts recently established being the definition of reliability improvement warranties (RIW). The discussion on this complex showed the necessity of well-substantiated reliability experience on which to base potential contractual reliability clauses for any new system generation. Once a reliability requirement has been contractually specified, particularly in the case of an RIW commitment, this may lead to a more rapid reliability growth due to an additional motivation of the supplier, as indicated by one paper. Yet another paper pointed out that this positive effect of RIW might be counterbalanced by the increased pressure on the supplier to achieve a positive outcome of reliability demonstrations, thus causing more problems with failure relevancy and possibly increasing the gap between demonstrated and operational reliability. This obvious lack of concurrence between test results and field experience gave rise to some additional discussion on failure classification rules during test and on the significance of test results in view of increased system complexity.

The paper by Mr Milner (13) covered practical considerations of and experience in reliability demonstration. He stressed the importance of investigating component failure mechanisms in an area of rapidly advancing component technology, and the great potential for improving reliability and reducing maintenance cost by early detection and rectification of design shortcomings.

The paper by Mr Weihe (14) presented an interesting procedure for production reliability assurance testing (PRAT), which could well develop into an alternative .o RIW in terms of the risks involved.

Mr J.C.Charlot's paper (15) described the problems of field data collection and evaluation for corrective action.

The two papers by Mr D.C.Fraser/Mr J.J.Deyst (16) and Mr S.J.Bavuso (17) covered fault tolerant system approaches leading to reliability improvement in advanced avionic systems.

The last paper of this session presented by Mr R.G.Hilton (18) was devoted to reliability considerations regarding the elements and the design principles of future actuation systems. This was, in turn, an excellent illustration for the requirements imposed upon the avionics due to the interconnections within the overall system.

Session III was devoted to the subject of Reliability Practices and Effects in Avionics Design, Development and Production. The aim of the session was to examine the methods being adopted today and the results of their application. This aim was not achieved completely though the session developed some very useful ideas. Seven papers were presented during the session.

The paper by Mr J.M.Girard/Mr M.Giraud (19) dealt with the initial selection of technologies during start-up of the design phase. The purpose of this paper was to present a methodology permitting to establish the existing relations between the technology selection and the operating parameters reliability, maintainability and cost effectiveness.

It is clear from the discussion of Mr J.J.Naresky's paper (20) upon a New Approach to Maintainability Prediction, that the real benefit of techniques will very much depend upon the way in which the data base is compiled and the test times used as a guide to the assessment of failure rates. A sincere attempt has been made by the author to ensure that as much relevant data as possible was included in the data base presented. However, the advent of new technology will require a continual update of this data base. Mr Naresky's paper generated the stimulating discussion which highlighted some other valuable issues. Visual scanning of components had been described and it was interesting to learn that new electronic tests are being studied as a possible alternative. The selection of key components and their characterizing after some 18 - 24 months on the market illustrated the need for a dedicated approach to reliability if such methods are to prove worthwhile. The stated acceptance of such qualifications by the US Forces is encouraging evidence of their success. It was also noted that software is qualified in detail as well as hardware.

An intriguing revelation by Mr L.J.Phaller in the presentation of Paper 21 was that the application of environmental testing during the design phase has identified the potential trouble spots to a marked degree. This is an exposure of the obvious to some extent. However, it should be recognized that the capital investment in suitable facilities can be considerable and the centralized facility will suffer inevitably from over booking. In response to a question, it was confirmed by the author that in-service modifications are handled by the cycling of units back through the factory for change. This can of course be a costly arrangement.

The paper by Mr Boardman (22) describing the reliability program of the Head-Up Display for the A7 aircraft revealed some of the lessons learned after initial problems in trying to meet a very stringent MTBF specification. Data was acquired from three main sources, namely (a) burn-in test (b) during an abortive reliability demonstration (c) early service usage.

The data pointed to a limited number of suspect circuit locations. Attention to these made a dramatic improvement to reliability. The closed loop approach has much to commend it. Again the question of a bottleneck created by test facilities and problems generated in the provision of testing and reporting staff were acknowledged. It was noteworthy that the feedback was obtained from the firm's own repair facility. Though no standards were available for high reliability ICs, a failure rate of 0.2 parts per million was measured. It was clear from the discussion that experience and accepted rules were used rather than the application of life-cycle costs curves for repair. The application of MIL-STD-781C was expected to improve reliability still further.

The interest generated by this paper emphasized the current dependence upon feed back rather than the direct application of theoretical methods presented in several papers at the meeting.

The paper by Mr G.E.Migneault (24) describing an emulation scheme based on a Markovian model was subjected to detailed questioning by several delegates. The application to the reliability analysis of reconfigurable highly reliable fault-tolerant computing systems for avionics is most topical since such systems are forming a significant part of new combat aircraft installations. It is evident that this aspect of reliability generated considerable interest within the audience and will require further research.

The paper by Mr White/Mr Pavier (25) provided the background to the Tornado experience and the management of reliability in complex strike aircraft systems generally. The high cost of testing was discussed and the Tornado was cited as an example where the rigorous application of MIL-STD-781C might be questioned on cost grounds.

Three papers by Mr J.P.Galves et al. (26), Dr H.H.Molter (27) and Dr M.B.Kline (28), respectively, dwelt essentially upon different theoretical models and their application to various selected systems as an illustration of their viability. Specialists within the audience questioned the way in which such models would be used and there was evidence of a desire to identify the most appropriate model to suit particular needs.

For Session IV related to the important topic of Software Reliability, a total of 8 papers had been presented, with an informal "Round Table Discussion" at the end of the session. A most interesting survey of approaches to achieve reliable computing systems was provided. Constructive and analytical methods were presented for software design, verification, reliability predictions and quality assurance.

The paper by Mr G.Heiner (29), "An Introduction to Software Reliability", tuned the audience to the subject by emphasizing the existing problems and possible approaches to software reliability.

Mr L.Mackie's paper (30) provided a number of vital recommendations regarding "The Hardware/Software People System, Software Requirement Specifications, Software Testing, Detection and Correction of Errors and Time necessary for Software Development", which could provide a basis for producing more reliable software at reduced cost.

The paper by Mr J.Goldberg (32) stressed the fact that future specifications for reliable avionic systems will require a level of confidence in program correctness that cannot be achieved by present programming methods. Newly developed formal methods for achieving reliable software were presented.

The paper by Mr J.C.Rault, Mr G.Memmi and Mr S.Pimont (33) provided a categorization and a description of approaches to quantitatively assessing software reliability. Methods of practical interest and of data that might help to

understand how often, when, where and why programmers introduce software errors and how these errors may be detected and corrected were outlined.

The paper by Mr A.N.Sukart (34) reported on a study undertaken by RADC, US to validate several mathematical models for predicting the reliability and error content of a software package against error data extracted from the formalised testing of four large software development projects.

The paper by Dr Ehrenberger/Mr Puhr-Westerheide (35) dealt with analytical software verification procedures to ensure correctness of software.

The paper by Mr P.Weigel (36) provided a good overview of the major quality characteristics of software and their assessment standards. The causes of failure and the development of software were discussed, and the technical means and measures for eliminating faults and impacting the software quality have been outlined. In addition to the technical measures, the organizational means of software quality assurance was discussed.

The last paper of this session, presented by Mr D.J.Harris (37), covered the methods and procedures adopted in the tri-national TORNADO Development Program. The concept of software documentation, testing and control associated with quality assurance standards in this project was described. Vital recommendations drawn from this experience have been made by the author with respect to software management, software definition, writing, testing and delivery.

Session V was dedicated to Avionics Logistic Support Aspects. Eight (8) papers had been prepared for this session, but only 6 were presented due to absence of two authors.

The paper by Mr R.M.Drake (38) covers the importance of ILS management techniques. The various elements of logistics services and their integration with matrix management were explained. Particular emphasis was placed on Integrated Quantitative Planning, ILS-Products and, last but not least, ILS Innovations.

Mr K.Lewandowski's paper (39) provided a survey of a new procedure for the development of maintenance policies applied in the TORNADO program and future GAF weapon systems. The procedure presented is based on a detailed collection and evaluation of the maintenance expenditure expected for the new weapon system. These data, after validation, present the basis for the defintiion of all logistic elements.

The paper by Mr R.C.Rassa (40) was devoted to the importance of Integrated Logistic Support considerations during the design phase of avionic equipment. The roles of the key personnel involved in the design cycle have been examined. The topics stressed were "design for testability" and the very early involvement of maintenance/logistics engineers during equipment design.

The paper by Mr S.E.Shapcott and Mr K.A.P.Brown (41) provided an overview on the Integrated Management of reliability and maintainability in UK MOD Procurement. Progess in the UK in laying sound foundations for developing and manufacturing equipment for the Services' needs, with the required performance, reliability and quality resulting in specific defense standards has been described.

The papers by Mr H.J.Moser (42) and Gp Capt. A.Andrews (43) described the collection and evaluation of statistical maintenance data of equipment in the Services. Problems and limitations in interpreting field data and relating it to experience in the design stages of new equipment were discussed. The problem of modifications to in-service systems and the consequent effect upon reliability was raised. Undesired side-effects might introduce further problems not demonstrated during test at the manufacturing stage. It was explained that test experience could be gathered during a demonstration, but if corrective action appeared to be necessary towards the end of the demonstration, then extra tests would be required.

The paper by Squadron Leader C.J.P.Haynes, UK (44) described a Computer Simulation Model of the Logistic Support for Electrical Engineering Test Equipment still under further development.

Mr G.Harrison's paper (45), one of the highlights in this symposium, was devoted to the application of reliability improvement warranty (RIW) in the F16 Multinational Fighter Program. A unique central spares supply management system used by all five partner nations was outlined which leads to significant cost savings. The successful application of the principles of RIW in this multinational program could mark the beginning of new procurement techniques in NATO with significant impact to obtain suitable reliability control and to reduce life cycle cost.

4. Conclusions and Recommendations

Session I

(1) Most avionic reliability predictions have been inaccurate, and factors other than listed part failure rates have a greater influence on observed equipment failure rates in service.

Emphasis on predictions may well begin to move away from the traditional numbers count procedure in the design phase, towards predicting from reliability growth modelling of operating experience during the development phase.

(2) For a given equipment, failure rate is a function of the type of aircraft and installation. There is little doubt that failures are not exponentially distributed in time. Failure rates are dependent on elapsed time following switch on, and decrease as the sortie progresses.

(3) The potential reliability of LSI is undoubtedly remarkably high, but failure rate prediction is as yet a doubtful procedure. In the future, if avionics predictions are to retain any credibility, the traditional approach will have to be modified.

(4) It is generally accepted that MIL-HBK-217 requires major alterations in the micro-electronics area. The RADC, US is working on the problem. As an objective, a NATO-wide agreement (i.e. among the major industrial partners) on an acceptable method for reliability analysis/prediction of advanced electronic systems is recommended.

The session fulfilled its purpose although perhaps some of the papers had to be "force fit" into a General Concept Session. The papers were, on the whole, of good quality, informative, and explored areas that were relevant to the reliability and maintainability disciplines.

Session II

(1) The RIW concept is primarily applicable to evolutionary-type equipment, whereas for completely new technology the establishment of a feasible reliability value which can be subject to a warranty clause remains problematic and must be approached through a somewhat questionable assessment of the reliability growth potential.

The future of RIW is promising if continued efforts are made to ensure that the concept is properly applied and implemented. It is also necessary for the military services to continue to support research in RIW and allied areas as technology, resources, and military demands change. The RIW concept that embodies the suitable form of contractor incentive for reliability and maintainability achievement will also be flexible enough to encompass most foreseeable changes provided the appropriate effort is made.

NATO-wide efforts for the implementation of the RIW concept as contractual commitments by avionic equipment suppliers are strongly recommended.

(2) The greatest potential for improving reliability and reducing life cycle cost lies in the early detection and rectification of design shortcomings. It is emphasized that more than two thirds of the potential life cycle cost have been determined by the time a system completes the concept formulation stage.

(3) For the purpose of demonstrating the achievement of contractual reliability requirements, practical methods of measuring reliability gain major attention. The general impression in this area was that a set of objective criteria for failure classification would lead to much more realistic results of reliability measurements, thus increasing confidence in the methods applied and overcoming problems with risk assessment.

(4) Testing digital systems which perform flight critical functions is not a feasible method for estimating system reliability. Analytic modeling of system reliability in conjunction with simulative techniques for coverage measurement appers to be the only alternative on the horizon. Accurate reliability estimates which account for such factors as latent faults, intermitted/transient faults and software errors require sophisticated fault tolerant/fault avoidance techniques which are to be developed in the near future.

(5) Avionics reliability requirements are strongly influenced by existing interfaces with non-electronic systems; complete consideration of these interfaces is, therefore, of utmost importance in all reliability studies.

Session III

(1) There is a current need for feedback to the manufacturer from the user, particularly during early months of service of new equipment.

(2) Tests can identify many faults before service, though tests need to be extended to accommodate the reliability of modifications required as a result of earlier tests.

(3) Adequate testing can be very expensive in equipment and man power. There is some evidence that high cost has limited the rigorous application of MIL-STD-781C.

(4) There are several theoretical models which can predict reliability, but most require practical quantitative data input which can only be obtained from practical experience of equipment. Further examination of the most promising models would be valuable.

(5) It appears that military forces are willing to accept the cost of characterisation of components in their desire to obtain higher reliability.

(6) Theoretical models to assess the likely repair downtime of equipment are now available. It remains for them to be considered by users and manufacturers.

(7) The rate of reliability growth depends not only on the degree of management commitment to the program, but also on the unit complexity and state of the art of the unit's design.

Reliability growth can be experienced in two basic forms, (a) growth in the design (permanent growth), (b) growth in the quality control procedures (short-term growth).

Reliability growth through environmental simulation with follow-on corrective action is a viable means to achieve improved field reliability.

(8) In order to arrive at a useful understanding of the causes of unreliability in order to reduce it, we will have to consider these systems very closely and be prepared to modify them if improved reliability is a genuine requirement. This improvement must recognize some of the long list of causes of unreliability. The most significant appear to be:

Firstly immature engineering design which is considered to be closely related to what, in many instances, appears to be an inadequate amount of development testing. This is compounded by the all too prevalent incorporation of unsuitable and defective components and materials into equipment.

There is a scope also for better manufacturing planning and control, vis-à-vis reliability, to ensure that the reliability levels achieved in design are maintained throughout production life.

Lastly it is recognized that failures induced by operators mistakes during manufacture and improper use in the field are in fact parts of the broader field of human induced failure.

Such ergonomic considerations of reliability are, as yet, in their infancy, but evidence is rapidly accumulating to indicate that design discipline in this area can be very rewarding indeed.

The Session III presentations on balance achieved the desired objectives. The papers tended to fall into two distinct categories, either descriptions of direct experience and the methods used to improve reliability on the one hand, and those which suggested more theoretical prediction and some evidence of likely accuracy of prediction on the other. It is evident that cost will play an important part in all of these considerations.

In conclusion, it is recommended that all techniques offered should have cost in mind. Methods that are too expensive or too restrictive in nature, for whatever cause, are unlikely to receive general acceptance.

Data feedback is an essential feature of reliability testing and reporting methods need continual attention. It is important that the consequent testing of modifications should be included.

Guidance is needed by manufacturers and procurement authorities in the use of the most appropriate models which may be applied to the accurate prediction of reliability probabilities for given sets of circumstances.

Session IV

(1) Future requirements of computing systems, in particular where flight safety is involved, result in a growing need for very high software reliability.

(2) Confusion in terminology and lack of consistency in the definition of the terms used in connection with software reliability require near term clarification, It has been recommended that AGARD should take the initiative of bringing the appropriate experts together, i.e. forming a Working Group.

(3) Significant problems are that most system specifications are ambiguous, incomplete, and inconsistent, due to the lack of generally agreed standards for design and documentation as well as adequate validation methods.

(4) The lack of knowledge and experience in the field of software contributes to the fact that an integral treatment of system reliability including hardware, software and human factors has not been achieved until now. Reference is made to the identified key features in paper 37 which contribute to improved software reliability and maintainability.

(5) Developing and maintaining software has become the dominant factor in digital systems. To improve in software reliability and cost, it was recommended to

adopt a *formal* system user oriented jargon-free language for the Functional Requirements aspect of Software Requirements Specifications to *achieve* definition and clarity.

eliminate unnecessary *error-prone processes*.

design for testing and development tools.

minimize structural *complexity*. *Preserve* the *problem structure*.

use more hardware as necessary for all the above.

allow *time* to do the *job properly*.

(6) Innovations in analytical software verification methods are necessary to cope with future requirements for reliability avionic systems. To achieve the required level of correctness and confidence, digital computers with fault tolerant organization seem to be the answer.

(7) A full AGARD symposium or specialist meeting on software integrity and reliability, possibly between the GCP and the AVP, is recommended to be held in the very near future.

The objective of this session to define significant problem areas and to elaborate possible methods and techniques to produce more reliable software at lower cost was reached. Follow-on activities, however, are urgently required.

To arrive at the above conclusions and recommendations, the papers presented and the round table discussions held have been regarded as an entity.

Session V

(1) Increased emphasis has to be placed on the logistic support aspects in avionic equipment due to the importance of operational readiness and ever increasing maintenance cost against the background of limitation of resources available to NATO.

(2) During Prime System Design Reliability Analysis, Maintainability Analysis, Design to Cost Analysis and System Engineering should be an integral part with Logistic Support Engineering, Maintenance and Support Analysis. A continued iterative feedback between the disciplines involved will lead to the optimization of design trade-offs with emphasis on improved maintenance concepts. Unless logistics is an inherent part of the design and development process and maintainability is designed into the product, we will continue to be faced with major logistic problems.

(3) In-service reliability data provide a powerful and flexible information source for statistical evaluation and reliability analysis. In fact, it provides the most valuable and realistic basis for corrective actions.

(4) Growing cost of advanced avionic systems in design, production and support requires application of the concept of life cycle costing. A NATO-wide implementation of the LCC approach in procurement of avionic equipment is recommended. This requires common efforts in the development of new methodologies for better visibility and control of LCC as well as new specifications, contract provisions and source selection procedures in procurement.

The goal of this session to highlight the importance of Logistic Support considerations for better equipment availability and stressing means to lower support cost was reached, although the subject is too complex to cover all aspects in the short time available.

AN ANALYSIS OF THE EVOLUTION OF THE RELIABILITY AND MAINTAINABILITY DISCIPLINES

M.B. Kline
Naval Postgraduate School
Monterey, California

J. Di Pasquale, T.A. Hamilton
Naval Weapons Center
China Lake, California

CDR. R.L. Masten
U.S. Navy

SUMMARY

This paper presents the results of a study of the development of the reliability and maintainability (R&M) disciplines in the years since World War II. The study was conducted primarily through an examination of the published (open) literature. The exponential rate of growth shown during this period is an indication of the dynamic nature and importance of these disciplines to system development, design, and operation.

Family trees of each discipline have been developed to indicate the growth and branching of the relevant subject matter. The direction and rate of growth of these disciplines in each of the decades of interest are analyzed along with projections of current and future trends. Applications of R&M in both the private and public sectors, including defense, space, energy, transportation, industrial and consumer items, are examined.

1. INTRODUCTION

Prior to and during World War II, the designer's primary goal was to satisfy a desired set of performance requirements. The rapid advances in technology which have occurred since then have been applied to military, space, and consumer needs. With this increase in technology and performance capability has come a corresponding increase in system complexity and the emergence of reliability and maintainability as engineering disciplines of equal importance as system performance in terms of system effectiveness and cost.

The effectiveness of a system is concerned with (1) the ability of the system to begin performing its mission when called upon (often called operational readiness or availability), (2) the ability of the system to perform satisfactorily for the duration of its mission (often called mission reliability or dependability), and (3) the actual performance of the system in terms of its performance parameters in the operational environment (often called capability).

The rapid growth of reliability and maintainability (R&M) as design parameters in the past 25 years is well evident by the numbers of professional symposia, books, and publications which have appeared. This very conference is an example of the growing international attention being given to reliability and maintainability.

This paper describes some of the results of a research investigation which took place in 1976 and early 1977 to trace and analyze the evolution of these engineering disciplines as evidenced by the published literature. More specific details of the research are given in reference 1. Growth curves were developed to determine where the emphasis in reliability and maintainability has been during the past three decades, what the emphasis is today, and to project probable future discipline emphasis. Secondary objectives were the development of a substantial data base of R&M documents for easy retrieval.

2. A BRIEF HISTORY OF THE EVOLUTION OF R&M

Reliability principles were used as early as 1916 by the Western Electric Company, the manufacturing unit of the Bell (Telephone) System. With production running at a high rate and rising rapidly, studies were initiated to discover means to produce trouble-free telephone equipment for public use. The Western Electric Company was among the first to realize that statistical sampling methods could be applied to industrial processes. The Bell System understood that durability must be a main goal, and that service history and optimization of the design for maximum quality were important factors.

During World War II, there was an urgent need to develop methods for the manufacture of uniform high-quality products at increased rates of production. Large quantity production in the rapidly growing electronic industries led to the development of standards for the application of statistical methods to the quality control of materials and manufactured products. Radar and other military developments of World War II introduced the need for specific consideration of reliability.

In 1946, the commercial airlines sponsored field studies of vacuum tube and electronic equipment failures to improve the reliability of aircraft communication and navigation

equipment. This led to an investigation of electron tube reliability in military applications. The effects of application, environment, and operating and maintenance conditions were shown to be so closely related that the study was redirected to emphasize system reliability as affected by electron tubes. Systems under study were expanded to include radio communication systems, radar systems, and bombing and navigation systems.

The recognition that many parts other than vacuum tubes were causes of problems led to the formation of the Ad Hoc Group on Reliability of Electronics Equipment (AGREE) in 1952. This group was instrumental in initiating an increasing number of studies in order to add to the knowledge of equipment failures.

In 1950, a study of the reliability of U.S. Navy shipboard electronic components and equipments resulted in the establishment of relative failure rates for component parts and in the development of an improved failure reporting system. In the late 1950's many other studies were carried out under military sponsorship directed toward the measurement of equipment reliability and the development of methods of predicting electronic equipment reliability while still in the design stage.

Mass production introduced the need for standardized tests to be used in the factory. Reliability standards had to be developed. Standardization of parts and circuits were stressed. Parts improvement programs were initiated as the quality of parts still left much to be desired. The critical importance of reliability was recognized both by the U.S. Department of Defense and by industry. This importance was emphasized with issuance of military standards, specifications, and handbooks for reliability programs, reliability prediction and reliability demonstration testing.

Prior to 1954, maintainability was not a defined discipline. Some manufacturers were starting to incorporate specific maintainability features into the design of their products. An example of this was the design and production of standardized rifles for the U.S. Army during World War II.

U.S. Government publications concerning maintainability did not exist during this period. Maintainability requirements were covered through specialized contractual exhibits and/or amendments to contracts. By 1959, formalized program specifications started to evolve. The 1960-1970 decade witnessed a rapid growth of the maintainability discipline. A realization that the best design from the reliability standpoint may be unaffordable created a new challenge for the design engineer. The need for maintainability was predicated on the basis that no system can be made totally reliable. Awareness of the need to consider reliability and maintainability together as design parameters early in system development evolved.

The development of new technologies, such as integrated circuits, and their application during this period increased the complexity and sophistication of hardware. Computerized failure history data banks for use in reliability predictions were developed. Reliability testing using statistically designed tests was recognized as a valid test method. The systems effectiveness concept was extensively explored. Standards, handbooks, and design guides were developed for maintainability program management, prediction, demonstration, and design.

Cost factors became dominant from the mid-1960's on and today emphasis is given to life cycle cost and design-to-cost as principal design trade-off parameters. The extended time in service of some older systems increased reliability and maintainability problems which in turn increased the costs for operation and maintenance.

Reliability methods and procedures which had been developed earlier were refined and their use was extended into space, consumer, energy, and nuclear power areas. The airlines pioneered the MTBF guarantee which requires that the equipment supplier guarantee a stated mean-time-between-failure in the operating environment. If the guarantee is not met, the supplier must provide corrective action and additional spares.

In today's atmosphere of increased cost consciousness, there is continued emphasis on reliability and added emphasis on maintainability. The defense community emphasizes the total cost of ownership, the largest component of which is operating and support costs. This is leading to a search for innovative approaches to improve equipment reliability and maintainability. Recent experiences with power outages and concerns for safety in nuclear power plants have dramatically increased the attention being given to reliability and maintainability in this industry.

3. RESEARCH METHODOLOGY

Literature published on reliability, maintainability, and quality control was researched to identify the emergence of these disciplines and to quantify their growth patterns. Quality control was included because some of the early concepts applied to the reliability discipline were an outgrowth from the quality control field. The Cumulative Book Index (reference 2) was used to identify books in these fields because it is a reasonably complete and comprehensive list of works published in the English language.

Figure 1 shows the evolution of quality control, reliability, and maintainability as evidenced by books published in these disciplines. The indicator for growth used is the number of pages published in three-year intervals beginning in 1944 and extending through 1975. It should be noted that any conclusions reached using these curves must be treated

with care since books have an inherent time lag of up to three years from original manuscript preparation. These curves give a qualitative feeling for the discipline growths. As Figure 1 indicates, publication of books on quality control peaked about 1951 or 1952 and declined thereafter. Books on reliability began to appear in the early 1950's and books on maintainability in the early 1960's. Research into the periodical literature provided a more detailed breakout of some of the events which have affected the development of these disciplines.

It is apparent that reliability and maintainability have evolved from rudimentary concepts into full scale scientific disciplines over the past thirty years. For analysis purposes, it was necessary to devise a means to measure their evolution and to identify data sources applicable to the measurement technique. It was desirable to employ a measure that would give as accurate a representation as possible across the full spectrum of both disciplines. Several measures of discipline development were considered. The measure ultimately chosen was the number of articles published in the open literature.

3.1. Discipline Taxonomies

Once the method for measuring discipline development had been selected, it was necessary to address questions concerning the breakdown of each discipline into subelements. The disciplines could, for example, be divided into functional and application-oriented subelements or they could be subdivided by the physical and mathematical sciences forming the core of underlying theory. Each of the subdivision alternatives had merit, and it was decided to incorporate them into a hierarchical classification system. As a result, a taxonomy was developed by which articles could be classified and the data stored for future analysis as well as providing a mechanism for article retrieval for research purposes. The taxonomy provided an excellent structure for analyzing the development of these disciplines. It was particularly useful in defining the main branches and emerging subbranches of the scientific core of the disciplines.

Keywords were initially selected by researching a representative sample of the available literature and, through successive refinement, were finally arranged into a classification structure. The structure was then presented to several persons with extensive experience in reliability and maintainability for comments. This process was iterated several times and resulted in the taxonomies presented in Figure 2 (Reliability) and Figure 3 (Maintainability). The top three levels, functions, applications (general), and applications (specific) were established to enable discipline growth to be measured along these dimensions as well as within the branches or elements of the discipline. In terms of growth measurement, it appeared worthwhile to provide a means for separating government-oriented applications such as defense and space systems from industrial, consumer, and other non-government applications.

As indicated in Figures 2 and 3, the keyword structures for reliability and maintainability have a great deal of similarity, particularly at the upper levels. This is not accidental nor is it inappropriate if one considers that the disciplines are heavily interdependent in terms of both application and functional dimensions. Emphasis--and hence growth stimulus--has varied greatly within the functional categories over the years. For example, until recently space and power generation applications have been primarily concerned with reliability whereas military applications have emphasized both reliability and maintainability in a more balanced sense.

In attempting to visualize the development of these disciplines, it is helpful to compare their development to that of a tree, as pictorially illustrated in Figure 4. One might represent the roots of the tree (which supply nutrients for growth) as the basic sciences of mathematics, chemistry, physics. The trunk represents the core of the disciplines, such as theory of failure and theory of repair. The branches represent growth of elements and subelements. As the tree grows, certain branches exhibit growth rates and then tend to stabilize or even stagnate and die. These same characteristics are exhibited in both the reliability and maintainability discipline developments. Thus, the analogy is helpful in visualizing the growth patterns.

3.2. Data Sources

In attempting to quantify the growth of the various branches within the R&M disciplines, it was decided to emphasize articles published in the open literature as opposed to books because articles were much narrower in subject matter and more specific in scope. Books tend to be tutorial in nature and, in general, are not representative of the taxonomy. It was necessary to examine and classify a large number of articles in order to develop a data base which was large enough to be significant and which, in the aggregate, would have minimum bias. Approximately 5000 articles were utilized in the study. There is, of course, a much larger data base available, and the challenge was to select those sources which would be most suitable for the purposes. It was desirable to include articles from sources which were somewhat continuous in nature and which, in the aggregate, covered the broad spectrum of both disciplines. It was important to ensure that a concentration of articles covering a narrow spectrum of the disciplines were not incorporated in any given time interval. Otherwise, conclusions based on the sample results would not be representative of the total population. In this regard, reliability and maintainability symposia proceedings and other publications devoted primarily to R&M were selected as the primary sources for articles since they are continuous in nature and tend to cover a broad spectrum of topics. Several different symposia were chosen which emphasized different branches of the disciplines, and together they appear to provide a well-

rounded coverage of each discipline. Technical reports and other sources not generally available to the public were not considered.

3.3. Taxonomy Validation

Throughout the development of the taxonomy, the classification scheme was subjected to a series of checks, primarily relating to the naturalness of the keyword groupings and their relation to the scientific basis for the underlying theory. The initial groupings were taken from the American Society of Quality Control (ASQC) classification system and were subsequently refined and modified as more insight was gained through research of the literature. As the taxonomies evolved, many ambiguities and inconsistencies surfaced which required resolution. This was accomplished by conversations with experienced practitioners in the fields of reliability and maintainability and by a trial run consisting of classifying a large number of articles to expose the broad spectrum of subbranches within the disciplines. As experience and depth of knowledge about the disciplines increased through exposure to the literature, it became progressively easier to resolve the ambiguities.

In a less formal, yet equally meaningful sense, the taxonomy was validated when articles began to be classified with relative ease. Finally, a classification scheme was developed that corresponded to the opinions of the practitioners and which appeared to fit the patterns established by an analysis of the literature content.

3.4. Data Storage and Retrieval

It was apparent from the start of the research that, because of the large number of articles to be read and classified, there would be a monumental task associated with the storage, control, and manipulation of the data. To maintain control, each article was assigned a series of codes which served to distinguish it from all others. A computer was used to store and manipulate the generated data. This allowed for data storage by article title, author, date of publication, publisher, keyword, and combinations thereof. Each article entered into the data bank was analyzed for content, and classified by keyword according to the taxonomy. These data (number of articles by keywords) were then recalled, totaled by year for each keyword, and plotted in the form of a histogram. The histograms provided considerable insight into significant changes in emphasis which occurred in the time period of interest. The data were then summed and plotted in cumulative form to gain a perspective of overall growth characteristics of each branch. This also provided a good indication of branches which had matured or which were in the process of maturing or increasing. Although not carried out in this research, the data base can easily be used to print a bibliography of the articles researched by keyword category or by correlation to applications.

4. RELIABILITY DISCIPLINE EVOLUTION

To gain perspective of the overall reliability discipline growth, a composite picture was formed using both books and short articles. Figure 5 presents this compositve overview, the unit of measure being the number of pages published in a three year interval. This unit of measurement was necessary in order to directly compare books and articles. It also provided a smoothing function so that the long term trend was discernible.

Figure 5 indicates that interest in reliability gradually increased from the early 1950's to about 1960, and then dramatically increased and held a high level until about 1970. After 1970 there appears to be a noticeable decline in growth. The growth during the 1960's in the U.S. can be largely attributed to the intense interest of the Space (NASA) and Defense (DOD) agencies and the accompanying financial resources which stimulated the aerospace industry in this time period. The decline noted during the early 1970's appears to be a combination of a maturing discipline and space and defense budget cutbacks.

4.1. Reliability Branch Growth Trends

Analysis of the main branches of the reliability taxonomy was performed utilizing the cumulative number of articles published during the time interval of interest. These data were presented in Figure 6 for the period from 1950 to 1976. The curves provide an overview of the growth patterns of the main branches of the discipline. The slopes of the curves indicate the emphasis that each branch received at a particular instant in time, and it appears that none of them have reached maturity. Maturity is assumed to be evidenced by a flattening of a curve, thus showing no further growth.

The starting points for the various curves indicate approximately when the literature began to emphasize each branch. They do not imply that nothing was published previously. Analysis and Management show strong tendencies toward increased emphasis in the immediate future. Within each main branch there are several subbranches which also have pronounced growth patterns. Examples of some of these are discussed in the following paragraphs.

4.1.1. Reliability Analysis Branch

The analysis branch was divided into three subbranches: (1) configuration, which consists of those analytical techniques dealing with the physical composition of systems or subsystems; (2) prediction; and (3) reliability growth. Configuration, which includes such analysis techniques as modeling, Failure Modes and Effects Analysis (FMEA), Fault

Trees, and Reliability Block Diagrams, emerged in the mid to late 1950's. It received a large growth stimulus in the early 1960's and has settled into a fairly stable, continuing growth pattern. (Figure 7.)

Prediction has exhibited characteristics similar to configuration but with somewhat lower emphasis. It appears to have received increased emphasis in the 1970's, primarily from defense and the service industries. Reliability growth has not shown significant development in comparison to the other subbranches. There have been a small and relatively consistent number of articles each year. Most of the articles are concerned with analytic techniques to predict whether a product will achieve a specified reliability level. Increased attention to this area shows up, however, in the technical report literature which was not included in this research.

Both the configuration and prediction subbranches have been further subdivided into twigs. The twigs which have grown out of the configuration are illustrated in Figure 8. Modeling/simulation and failure analysis experienced an extremely sharp increase in growth starting in 1962. The other areas appear to have experienced a reasonably consistent growth pattern with the exception of design review, which appears to have matured in the late 1960's. Fault tree analysis, developed originally for safety analysis, appears to be receiving increased emphasis in the 1970's. This is partially caused by the growing interest in safety stimulated by the electrical power industry. The technique is presently one of the strong bridges between reliability and safety analysis.

4.1.2. Reliability Management Branch

This branch is very broad in scope and is the most complex to discuss since numerous subbranches and twigs have emerged since the 1950's. The major subbranches of management are illustrated in Figure 9. All of the subbranches and twigs, with the exception of product liability, have received major impetus to their growth from defense and space activities.

The reliability program management, cost, failure, recurrence control, and procurement subbranches all emerged very early in the literature. Reliability program management has received far more emphasis than any of the other subbranches. Failure recurrence control (efforts by management to reduce recurrence of reliability failures) indicates signs of approaching maturity in the mid-1970's, whereas both procurement and cost show signs of increasing emphasis. Product liability emerged in the early 1970's and is giving every sign of growing into a major area in the near future as a result of current consumer protection emphasis in the U.S. In the procurement subbranch of management, warranties emerged in the early 1960's and exhibits a pattern of increased growth as a result of airline and defense interest in failure-free and reliability improvement warranties.

4.1.3. Reliability Test and Evaluation Branch

The test and evaluation (T&E) branch, illustrated in Figure 10, is divided into three main subbranches: methods, statistics, and reporting/evaluation. The T&E branch emerged very early in the literature as an extension of the quality control discipline. The literature gradually transitioned from quality control related issues into reliability during the 1950's. Defense and space activities appear to have provided the major motivation for the rapid increase noted for methods of testing during the early 1960's.

Development of several twigs in the statistics subbranch are noted in the literature, as illustrated in Figure 11. Sampling plans and design of experiments are the most direct carry overs from quality control. It appears that design of experiments has matured in the 1970's. Bayesian techniques and parameter estimation have exhibited the most active growth patterns.

4.1.4. Reliability Design Branch

Design has been divided in Figure 12 into component related activities and engineering analysis activities to separate the curves for ease and clarity of data presentation. Part/material selection and component testing have been the most active areas in this branch. Both of these have exhibited dramatic increases in growth starting about 1964, and the increased growth rate has continued to the present. Redundancy and high reliability parts have exhibited smaller growth patterns. In fact, the high reliability parts twig appears to have reached maturity. Redundancy appears to be receiving increased emphasis in the 1970's, and this seems to be primarily due to space applications.

4.1.5. Reliability Theory Branch

This branch emerged in the literature in the late 1950's. The primary focus appears to be theory of failure, and this subbranch has exhibited a relatively constant growth pattern since 1961. Renewal theory has exhibited a small but consistent growth pattern.

4.2. Reliability Branch Correlation With Specific Applications

Only the four most active branches (Design, Analysis, Management, and Test) have been selected for correlation of reliability with applications because these reflect the major emphasis of the reliability discipline and, hence, contain sufficient data to establish some meaningful trends. Figure 13 presents the correlations of design and analysis with specific applications. The curves correlating design with specific applications indicate some rather dramatic changes in emphasis over the last two decades. Design applied to

space transportation has received more emphasis than any other application during the 1960's. It appears to have matured in the 1970's. Air transportation design applications were mentioned in the earliest literature. It has displayed a slow but steady growth and seems to have reached maturity. Communication and medical applications also seem to have approached maturity in the mid 1970's. Computer applications (including both software and hardware) emerged in the literature in the late 1950's and has experienced increasing emphasis ever since. Ground transportation (primarily rapid transit) emerged in the mid 1960's and is receiving increasing emphasis in the mid 1970's. Power generation (including both nuclear and conventional) emerged in the early 1970's and is showing signs of significant growth during the 1970's if the present trend continues.

The same general trends are evident in the analysis branch as were noted for the design branch. However, it appears that analysis has lagged design by two or three years. None of the analysis applications appear to have matured, although space and air transportation are exhibiting signs that this may happen in the not too distant future. Analysis applications emphasis, like design, appears to have shifted to computer and power generation in the 1970's. Both of these application areas have received very high growth stimulation in the last few years and there is every reason to believe that these trends will continue.

Figure 14 presents the test and evaluation and management branches correlated with the same specific application areas. Some of the trends established for analysis and design are also evident. For example, space transportation applications are approaching maturity for both T&E and management. All other applications presented in the T&E portion of Figure 14 indicate stable growth patterns in the 1970's. T&E application in power generation has just recently emerged in the mid 1970's. It is likely that it will experience the same significant growth exhibited in the other branches.

Management applications other than space transportation and communications appear to be exhibiting strong, stable growth patterns. Ground transportation emerged in the mid 1960's and has achieved a strong growth pattern. Most of the emphasis here has been in rapid transit considerations for cities. Power generation again indicates a very strong growth pattern.

Management applications in the medical field emerged in the late 1960's and appears to have established a stable growth pattern. A large part of the emphasis here has dealt with the effect of medical device failures on patient safety.

5. MAINTAINABILITY DISCIPLINE EVOLUTION

A composite picture using both books and periodicals was formed in order to gain a perspective of the overall growth of maintainability. As for reliability, the unit of measure is the number of pages published in a three-year interval.

Figure 15 indicates that interest in maintainability began about 1957, sharply increased to about 1966, and then dramatically decreased until about 1972. The growth in complexity of systems has led to the realization that equipment reliability cannot be improved to the extent that the need for maintenance could be economically eliminated. Significant increases in operating and support costs have provided additional support for the rising interest in maintainability. The overall growth of maintainability has been strongly influenced by recognition of this engineering discipline as a critical element of system effectiveness.

5.1. Maintainability Branch Growth Trends

Figure 16 indicates that management has received far more emphasis than any of the other maintainability branches. This stems from the imposition of firm military standards for maintainability programs in the U.S. in the 1960's. The analysis and design branches emerged in the 1950's when maintainability criteria were being developed. Their growth then increased as maintainability requirements became better defined in the early 1960's. Analysis and design seem to have approached maturity in the 1970's. On the other hand, data growth rose in the early 1960's, began to flatten out in the late 1960's and appears to have matured in the 1970's.

Maintainability test and evaluation emerged in the late 1950's, rose slowly in the middle 1960's, and has received increased emphasis since. The growth increase noted in the late 1960's appears to be due to military requirements for maintainability demonstration tests. Growth tendencies within each of the above main branches are developed in the following sections.

5.1.1. Maintainability Analysis Branch

The analysis branch has been divided into simulation/modeling, prediction, maintenance engineering analysis (MEA), and design reviews. (Figure 17.) Prediction received a large growth stimulus in the early 1960's and then slowed down. This stimulus was primarily due to interest from the military services in developing methods for quantifying measures of maintainability. Simulation/modeling growth increased from the middle to late 1960's and appears to have settled into a stable, rapidly increasing growth pattern. Maintenance engineering analysis (MEA) has also shown a strong emphasis in the 1960's with a lower but steady growth in the 1970's. MEA growth in the 1960's can be linked to interest in Integrated Logistic Support (ILS) concepts.

5.1.2. Maintainability Design Branch

This branch has produced several twigs. Figure 18 shows the relative emphasis each has received. The literature indicates that human factors engineers have been primarily responsible for the development of this branch through the development of maintainability design guides. This interest began in the 1950's and continued through the following decade.

Quantitative maintainability design criteria relate to equipment features that enhance maintenance time reduction. These requirements have been specified for defense electronic, aircraft, and missile systems for about 15 years. The rapid growth of quantitative design requirements is portrayed in Figure 18. The early quantitative maintainability specifications, established in 1960, stimulated defense contractors to put greater emphasis on design requirements. The effort to meaningfully quantify requirements continues to the present day. Trade-off studies as a design technique sustained a high growth rate in the 1960's and appear to have matured in the early 1970's.

5.1.3. Maintainability Test and Evaluation Branch

The test and evaluation branch can be split into three main subbranches: methods, statistics, and reporting/evaluation as shown in Figure 19. Greater emphasis appears to have been given to test methods than to either of the other two subbranches. This emphasis started in the early 1960's and has continued to the present.

Interest in maintainability testing evolved in the early 1960's and has continued into the 1970's with publication of improved methods for maintainability demonstration. A normal carry-over from demonstration testing is the need for feedback data and other reporting mechanisms. The reporting and evaluation subbranch received emphasis in 1965 and leveled off in the early 1970's. Concern with the accuracy and application of feedback data has stimulated new growth in the mid 1970's.

The application of statistics in developing test methods has been a natural development. Maintainability demonstration, for example, is primarily concerned with the measurement of active maintenance downtime. Measurement of both preventive and corrective maintenance downtime is generally conducted using statistical test methods under carefully defined conditions. Interest in statistical techniques shows a steady increase starting in 1965 and maturing about 1970.

5.1.4. Maintainability Management Branch

This branch is the most difficult branch to assess as it has the most complex structure. Figure 20 shows the five main subbranches: (1) organization and management, (2) logistics, (3) availability, (4) procurement, and (5) cost. All subbranches showed growth patterns in the 1960 - 1970 decade. Organization and management's growth can be traced to the military's desire to formalize maintainability into a firm contractual requirement. Logistics concerns grew dramatically in the U.S. in the 1960's as the Defense Department put increased emphasis on lowering operation and support costs.

Availability deals not only with failure (reliability), but also with repair (maintainability). After its original emphasis in the mid 1960's, availability appears to have experienced little or no growth in the 1970's until the past several years. Procurement's rise through the 1960's is traced to specification of maintainability requirements in military contracts. The need to specify quantitative requirements forced more consideration of maintainability in specifications.

5.2. Maintainability Branch Correlation With General Applications

Correlation of the most active main branches of maintainability with general applications indicates a strong influence of maintainability interest within the U.S. Department of Defense and a lesser influence by NASA.

NASA interest in maintainability is primarily in the space environment. The literature has been sparse with respect to number of published articles. Awareness of the importance of maintainability by NASA did not really emerge until the middle 1960's. It exhibits a peak in the late 1960's when the space program design effort was at its peak. Maintainability interest in the 1970's can be expected to increase with renewed emphasis on longer manned space missions.

The data base does not show a large interest in maintainability in consumables, construction, or industrial short-life equipment. However, the literature indicates a concern for maintainability by those companies manufacturing equipments which they sell or lease but continue to maintain throughout the equipment's service life. Computer and copying machine manufacturers are examples. The thrust for consumables has been more towards reliability, product liability, and quality control. Maintainability interest with respect to industrial long-life equipment appears to have been a significant concern for commercial aircraft. Here again, the reasons appear to be economics and consumer satisfaction since the airlines perform their own maintenance.

6. CONCLUSIONS

The preceding discussions examined the growth of the reliability and maintainability disciplines by developing growth curves of the various branches within each discipline.

Figure 21 summarizes how the emphasis has shifted within each discipline from 1960 to 1975. Reliability had a high emphasis on management and test & evaluation in the late 1950's and early 1960's. However by 1965 the emphasis had started shifting more to analysis and design, and this trend has continued to the present. Maintainability, on the other hand, had high emphasis on design in the early 1960's, but this has subsequently shifted to management and test & evaluation.

Several conclusions regarding present and future emphasis of reliability and maintainability have been reached as a result of the research. As noted earlier, military and space activities have strongly influenced the development of both reliability and maintainability in the U.S. Current trends, however, indicate that the energy (power) and other service industries will play a major role in influencing the growth trends in the future.

In recent years more attention has been devoted in the literature to the implications of equipment reliability on safety issues. Analysis techniques, such as Fault Trees, have been developed which are useful in both reliability and safety analysis. Many articles have dealt with these techniques. This area has been growing noticeably in the recent past and appears to have potential for continued rapid development in the future.

Product liability is potentially a very complex area with safety and reliability issues combined with the complicated legal implications involved with consumer protection. It is experiencing a substantial growth rate. These issues are important because court decisions could have significant effects on the amount of analysis and testing a manufacturer may have to undertake in order to demonstrate that his product is safe and reliable and to precisely define the environmental rating.

Finally, a structure has been evolved for the reliability and maintainability disciplines which the authors believe can be used for organizing symposia, classifying papers, and producing useful bibliographies.

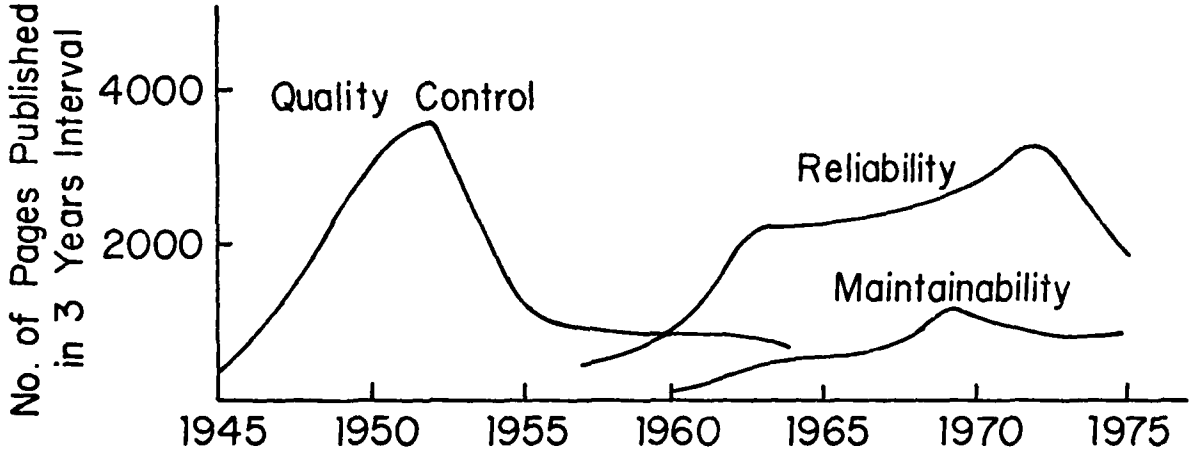


Fig.1 Emphasis on R & M from books

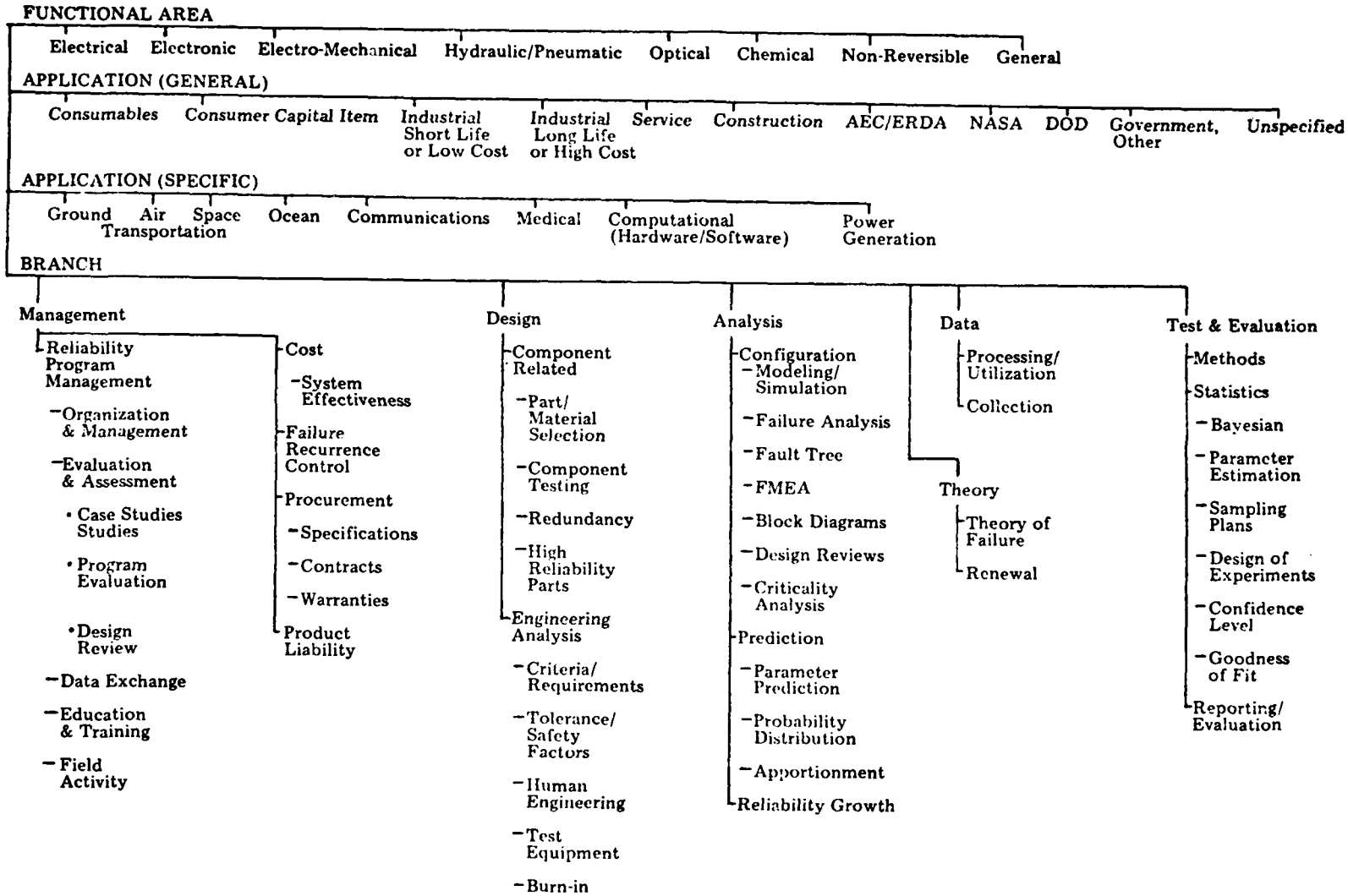


Fig. 2 Reliability taxonomy

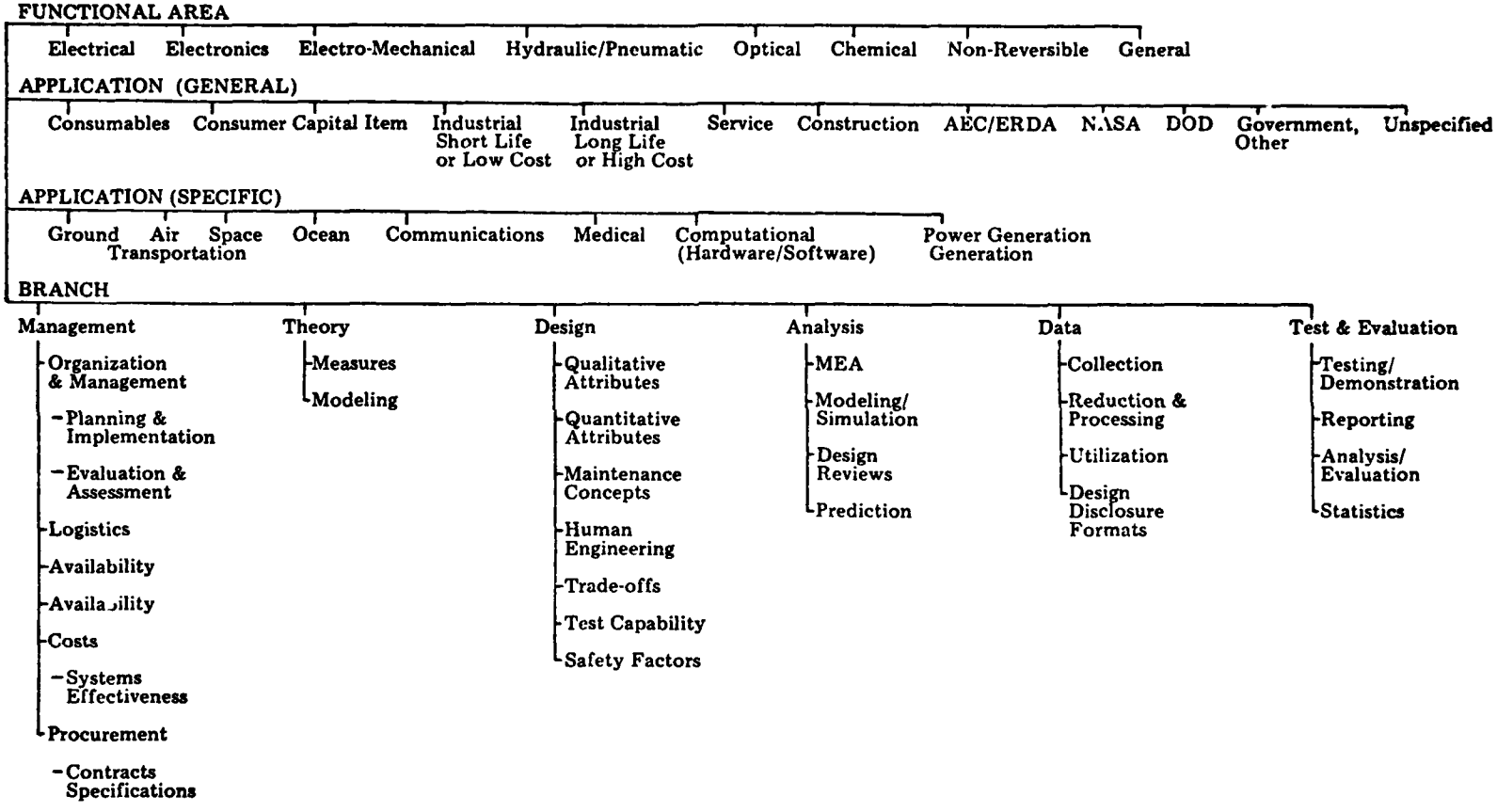


Fig.3 Maintainability taxonomy

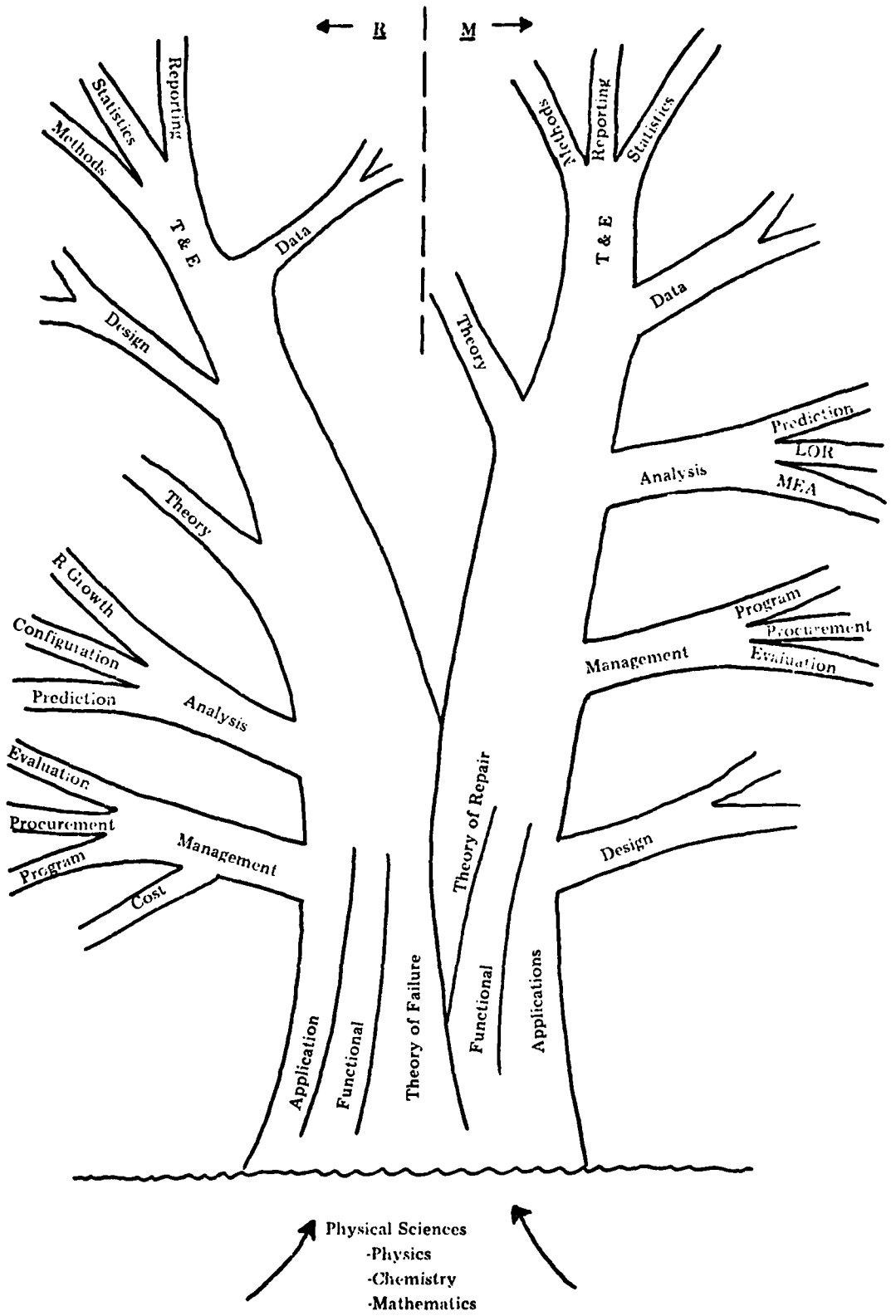


Fig.4 R and M disciplines described by tree analogy

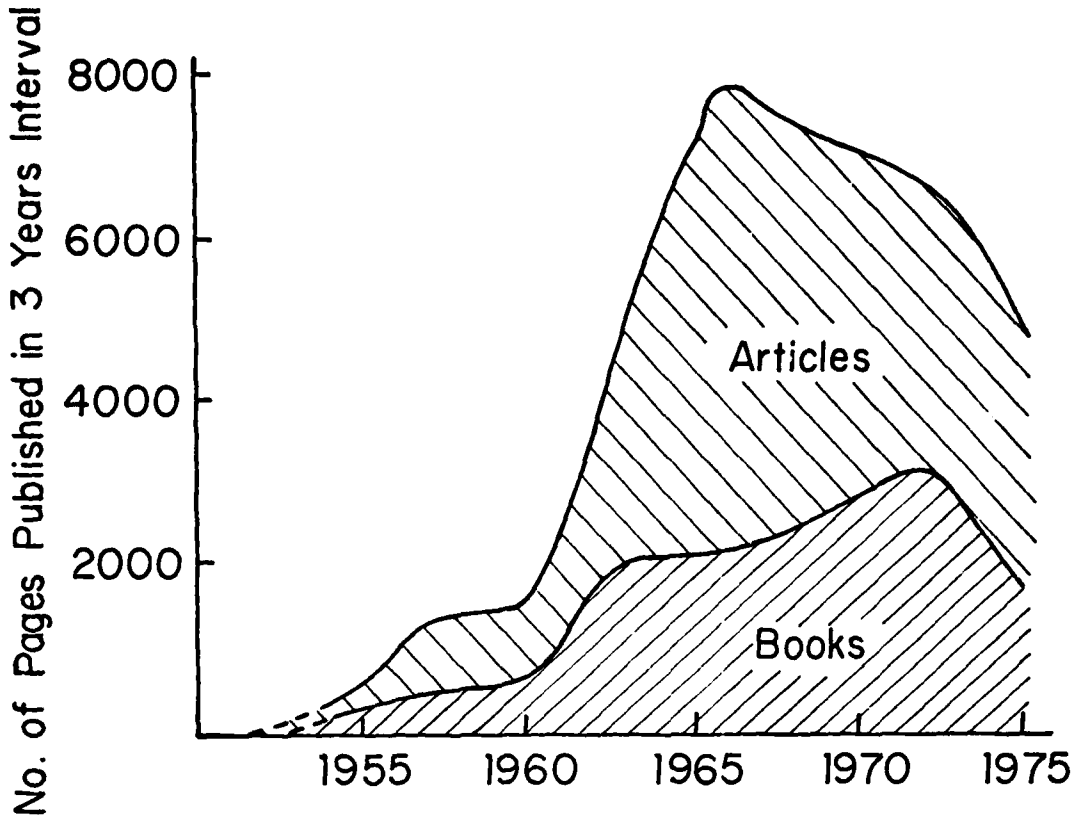


Fig.5 R discipline emphasis

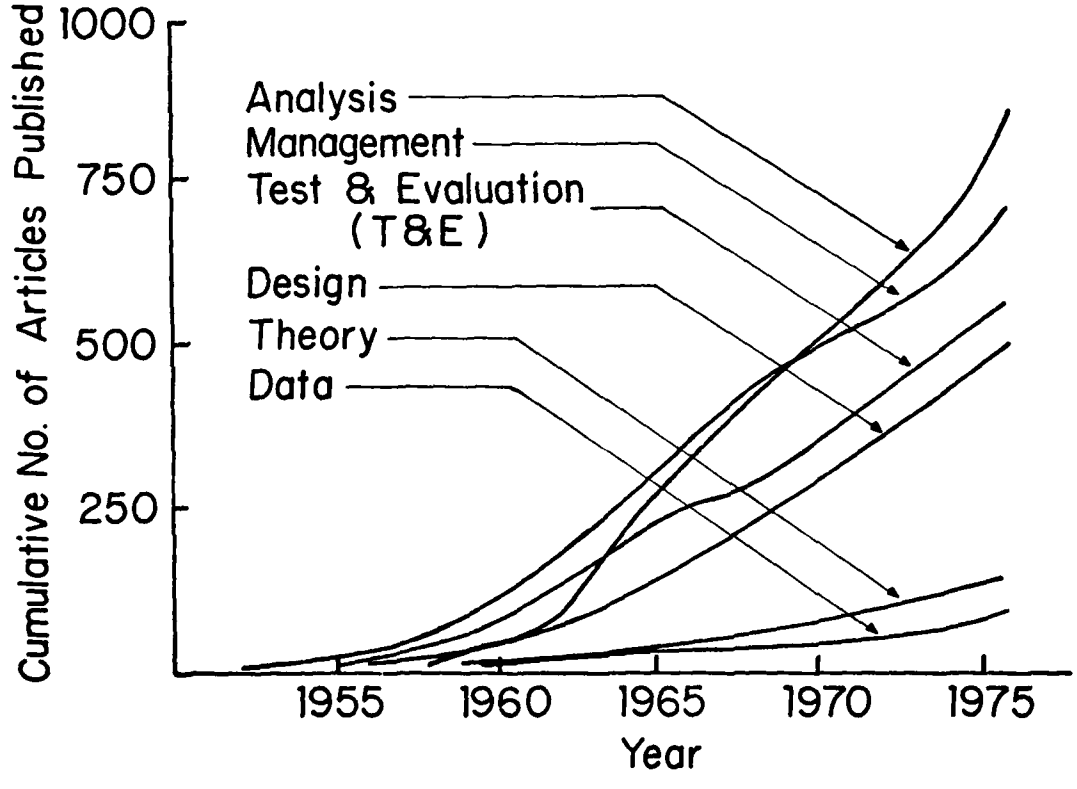


Fig.6 R branches

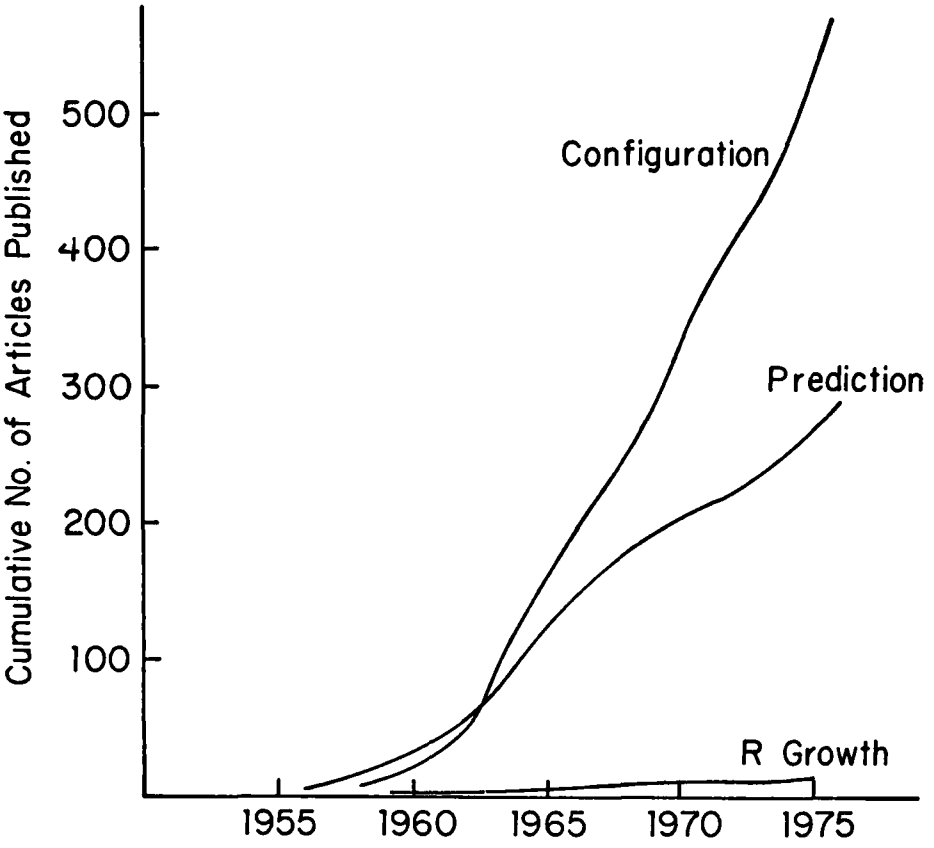


Fig.7 R analysis branch

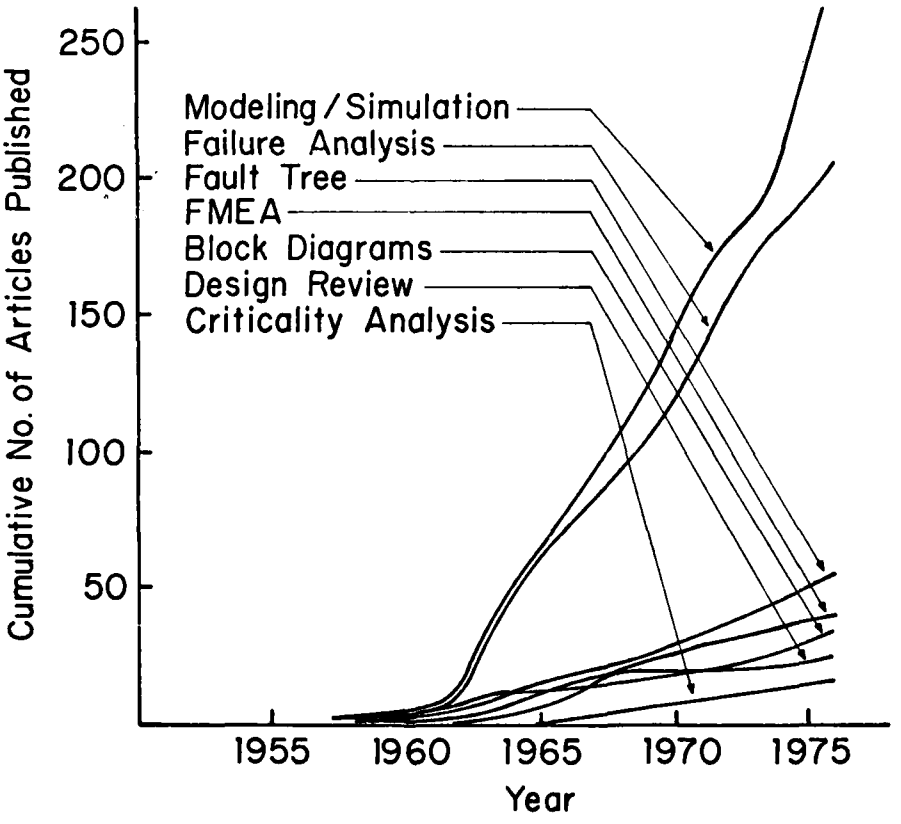


Fig.8 R configuration subbranch

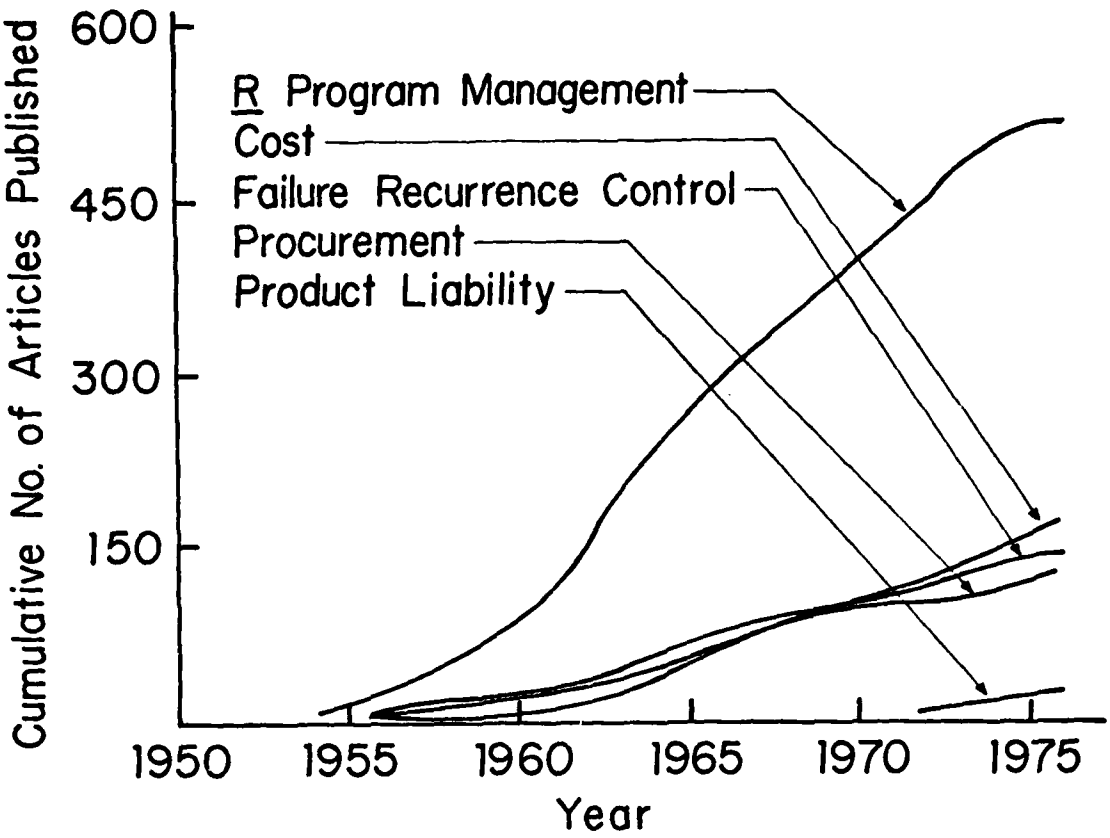


Fig.9 R management branch

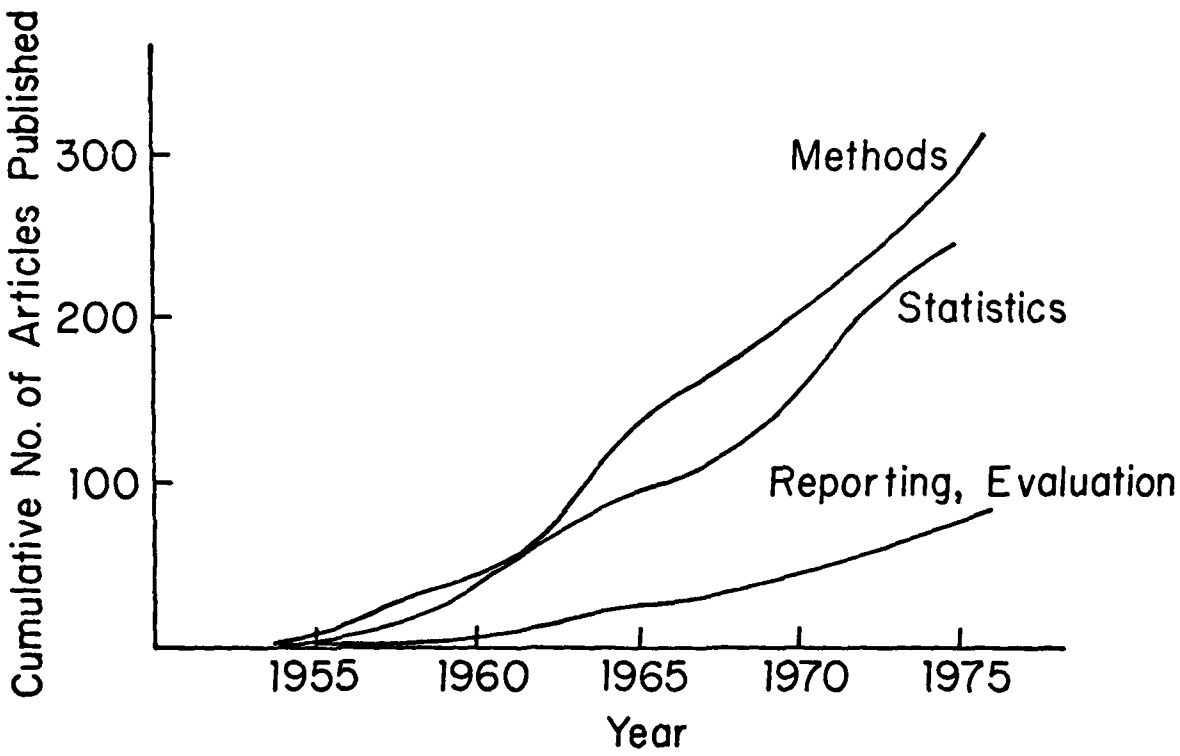


Fig.10 R test and evaluation branch

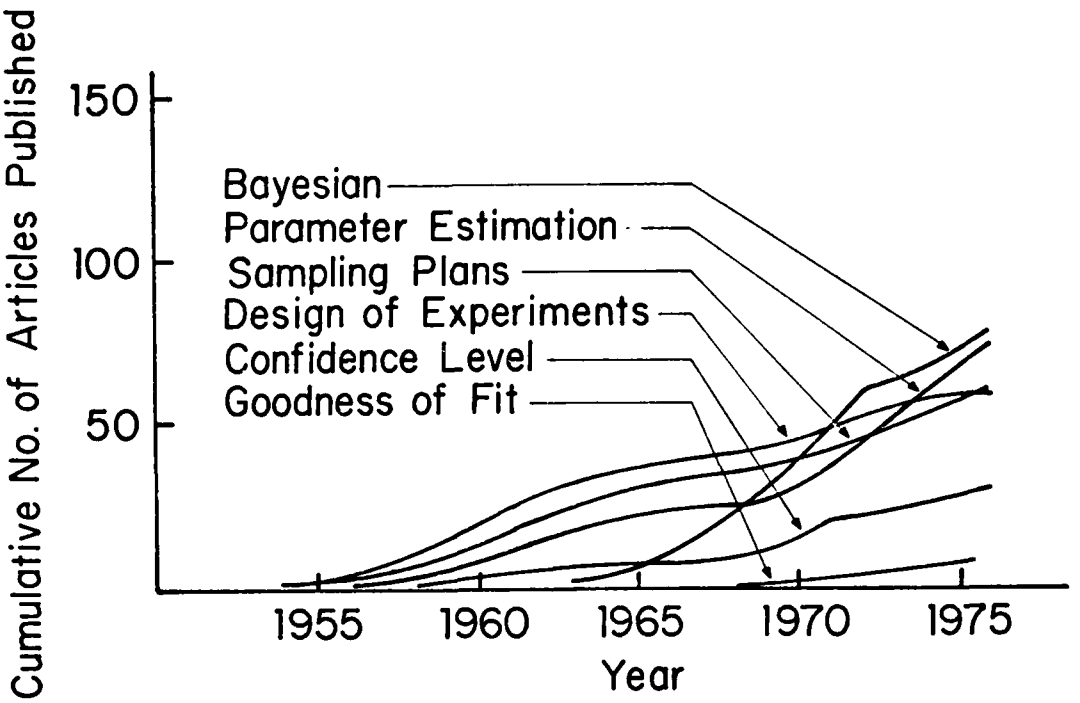


Fig.11 R statistics subbranch

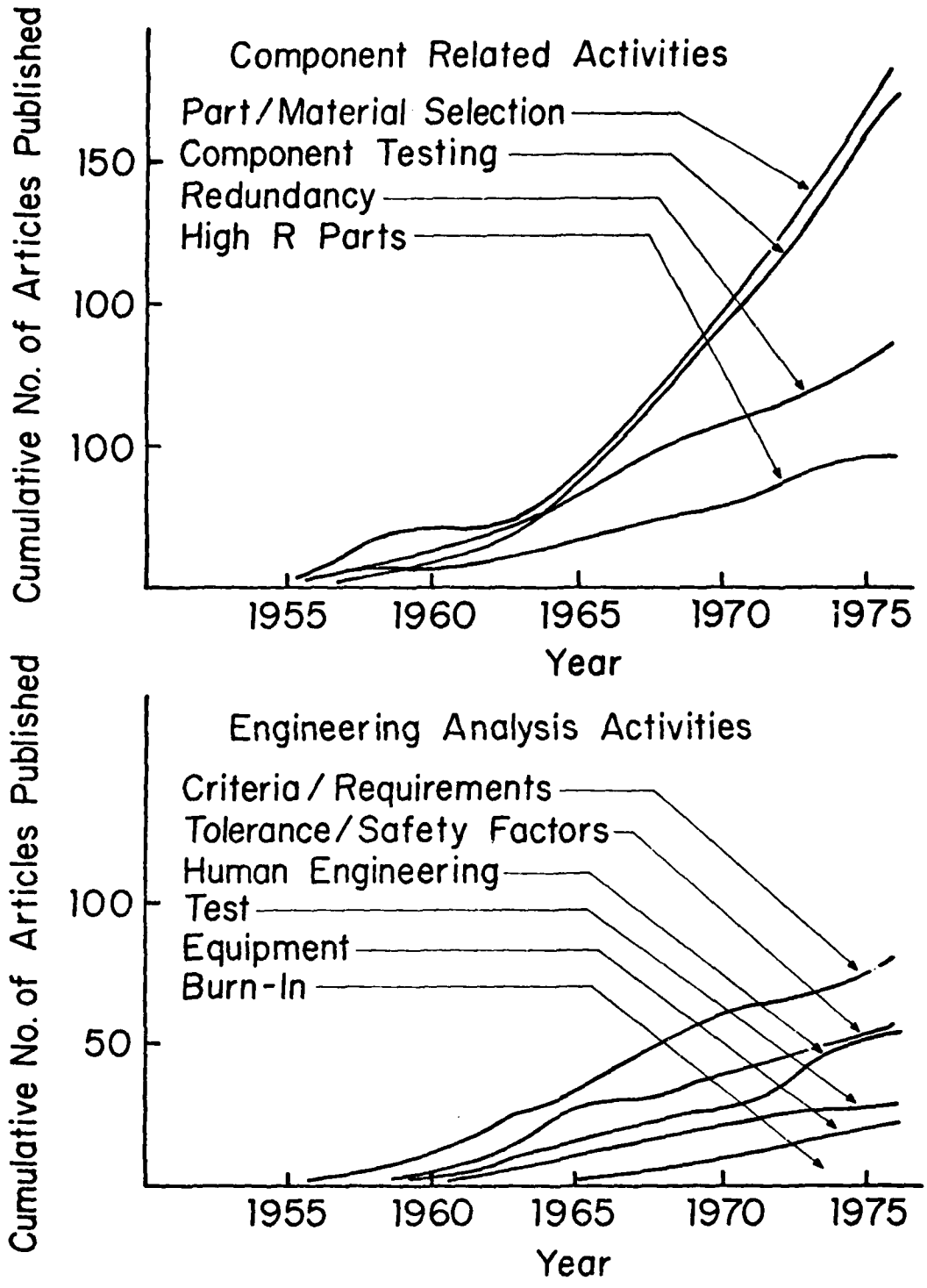


Fig.12 R design branch

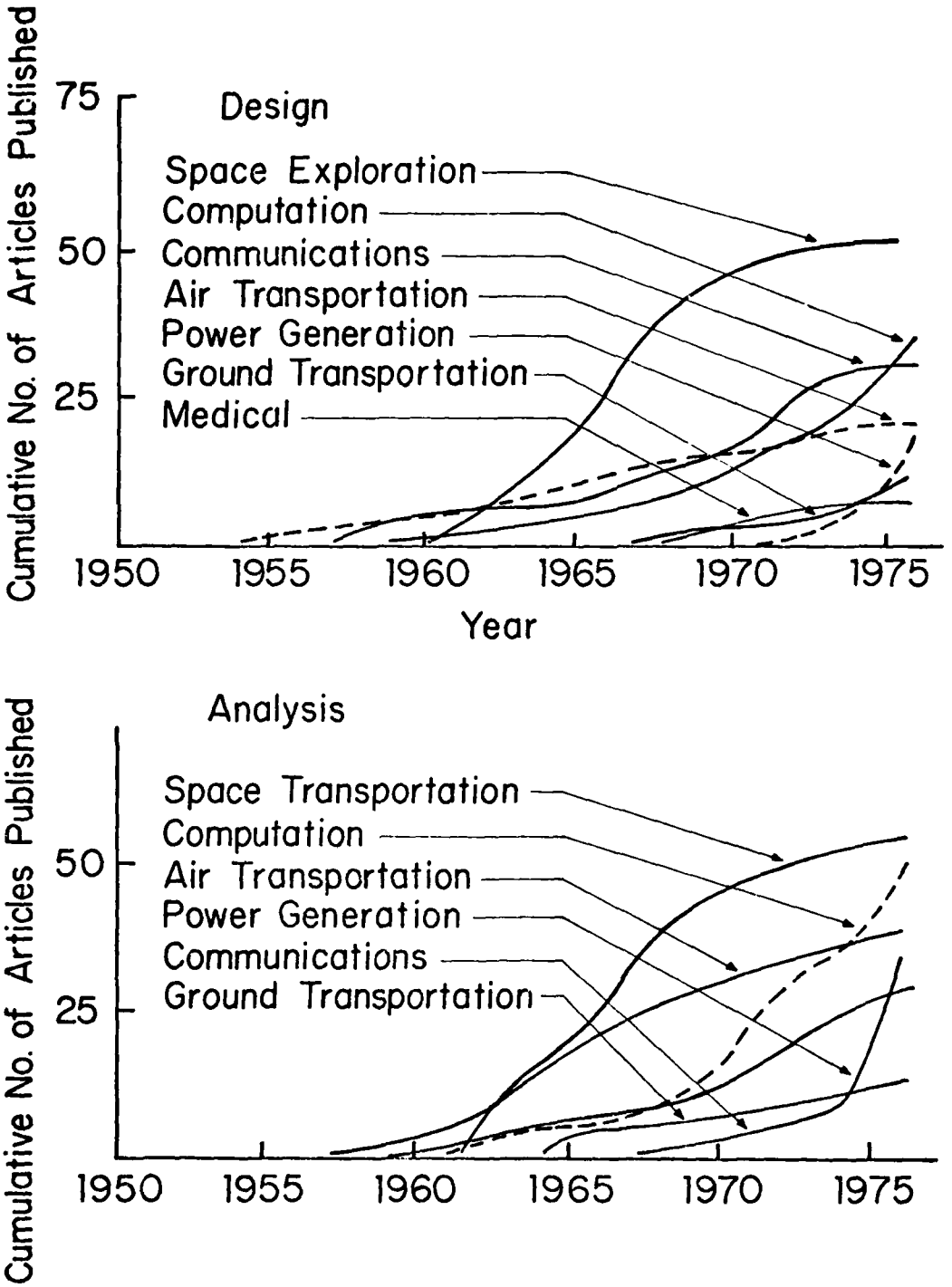


Fig.13 Correlation of design and analysis with specific applications (R)

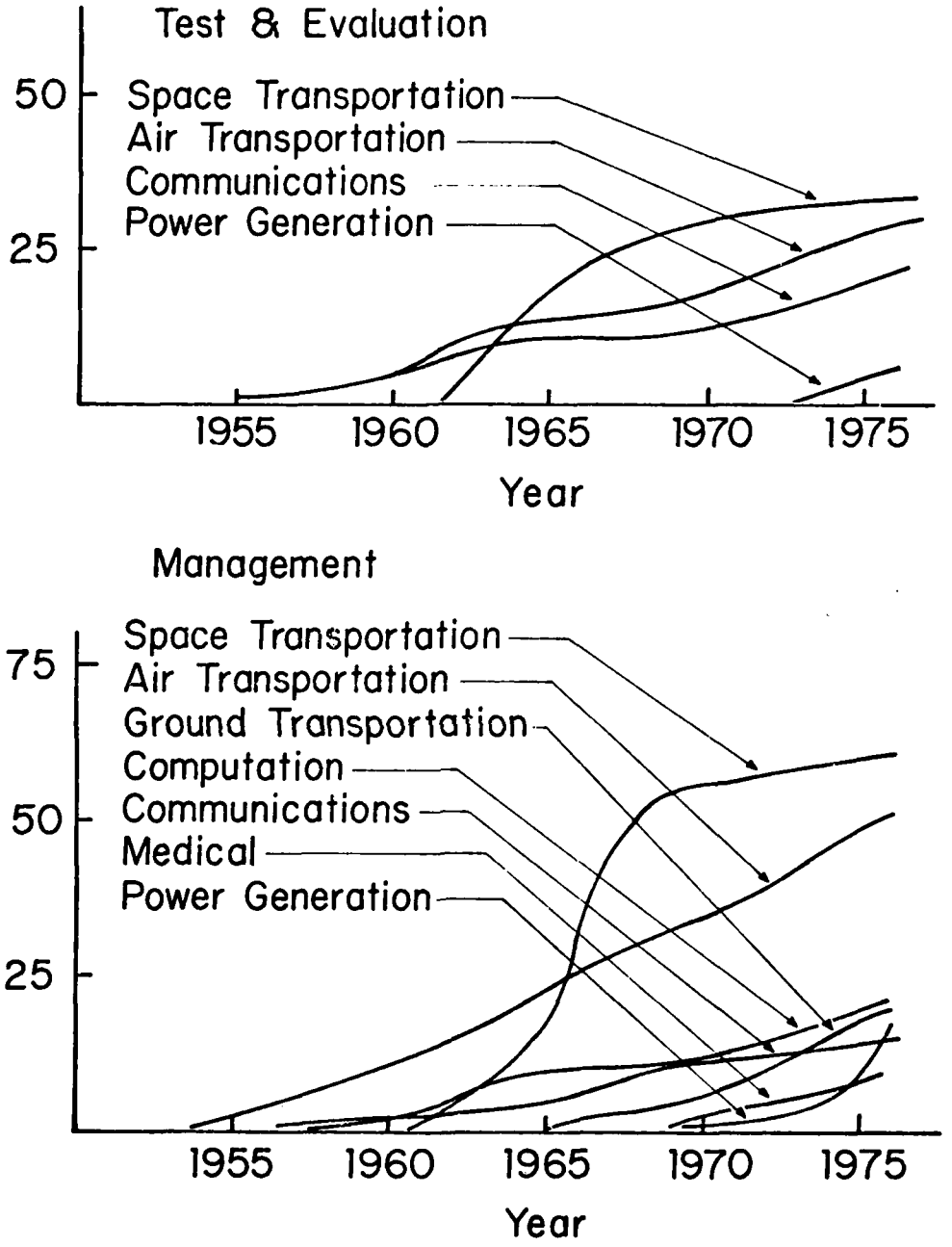


Fig.14 Correlation of T & E and management with specific applications (R)

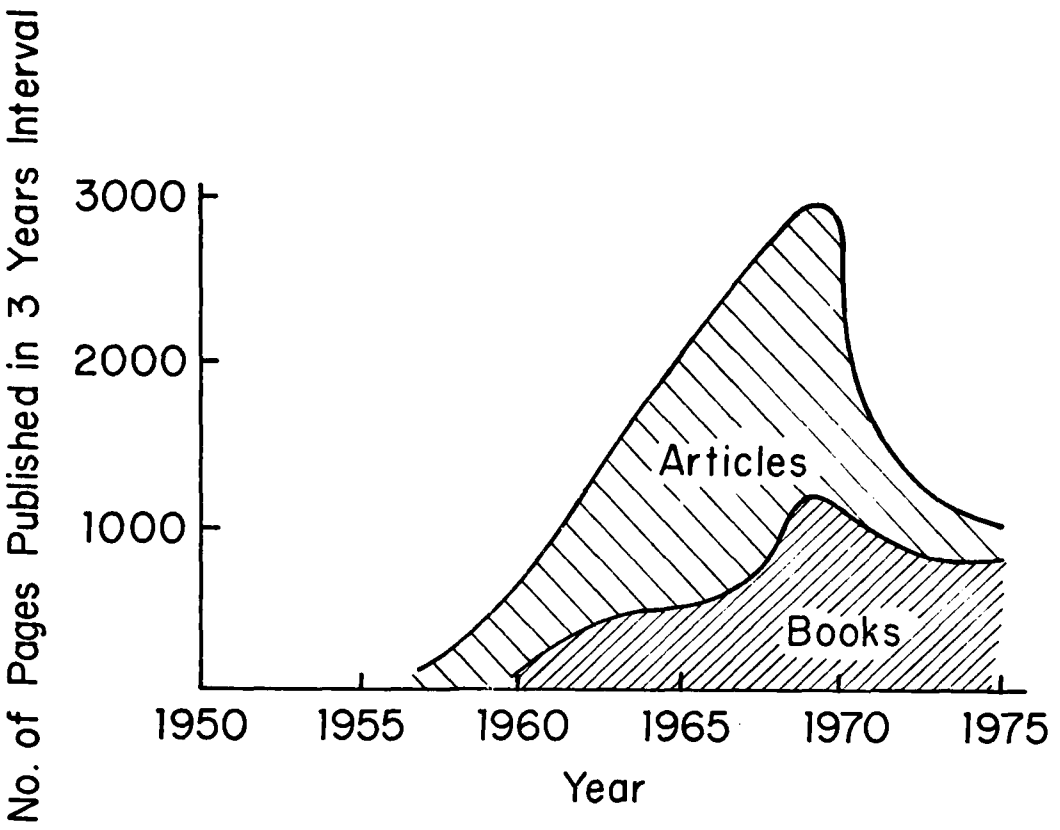


Fig.15 M discipline emphasis

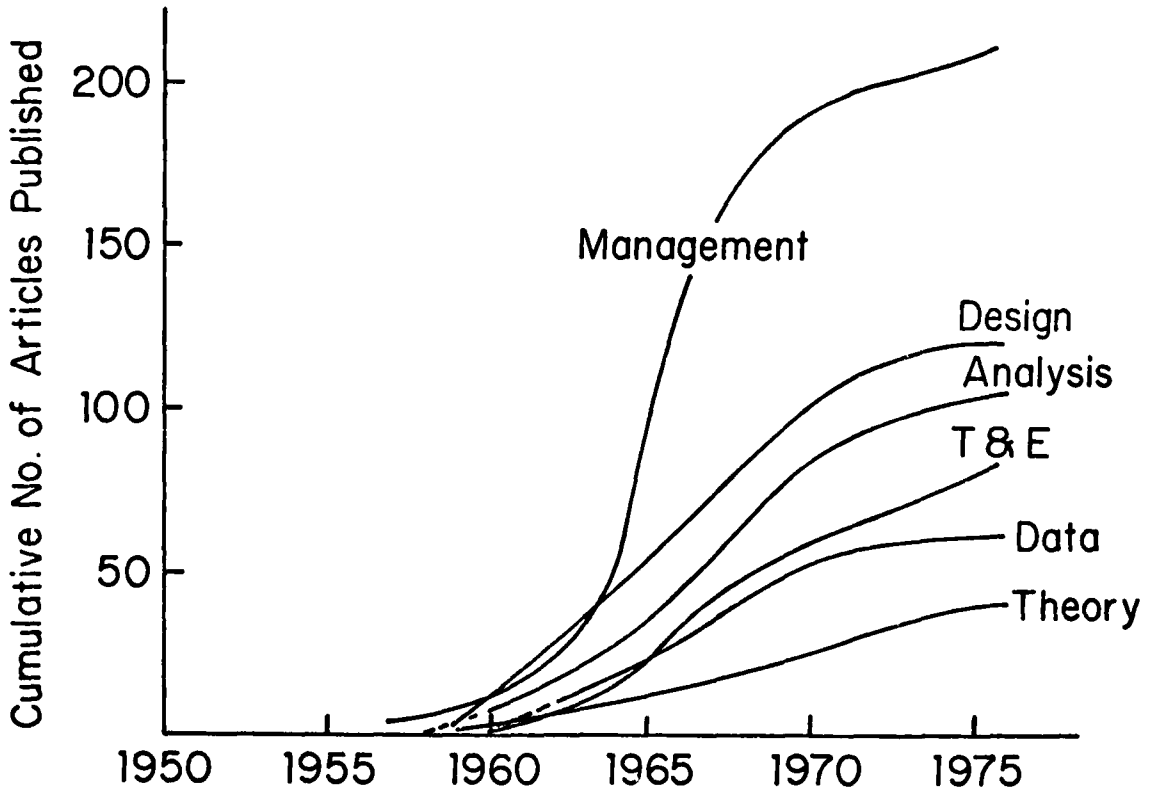


Fig.16 M branches

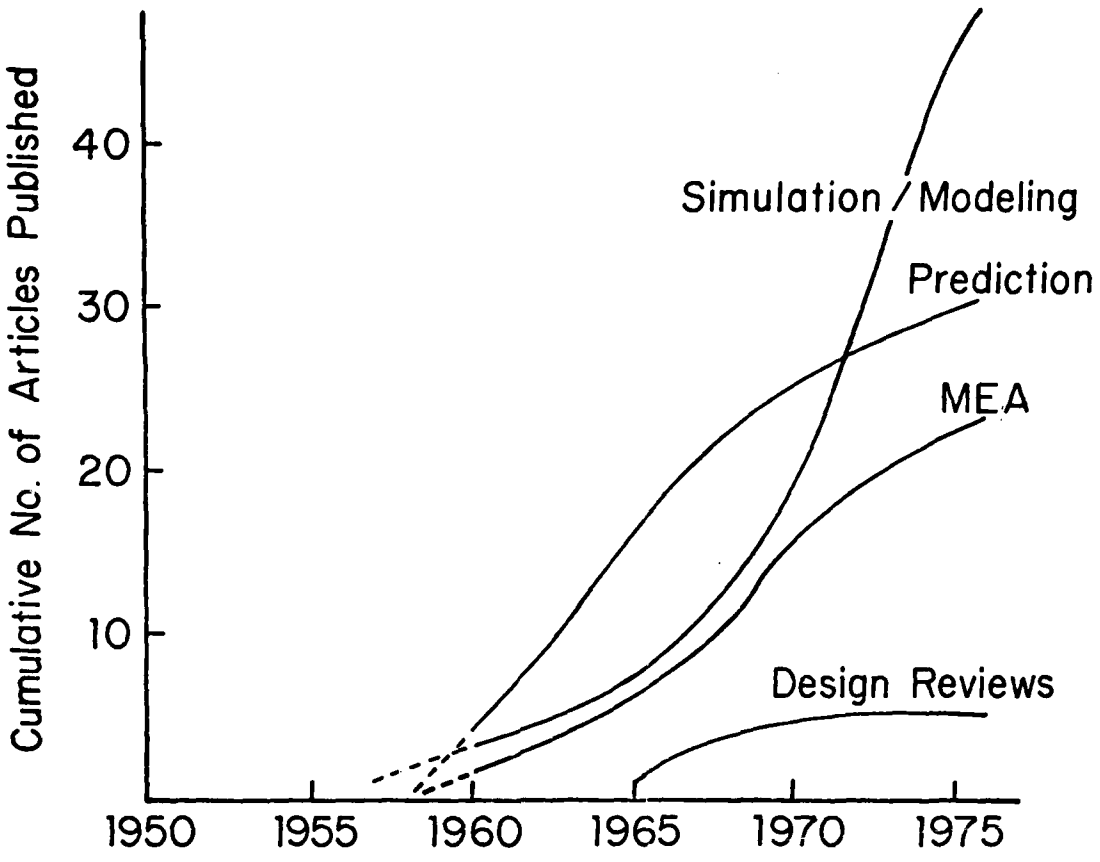


Fig.17 M analysis branch

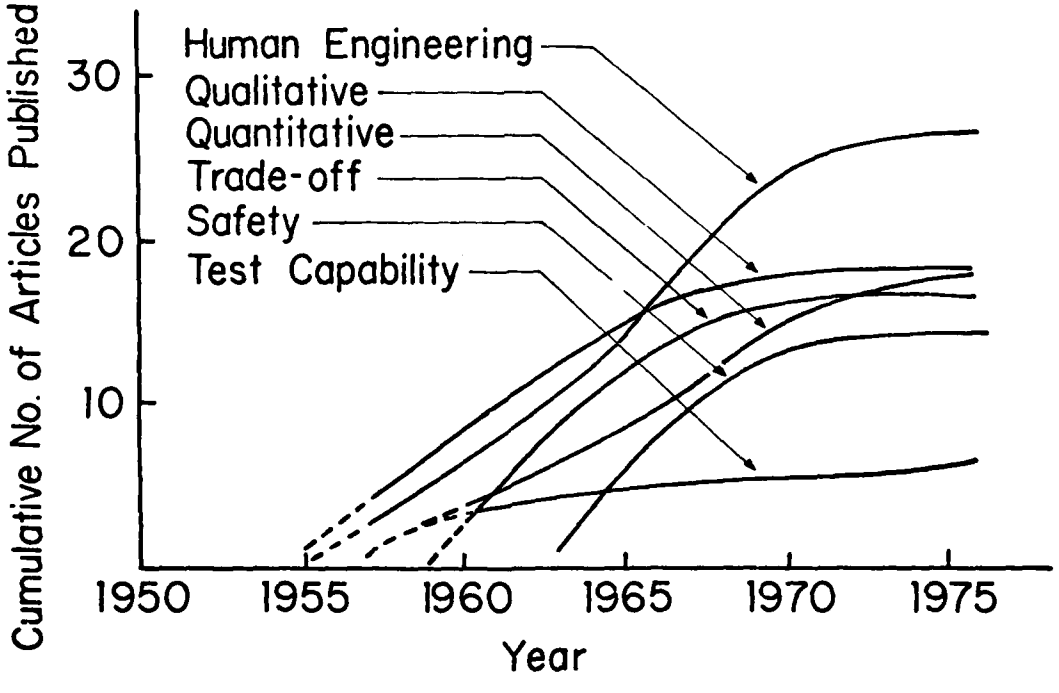


Fig.18 M design branch

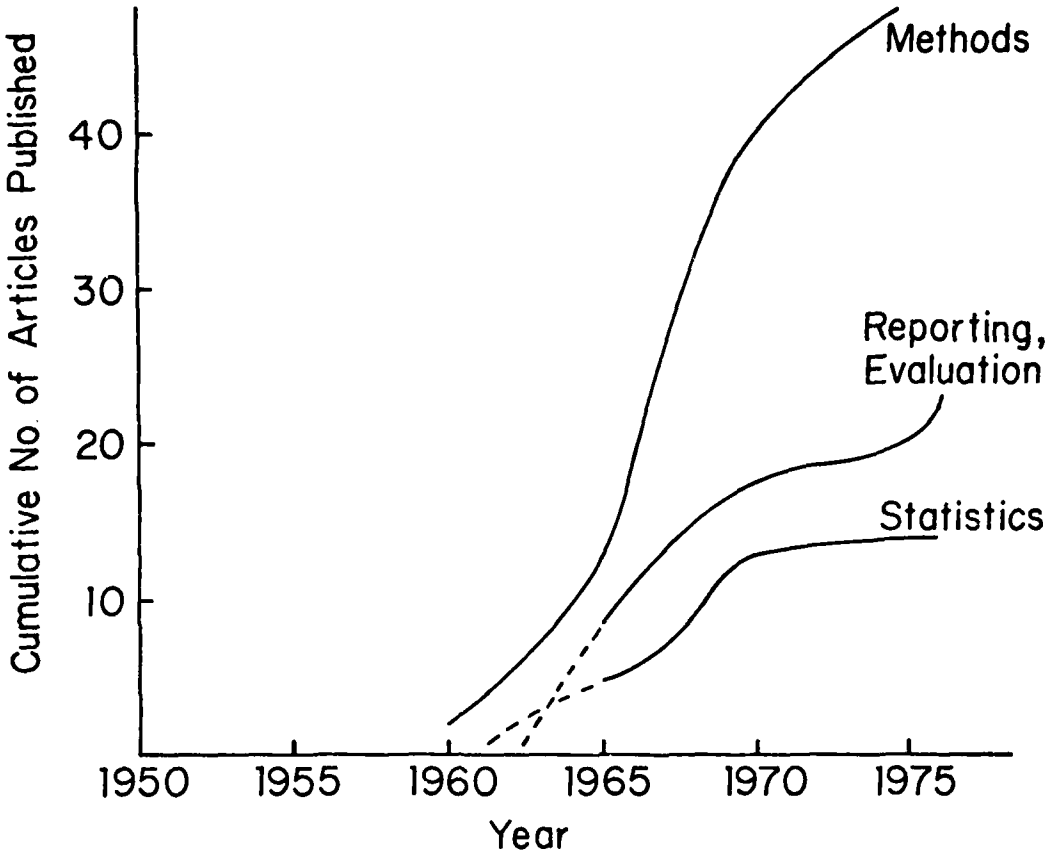


Fig.19 M test and evaluation

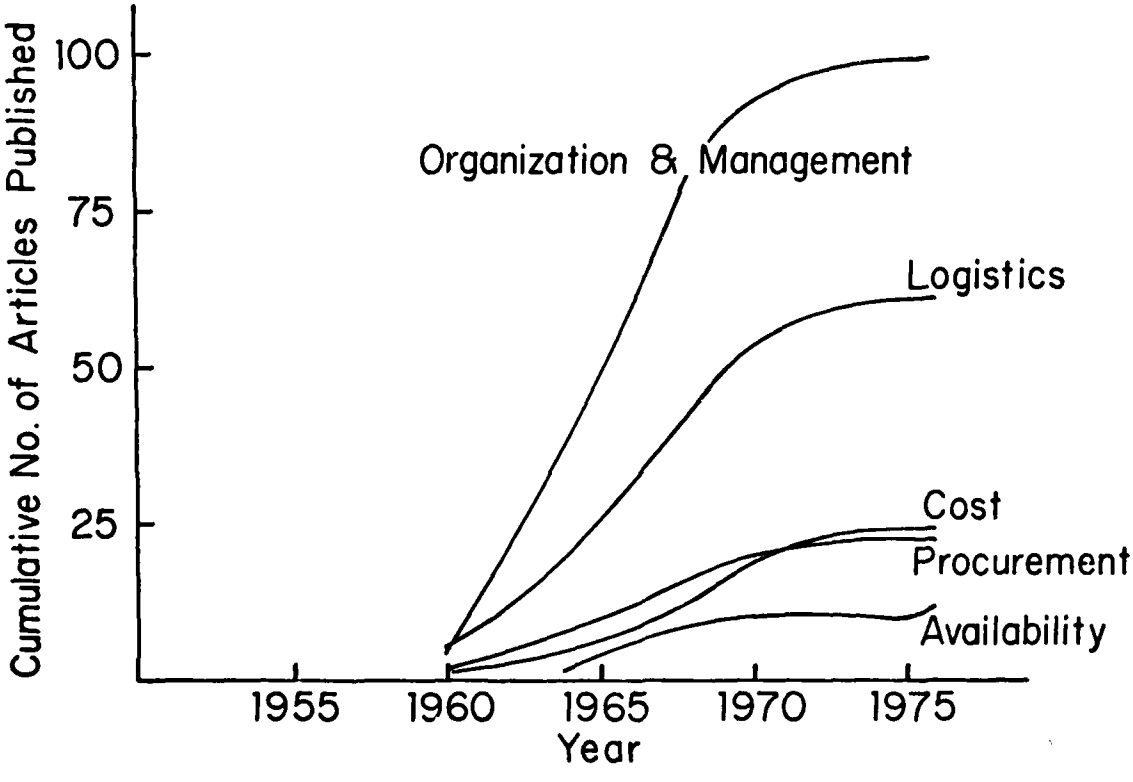


Fig.20 M management branch

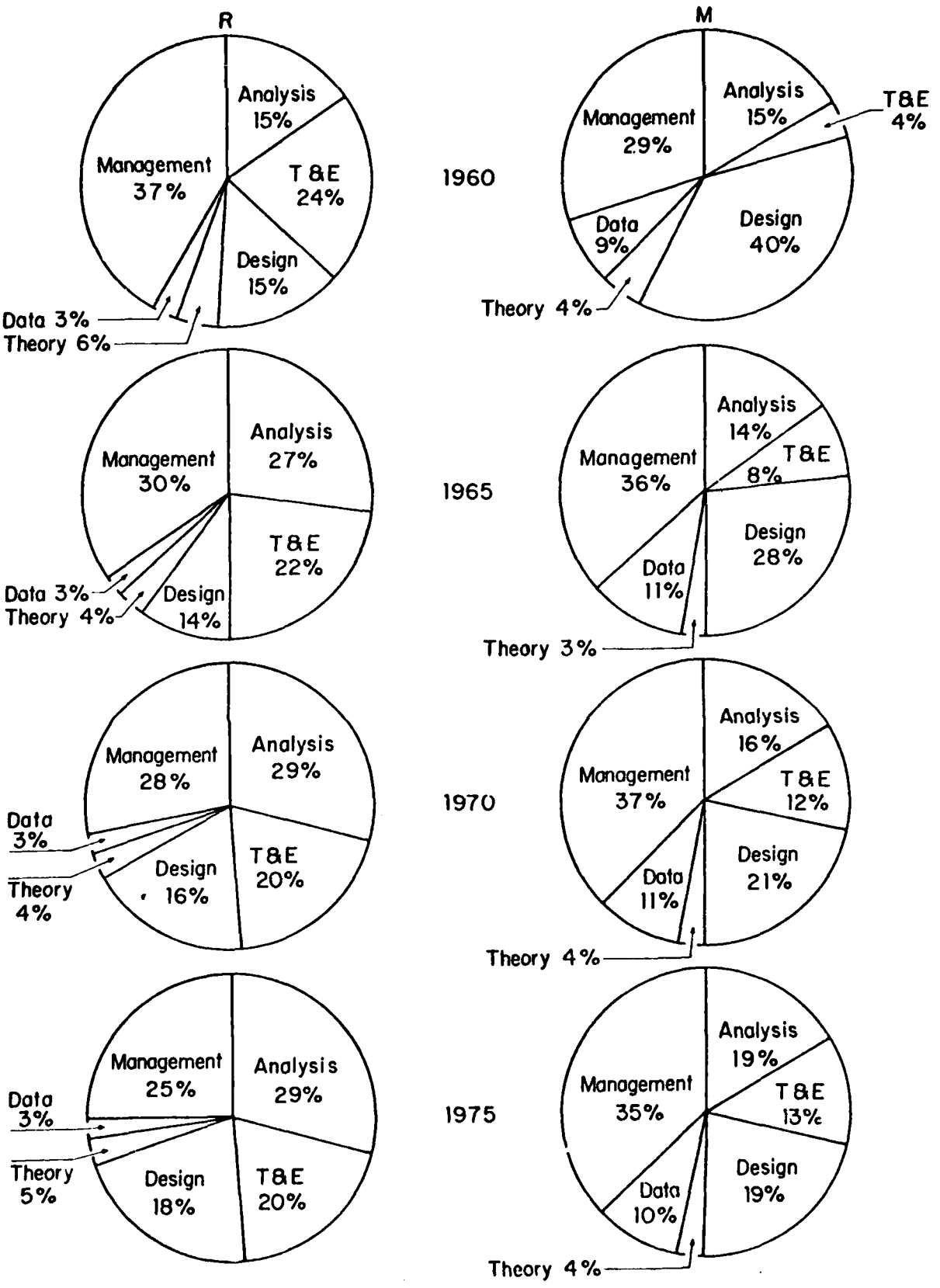


Fig.21 Comparison of R and M evolutions

DISCUSSION

H.S.Balaban, US

Another indicator of trends in reliability and maintainability could be the number of R & M courses taught in universities. Has any research been performed in this area?

Author's Reply

While there appears to be courses in some universities on reliability these tend to be on mathematical (statistical) reliability. To the best of my knowledge, they do not consider the engineering aspects of reliability (physics of failure, stress and strain, prediction and demonstration etc.). I am not aware of any courses in Maintainability Engineering in civilian universities. The Naval Postgraduate School, the Air Force Institute of Technology and the Army Management Engineering Training Activity have courses in R & M Engineering and a number of short courses are also given on these subjects by various organisations.

Questioner Unknown

Did you use Citation Analysis to determine the impact of significant particles in generating literature?

Author's Reply

References in the papers were used as a lead. The difficulty in citation involves different phenomena such as classification.

The annual conference on reliability and maintainability held in the US uses a number of key-words for indexing developed by the American Society for Quality Control. No citation index is useful however unless it is periodically updated to keep pace with technology. Citation Analysis at the moment is greatly outdated for present needs.

H.A.T.Timmers, Ne

You have given several statistics on the attention paid to Reliability and Maintainability, divided over several industrial activities and within the branches which can be distinguished. Are statistics also available on the Hardware versus Software aspects of systems? If so, what is your analysis/opinion?

Author's Reply

We did not track Software Reliability and Maintainability in our study since we did not have ready access to some of the software publications at the time.

My own personal opinion is that Software Reliability has been in a rapid growth pattern since about 1970 and there are new journals and symposia devoted to it. Software Maintainability (really Software Configuration Management, in my opinion) has begun to appear in the last two or three years as a subject of papers.

W.Ehrenberger, Ge

According to your graphs the overall number of pages has shown a decrease in recent years but there has been no fall in the actual number of published articles. Is this because papers are becoming shorter?

Author's Reply

I am sorry if I misled you in my talk. The number of pages published in three-year intervals was used *only* for books (Figures 5 and 15). The other figures for periodicals were the *cumulative* number of articles and thus do not fall but eventually reach saturation. The number of pages is not, however, a good indicator of articles published.

DIFFICULTIES IN PREDICTING AVIONICS RELIABILITY

by

J.E. Green
Royal Signals and Radar Establishment
Great Malvern, U.K.

SUMMARY

Conventional numbers count predictions when viewed in retrospect have often been misleading, and the reasons are considered. Several factors which significantly influence reliability are normally ignored, such as the type of aircraft, the duration and type of sortie, the frequency of use, and the incidence of reported but unconfirmed failures which nevertheless result in maintenance actions.

Failures are non-exponentially distributed during a sortie, although conventional predictions are based on the exponential distribution. Against this difficulty, various proposals for predicting avionics reliability are considered.

The difficulties in making predictions for the latest microelectronic devices are also considered with reference to MIL-HDBK-217B, in particular reviewing the potential reliability of LSI and microprocessors. The significance of the choice of quality factors is noted.

The paper concludes with some thoughts on the future approach to the prediction problem if any credibility is to be retained.

INTRODUCTION

When an avionics reliability prediction is called for, most people immediately think of a numbers counting exercise, assuming a constant failure rate for each type of component and using data from a recognised source such as MIL-HDBK-217B. This type of activity has continued since the earliest days of reliability engineering for electronics, and is dependent on the assumption that the probability of failures during the useful life period of an equipment can be described by an exponential distribution. Over the years there have been many equipments which have failed in service to meet the reliability prediction, often by a substantial amount,[1] and I feel that the time has come to review our whole approach to the subject of predictions with the object of seeking improvements.

Those involved with avionics projects who are not practising reliability engineers often seem to think that predictions are our major preoccupation, which is not true. However, I would suggest that predictions have caused more damage to the status of reliability engineers than any other of their activities. One remembers such frank papers as "The Numbers Game - Who Wins?"[2], and "Reliability Prediction - Help or Hoax?"[3] which appeared a decade ago, yet since then the basic method of making avionic predictions has hardly changed. So, what can we do to improve the situation for future avionics?

Requirements for predictions are not limited to the numbers count procedure of course. Indeed, the first estimate is required at the time of formulating the Staff Requirement when there is no hardware. Then come the definition and design phases, and assembly of the prototype models, when the component populations become known and conventional predictions are made. During development there is a progressively improving situation, and reliability growth modelling techniques are applied to predict the future standards which should be achievable. Then there are formal demonstrations on late development and/or early production models, where the outcome is viewed by the customer as an assurance of the standard which will be realised in service use. It can be regarded as a late prediction, but actual reliability on the aircraft may turn out to be embarrassingly different, and usually lower! That is when the moment of truth has to be faced!

Fig. 1 shows the degree of uncertainty which is associated with reliability predictions made at various stages in an equipment life cycle. Uncertainty tends to increase with equipment complexity and with the introduction of new technology.

Fig. 2 shows the typical way in which the reliability standard improves as the project proceeds. After introduction to service, further improvement continues as a function of user experience, maturing production and often incorporation of late modifications. So, referring back to the numbers count prediction made during the design phase, where was it supposed to apply in the life cycle? In the U.K., avionics predictions normally refer to a time 18 months after first introduction of a new equipment to Service squadrons. As a general guide, by that time it will have settled down in service and the MTBF reported will often approximate to that calculated by reliability growth modelling of development operating experience. The initial MTBF reported in service tends to be only about half that figure.

ADDITIONAL FACTORS AFFECTING AVIONICS RELIABILITY

Although the introduction to MIL-HDBK-217B includes the very true words that "other variants can affect the stated failure rate of a system", in practice most predictions will be limited to quantifying the variables included in the failure rate models for each part of the system. The environment of an avionic installation will be allowed for by a single factor for each part, simply according to whether an

equipment is in an inhabited or an uninhabited region of an aircraft. The factor will be the same irrespective of the type of aircraft and the sortie duration.

Is it reasonable to expect that a given equipment installed in the crew area of a long range Jumbo airliner will have the same failure rate as when it is installed in the cockpit of a short duration high performance military fighter aircraft? I think not, and I feel that few would disagree that additional factors must influence avionics reliability.

Table 1 lists the more important of these factors.

Type of aircraft
Installation environment
Sortie duration
Frequency of operation
Non-exponential failure distribution
Reporting system
Experience and skill of maintenance personnel

TABLE 1. ADDITIONAL FACTORS AFFECTING AVIONICS RELIABILITY

The problem is to obtain sufficient data from a range of different aircraft and equipments which will permit the derivation of a mathematical model allowing for these pertinent factors. In the U.K. we have made some progress in this direction, in fact as far back as 1969 at the NATO Conference on Operations Research and Reliability, Mr R. Chaplin of the U.K. Royal Aircraft Establishment presented a very relevant and interesting paper.[4] He had used regression analysis methods in order to derive K factors according to type of aircraft, and what he called an "equipment characteristic defect rate λ". The latter, although based on the conventional numbers count prediction from parts, using the then current MIL-HDBK-217A was a non-linear function of that calculated prediction. λ became progressively less than the predicted figure as equipment complexity increased.

Chaplin also presented data to show that the probability of failure decreased as a function of elapsed sortie time, and he noted that reliability was influenced by system duty cycle. Indeed most of his deductions are still correct a decade later, yet little attention seems to have been given to them outside the U.K.

Let us examine the factors listed in Table 1 in more detail.

Type of Aircraft and Installation Environment

It is convenient to consider these two factors together. The environmental stress levels which can promote physical and/or chemical degradation processes leading to equipment failure will clearly be a function of the aircraft installation. Although in the conventional prediction an allowance is only made for temperature, and then often only for a constant temperature, in practice there is vibration and perhaps significant levels of acoustic noise, together with humidity and condensation. Ideally, adjustments to the prediction would be made for each of the different kinds of environmental stress, but in reality any adjustments attempted can often only be based on previous experience with a similar class of aircraft and will include a degree of personal judgement. Even then attention has to be given to the experience and anticipated capability of the designers to evolve a design which will afford protection from potentially damaging environmental hazards; perhaps another K factor for quality of design!

Currently we are using a K factor according to type of aircraft and installation. Fig 3 shows the general trend of this factor (K_A) when plotted against the scheduled flying hours per annum of various aircraft. Those with a high schedule are nearly always large aircraft mainly involved on long duration flights, whereas the low schedules are predominantly high performance fighter type aircraft flying relatively short sorties.

Where the K_A factor specified for a particular installation deviates from the graph, there will be logical reasons for the difference, although the actual adjustments will often depend on personal experience with other equipments and aircraft.

For example, a high speed aircraft used for low level operations will tend to have a higher K_A due to buffeting effects on the airframe, a long duration aircraft on a low duty cycle will tend to have a lower K_A on account of longer operating periods, and an electronic installation on an extremity of an aircraft such as the top of the tail fin would tend to have a higher K_A because of high environmental stresses.

With a new aircraft project, decisions about K_A factors will inevitably be tentative, but predictions have to be made, and if attention is drawn to past experience of environmental hazards associated with similar installations then action can be stimulated to investigate and attempt to minimise the risks. It seems paradoxical in the case of MIL-HDBK-217B to use electronic part failure rates which are to several decimal places implying high accuracy, and then only apply relatively coarse factors to allow for the aircraft installation ie Inhabited or Uninhabited.

Turning to the temperature dependence of part failure rates, everyone accepts that at high temperatures failure rates will increase. But in the case of avionics, what about the impact of low temperatures? At increasingly low sub-zero temperatures differential expansion effects will lead to increasingly high mechanical stress levels in parts, and the properties of semiconductors often change to an extent that circuit performance is affected. The risk of failure must increase with increasing mechanical stress and abnormal electronic functioning. Indeed MIL-HDBK-217B devotes one page in capital letters to impress caution;

"EXTRAPOLATION OF ANY OF THE BASE FAILURE RATE MODELS BEYOND THE TABULATED VALUES, SUCH AS HIGH OR SUB-ZERO TEMPERATURE ------ IS COMPLETELY INVALID".

This caution about sub-zero temperatures is nearly always ignored in the case of avionic predictions. Moreover, the contractor is usually called upon to carry out a reliability demonstration involving repeated temperature cycling down to -54°C with cold switch-on.

Fig. 4 illustrates one of our current methods of adjusting the prediction for temperature, using a factor K_T. Failure rates increase at sub-zero and at elevated temperatures, and are at a minimum in the range 0-20°C. In order to allow for temperature changes during a typical sortie, K_T is integrated for the time/temperature profile of the operating period and a mean $\overline{K_T}$ is calculated. Temperature refers to the typical ambient inside the equipment. $\overline{K_T}$ is applied as an overall adjustment for temperature in the prediction, and will apply to all parts operating at not more than specified moderate electrical stress levels.

Sortie Duration

Kujawski and Rypka of Litton Systems[5] studied the relationship between the operating cycle and reliability of four types of solid state electronic equipment and concluded that the observed failure rate was a function of the operating cycle. They proposed that predicted failure rates based on MIL-HDBK-217B should be adjusted by a factor $\frac{3}{\sqrt{T}}$ where T = length of the operating cycle. If the operating cycle approximates to sortie duration, this means that an avionic prediction should include an additional factor of $T^{-0.5}$. Our approach has been to use a K_A factor which is specific to each type of aircraft, and sortie duration was only one of several contributory factors which were considered. Although our figures are derived from empirical data it seems that an adjustment of $T^{-0.5}$ for sortie duration is not unreasonable.

The important point is agreement that the prediction must include some form of adjustment for sortie duration. Refinement of any proposed model can only come from careful study of sound field data.

The evidence indicates that in most cases the contribution to overall failure rate is relatively small for the period when the equipment has warmed up and is operating under fairly stable moderate thermal conditions. Risk of failure is concentrated in the early part of the equipment operating period. This concept is readily understood, and it follows that avionics on short duration aircraft will exhibit a higher overall failure rate.

Frequency of operation

There is abundant evidence that as electronic equipment is operated less frequently, any failures are more likely to occur either at power switch-on or during the early part of the operating period. Attempts to quantify the risks have been directed mainly towards trying to establish dormant failure rates, and for avionics, figures in the range $\frac{1}{30}$ to $\frac{1}{100}$ of assumed constant failure rates when operating are typical. We normally use a figure of $\frac{1}{40}$.

In addition the risk of failure attributed to the action of switching on power has been tackled on the assumption that each switch-on could be equated to a number of operating hours. A U.K. study concluded that the equivalence "varied from 1-10 hours, with a median value of 3 or 4 hours. If this value is taken, it is considered unlikely to be more than a factor of two in error for most military electronic equipment ----". For avionics, a tentative figure of 3 hours has been assumed.

Kujawski and Rypka considered the effects of frequency of operation on equipment reliability and concluded that each switching cycle was equivalent to an average of about 6 hours continuous operation with a standard deviation of 2 hours approximately. In this case the switching cycle covered the actions of switching on and off and the first 10 minutes of operation and presumably all time when the equipment was not energised. Their 6 hour figure is not greatly dissimilar to the figure we would derive by adding failure risks attributable to the dormancy period and to power switch-on. We concur with their observation that identical equipments operating under different "on-off" duty cycles exhibit different failure rates. One U.K. example was a computer which had been operated continuously and was then changed to operating 16 hours per day with close down over week-ends. The overall failure rate then increased by about 20% and the Monday morning situation was often the worst. The operators used the expression "Monday morning sickness" to describe the computer's state!

Non-exponential failure distribution

There are several reasons for expecting a higher probability of failures during the early part of an operating period:-

(a) Any deterioration during the preceding non-operating period will result in particular components being more vulnerable to stresses in the operating mode. Parts may be mechanically stressed during cool down following switch-off, and chemical and physical processes may be active during dormancy as a function of environmental conditions.

(b) During aircraft maintenance, avionic equipment may be subjected to mechanical stresses.

(c) At switch on, and again as other aircraft systems are switched on and tested, high transient electrical stress levels may occur.

(d) During take-off at maximum engine power, higher mechanical stresses will tend to be applied than those occurring during steady flight.

It is virtually impossible to quantify dormant failure rates, but data can be collected on the times after switch-on when failures occur. All the evidence indicates that failure rates decrease as a function of operating time, and various mathematical models have been proposed.

Examples were given by first Peacore[6] and then Shurman[7] with data from the operation of AWACS on Boeing 707-320 aircraft. Fig. 5 is a reproduction with acknowledgement to Peacore, showing the differences in reliability which would be predicted depending on whether the exponential model or the Boeing time-dependent model was used. The corresponding reliabilities for the normal 10 hour flight duration were 0.86 and 0.93. The models predicted the same reliability at 3.5 hours. In terms of failures, the time dependent model predicted only about 27% more failures in 10 hour flights than in 3.5 hour flights. Even more impressive is the result after refuelling in flight, when "the failure rate for the refuelled AWACS flight is typically 1/5 to 1/6 the average failure rate of the flight period prior to refuelling".

Peacore proposed the following reliability model to fit the observed experience with AWACS:-

$$R_{(t)} = \left(\frac{0.08}{t + 0.08}\right)^{14\lambda_o} \quad \ldots\ldots (1)$$

where t was the elapsed sortie time, and λ_o was the equipment failure rate after 14 hours flight when it had become virtually constant.

Three years later, based on more experience and studies, Shurman proposed the following amendment

$$R_{(t)} = \left(1 + \frac{t}{0.08}\right)^{-0.45\lambda_o T} \quad \ldots\ldots (2)$$

where T was the normal sortie duration.

Shurman's model can be rearranged to the same form as Peacore's model, ie

$$R_{(t)} = \left(\frac{0.08}{t + 0.08}\right)^{0.45T\lambda_o} \quad \ldots\ldots (3)$$

If the additional variable T is now substituted to correspond with Peacore's equation ie T = 14, the reliability exponent becomes $6.3\ \lambda_o$, which is different to that in Peacore's model. The explanation is that Shurman's λ_o is the failure rate after flight durations of 0.7T and not 14 hours. Apart from this complication, the models are essentially the same. However it does raise the question as to what flight duration predictions are supposed to refer?

Our approach was to make a prediction λ for the normal sortie duration using our aircraft factor K_A and constant failure rates for parts. Then we applied the following model to predict reliability for various times into the sortie, or differing sortie durations.

$$R = \exp(-\lambda T^{0.7} t^{0.3}) \quad \ldots\ldots (4)$$

where T = normal sortie duration

t = elapsed operating time on sortie

When t is the normal sortie duration, reliability becomes the usual exp $-\lambda T$.

Of course there is a contradiction in applying constant part failure rates to make a prediction, when it is clear that failure rates are not constant during a sortie; but how else can the prediction problem be tackled when all known parts failure rate lists assume the exponential distribution? That has always been the basis of conventional predictions for electronics.

The reporting system

Predictions are compared eventually with the reported failure rates in service, and any differences are highlighted. It is important to review carefully the basis and validity of the reported field experience, because adjustments are often necessary before a true comparison can be made. Considerable errors can occur in the accounting of both failures and time. Kern of Hughes Aircraft[1] re-examined reported MTBFs for 16 avionic equipments, and the revised MTBFs showed increases by factors ranging from 1.2 to 2.7 with an average of twice the reported figures.

Historically most aircraft defect reporting systems were planned to provide logistic support data in terms of airframe flying hours, spares used and maintenance man-hours. However, flying hours are rarely the same as avionic operating hours. For example an equipment is switched on for checks before take-off, and with complex systems this ground running time can be a significant addition to the flying time. In contrast a few equipments are operated only when required during a particular phase of a flight eg during landing approach. Other errors can arise in the reporting system for repair activities, especially if elapsed time indicators are not fitted. Defective equipments will be removed from the aircraft and will

undergo additional operation in workshops during repair and check out, and usually again after re-installation. Amongst equipments with high failure rates, there are examples where this additional ground operating time has exceeded the actual flying time. If flying hours only are counted, and all events which result in consumption of spare parts, then reported failure rates will be misleadingly higher.

The rules for accounting failures are an increasing problem as avionics become more complex. Whereas the failure of a part can be defined in terms of inability to meet specified performance parameters, such an inability does not necessarily result in equipment failure. Moreover some degradation in equipment performance may be tolerated before a failure is declared.

Then there is the well known category of "No fault found" which for some units can be as high as 40%. Aircrew report a real malfunction which is not evident later in the different environmental conditions of the workshop, and despite close examination maintenance personnel are unable to find anything wrong. An equipment in this category can be moved several times between workshops and perhaps different aircraft before the cause of the original failure is diagnosed. Several failures may be counted instead of one, furthermore operating time in workshops may not be counted.

In view of the many criticisms of avionic predictions, it is important from the outset to ensure that the same rules will apply for acccunting of time and failures both for the prediction and for the reporting system. Don't leave it until there is an enquiry into the reasons why the in-service reliability is so disappointingly low. Retrospective adjustments to failure rates reported by the user are likely to lead to further disenchantment with predictions.

Experience and skill of maintenance personnel

The reliability of any electronic equipment which experiences failures will be influenced by the capability of people to identify failed or defective parts, and their skill in restoring the equipment to a fully serviceable state. Unfortunately the complexity of modern equipment is such that slightly defective parts are not always detected. Sometimes these are parts which have suffered a degree of overstress as a secondary effect when other parts failed. In many cases the intricacy of the assemblies makes them vulnerable to inadvertent damage during repair activities. Repeated maintenance work on an equipment certainly lowers its reliability. Kern found that after allowing for discrepancies caused by the reporting system, almost half of the remaining differences between field and either predicted or demonstrated MTBFs were attributable to maintenance handling!

When making a prediction for a new avionic equipment, only a tentative adjustment can be proposed for this factor. It depends on skill and training, and on the accessibility provided by the designers, but an attempt should be made to assign a contingency factor for the impact of maintainability on equipment failure rate. Maybe if Kern's figure quoted above was used, it would motivate those responsible for maintenance to do better!

LSI

It is appropriate to review briefly the current situation with predictions for integrated circuits, because they form an increasingly important part of avionics. MIL-HDBK-217B is the recognised reference document used when predicting failure rates for LSI devices, memories, microprocessors, etc., and most reliability engineers will be familiar with the mathematical models. Few who have used them will not have reflected on the contrast between the implied accuracy in some of the figures which are given to several decimal places, and the compounding multipliers for quality which are quite coarse and can range from 1 to 150; also if the device is a newly introduced production item a further multiplier of 10 is applied.

There is certainly a problem in deciding how best to deal with the quality factor. The manufacturing processes require an extremely high level of quality control before there is any economic yield from the production line. In general as production experience accumulates, the product improves, and any risk of failures due to initial manufacturing lapses lies mainly with the occasional sub-standard batch. Given a long production run, most delivered silicon devices are potentially failure free, and the future reliability prospects are extremely good. The quality factors attempt to reflect the extent of screening.

Circuit complexity was an important influence on failure rate in the past, but it has rapidly diminished. Successive amendments to MIL-HDBK-217B have introduced changes to the adjustment factor for complexity in an attempt to keep pace with the continually improving reliability standards. But progress in technology is so rapid that today even predictions made in accordance with the 1978 amendment seem rather high for devices with many thousands of gates. Mr P.D.T. O'Connor has studied the problem and in Paper 1-5 will be presenting a new model for complexity based on the square root of the number of gates.

At present I feel that reliability predictions for LSI devices are still very tentative and that further changes to the models are inevitable as even greater complexities are achieved and new technologies are introduced. The extensive effort by RADC to evolve suitable models has been most commendable. For avionics it is rather fortunate that LSI devices have in most cases lived up to their initial promise and made so little contribution to overall failure rates.

FUTURE TRENDS

At the root of the problem of making realistic predictions for modern electronics is the fact that the potential failure rates of the parts have progressively reduced to the point where they are extremely low, approaching zero, and we now have units of 10^{-9} failures per hour. At this level any occasional lapse in quality control during manufacture, or in equipment design application, or any overstress at

any time, can lead to observed failure rates which are orders higher. Therefore factors other than the potential failure rates of the electronic components dominate the reliability experience with the equipment, and this is particularly so in the case of military avionics.

Looking into the future, the situation seems unlikely to change except that the potential failure rates of the electronic components will become even lower. If predictions are to retain any credibility, more attention will have to be given to quantifying the external factors which influence reliability at equipment level. Even then, it will be advisable to emphasise the degree of uncertainty attached to predictions, and not present figures at equipment level which imply unjustified high accuracy.

Emphasis on predictions may well begin to move away from the traditional numbers count procedure in the design phase, towards predicting from reliability growth modelling of operating experience during the development phase. Greater attention is likely to be given to long term operation of prototype avionics in combined environmental test facilities. These will endeavour to simulate conditions monitored at the aircraft installation during varying types of sortie and when on the ground.

CONCLUSIONS

1 Most avionic reliability predictions have been inaccurate, and factors other than listed part failure rates have a greater influence on observed equipment failure rates in service.

2 For a given equipment, failure rate is a function of the type of aircraft and installation, and not solely whether the installation is in either an inhabited or uninhabited region.

3 There is little doubt that failures are not exponentially distributed in time. Failure rates are dependent on elapsed time (t) following switch on, and decrease as the sortie progresses. A provisional equipment reliability model is of the form $R = \exp(-C\lambda t^{0.3})$.

4 For a given sortie duration, if the aircraft is flown more frequently, the observed failure rates will tend to decrease.

5 Retrospective comparison of predictions with in-service experience is complicated by incompatibilities of the reporting system.

6 The potential reliability of LSI is undoubtedly remarkably high, but failure rate prediction is as yet a doubtful procedure.

7 In the future, if avionics predictions are to retain any credibility, the traditional approach will have to be modified.

REFERENCES

1 G.A. Kern, Operational Influences on Avionics Reliability, Proc 1978 Annual Reliability and Maintainability Symposium.

2 J.B. Brauer, The Numbers Game - Who Wins?, Proc 1967 Annual Symposium on Reliability.

3 E.O. Codier, Reliability Prediction - Help or Hoax?, Proc 1969 Annual Symposium on Reliability.

4 R. Chaplin, Reliability Prediction in Cost Effectiveness Analyses, Proc 1969 NATO Conference on Operations Research and Reliability.

5 G.F. Kujawski and E.A. Rypka, Effects of "On-Off" Cycling on Equipment Reliability, Proc 1978 Annual Reliability and Maintainability Symposium.

6 E.J. Peacore, Reliability Developments - AWACS, Proc 1975 Annual Reliability and Maintainability Symposium.

7 M.B. Shurman, Time-Dependent Failure Rates for Jet Aircraft, Proc 1978 Annual Reliability and Maintainability Symposium.

Copyright © Controller HMSO, London, 1979.

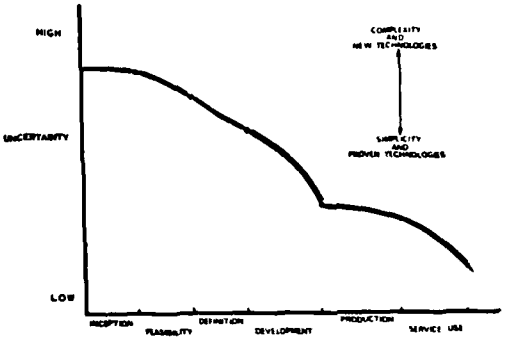


FIG. 1 **DEGREE OF UNCERTAINTY IN ESTIMATING EVENTUAL RELIABILITY**

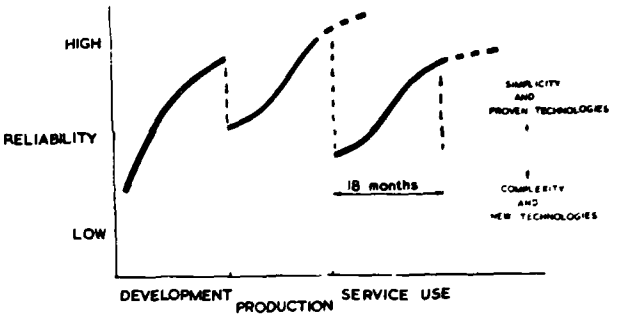


FIG. 2 RELIABILITY IMPROVEMENT

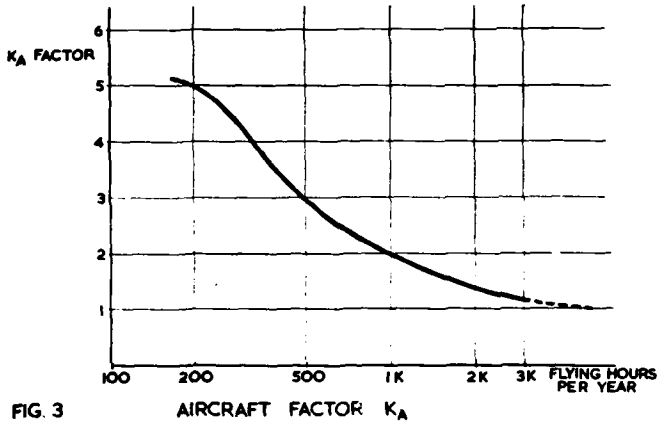


FIG. 3 AIRCRAFT FACTOR K_A

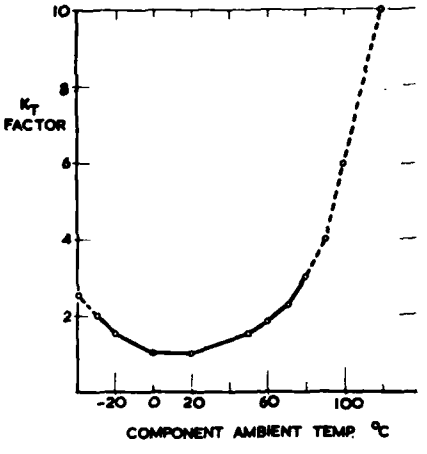


FIG. 4 TEMPERATURE FACTOR K_T

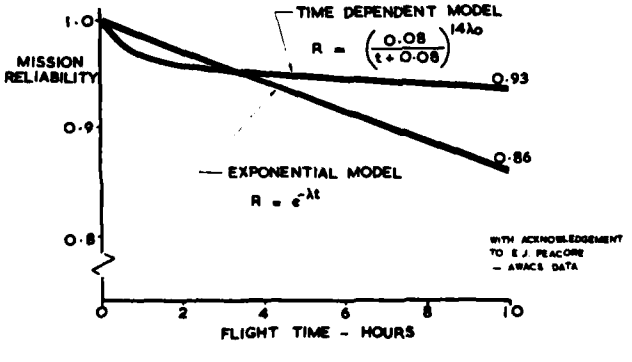


FIG. 5 RELIABILITY MODEL COMPARISON

RELIABILITY GROWTH MODELS

W. M. Woods

Naval Postgraduate School
Monterey, California, USA

SUMMARY

The purpose of this paper is to introduce the concept of reliability growth models, indicate how they are used in the system acquisition cycle, and show how they can be evaluated for accuracy. Two reliability growth models are presented - one for time data and one for attributes data. Their uses are discussed and the methods for evaluating them are presented in graphical and tabular form. Both models show reasonable accuracy for reasonable amounts of testing under a wide variety of actual reliability growth and non-growth.

1. INTRODUCTION

Newly designed components of commercial and military systems frequently display low initial reliability in early development. Engineering evaluation tests early in the development phase hopefully result in design changes that improve this low initial reliability. Subsequent test programs such as qualification test and subsystem performance test may result in additional changes in design, fabrication, and production that provide additional reliability growth. Field test may generate additional changes that initiate still further growth in reliability. The growth that occurs in component reliability is due to changes in hardware or in a process. Reliability growth models are designed to estimate this rate of growth. They can provide an impartial measure of growth or non-growth in reliability during all or part of the acquisition cycle.

Reliability growth models should be used as a management tool. They are intended to assess reliability trends rather than precisely measure current reliability. For this reason, they are not the best device to assess or demonstrate reliability against a contractual goal. They should be used as an unbiased source of information to detect and track reliability growth and to point out problem components.

Since the underlying reliability progress pattern may vary greatly from one component to another, useful reliability growth models should be able to accomodate widely different "growth" patterns with a reasonable degree of accuracy. Since components differ in type with respect to operating characteristics; e.g., cycle type vs continuously operating type, different reliability growth models are needed to be compatible with the type of hardware and corresponding test programs.

In this paper, a reliability growth model based on time data and a discrete reliability growth model based on attributes data are presented and evaluated. Their applications are discussed and quantitative measures of their accuracy are provided. Their accuracy is displayed not only for "nice" underlying reliability progress paths; but also for underlying patterns that exhibit reliability stagnation and degradation prior to continued growth and patterns demonstrating no reliability progress.

The two reliability growth models presented can be employed for reliability tracking at the system level, sub-system level, and/or the individual component level during any phase of any acquisition cycle including retrofit programs. "Component" and "item" are used to reference an entire system, or a sub-system, or an individual component.

2. TWO RELIABILITY GROWTH MODELS

2.1 CONTINUOUS RELIABILITY GROWTH MODEL

The continuous reliability growth model presented here assumes an exponential failure distribution. It provides failure rate estimates of each of several versions of a component which evolve from changes during its development process. The symbol λ_i denotes the failure rate per specified time unit for the i^{th} version of the component or correspondingly during the i^{th} testing phase. The phases are determined by decisions or changes in the acquisition process that are believed to affect reliability significantly. These phases can be determined by the user as the acquisition process develops. The data needed to employ the model are the number N_i of components tested in phase i, total accumulated test time T_i in phase i, total # of failures F_i observed in phase i, and the number of test phases K or changes in the acquisition process to date.

It should be noted that λ_i denotes failure rate in the same time units as T_i. The model for the failure rate λ_{TT_K} after a total of TT_K units of time have been accumulated over K phases is

$$\lambda_{TT_K} = (1-a)b(TT_K)^{-a} \qquad (2.1)$$

where $TT_K = T_1 + T_2 + \ldots + T_K$. That is, equation (2.1) is used to model the growth curve established by the true values $\lambda_1,\lambda_2,\lambda_3,\ldots,\lambda_K$. Estimates of the parameters a and b in equation (2.1) are updated each time a new phase is entered. This can be at any arbitrary point in accumulated test time TT_K. That is, at any time point TT_K of accumulated test time, the parameters a and b are reestimated from the observed total test times $T_1,T_2,\ldots,T_K$ and the observed number of failures $F_1,F_2,\ldots,F_K$ in each of the observed K phases. Any test plan may be used for this model. The one for which the accuracy evaluations

were performed in this paper is truncated in time. That is, each item tested in any phase is tested until failure or until a predetermined test time which may be unique for each item.

The procedure for obtaining current estimates $\hat{a}_K$ and $\hat{b}_K$ for a and b respectively in the K^{th} phase uses the nearly unbiased modification of the maximum likelihood estimate $\hat{\lambda}_K$ for λ_K defined by

$$\hat{\lambda}_K = \begin{cases} \frac{2N_K}{1+2N_K} \cdot \frac{F_K}{T_K} & \text{if } F_K > 0 \\ \frac{2N_K}{1+2N_K} \cdot \frac{1/2}{T_K} & \text{if } F_K = 0 \end{cases} \qquad (2.2)$$

It also uses all corresponding estimates $\hat{\lambda}_1,\ldots,\hat{\lambda}_{K-1}$ for the previous K-1 phases. The update estimate $\hat{\lambda}_{TT_K}$ for λ_{TT_K} at time TT_K in the K^{th} phase is

$$\hat{\lambda}_{TT_K} = (1-\hat{a}_K)\hat{b}_K(TT_K)^{-\hat{a}_K} \qquad (2.3)$$

To obtain the ordinary least squares regression estimates $\hat{a}_K$ and $\hat{b}_K$ for a and b at the end of the K^{th} phase, the data pairs $(\ln \hat{\lambda}_i, \ln TT_i)$ are employed as follows: Let $Y_i = \ln \hat{\lambda}_i$, $X_i = \ln TT_i$, $\overline{Y}_K = (Y_1+Y_2+\ldots+Y_K)/K$ and $\overline{X}_K = (X_1 + X_2 + \ldots + X_K)/K$ for $i = 1,2,\ldots$ and $K = 1,2,\ldots$.

Then

$$\hat{a}_K = \frac{\sum_{j=1}^{K} X_jY_j - \overline{Y}_K \sum_{j=1}^{K} X_j}{\overline{X}_K \sum_{j=1}^{K} X_j - \sum_{j=1}^{K} X_j^2} \quad \text{and} \qquad (2.4)$$

$$\hat{b}_K = \frac{1}{1-\hat{a}_K} \exp(\overline{Y}_K + \hat{a}_K\overline{X}_K) \qquad (2.5)$$

for $K = 2,3,\ldots$. Note that the regression methods requires observations on the results of two test phases; thus, model parameter estimates are made for the second thru the K^{th} test phase. The instantaneous failure rate estimate given by equation 2.2 for $K = 1$ is used for the first phase of testing.

The ability of this model to track actual reliability growth patterns (i.e., decreasing failure patterns) was evaluated by computer simulation. The results are provided in Appendix A. A wide variety of growth patterns were used. Failure rate values were used that are indicative of early development. The sample sizes used were quite small in many of the cases simulated. In nearly all cases the estimated growth curve provided a reasonable track at the actual failure rate curve. As the amount of testing increases, the estimated growth curve approaches the true growth curve. The accuracy of the model can be visually assessed by examining the figures in Appendix A that show the true growth curve and the estimated growth curve obtained from the model.

Another characteristic that depicts the model's accuracy is the standard deviation $s_{\hat{\lambda}_{TT_K}}$ of the failure rate estimates $\hat{\lambda}_{TT_K}$ obtained in the simulations. A normalized version of this is given in Table 3.2 and provides an indication of how likely the estimated value of $\hat{\lambda}_{TT_K}$ is to stray from the true λ value on any <u>one</u> computation.

2.2 DISCRETE RELIABILITY GROWTH MODEL

The discrete model provides reliability estimates of each of several versions of a component which evolved from changes during its development process. The reliability growth pattern being estimated is the growth in the probability of passing a designated test. If these are mission environmental tests then the model provides estimates of mission reliability. If the environmental levels do not approximate mission environments, then it estimates something other than mission reliability.

The model assumes that a type of component will see several phases of improvement during that portion of its life cycle to which the model is being applied. R_K denotes reliability in the K^{th} phase or modification to the type of component. Specifically, the model is

$$R_K = 1 - \exp\{-(\alpha + \beta K)\} \qquad (2.8)$$

for $K = 0,1,2,\ldots$. The K=0 phase denotes the phase prior to any modification. The model assumes testing is performed in each phase until a specified number of failures are observed at which time a change may be made. No assumption is made about the distribution of the time to failure. Only attributes data are used with this model.

The model provides a current estimate of component reliability at any point in the testing program. It uses past estimated values and current test data to obtain estimates $\hat{\alpha}$ and $\hat{\beta}$ in the model. This in turn provides an estimate of current reliability.

The estimates $\hat{\alpha}_K$ and $\hat{\beta}_K$ for α and β at the end of the K^{th} phase are obtained using linear regression methods and an unbiased estimator for $(\alpha+\beta K)$. The data collected during testing in the K^{th} phase is the following: Let F_K = the total number of failures during the K^{th} phase, and $N_{j,K}$ = the number of tests between the $(j-1)^{st}$ failure and the j^{th} failure, including the j^{th} failure, in the K^{th} phase, $j = 1,2,\ldots,F_K$.

An unbiased estimator Y_{jK} of the quantity $(\alpha+\beta K)$ using the j^{th} sequence of tests in phase K is given by (Chernoff and Woods, 1965):

$$Y_{jK} \equiv \widehat{(\alpha+\beta K)}_{jK} = \begin{cases} 0 & \text{if } N_{j,K} = 1 \text{ (first test was failure)} \\ 1 + \frac{1}{2} + \ldots + \frac{1}{N_{j,K}-1} & \text{if } N_{j,K} \geq 2 \end{cases} \tag{2.9}$$

for $K = 0,1,2,\ldots$ and $j = 1,2,3,\ldots,F_K$. Since $N_{1,K},N_{2,K},\ldots,N_{F_K,K}$ are independent random variables, then $\bar{Y}_K = (Y_{1K} + Y_{2K} + \ldots + Y_{F_K K})/F_K$ is unbiased. Then least squares estimates $\hat{\alpha}_K$ and $\hat{\beta}_K$ for α and β at the K^{th} phase are

$$\hat{\beta}_K = \frac{\sum_{j=0}^{K} (j - \bar{K})\, \bar{Y}_j}{\sum_{j=0}^{K} (j - \bar{K})^2} \quad \text{and} \tag{2.10}$$

$$\hat{\alpha}_K = \bar{Y} - \hat{\beta}_K \bar{K} \tag{2.11}$$

for $K = 1,2,3,\ldots,$ where $\bar{Y} = (Y_0 + Y_1 + \ldots + Y_K)/(K+1)$ and $\bar{K} = (1 + 2 + \ldots + K)/(K+1)$.

Finally, these $\hat{\alpha}_K$ and $\hat{\beta}_K$ estimates are utilized in the discrete reliability model, equation 2.8, to produce sequentially the model estimates of the modified component reliabilities $R_1,R_2,\ldots$ from the equation

$$\hat{R}_K = 1 - \exp\{-(\hat{\alpha}_K+\hat{\beta}_K K)\} \tag{2.12}$$

for $K = 1,2,3,\ldots$. Note that since the regression procedure requires a minimum of two observations, model reliability estimates are produced from the first modification thru the last modification. A reliability estimate for original version of the component is

$$\hat{R}_0 = 1 - \exp\{-\bar{Y}_0\} \tag{2.13}$$

3. EVALUATION METHOD

Performance evaluation of the continuous failure rate reliability growth models and the discrete reliability growth model was accomplished using Monte Carlo simulation. Computer simulation permits the analyst to specify and control the underlying reliability path of growth. To evaluate the accuracy of a proposed reliability growth estimation model against a given growth path, the following steps are followed:

1) Specify the sequence of reliability values (or failure rate values) that reflect the desired growth path.
2) Specify other parameters such as sample sizes and number of failures before a fix in the test procedure.
3) Generate the test data via computer simulation and compute the reliability estimates using the proposed growth method for each phase.
4) Compare the reliability estimates obtained from the model with the actual values specified in 1) and analyze their differences.
5) Repeat 2), 3), 4) for different parameter sets and test plans to determine behavior of the model.
6) Repeat 1) through 5) for different reliability growth paths.

In this paper, to assess the models' accuracies, the simulations were replicated one-hundred times for each underlying reliability progress path specified. The average values (arithmetic means) of the models' estimates of the parameter utilized in the characterization of the underlying reliability progress path were computed to provide a measure of the models' accuracy. Average values of reliability or failure rate estimates were computed at the end of a test phase. For each particular reliability growth model, test structure, and specified underlying reliability progress path, a graph was prepared depicting the true underlying path and the model's mean estimated value for the reliability or failure rate at the end of each test phase. A selection of these graphs is provided in Appendix A.

Managers are concerned with only one "replication" of the reliability progress path estimation. Given that the mean performance of a reliability growth model is satisfactory, managers need to know if the model can

be trusted to deliver satisfactory performance on that single "replication". This concern equates to the question of variability (precision) in the model estimates; i.e., does the model deliver "tight groups" around the mean values of its estimates? To measure variability, the standard deviation of each model's reliability or failure rate estimates from the mean value performance was computed for all phases of all simulations. Variability performance of the reliability growth models is presented in tabular form as a percentage standard error (PSE).

Formulae for the sample (100 replications) mean, sample standard deviation, and percentage standard error (PSE) for the continuous reliability growth model are given below where NSIMS = 100 = number of replications.

$$\bar{\hat{\lambda}}_{TT_K} = \frac{1}{NSIMS} \sum_{r=1}^{NSIMS} \hat{\lambda}_{TT_{K,r}} \tag{3.1}$$

$$S.D.\hat{\lambda}_{TT_K} = \sqrt{\frac{1}{NSIMS-1} \sum_{r=1}^{NSIMS} (\hat{\lambda}_{TT_{K,r}} - \bar{\hat{\lambda}}_{TT_K})^2} \tag{3.2}$$

$$P.S.E.\hat{\lambda}_{TT_K} = \frac{S.D.\hat{\lambda}_{TT_K}}{\bar{\lambda}_{TT_K}} \times 100 \tag{3.3}$$

Similar estimates in the discrete reliabiiity growth model are as follows:

$$\bar{\hat{R}}_K = \frac{1}{NSIMS} \sum_{r=1}^{NSIMS} \hat{R}_{K,r} \tag{3.4}$$

$$S.D.\hat{R}_K = \sqrt{\frac{1}{NSIMS-1} \sum_{r=1}^{NSIMS} (\hat{R}_{K,r} - \bar{\hat{R}}_K)^2} \quad \text{and} \tag{3.5}$$

$$P.S.E.\hat{R}_K = \frac{S.D.\hat{R}_K}{\bar{\hat{R}}_K} \times 100 \tag{3.6}$$

for $K = 0,1,2,\ldots$.

Figures 3.42 thru 3.77 in Appendix A depict the accuracy of the continuous instantaneous failure rate reliability growth model for the failure rate (lambda) sets listed in Table 3.1. In each graph the specified underlying instantaneous failure rate λ_K from Table 3.1 (———, solid line) and the mean model determined instantaneous failure rate $\bar{\hat{\lambda}}_{TT_K}$ from equation 2.3 (○, circles) are all plotted against the mean total accumulated test time TT_K for each phase of the specified reliability testing procedure. These graphs were taken from Neil, 1978.

The variability performance of the continuous instantaneous failure rate model is presented in Table 3.1. Entries in these tables are the percentage standard errors as computed in equations 3.3.

Figures 3.79 thru 3.98 in Appendix A present the ability (for the ten reliability growth sets) of the discrete reliability growth model to estimate the reliability progress paths in Table 3.3 under a testing procedure where a change is made after F_K failures are observed for F_K = 1 and 5. These graphs were selected from Neil, 1978.

The graphs depict the specified true underlying reliability R_K progress path (———, solid line) and the mean model determined reliability. $\bar{\hat{R}}_K$ from equation 3.4 for each modification version of the component under test (○, circles) plotted versus the modification number $K = 0,1,2,3,4,5$. Note that the point plotted for the mean reliability of the original version (K = 0) of the component under test is not model determined; rather, it is the mean value of the estimate given by the estimator of equation 2.13. This point allows the accuracy of the reliability estimator utilized to be examined also.

Table 3.4 presents the discrete reliability growth model's variability performance for determining the reliability status of a component as it undergoes modification in a system acquisition cycle. Entries in the tables are percentage standard errors as computed in equation 3.6.

TABLE 3.1

LAMBDA SETS
CONTINUOUS RELIABILITY GROWTH MODELS

	LAMBDA SET								
PHASE	1	8	10	12	14	MOD 3	MOD 8	MOD10	MOD14
1	.7020	.7000	.7000	.7000	.7000	.7000	.7000	.7000	.7000
2	.4340	.7000	.0500	.5500	.4500	.1800	.7000	.0500	.2250
3	.3200	.7000	.0500	.4250	.3000	.1060	.7000	.0500	.3000
4	.2550	.7000	.0500	.4050	.2250	.0760	.7000	.0500	.5500
5	.2130	.7000	.0500	.4000	.2000	.0600	.7000	.0500	.6100
6	.1830	.7000	.0500	.4000	.2250	.0500	.7000	.0500	.0500
7	.1610	.7000	.0500	.400	.3000				
8	.1440	.7000	.0500	.4000	.4000				
9	.1300	.7000	.0500	.4000	.4750				
10	.1180	.7000	.0500	.4000	.5500				
11	.1090	.7000	.0500	.4000	.6100				
12	.1010	.7000	.0500	.4000	.6250				
13	.0936	.7000	.0500	.3750	.6100				
14	.0876	.7000	.0500	.2000	.5500				
15	.0823	.7000	.0500	.1000	.3000				
16	.0776	.7000	.0500	.0500	.0500				

TABLE 3.2

INSTANTANEOUS RELIABILITY GROWTH MODEL VARIABILITY PERFORMANCE ($PSE\hat{\lambda}_{TT_K}$)
FIRST ENTRY FOR 5 TESTS/PHASE, SECOND ENTRY FOR 20 TESTS/PHASE

	LAMBDA SET															
PHASE	1		8		10		12		14		MOD3		MOD8		MOD10		MOD14	
1	62	42	62	42	62	42	62	42	62	42	65	37	65	37	65	37	65	37
2	76	45	81	40	96	42	79	40	76	41	76	44	76	48	90	43	78	44
3	67	36	76	43	62	29	66	38	64	38	67	39	81	43	62	35	56	30
4	68	36	99	47	56	26	84	33	69	38	58	33	69	43	54	27	50	27
5	49	31	86	37	40	22	60	26	50	30	53	30	83	44	46	24	58	27
6	47	29	77	48	36	22	49	25	42	24	52	23	88	45	44	19	77	37
7	47	25	72	40	38	20	47	22	38	20								
8	45	26	81	40	35	19	48	22	38	19								
9	43	23	84	43	34	16	48	20	42	22								
10	40	22	78	43	29	17	45	21	40	26								
11	38	22	82	44	31	15	54	18	46	22								
12	41	20	100	42	36	16	72	20	58	22								
13	38	22	126	43	38	16	91	22	54	24								
14	32	19	88	37	25	15	48	34	36	17								
15	33	19	75	45	26	15	63	37	31	17								
16	35	18	99	38	32	15	60	33	72	34								

TABLE 3.3

RELIABILITY SETS

DISCRETE RELIABILITY GROWTH MODEL

RELIABILITY SET

PHASE	1	4	7	8	9	10
0	.200	.200	.200	.200	.200	.200
1	.925	.550	.225	.500	.650	.200
2	.950	.750	.275	.550	.650	.200
3	.950	.875	.350	.550	.350	.200
4	.950	.925	.475	.700	.600	.200
5	.950	.950	.950	.950	.950	.200

TABLE 3.4

DISCRETE RELIABILITY GROWTH MODEL VARIABILITY PERFORMANCE ($\hat{PSER}_K$)

1 FAILURE/PHASE, 5 FAILURES/PHASE

RELIABILITY SET

PHASE	1		4		7		8		9		10	
0	231	79	195	71	170	87	161	73	170	80	184	82
1	32	5	100	34	166	69	98	42	69	22	195	79
2	12	2	53	12	170	65	76	26	47	18	214	81
3	8	1	26	6	116	42	62	21	49	21	213	75
4	4	1	16	3	84	33	38	17	48	17	172	76
5	4	1	6	2	19	6	16	4	21	4	137	65

BIBLIOGRAPHY

1. Chernoff, H. and Woods, W. Max, 1962, "Reliability Growth Models - Analysis and Applications", CEIR, Inc.

2. Codier, E. O., 1968, "Reliability Growth in Real Life", IEEE Proceeding, 1968 Annual Symposium on Reliability.

3. W. J. Corcoran and R. R. Read, 1967, "Comparison of Some Reliability Growth Estimation and Prediction Schemes", United Technology Center Report Addendum UTC2140.

4. Neal, R. O., 1978, "An Evaluation of Three Reliability Growth Models", Naval Postgraduate School Thesis.

APPENDIX A

PERFORMANCE GRAPHS

FOR

CONTINUOUS RELIABILITY GROWTH MODEL

AND

DISCRETE RELIABILITY GROWTH MODEL

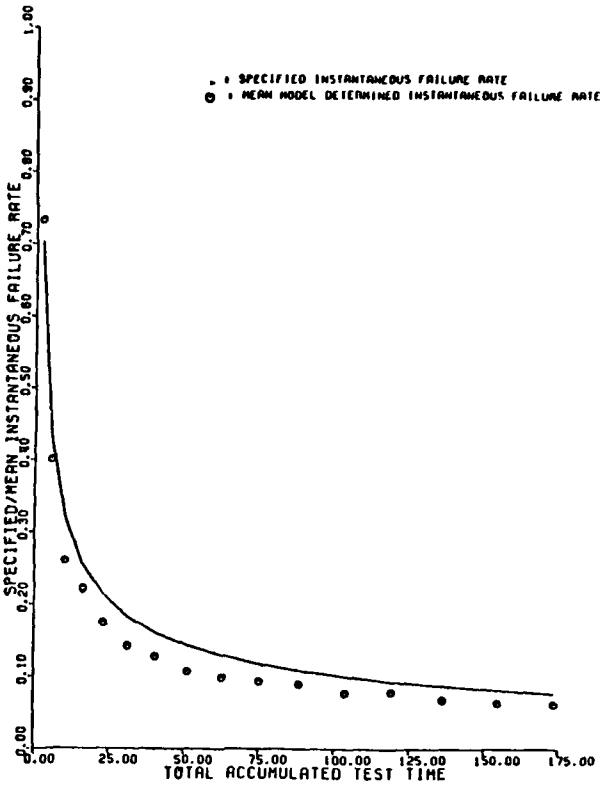


Fig.3.42 Instantaneous reliability growth model performance lambda set 1: 16 phases, 5 tests/phase

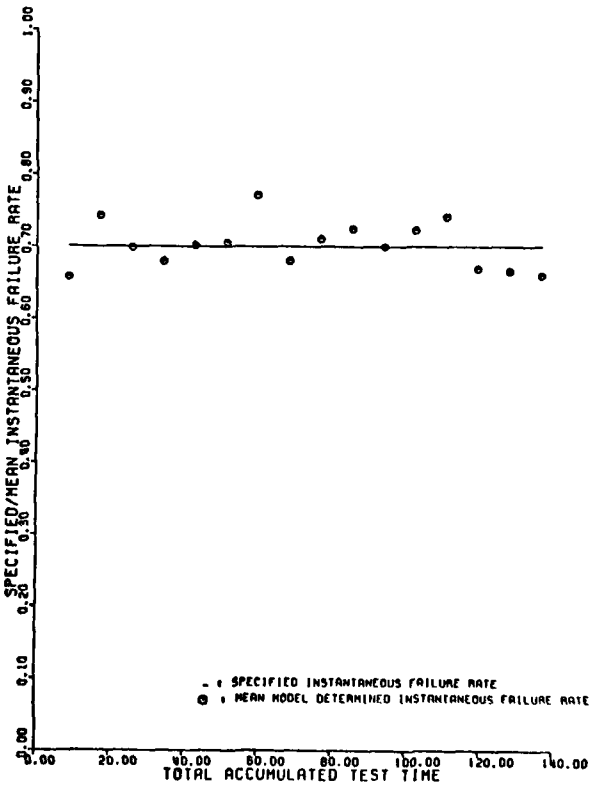


Fig.3.49 Instantaneous reliability growth model performance lambda set 8: 16 phases, 20 tests/phase

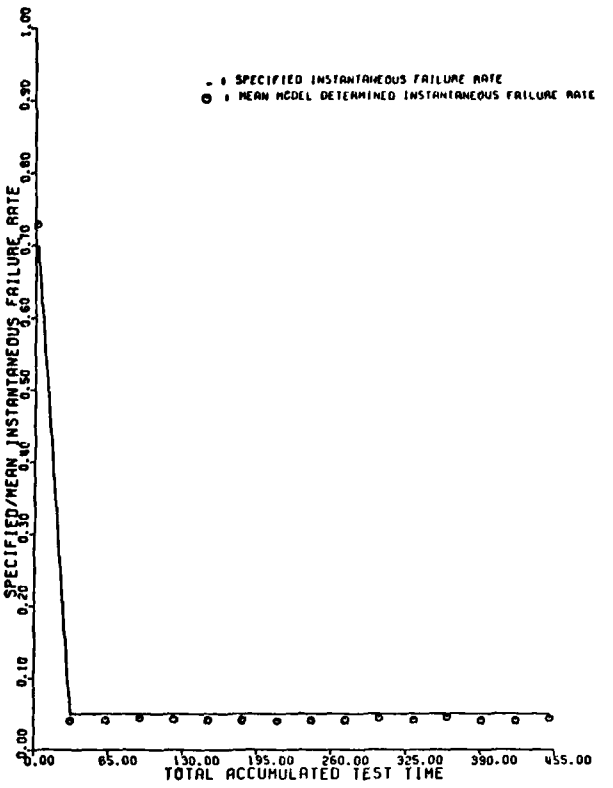


Fig.3.52 Instantaneous reliability growth model performance lambda set 10: 16 phases, 5 tests/phase

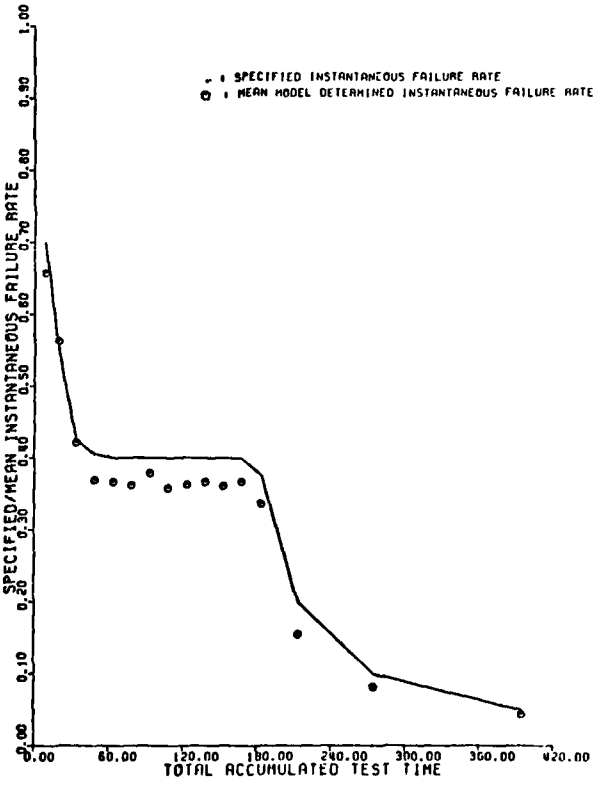


Fig3.54 Instantaneous reliability growth model performance lambda set 12: 16 phases, 20 tests/phase

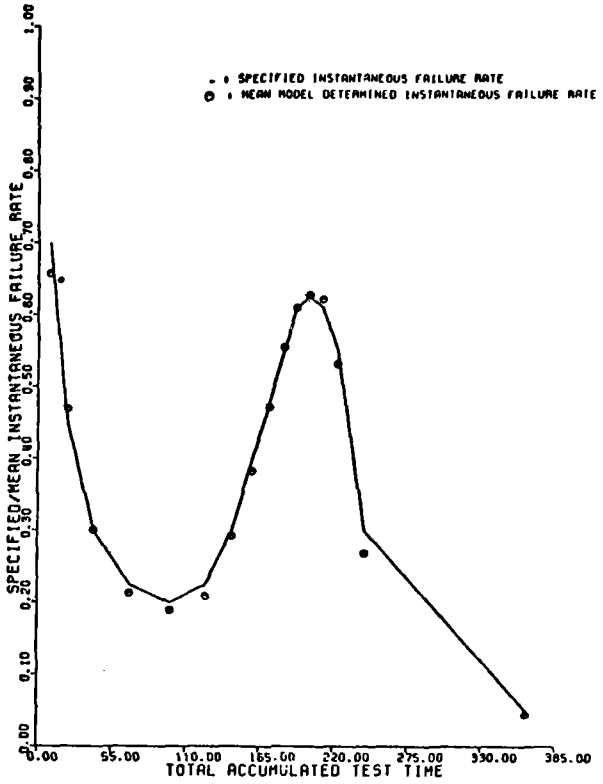


Fig.3.59 Instantaneous reliability growth model performance lambda set 14: 16 phases, 20 tests/phase

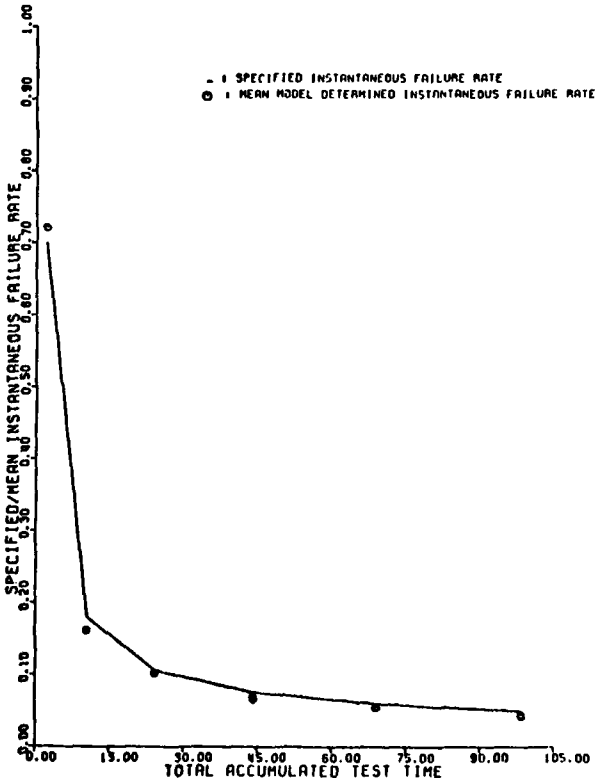


Fig.3.62 Instantaneous reliability growth model performance lambda set mod3: 6 phases, 5 tests/phase

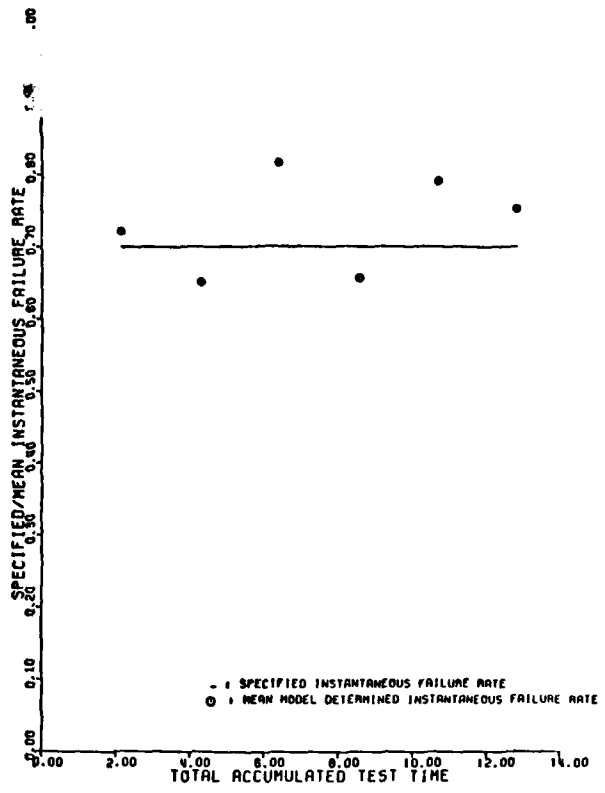


Fig.3.68 Instantaneous reliability growth model performance lambda set mod8: 6 phases, 5 tests/phase

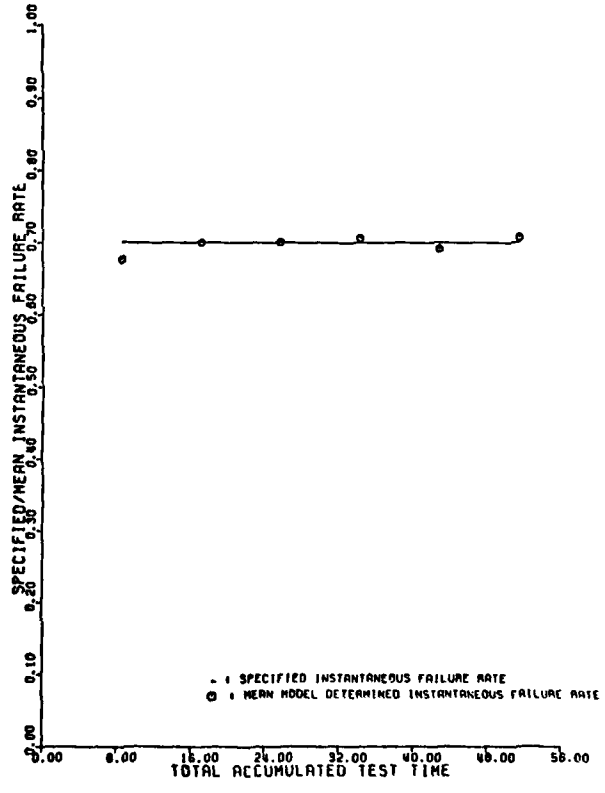


Fig.3.69 Instantaneous reliability growth model performance lambda set mod8: 6 phases, 20 tests/phase

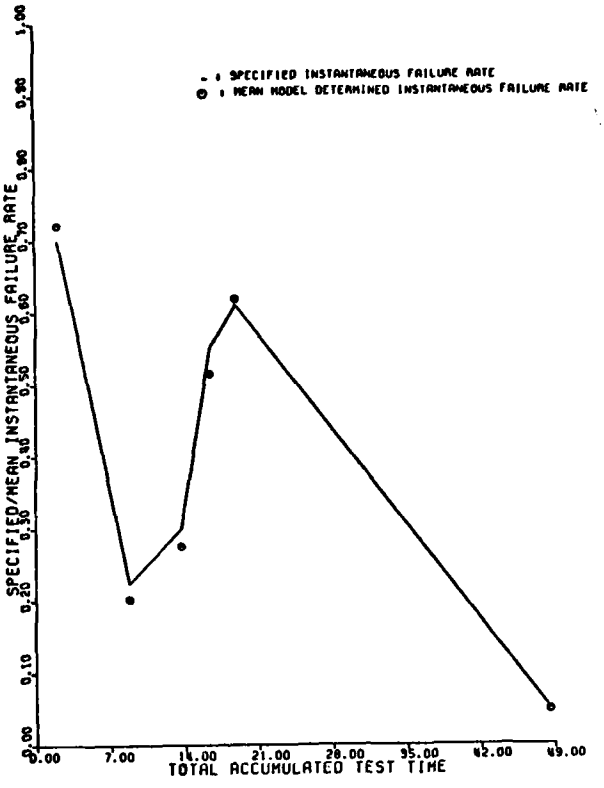


Fig.3.71 Instantaneous reliability growth model performance lambda set mod10: 6 phases, 5 tests/phase

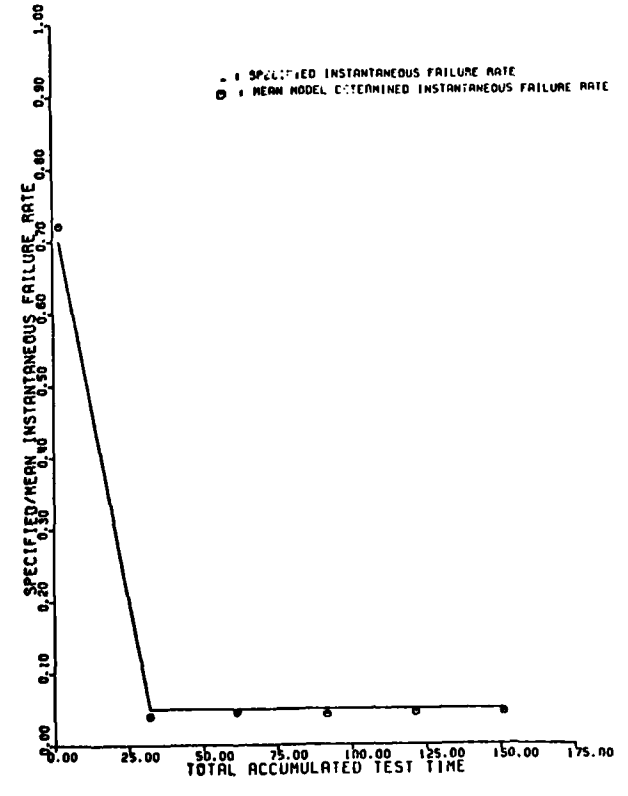


Fig.3.76 Instantaneous reliability growth model performance lambda set mod14: 6 phases, 5 tests/phase

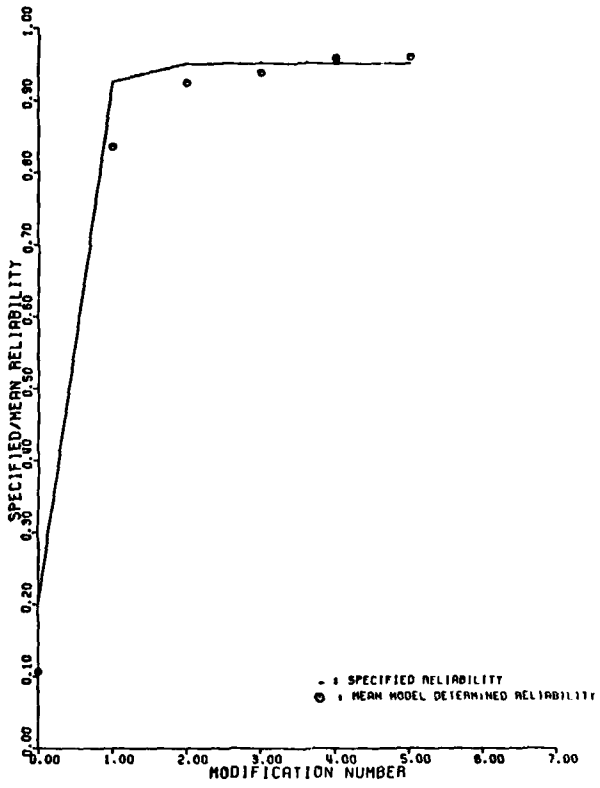


Fig.3.79 Discrete reliability growth model performance reliability set 1: 5 mods, 1 failure/mod

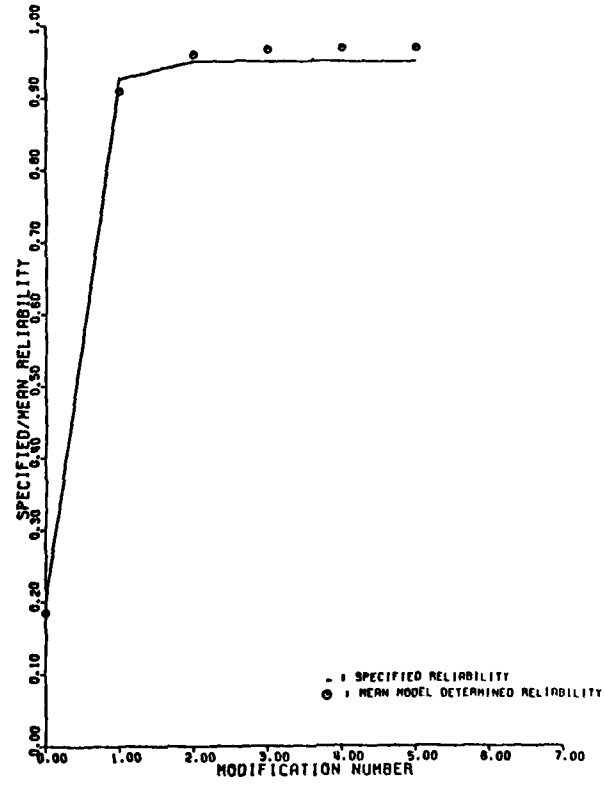


Fig.3.80 Discrete reliability growth model performance reliability set 1: 5 mods, 5 failures/mod

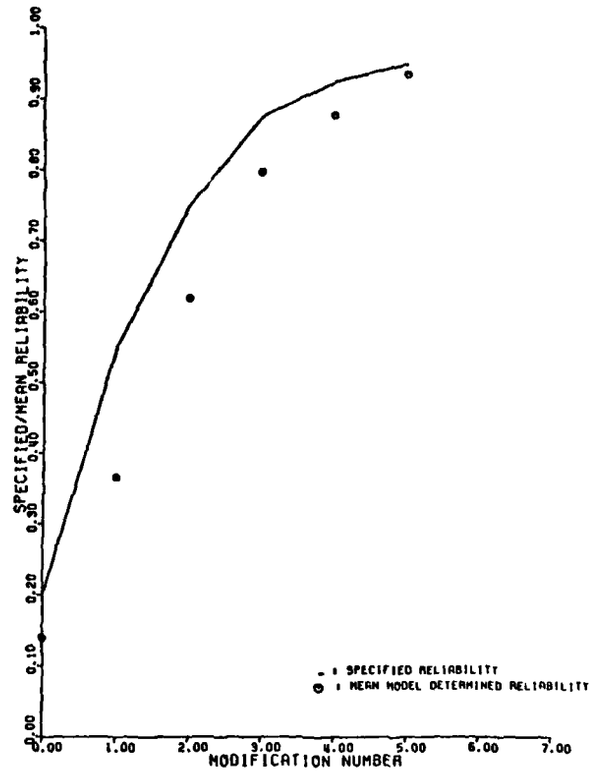


Fig.3.85 Discrete reliability growth model performance reliability set 4: 5 mods, 1 failure /mod

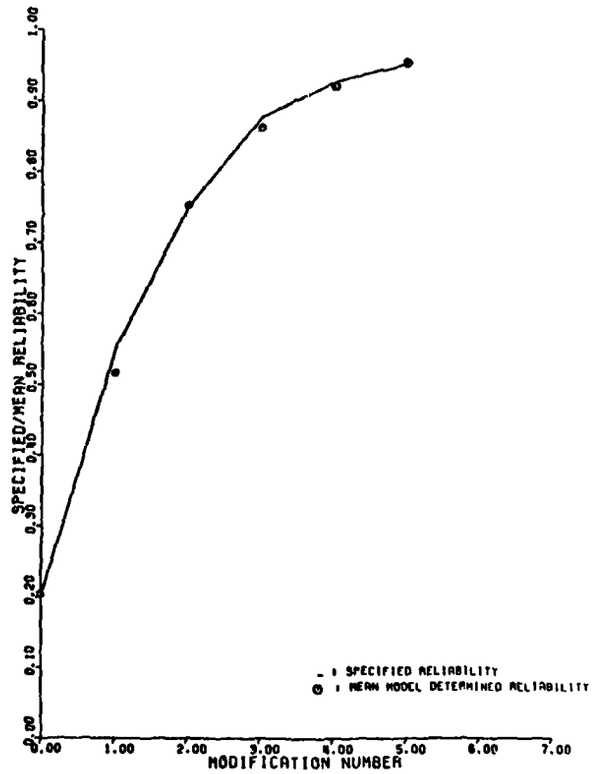


Fig.3.86 Discrete reliability growth model performance reliability set 4: 5 mods, 5 failures/mod

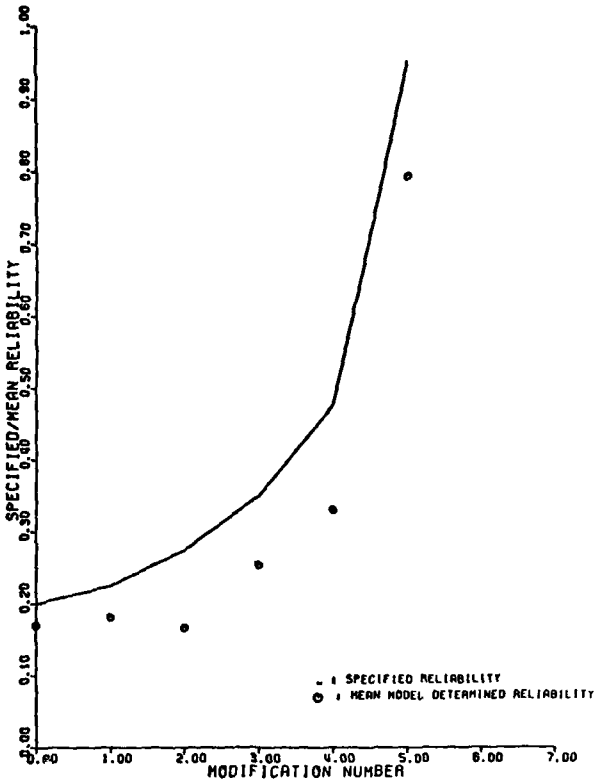


Fig.3.91 Discrete reliability growth model performance reliability set 7: 5 mods, 1 failure/mod

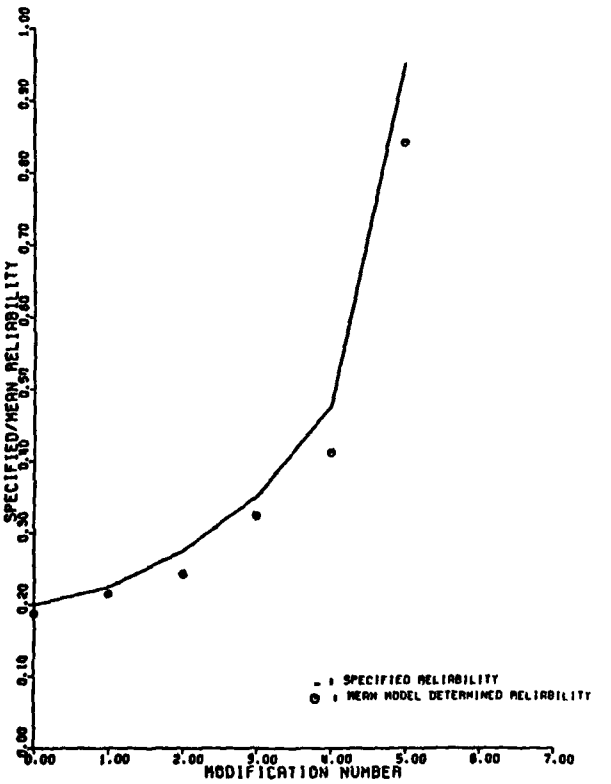


Fig.3.92 Discrete reliability growth model performance reliability set 7: 5 mods, 5 failures/mod

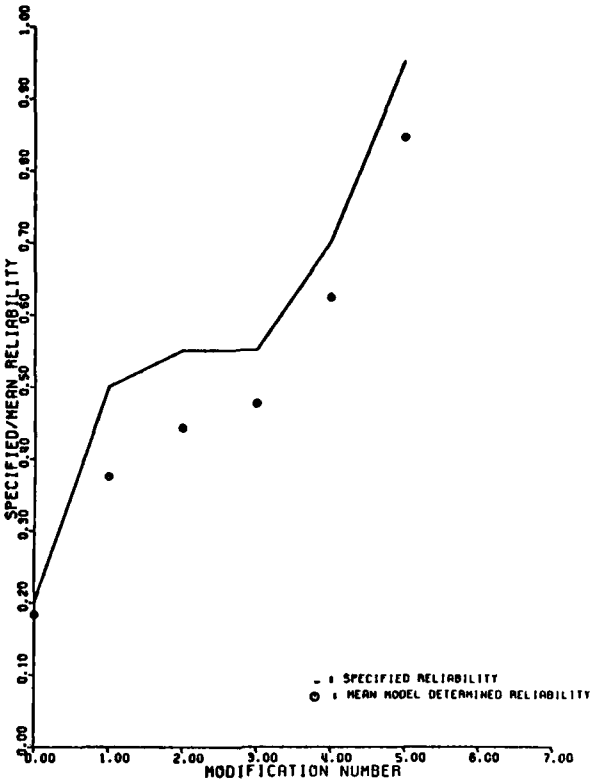


Fig.3.93 Discrete reliability growth model performanc reliability set 8: 5 mods, 1 failure/mod

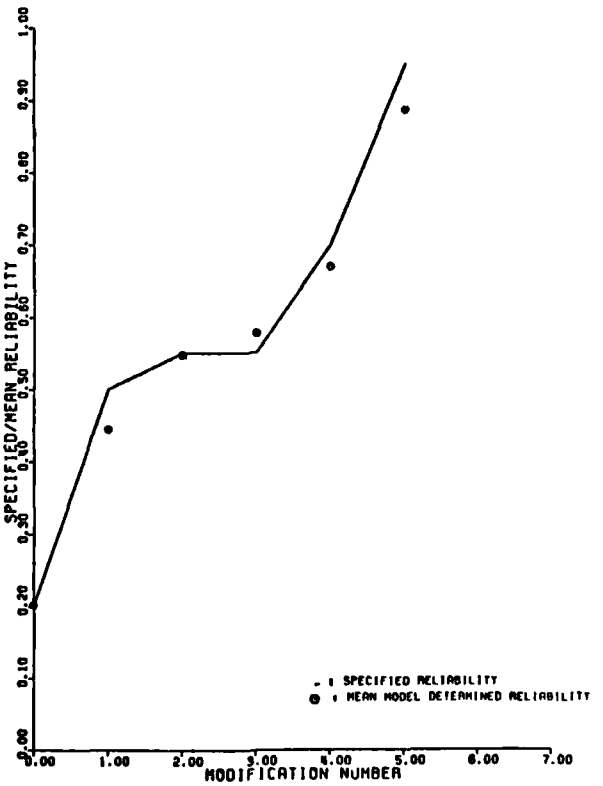


Fig.3.94 Discrete reliability growth model performance reliability set 8: 5 mods, 5 failures/mod

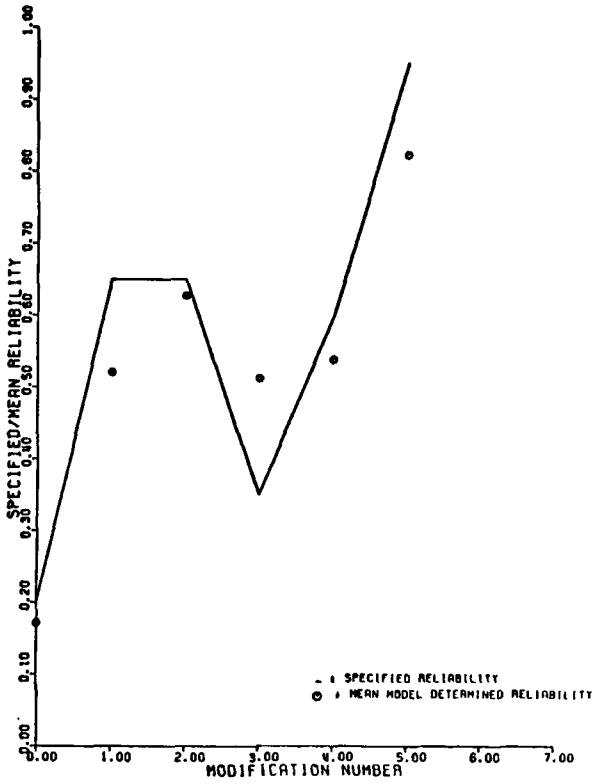


Fig.3.95 Discrete reliability growth model performance reliability set 9: 5 mods, 1 failure/mod

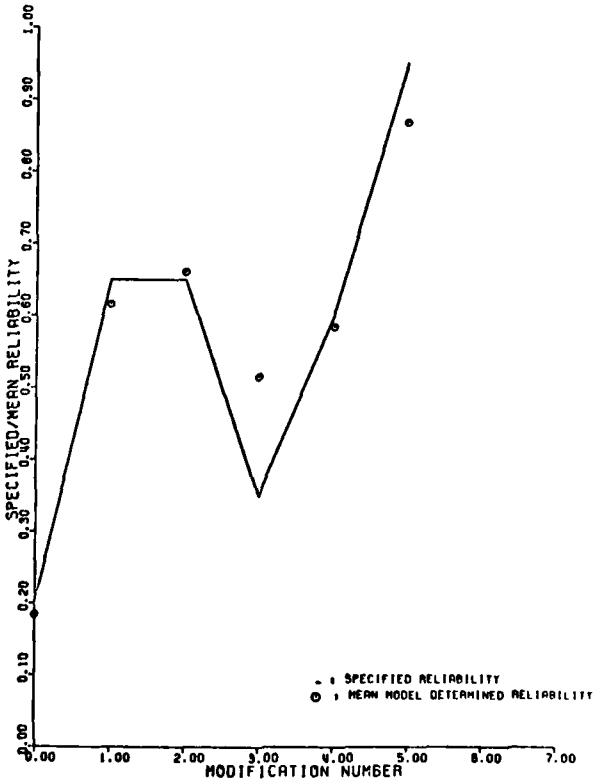


Fig.3.96 Discrete reliability growth model performance reliability set 9: 5 mods, 5 failures/mod

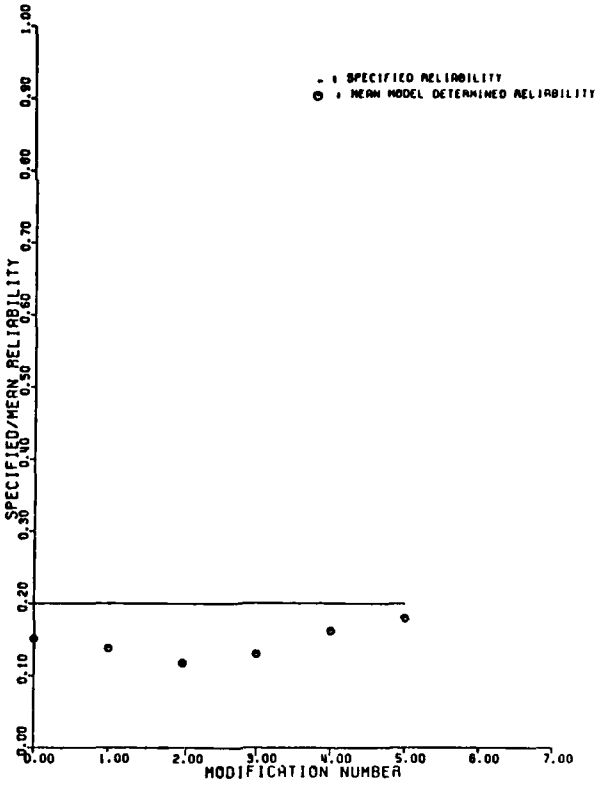


Fig.3.97 Discrete reliability growth model performance reliability set 10: 5 mods, 1 failure/mod

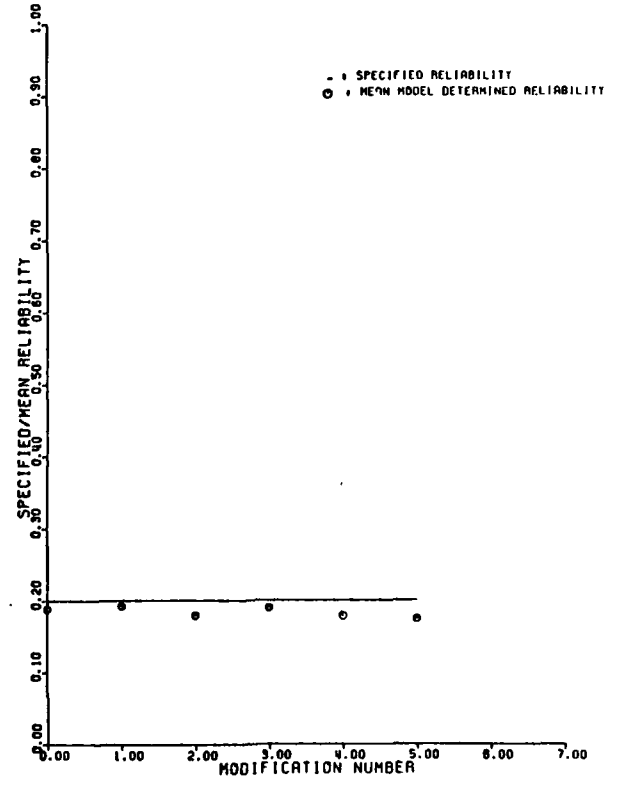


Fig.3.98 Discrete reliability growth model performance reliability set 10: 5 mods, 5 failures/mod

DISCUSSION

M.Jacobsen, Ge

Do you consider the debugging process during equipment burn-in as reliability growth per definition?

Author's Reply

Reliability has a specific definition, namely: "The probability that an item selected at random from a population will perform satisfactorily for a specified period of time under a specified environment." It is true that removing outliers from a population of items by burn-in can increase reliability by this definition, but the burn-in action is more of a gravity control action or a sampling inspection action. I personally prefer to keep its contribution to reliability improvement in the quality sampling inspection area where many other actions are taken which also improve reliability e.g. sampling inspection by Mil. Std 414. Let's preserve the more explicit meaning of reliability growth for changes which affect all future items.

This is only a personal view.

A SIMULATION PROGRAM FOR THE DETERMINATION OF SYSTEM RELIABILITY OF COMPLEX AVIONIC SYSTEMS

Christian Krause - Hubert Limbrunner
Elektronik-System-Gesellschaft
Vogelweideplatz 9

8000 München 81
Germany

SUMMARY

The simulation program SIMZUV which will be described in the lecture, computes system reliability according to the Monte-Carlo method.

This program enables the realistic consideration of complex failure logics of meshed systems, couplings of failures of different units and different mission phases, which is possible by the selection of a certain formulation of these marginal conditions.

In its capacity as a simulation program, SIMZUV is certainly capable of considering condition- and time-dependent failure rates and various unit reliability functions.

Owing to the low failure rates common in electronics, and to the system structures of high reliability typical of the avionic system, numerous computer runs are required in the case of reliability simulations in order to generate the requisite number of system failures necessary in order to obtain statistically reliable information. This requires much computer time, also in the case of high-speed computer facilities.

In order to avoid this situation, it is possible to perform the simulation of a "substitute system" of higher failure rates with shorter computer time, after a transformation of the failure rates.

It can be demonstrated that the reliability curve of the actual system can be computed from the reliability curve of the "substitute system" via a linear system of equations.

1. DESCRIPTION OF THE SIMZUV SIMULATION PROGRAM

Analytical procedures for the computation of system reliability can only be applied to a limited extent, because under practical conditions there are repeatedly requirements regarding reliability programs (e.g. multi-phase missions, cold redundancy, time-dependent failure rates etc.) which cannot be met by the analytical procedures available.

A simulation program can duplicate all these special cases in a realistic manner, and can thus be generally used. It also enables pinpointed program adaptations to further special cases. The SIMZUV described in the following was developed for the reasons stated above. It is written in FORTRAN IV and can thus be run on every large computer system.

1.1 Program Sequence

The program is divided into three sections:

- Reading of the input data and printing of the input data log
- Performance of the simulation and storing of the results

  At the beginning of each mission phase a decision is made for every equipment in accordance with the Monte-Carlo method as to if and when it will fail during that phase.

  By subsequently using the failure logic a check is made whether and when the overall system must be considered as failed as a result of these failures. System failure in this connection may already have occurred with the change in phase by modification of the system requirements.

  If a system failure is determined, or if the end of the mission is reached, the results will be stored and the simulation loop is repeated.

- Statistical evaluation and output of the results

  The program sequence is shown in Fig. 1-3. The subroutines called up are a random number generator for equally distributed random numbers, a program for computation of the failure-rate-dependent equipment down-times and a report program for the results.

1.2 Possibilities of Program Application

At present, the program can be used under consideration of the following system properties:

- meshed systems, i.e. in the case of the logic diagram a resolution according to series connection and parallel connection is not possible;
- cold redundancy;
- constant and time-dependent failure rates;
- multi-phase missions with different failure logics per phase;
- consideration of failure rates in the case of equipment switch on/off.

The limits with regard to the possibilities of application are only determined by the computing time and by the accuracy attainable by simulation. As the computing time is mainly influenced in negative form by very little failure rates, a procedure for the reduction of computing time is described in para. 2.

1.3 Input Quantities

For compilation of the input quantities, the logic diagram and the system mission description must be available. In detail, the following input quantities are required for the program:

- Number of equipments
- Number of mission phases
- Number of equipment statuses
- Number of simulation steps
- Duration of phases
- Equipment failure rates in the individual statuses, either in constant form or as a function of time
- Failure probability of the equipments in the case of status change (switching)
- Equipment status prior to mission begin and in the individual phases
- Equipment reliability function in the various statuses
- Failure logic for each phase
- Logic for the coupling of failures of different equipments
- Cold redundancies

1.4 Presentation of the Failure Logic

The failure logic is presented in the input data in the form of a matrix. Each matrix line contains the designation of a set of equipments, which, as a minimum, must have failed together in order to have caused a system failure. The whole matrix contains all those possible minimum combinations. Thus it is also possible to demonstrate complicated intermeshings.

Fig. 4 shows a meshed system with the associated failure logic.

1.5 Computing Results

The following is computed by the program:

- the system reliability curve, whereby the support elements can be indicated by the user;
- the equipment reliability in the individual phases and
- the conditional and absolute system reliability in the individual phases.

The absolute reliability of a system is the functioning probability on the assumption that the system was intact at the beginning of the mission.

The conditional reliability of the system is the functioning probability on the assumption that the system was intact at the beginning of the phase.

2. REDUCTION OF THE COMPUTING TIME BY TRANSFORMATION OF THE FAILURE RATES

The occurrence of very low failure rates of the individual components and of the overall system is typical of system reliability simulation. Therefore, it is necessary to simulate very many missions in order to obtain a statistically significant ratio between system failures and the number of missions, which can have a positive influence on the computing time. Therefore, a system was developed for performing the simulation on a substitute system with higher failure rates on a computer time-saving basis, followed by subsequent definition of the reliability of the system actually intended for investigation with the help of a linear equation system.

2.1 The Status Tree

Starting from status Z_o, i.e. absolute freedom from faults, all statuses are defined, which can result by any individual event from that status. The follow-on statuses again are determined from these statuses. Thus a directed graph is obtained, the (numbered) circles of which represent the (numbered) statuses, and the lines the respective failure rates of the equipments not yet failed in the original status (Fig. 5).

The status tree is completed when every chain starting from Z_o ends at a status which represents the failure of the system.

2.2 Computation of Reliability from the Status Tree

The following definitions can be introduced in a status tree as described:

s number of final statuses (failure statuses)

n number of statuses as counted from the original status (freedom from faults) to a final status (failure status) on the path between these statuses

λ_i Failure rates of the equipments along the transitions on the path

a_i Total of failure rates of all transitions starting from a status of that path

$$K_{in} = \prod_{\substack{k=1 \\ k \neq i}}^{n} (a_k - a_i)$$

t time

Thus the probability of each final status versus t (time) can be computed. The failure probability P (t) is the total of the probabilities of all final statuses.

$$P(t) = 1 - \sum_{j=1}^{s} \left(\lambda_1 \cdot \lambda_2 \cdot \ldots \cdot \lambda_{n-1} \cdot \lambda_n \cdot \sum_{i=1}^{n} \frac{e^{-a_i \cdot t}}{a_i \cdot K_{in}} \right)_j$$

Thus the following results for system reliability R (t):

$$R(t) = \sum_{j=1}^{s} \left(\lambda_1 \cdot \lambda_2 \cdot \ldots \cdot \lambda_{n-1} \cdot \lambda_n \cdot \sum_{i=1}^{n} \frac{e^{-a_i \cdot t}}{a_i \cdot K_{in}} \right)_j$$

2.3 Transition to the Substitute System

For all failure rates a factor β ($\beta \ll 1$) is selected such that the following shall apply:

$$\lambda = \beta \cdot \bar{\lambda}$$

For a_i and K_{in} the following results thereof:

$$a_i = \beta \cdot \bar{a}_i \qquad K_{in} = \beta^{n-1} \cdot \bar{K}_{in}$$

If these interrelations are inserted in the formula for R (t), with the exponential function expanding into series, the following equations result; the abbreviations A, B, C ... have to be chosen accordingly.

$$R(t) = \sum_{j=1}^{s} \left(\beta^n \cdot \bar{\lambda}_1 \cdot \bar{\lambda}_2 \cdot \ldots \cdot \bar{\lambda}_{n-1} \cdot \bar{\lambda}_n \sum_{i=1}^{n} \frac{e^{-\beta \cdot \bar{a}_i \cdot t}}{\beta^n \cdot \bar{a}_i \cdot \bar{K}_{in}} \right)_j =$$

$$= \sum_{j=1}^{s} \left(\bar{\lambda}_1 \cdot \bar{\lambda}_2 \cdot \ldots \cdot \bar{\lambda}_{n-1} \cdot \bar{\lambda}_n \cdot \sum_{i=1}^{n} \frac{e^{-\beta \cdot \bar{a}_i \cdot t}}{\bar{a}_i \cdot \bar{K}_{in}} \right)_j =$$

$$= \sum_{j=1}^{s} \left(\bar{\lambda}_1 \cdot \bar{\lambda}_2 \cdot \ldots \cdot \bar{\lambda}_n \cdot \sum_{i=1}^{n} \frac{1}{\bar{a}_i \cdot \bar{K}_{in}} \right)_j - \sum_{j=1}^{s} \left(\bar{\lambda}_1 \cdot \bar{\lambda}_2 \cdot \ldots \cdot \bar{\lambda}_n \cdot \sum_{i=1}^{n} \frac{\bar{a}_i}{\bar{a}_i \cdot \bar{K}_{in}} \right)_j \cdot \beta t +$$

$$+ \sum_{j=1}^{s} \left(\bar{\lambda}_1 \cdot \bar{\lambda}_2 \cdot \ldots \cdot \bar{\lambda}_n \cdot \sum_{i=1}^{n} \frac{\bar{a}_i^2}{\bar{a}_i \cdot \bar{K}_{in}} \right)_j \cdot \frac{1}{2!} (\beta t)^2 - \ldots =$$

$$= A - B\beta t + C(\beta t)^2 - \ldots$$

The residual term after m terms results

$$G_{m+1} = \sum_{j=1}^{s} \left(\bar{\lambda}_1 \cdot \bar{\lambda}_2 \cdot \ldots \cdot \bar{\lambda}_n \cdot \sum_{i=1}^{n} \frac{1}{\bar{a}_i \cdot \bar{K}_{in}} \cdot (-1)^{m+1} \cdot \frac{(\beta \cdot \bar{a}_i \cdot t)^{m+1}}{(m+1)!} \cdot e^{-\xi} \right)_j , \quad 0 \leq \xi \leq \bar{a}_i \cdot t \cdot \beta$$

If the reliability of the substitute system is computed by means of the increased failure rates, the reliability function R'(t) is as follows:

$$R'(t) = \sum_{j=1}^{s} \left(\bar{\lambda}_1 \cdot \bar{\lambda}_2 \cdot \ldots \cdot \bar{\lambda}_n \cdot \sum_{i=1}^{n} \frac{e^{-\bar{a}_i \cdot t}}{\bar{a}_i \cdot \bar{K}_{in}} \right) = A - B \cdot t + C \cdot t^2 - \ldots$$

with the residual term:

$$G'_{m+1} = \sum_{j=1}^{s} \left(\bar{\lambda}_1 \cdot \bar{\lambda}_2 \cdot \ldots \cdot \bar{\lambda}_n \cdot \sum_{i=1}^{n} \frac{1}{\bar{a}_i \cdot \bar{K}_{in}} \cdot (-1)^{m+1} \cdot \frac{(\bar{a}_i \cdot t)^{m+1}}{(m+1)!} \cdot e^{-\xi} \right)_j , \quad 0 \leq \xi \leq \bar{a}_i \cdot t$$

The following equation system results from the expansion into series. R'(t) is computed by simulation at the support points desired.

$R'(t = o) = A \rangle A = 1$ as the reliability vs time t = 0 is value 1

$R'(t_1) = A - Bt_1 + Ct_1^2 - \ldots$

$R'(t_2) = A - Bt_2 + Ct_2^2 - \ldots$

etc.

This equation system enables computation of the unknown A, B, C .... via a sufficient number of support elements t_i.

2.4 Practical Application

A factor β is determined from the failure rates of the system to be investigated such that the residual terms G_{m+1} and G'_{m+1} are small enough, and that the failure rates of the substitute system become as high as possible.

The simulation result is the reliability of the substitute system at several support elements; then the linear equation system for the values of A, B, C ... is solved, followed by a computation of the reliability of the system to be investigated in accordance with the following formula:

$$R(t) = A - B\beta t + C \cdot (\beta t)^2 - D \cdot (\beta t)^3 + \ldots .$$

In practical applications, this transformation enabled the selection of values down to 10^{-3} for factor β, thus providing for a reduction of the computing time by that factor, while keeping accuracy at the same level.

SOURCE:

Keppeler, Krause, Dr. Reichinger, Dr. Troetsch, Broschk, January 1976
Untersuchungen zu einsatztechnischen Kenngrößen

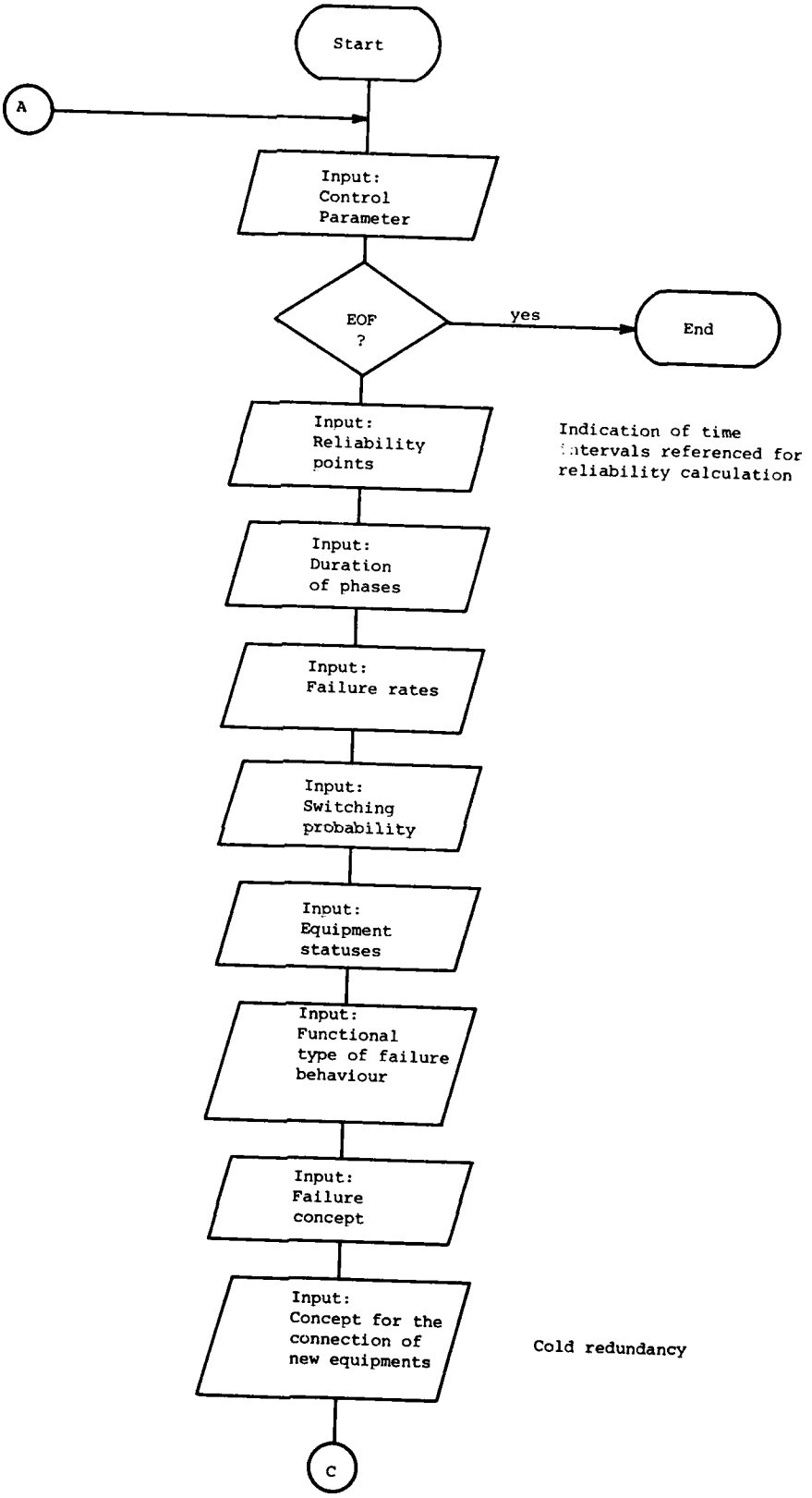


Fig.1 Flowchart, Part I

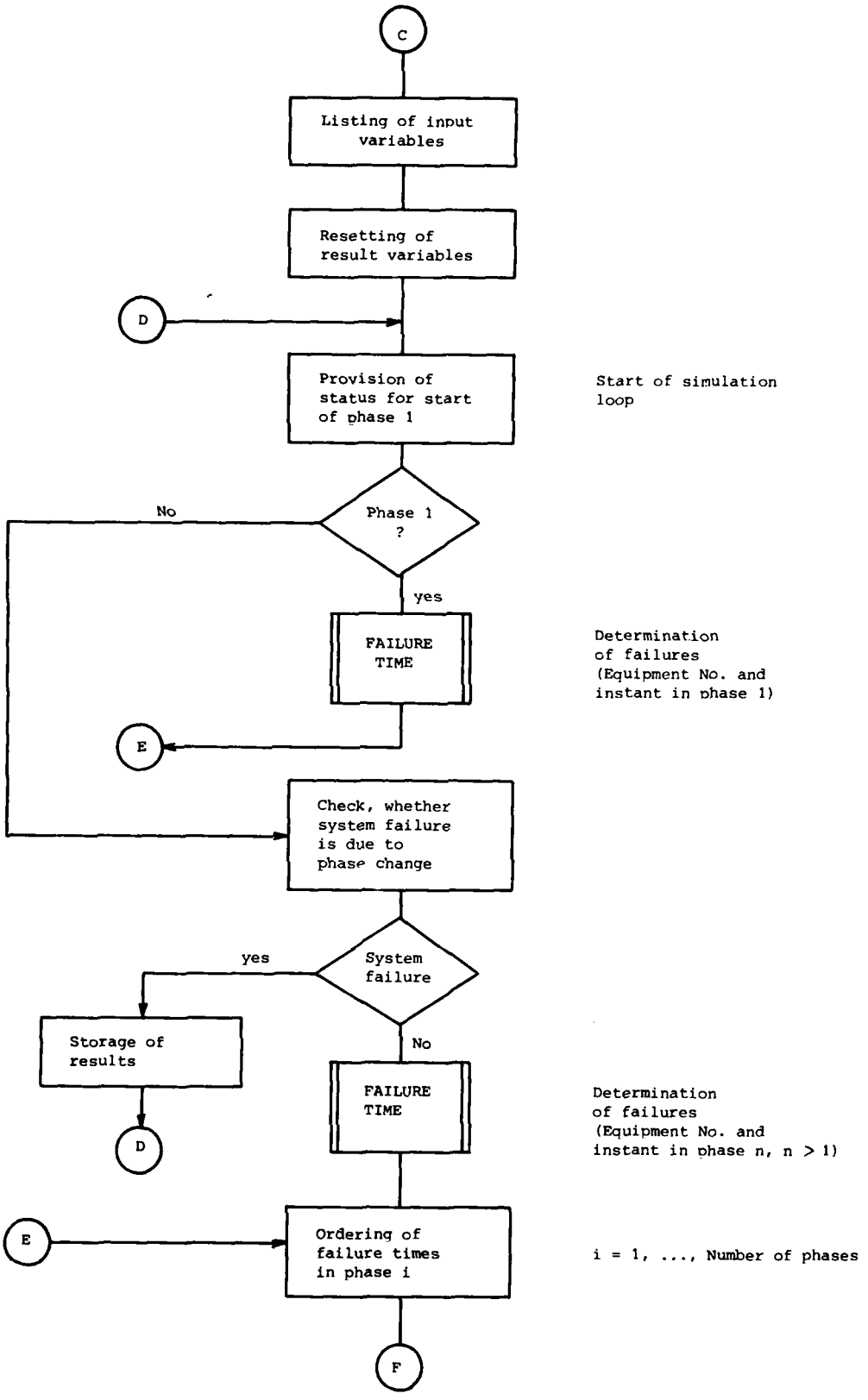


Fig.2 Flowchart, Part II

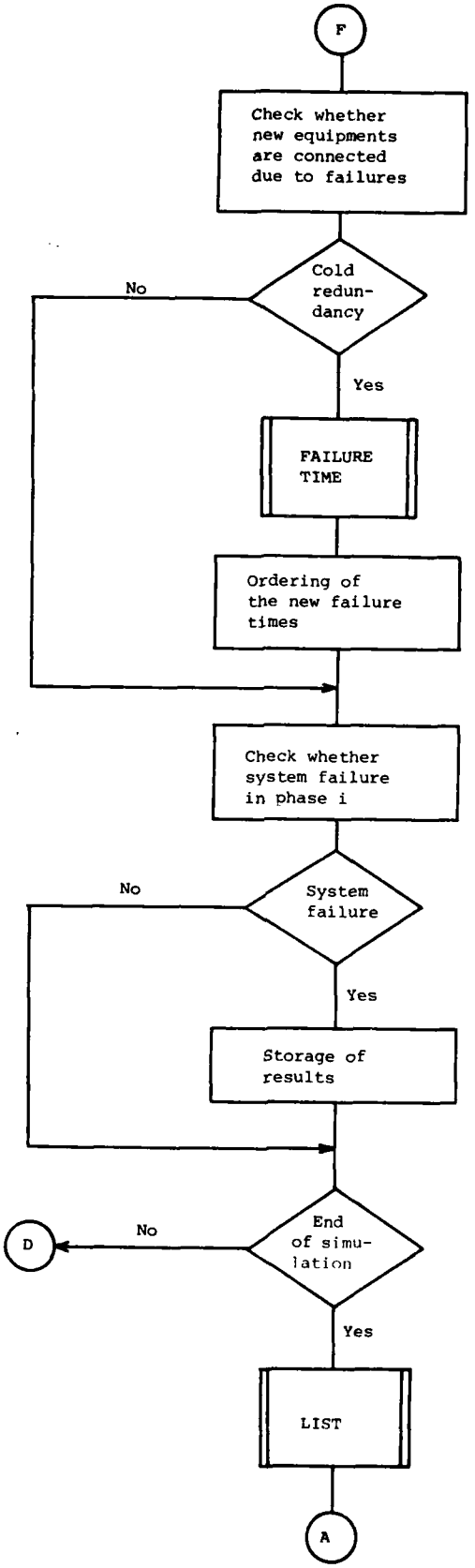
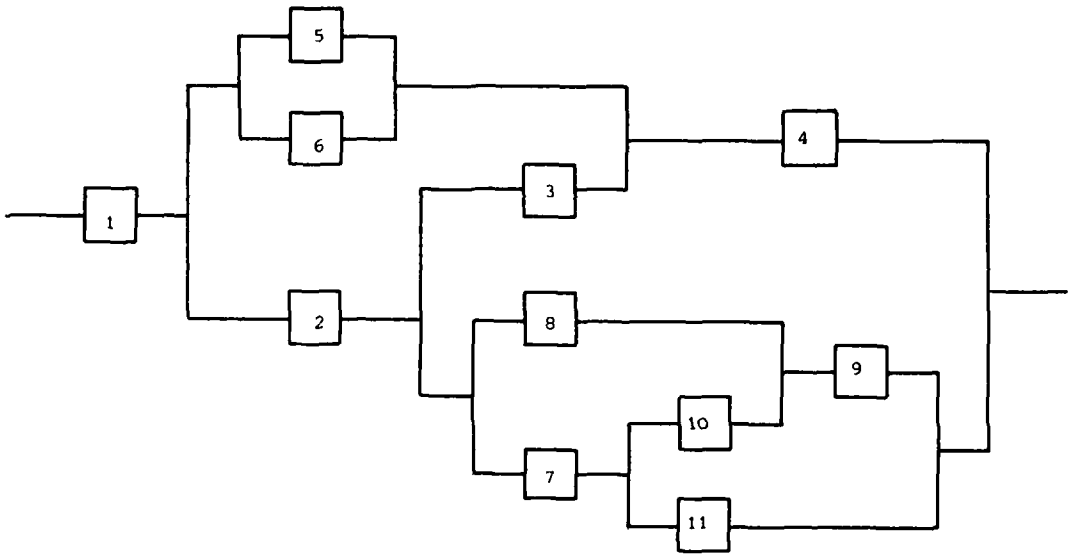


Fig.3 Flowchart, Part III



1
2 - 5 - 6
2 - 4
3 - 5 - 6 - 7 - 8
3 - 5 - 6 - 7 - 9
3 - 5 - 6 - 9 - 11
4 - 7 - 8
4 - 7 - 9
4 - 9 - 11
4 - 8 - 10 - 11

Fig.4 Failure logic of a meshed system

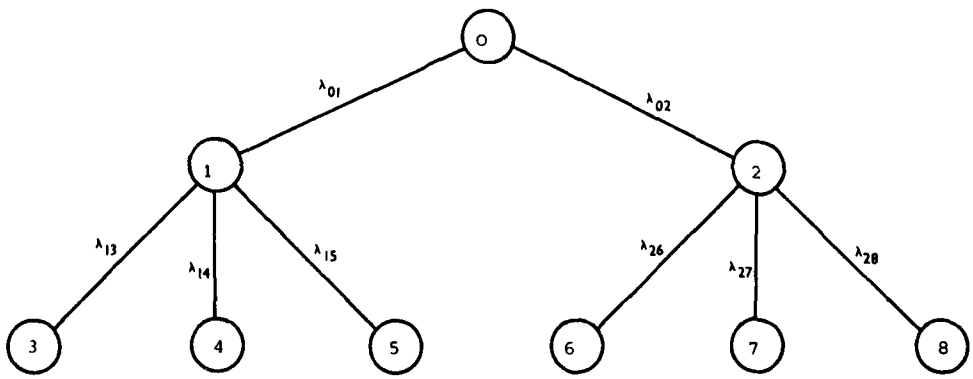


Fig.5 Status tree

DISCUSSION

R.Voles, UK

At the beginning of your paper, you state that the software package is written in FORTRAN and imply that it is transportable. Have you made this package generally available and, if so, from where can it be obtained?

Author's Reply

The program is available for sale or for rent from: Elektronik System GmbH, 8000 Munich, Vogelweideplatz 9, FRG. Program maintenance and adaptation will be conducted by ESG.

W.Ehrenberger, Ge

(1) Can your system also be applied to systems including repairs?

(2) How do you get the "failure logic" e.g. as in Figure 4? Is this obtained by hand?

(3) Does your method imply that the equation system received can be solved?

Author's Reply

(1) As it is a simulation program, it can be adapted to the special requirements of the system, for instance repairs with some distribution functions of repair time.

(2) The data concerning "failure logic" and others (e.g. logic of cold redundancy) are input data of the program. This data acquisition must be conducted by hand.

(3) The system of linear equations can be solved (Van der Monde Determinant).

W.M.Woods, US

Have you done any sensitivity analysis to determine the accuracy of your results when the exponential distribution does not quite hold?

Author's Reply

We have experience in the project we did and we made comparisons with another simulation program. A special feature of the system is that any system failure logic can be introduced into the program, but this must be described at the time.

MICRO-ELECTRONIC SYSTEMS RELIABILITY PREDICTION

P.D.T. O'Connor
Chief Quality Engineer, New Projects, British Aerospace Dynamics Group, Stevenage-Bristol Division, Stevenage, UK.

The paper reviews existing methods of parts stress analysis failure rate predicition, based upon US-MIL-HBK-217C, as applied to micro-electronic logic and memory devices. The extent to which the failure rate prediction formulae used in MIL HBK 217C are compatible with the physics of the various failure modes experienced, and with the failure statistics available, are investigated. The paper considers the effects of the failure rate distributions of the various failure modes, in relation to the objective of deriving a simple constant-failure-rate prediction model. Proposals are made for methods of improving the effectiveness of micro-electronics reliability prediction both as a design aid and for forecasting, and for areas of further study. A proposed alternative model is presented, with an example of its use to predict the failure rate of a typical system.

INTRODUCTION

The standard method of prediction of failure rates for electronic devices is US MIL HBK 217C. MIL HBK 217C has effectively become the international standard for this work, because of the attempts made by its sponsors to maintain the currency of the method in line with the rapid advances in device technology, and because it provides a very effective design review method for reliability analysis of circuits, particularly thermal analysis. However, there are some aspects of the MIL 217C method which do not adequately relate to the failure modes of devices and systems.

MIL HBK 217C micro-electronics failure rate prediction is based upon four principal assumptions:-

1. Failure rates are functions partly of time and temperature, following an Arrhenius relationship:

$$\lambda(t) = \lambda(T_R) \exp (E_A/k) (T_R^{-i} - T^{-1})$$

$\lambda(t)$ = Device failure rate

T = Device operating (junction) temperature ^{o}K

T_R = Reference temperature ($298^{o}K$)

E_A = Activation energy

k = Boltzmann's constant (8.62×10^{-5})

2. Failure rates have a second component, related to the overall environmental condition (e.g. airborne, space, ground mobile, etc.).

3. Failure rates are proportional to complexity (e.g. gate count).

4. The failure rate for a given device is constant with time.

This paper examines the validity of these assumptions, and relates them to the physics and statistics of device failures. It proposes alternative methods of failure rate prediction in line with the conclusions.

MIL HBK 217C FORMULAE

The MIL HBK 217C formula for micro-electronic device failure rate is of the form:

$$\lambda p = \pi_p \pi_L \pi_Q (C_1 \pi_T + C_2 \pi_E)$$

λp = Device FR

π_p = Pin factor dependent upon the number of external pins.

π_L = Learning factor, related to time device in production.

π_Q = Quality factor.

π_T = Temperature factor

π_E = Environment (application) factor

C_1 and C_2 = Complexity factors.

π_T is an Arrhenius function. It is shown plotted for bipolar digital ICs on Figure 1.

C_1 and C_2 are empirical exponential functions based on device complexity e.g. number of gates, transistors or memory bits. For example, for bipolar digital ICs:

$C_1 = .00129G^{(0.677)}$, $C_2 = .00389G^{(0.359)}$ (for $G \leq 1000$)

$C_1 = .051\exp(.001G)$, $C_2 = .0171\exp(.001G)$ (for $G > 1000$)

(G = number of gates)

These relationships are shown plotted on Figure 2 for bipolar digital ICs. To summarise, the physical factors that affect the derived failure rates are temperature and (indirectly) packing density. The "1000 gate discontinuity" shown in Figure 2 should be noted.

FAILURE PHYSICS

The main causes of failure of micro-electronic devices are shown in table 1 with typical percentage contributions. The 'out-of-spec overload' failure mode is included for completeness, but these failures are not due to device properties. Rather, they reflect system design and maintenance considerations. The percentage contributions can vary by large amounts, so it is not safe to generalise. Factors that affect the relative contributions are device technology, production techniques, quality control, screening methods used, circuit design, and device maturity.

TABLE 1

Failure Mode	Typical Failure Percentage (From ref. 9)
(Oxide/Diffusion/Metallisation) (Inversion/channeling)	50
(Contamination) (Foreign Material/particles)	5
Bonds	25
Hermeticity	10
Other	5
Out of specification overload	?

DEVICE COMPLEXITY

Referring to Table 1, the major causes of device failure are usually defects induced during the wafer fabrication processes. Failures due to this type of defect (chip related failures) can occur when cross sectional areas are reduced to the extent that physical processes such as electromigration or localised overheating can change the device characteristics within the operating lifetime of the item. Normally device geometries are such that these processes would not affect electrical characteristics in the normal lifetime of a good device. However, imperfections due to non-uniformity of diffusion, oxidation or metallisation, crystal flaws, etc. can lead to reduced cross sections in devices which otherwise pass initial visual, electrical and burn-in tests. Subsequent deterioration due to these processes can then result in failure. The assumption of an Arrhenius relationship to describe the time to failure due to these causes is justifiable, since the processes are obviously temperature and time dependent. However, whilst it is possible to derive a formula for the time to failure for a particular situation, in terms of cross sectional area and the physical constants applicable, (c.f. Alexanian I.T. - 1977) one never knows the geometries of imperfections which may exist in a supposedly good device. Therefore it is impossible to predict the time to failure of a given device, and the π_T formula given in MIL 217B can only be an empirical relationship based upon regression analysis of failure data.

The probability of the existence of failure inducing imperfections must to some extent increase with device complexity and packing density. However, this aspect tends to be offset by improvements in device fabrication technology. For example, updates of MIL HBK 217 regularly indicate progressive reductions in failure rate per gate for complex devices. In fact the empirical complexity factors of MIL HBK 217C appear to be more related to device maturity than to any other factor. The "1000 gate discontinuity" is probably a manifestation of this, as is the presently pessimistic failure rate model for memories. As a rough guide, the complexity of a LSI chip as expressed by the gate count is inversely proportional to the square root of the widths of the tracks on the chip, and a chip complexity factor related to $\sqrt{}$(gate count) would show a better fit to the failure rate data given in ref. 1.

The other two main internal failure modes (bond failure and hermeticity) (package related) are also obviously time-dependent. For example reference 3 shows a Weibull slope parameter of 0.5 for failures of gold bonds on hybrid circuits. However, temperature is unlikely to be a significant factor in a purely mechanical process as are many bond failures, but would be a factor in intermetallic processes. Temperature effects would also obviously affect the time to failure of a leaky device, but the relationship is unlikely to be a simple one, since temperature cycling, and the nature of impurities introduced, are likely to be the dominant factors. These failure modes are not related to chip complexity.

Failures due to loose inclusions would obviously be randomly distributed, as would failures due to out of specification overload. The rate of occurrence of the former is dependent upon QC screens during device fabrication and test. Overload (e.g. 'zap') failures are not the fault of the device, but of the way it is used. Such failures should not have been included in the data used to derive the MIL HBK 217C relationships, but they obviously are a factor of the reliability of systems involving microcircuit devices.

Plastic encapsulated ICs are subject to long-term hermeticity problems due to moisture absorption by the encapsulating material. This is a failure mode which should be considered separately in any predicition for systems using such devices, since it is a significant life-dependent feature affecting the whole population.

Since several failure modes contribute significantly to device reliability, and they have different underlying physical causes, any failure rate model should take account of this fact.

SYSTEM LEVEL CONSIDERATIONS

When one considers the pattern of failures in an electronic system due to the causes described above it is obvious that, whilst the time to failure of individual devices can be described by Arrhenius functions, the failure rate of the system as a whole must decrease with time. This is due to the fact that only a small proportion of the devices in any application will have defects that can lead to failure due to the action of these processes. Typically about .1% to 5% of devices in a circuit may be imperfect in this sense, and burn-in either at the device or system level is the recognised way of eliminating them, by accelerating the processes which lead to failure of defective devices, without damaging the much higher proportion of good devices. In such a situation, at the circuit level, as each failed defective device is replaced by another which is unlikely to be defective, the system failure rate will decrease with time. Kuehn (1974) has demonstrated this. However, this represents a controlled situation, untypical of maintained systems in the field. In-use electronic equipment often shows a nearly constant failure rate, due to imperfect maintenance, different ages of modules, etc. "Loading roughness" (Carter ADS - 1977) becomes a more significant factor in the equipment's reliability, and this causes the failure rate slope to decrease. However, the objective of a reliability programme is to identify and eliminate as many sources of loading roughness as practicable.

What is required therefore is a failure rate formula that combines the device level considerations (failure physics, device quality) with the system considerations (design, environment, reliability programme effects). Such a formula must also retain the advantages of the existing MIL HBK 217C model in relation to component stress analysis as a design discipline, and ease of computation.

PROPOSED FAILURE RATE MODEL (DEVICE LEVEL)

QUALITY EFFECTS

At the device level, the model should be based on the number of expected defective components in the system, expressed as a percentage (Table 2). This factor would be related to the present π_Q. It is not considered feasible to break this down to the different types of inherent defect, since, as stated above, the percentage contributions vary considerably in the data available. However, the proposed overall percentage defective values are supported by most of the recent published data on device screening and reliability (e.g. Kuehn R.E. 1974, Hnatek E.R. 1978, and ref. 9; see Annex A)), and the relationships between the different quality levels are generally in line with those for π_Q in MIL HBK 217C.

TABLE 2

Screening level	Present π_Q	Proposed π_Q (% defective)
A	1.0	0.1%
B	2.0	0.2%
B-1	5.0	0.5%
B-2	10.0	1.0%
C-1	16.0	3.0%
D (Commercial)	150.0	15.0%

No π_Q factor is allocated for plastic encapsulated devices, since a dominant (life-dependent) failure mode is a feature of the method of encapsulation. It is suggested that a separate model should be used for PEDs, to reflect the increasing failure rate property. Alternatively, the model proposed could be used, with an appropriate π_Q value, with a stipulation on maximum useful life expectancy.

COMPLEXITY

It is proposed that only one complexity factor should be used instead of two. It is doubtful if the failure data available would ever be sufficient to allow a credible derivation of two complexity factors, related separately to chip defects/temperature stress and package defects/application environment. The complexity factor should be applied only to chip related failures, since the package failure modes (hermeticity, contamination, bond failures) are largely independent of gate count.

GENERAL DEVICE-LEVEL MODEL

The general device level failure rate model would then take the form:

$$\lambda_d = \pi_Q\% \cdot C.\exp \cdot \left[K\left(T_R^{-1} - T^{-1}\right)\right] \quad \text{per } 10^6 \text{ hours.} \quad \left(K = \frac{Ea}{k}\right)$$

Kasouf & Mercurio (1978) propose a model for C based upon gate and bond complexity, and supported by failure data:

$$C = 0.0037 + (0.00006)P + (0.00000925)G$$

where P = number of pins, G = number of gates.

The constant term can be considered to cover the other failure modes.

It is suggested that an amended version of the Kasouf and Mercurio model be used in conjunction with a percentage defective factor (revised $\pi_Q\%$) (table 2), to indicate the failure rate of defective devices in the system, with "P" fixed at 16, since for most ICs the pin count is 16, with a small proportion having a slightly different number. Where the pin count is considerably in excess of 16 (e.g. microprocessors,) the 'pin' factor becomes a relatively insignificant term in relation to the chip complexity factor. For typical IC types, as make up the great majority of most system IC populations, the inclusion of a pin count complicates the prediction without adding any credible increase in accuracy. The revised λ_d formula proposed is:

$$\lambda_d = \pi_Q(\%) \left[4.67 + 0.1\sqrt{G}\right] \exp \left[K\left(T_R^{-1} - T^{-1}\right)\right] / 10^6 \text{ Hours.}$$

where K is as stated in MIL HBK 217C, i.e. 4794 for TTL, 8121 for MOS, and $T_A = 348^oK$, as proposed by Kasouf and Mercurio.

The 'C' term $\left[4.674 + .1\sqrt{G}\right]$ is shown plotted on Figure 2, compared with the MIL 217C C_1 and C_2 terms.

A constant failure rate (CFR) model is assumed, because:

- It is simpler than a decreasing failure rate model, whose general parameters would be very controversial.
- For most electronic equipment, particularly where a reliability programme including burn-in of production equipment has been applied, the failure rate approaches a level at which such parameters are difficult to derive with confidence from in-use data.
- For maintained equipment the CFR model is usually appropriate.

PROPOSED FAILURE RATE MODEL - SYSTEM LEVEL

'System level' failures are those due to inadequate design and manufacture, unsatisfactory handling or maintenance, and to environmental aspects not covered by the device-level formula. These include vibration, humidity, switching effects, etc. The system level model should therefore allow for system level reliability and QC activities, such as FMEA, reliability growth planning, and burn-in. The effect of such activities is very well documented (e.g. Anderson, 1978). The system level effect of overall environment is covered in MIL HBK 217C by the environmental factor π_E, applied to each device. Such a factor should, however, be applied at the system level. Also, there is no physical justification for multiplying it by a device complexity factor. On the other hand, at the system level the number of devices is a more significant reliability determinent than the complexity within individual devices.

It is proposed therefore that the system level model be of the form:

$$\lambda_S = B \cdot \pi_R \cdot \pi_E \cdot N \text{ per } 10^6 \text{ hours}$$

where N = number of electronic components, B = a constant to be derived from further analysis of system failure data.

Again a CFR is assumed, for the reasons given above.

The values given to π_R should be as shown in Table 3.

TABLE 3

		π_R
1.	Normal design activity.	4
2.	Augmented design (FMEA, design review, full failure data analysis and corrective action)	1
3.	As 2, but with reliability growth management using Duane methods (λ_i = initial failure rate, λ_a = achieved failure rate)	$\frac{\lambda_a}{\lambda_i}$

Burn-in of at least 50 hours to appropriate MIL STD 781C conditions is assumed in every case, for all production equipment.

π_E values could be as currently in MIL HBK 217C.

The system level model would cover all components in the system, and not only the micro-electronic content.

COMPLETE MODEL

The total system failure rate would be the sum of the failure rates due to the defective components and to system influences, i.e.

$\lambda = \Sigma\lambda d + \lambda s$ (In general, $\Sigma\lambda_d$ would include the failure rates of all device types in the system, not only ICs).

Such a model would have the following advantages over the existing MIL HBK 217C model:

1. It would allow component aspects (quality, application, etc.) and system considerations to be evaluated separately, whereas at present system level reliability considerations (apart from application environment) are ignored.

2. Elimination of the two MIL HBK 217C IC complexity factors, and the use of a simple gate count instead, would simplify prediction calculations.

3. It would provide a better basis for comparing tradeoffs in terms of component quantity and complexity, e.g. discrete MSI, custom LSI, etc.

Example

Taking the system described by Kasouf and Mercurio, and using the thermal functions they proposed (i.e. $T_R = 348^oK$), the results shown in Table 4 are obtained. ($\pi_Q = 0.5\%$ (Class B-1)).

TABLE 4

Part Description	Gates	N	T_j (^oC)	λ 217B x10^{-6}	λ Kasouf & Mercurio	λ Author x10^{-6}
Dual 1024-Bit MOS Shift Register	2048	1130	66	2.365	0.249	.0247
1024 x 1 Bit Static MOS RAM	1125	283	57	2.02	0.083	.0112
1024 x 1 Bit TTL RAM	1125	157	72	1.67	0.267	.0356
2048 (412 x 4) TTL Bit PROM	2048	50	102	2.365	1.249	.1240
1024 (256 x 4) Bit TTL PROM	1024	21	87	1.152	0.445	.0623
Hex D Type F/F	38	386	65	0.330	0.067	.0176
Quad 2 Input MUX	19	178	63	0.249	0.060	.0156
4-Bit Arithmetic Logic Unit	63	100	75	0.445	0.105	.0273
Sync 4-Bit Binary Counter	57	98	72	0.430	0.094	.0241
16-Input Multiplexer	26	111	54	0.285	0.046	.0107
8-Bit Bidirectional S/R	87	64	72	0.509	0.107	.0249
64 (16 x 4) Bit	95	27	70	0.515	0.091	.0231
$\Sigma N\lambda$				3985.3	488.55	66.91

Note the proportionately greater reduction of failure rates of the complex devices (>1000 gates) relative to the values derived by Kasouf and Mercurio, due to use of the $\sqrt{G}$ term.

For the programme described in reference 8, $\pi_R = .25$, is probably appropriate. $\pi_E = 4$ (airborne inhabited). N = 2605. B = .65, say.

Therefore $\lambda_S = .65 \times .25 \times 2605 = 421.64/10^6$ hours.

And the total failure rate = $488.55/10^6$ hours. ($\lambda_S + \Sigma N\lambda_d$).

The total failure rate predicted is the same as that predicted by Kasouf and Mercurio and supported by the data quoted therein. The value of B of .65 was selected to ensure this result. However, it is suggested that more analysis of failure data, covering many more projects, is required so that a more widely acceptable relationship between device and system reliability can be derived.

CONCLUSIONS

The model proposed, whilst it does not comply with all the requirements of the ideal failure rate model, particularly with respect to the decreasing failure rate characteristic of electronic systems is nevertheless a better analogue of their reliability determinants than is MIL HBK 217C. By considering separately the failure physics of devices and the system level determinants, reliability prediction can be made a more useful tool both for device stress analysis and for reliability programme management. Also, it should enable better correlations to be made between device test results and system test results. More work needs to be done to refine the parameters of the method proposed. For example, it is unlikely that one device level model is appropriate for all device types using TTL or MOS technology (i.e. logic, ROMs, RAMs, microprocessors, etc.). Also, for high-reliability programmes with closely controlled maintenance or with no maintenance, the use of a decreasing failure rate model for device failures should be considered. A Weibull slope parameter of .5 seems typical for such equipment. Therefore the model proposed is recommended as a basis for further work to improve the effectiveness and usefulness of micro-electronics reliability predicition.

ACKNOWLEDGEMENTS

I gratefully acknowledge the helpful comments provided by my colleagues at British Aerospace Dynamics Group, by George Kasouf of the General Electric Company, and by Mark Klein of the Reliability Analysis Centre (RADC). The conclusions derived are my own, and do not necessarily reflect the views of British Aerospace. I am indebted to the Group for permission to publish the paper.

REFERENCES

1. Alexanian I.T., Brodie D.E. - A Method for Estimating the Reliability of ICs - IEEE Trans. Reliability Dec. 77.

2. Anderson, R.T. - Reliability Design Handbook (RDH 376) - Reliability Analysis Center (RADC).

3. B.Ae. Report No. ST 15579 for DCVD Contract RP42-1.

4. Carter, A.D.S. - Reliability of Mechanical and Electronic Components. Is a Different Approach Necessary? - Proc. Sym. on Q & R in Aerospace R.Ae.S. (Stevenage) 1977.

5. Davies, C - Micro-circuit Failure Characteristics and the Implications for use with redundancy in logic systems - Proc. Sym. on Quality and Reliability in Aerospace, R.Ae.S (Stevenage) 1977.

6. Hnatek, E.R. - Microprocessor Device Reliability - Microelectronics and Reliability Vol. 17 1978.

7. Kasouf, G, Mercurio, S - Evaluation of LSI/MSI Reliability Models - Proc. IEEE Reliability Symposium 1978.

8. Kuehn, R.E. - Results of Production Thermal Cycle Screening - IEEE Trans. Reliability Oct. '74.

9. Reliability Analysis Center (RADC) - Micro-circuit Device Reliability - Memory/LSI/Data (MDR-7, MDR-8).

$\pi_{T1} = .1e^{X}$

$X = -4794\left(\frac{1}{T_j + 273} - \frac{1}{298}\right)$;

T_j = worst case junction temperature

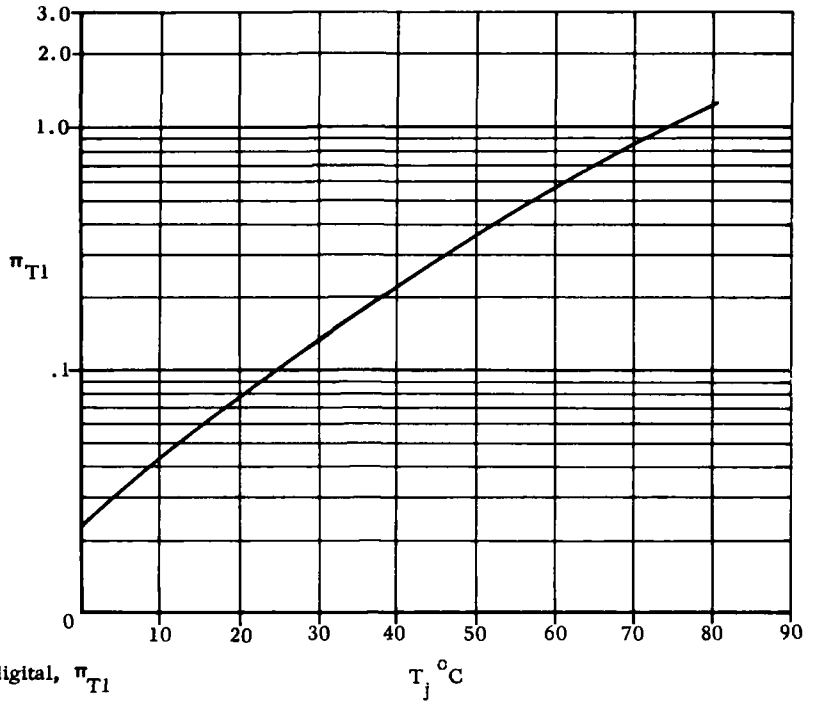
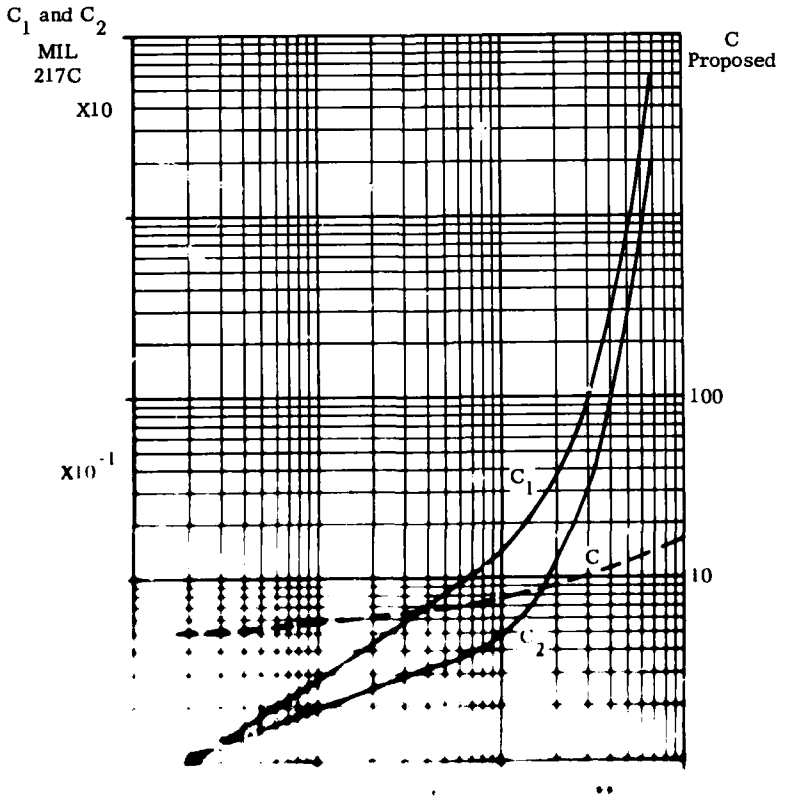


Figure 1 Bi-Polar digital, π_{T1}



ANNEX A - SUMMARISED IC FAILURE DATA

Reference 1 (MDR-7)

Field Data - Memory Devices

Quality Level	Percentage Defective		
	0-256 Gates	1024 Gates	2048/4096 Gates
B 1	.07	-	.10
B 2	.01	-	1.5
C1	.33	-	-
Commercial	.31	.40	2.2

Reliability Demonstration Data - Memory Devices

Quality Level Level	Percentage Defective		
	0-256 Gates	1024 Gates	2048 Gates
B 1	.05	0.03	-
B 2		0.15	0.01
C	2.85	3.14	-
Commercial	-	1.04	-

Life Test Data - Memory Devices (Quality Level: Commercial)

Temp ^{o}C	Percentage Defective
70	0.188
125	0.531

Reference 4 (Kuehn) (SSI Data from system test, quality level B2 (approx.) 1973 data.

.035% of devices failed in 200 hour system burn-in.
.095% of devices failed in field use.

Reference 6 (Hnatek) Life Test Data (Quality levels not stated).

Technology	Percentage Defective
Bipolar LSI	.34
PMOS LSI	.31
NMOS LSI	.22
CMOS LSI	1.5

DISCUSSION

R.Voles, UK

As the parameters characterising the failure rates of electronic devices are dependent on rapidly advancing technology, what prospects do you see for persuading the device manufacturers to publish these parameters at the time when the devices are put on the market?

Alternatively, how do you feel these parameters could be measured in a timely manner by a third party?

Author's Reply

I consider that the parameters should be derived and published by independent bodies, such as RADC in the USA, and not by the manufacturers. If a manufacturer can provide life test data to support this work, this would be helpful and could be used in special cases. In either case, the model I propose would allow the parameters to be kept up-to-date more easily since it separates the time device level failures from failures due to other causes, and also because it is based upon the physics of failure, and can, therefore, be more accurately extrapolated to take account of technology changes.

F.S.Stringer, UK

Your paper is entitled Micro-Electronic Systems Reliability Prediction. Your analysis does not refer to the effects of splitting up the system into say microprocessor units. Can such an addition be made easily to your program? If not, do you propose to try to include it?

Author's Reply

I agree that this aspect of system design can have reliability implications. However, my proposal is a tentative one, to indicate what I consider might be a better approach to the problem, and I have not yet considered extending it in the way you suggest. I think that for LSI and VLSI, particularly microprocessor systems, one would have to take into account system architecture and construction, and also firmware aspects (e.g. PROM programming).

MARKOVIAN AVAILABILITY MODEL FOR A NETWORK OF COMMUNICATING COMPUTERS

Dr. Thad L. Regulinski
Air Force Institute of Technology
Department of Electrical Engineering
Wright-Patterson Air Force Base
Dayton, Ohio 45433 USA

SUMMARY

In classical Availability modeling the steady state function is generally quantified from the ratio of expected time a given equipment spends in working state to the expected time the equipment spends both in the working and the repair states. In modeling the Availability function for a system of computer networks, the state space must necessarily be expanded to include factors other than failure and repair such as channel and processor overloads, and channel interference. The objective of this inquiry is to model the Availability function from a Markovian state-discrete, time-continuous formulation encompassing those adverse conditions which contribute to the network's total down time. The results of three and four state models are derived under assumption of temporally homogeneous, first order, Markovian process. Using sensitivity analysis, the effect of variation of state transition probabilities on the steady state Availability function is examined and illustrated by an example.

1. INTRODUCTION

A computer network can be viewed as an aggregate of computers and terminals connected together by communication channels over which information is interchanged between the information processing system, or between terminals and the processing systems. Such networks can be totally ground fixed, totally avionic or some combination of ground fixed and avionic.

Numerous metrics of performance are commonly in use to measure the efficacy of computer networks (Grubb, D. S. and Cotton, I. W., 1975). Salient among the more frequently used are the following five metrics: Transfer Rate, Channel Establishment Time, Network Delay, Reliability/Maintainability, and Availability. In quantification of these metrics, Availability is strongly affected by the remaining four and is the principal focus of this inquiry.

Availability denoted by A(t) is defined by the probability that the network is performing within its parameter limits at time t. Closely related to Availability are the metrics of Reliability and Maintainability. Reliability is defined by the probability of network performance within its parameter limits over some time interval (0,t) subject to specified environmental conditions. In the event of network unreliability, i.e., failure, malfunction or out of limits performance of one or more of the network equipments, the network becomes unavailable during that period of time it is subject to corrective maintenance. Hence, maintainability is defined by the probability that restorative maintenance will be completed by some t time (Regulinski, T. L., 1970).

Most common algorithms formulate the Availability function from the ratio of the expected time any given system is in working (UP) state to the expected time the system spends in both the working state and the repair (DOWN) state (Locks, M. O., 1973). Although unreliability of computer network equipments is an important contributing factor to network unavailability, other factors must necessarily be considered particularly those which are derived from the operational characteristics of computer processors and communication channels. Specifically, exceeding channel capacity or having too many messages contending for a channel can cause an overload condition leading to network unavailability. There may be other conditions of overload such, for example, the inability of the processing equipment to handle all message traffic at any given instant of time. It follows then that the formulation of Availability function must not only include network equipment failures, but other factors which contribute to the network unavailability.

The objective of this inquiry is to model the Availability function from a Markovian state—discrete, time—continuous formulation which would include all relevant conditions that contribute to the network's unavailability.

2. UNDERLYING THEORY OF MARKOVIAN MODELING

Markov processes are stochastic processes characterized by random variables referred to as the states of the process; transition probabilities by which the process changes state; and process parameter, usually time, by which the dynamics of the process is measured (Cramer, H., 1967). A stochastic process denoted by X(t) is said to possess a Markovian property if the conditional probability of any future event, given any past event(s) and the present state, is independent of the past event(s) and depends only on the present state of the process. Mathematically, this can be expressed as

$$P\{X(t) = j \mid X(t-\Delta t) = i,\ldots,X(0) = a\} = P\{X(t) = j \mid X(t-\Delta t) = i\} \quad (1)$$

where a,...,i and j,...,n represent the states assumed by the process at time 0,...,t-Δt, t,..., t_n. The probability that a Markov process will make a transition from any state i to any state j in Δt time is

$$P\{X(t) = j \mid X(t-\Delta t) = i\} = p_{ij} \quad (2)$$

Additionally Markov process is said to be temporally homogeneous, that is the probability of transition from state i to state j in time interval (t,t+Δt) is constant, if:

$$P\{X(t) = j \mid X(s) = i\} = P\{X(t-s) = j \mid X(0) = i\} \quad (3)$$

This in essence states that in a homogeneous process, the conditional probability of being in state j at t time, given that the process was in state i at time s, depends only on the difference between the time t and s, and that difference is Δt. Given then the constant transition probability λ, it follows that

$$P_{ij}\Delta t = \lambda_{ij}\Delta t \qquad (4)$$

$$P_{ii}\Delta t = \lambda_{ii}\Delta t$$

where

$$\lambda_{ii} = 1 - \sum_{j=0}^{n} \lambda_{ij} \qquad j \neq i$$

Calling upon the independence-of-transition occurence property of Markov process it will be assumed in the development which follows that the probability of transition of two or more occurrences in Δt interval of time is negligible. Further, it will be assumed that the process is a homogeneous one.

3. MARKOVIAN MODEL FORMULATION

Consistant with the observed behav.or of computer network equipments and interconnecting communication channels, the states of the process are defined as follows:

STATE	STATE SPACE DESCRIPTION
0	Network equipment performing and there are no overloads (network available).
1	Network equipment performing but there are overloads (network unavailable).
2	Network equipment not working and under repair (network unavailable).

The state probability vector defined as a row vector of nonnegative components summing to 1 consists of the probabilities that the network is in state i at t time. Hence

$$\overrightarrow{P(t)} = \{P_0(t),\ P_1(t),\ P_2(t)\} \qquad (5)$$

and

$$\sum_{i=0}^{2} P_i(t) = 1 \qquad (6)$$

Recalling that the Availability function was defined by

$$A(t) = P\{\text{Network performing within parameter limits at t time}\} \qquad (7)$$

it follows then that

$$A(t) = P_0(t) \qquad (8)$$

The transition probabilities are now considered in light of the defined network states. If the network is in state 0 at t time it can remain in state 0 in (t, t+Δt) interval of time or it can transition to either state 1 due to overload, or to state 2 due to failure. Further, if the network is in state 1 at t time, either it can remain in state 1 in (t, t+Δt) interval of time provided the network did not recover from overload and it did not transition due to failure; or it can transition back to state 0 due to recovery or to state 2 due to failure. Lastly, if the network is in state 2, it can remain in state 2 in (t, t+Δt) interval of time because recovery from failure has not been made, or it can transition back to state 0 upon repair. The transition probabilities are thus defined as follows:

$\lambda_1\Delta t$ = P (Transition due to overload in t, t+Δt given network in state 0 at t time)

$\lambda_2\Delta t$ = P (Transition due to failure in t,t+Δt given network in state 0 at t time)

$\lambda_3\Delta t$ = P (Transition due to failure in t,t+Δt given network in state 1 at t time) (9)

$\mu_1\Delta t$ = P (Transition due to recovery from overload in t,t+Δt given network in state 1 at t time)

$\mu_2\Delta t$ = P (Transition due to repair in t,t+Δt given network in state 2 at t time)

The concommitant state transition diagram is given in Fig. 1 showing the performi[illegible] the overloaded and the failed states and their respective transition probabilities as defined in expressio[illegible] f (9). For computational tractability the transition probabilities can be exhibited in a special [illegible]f the matrix form, thus

$$P_{ij} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{bmatrix} 1-(\lambda_1+\lambda_2)\Delta t & \lambda_1\Delta t & \lambda_2\Delta t \\ \mu_1\Delta t & 1-(\mu_1+\lambda_3)\Delta t & \lambda_3\Delta t \\ \mu_2\Delta t & 0 & 1-\mu_2\Delta t \end{bmatrix} \end{array} \qquad (10)$$

The product of the state probability vector and the transition probability matrix leads to the following set of ordinary-first order differential equations:

$$\begin{aligned} P_0'(t) &= -(\lambda_1+\lambda_2)P_0(t) + \mu_1 P_1(t) + \mu_2 P_2(t) \\ P_1'(t) &= \lambda_1 P_0(t) - (\mu_1+\lambda_3)P_1(t) \\ P_2'(t) &= \lambda_2 P_0(t) + \lambda_3 P_1(t) - \mu_2 P_2(t) \end{aligned} \qquad (11)$$

Taking LaPlace transforms of the three equations in expression (11), and assuming initial conditions such that $P_0(0) = 1$, $P_1(0) = 0$, and $P_2(0) = 0$, leads to

$$\begin{aligned} sP_0(s) + (\lambda_1+\lambda_2)P_0(s) - \mu_1 P_1(s) - \mu_2 P_2(s) &= 1 \\ sP_1(s) - \lambda_1 P_0(s) + (\mu_1+\lambda_3)P_1(s) &= 0 \\ sP_2(s) - \lambda_2 P_0(s) - \lambda_3 P_1(s) + \mu_2 P_2(s) &= 0 \end{aligned} \qquad (12)$$

To solve for $P_0(s)$ it is convenient to apply Cramer's rule, hence

$$P_0(s) = \frac{\begin{vmatrix} 1 & -\mu_1 & -\mu_2 \\ 0 & s+(\mu_1+\lambda_3) & 0 \\ 0 & -\lambda_3 & s+\mu_2 \end{vmatrix}}{\begin{vmatrix} s+(\lambda_1+\lambda_2) & -\mu_1 & -\mu_2 \\ -\lambda_1 & s+(\mu_1+\lambda_3) & 0 \\ -\lambda_2 & -\lambda_3 & s+\mu_2 \end{vmatrix}} \qquad (13)$$

The solution of expression (13) is of the form

$$P_0(s) = \frac{s^2 + sA + B}{s\{s^2 + sC + D\}} \qquad (14)$$

where

$$A = \lambda_1 + \mu_1 + \mu_2$$

$$B = \mu_2(\mu_1 + \lambda_3)$$

$$C = \lambda_1 + \lambda_2 + \lambda_3 + \mu_1 + \mu_2$$

$$D = \lambda_1\lambda_3 + \lambda_2\lambda_3 + \lambda_1\mu_2 + \lambda_2\mu_1 + \lambda_3\mu_2 + \mu_1\mu_2$$

By inspection it is seen that the inverse LaPlace teansforms of (14) will yield a steady state and transient solutions. However, for computer networks the steady state availability function is of principle interest and to obtain it, the most direct procedure is to call upon LaPlace's final value theorem:

$$\lim_{s\to 0} sF(s) = \lim_{t\to\infty} f(t) \tag{15}$$

which when applied to the equation of expression (14) gives

$$\lim_{s\to 0} sP_0(s) = \lim_{s\to 0} \frac{\{s^2 + sA + B\}}{\{s^2 + sC + D\}} \tag{16}$$

The steady state Availability function denoted by A_{ss} follows directly.

$$A_{ss} = \frac{\mu_2\{\mu_1+\lambda_3\}}{\lambda_1\lambda_3+\lambda_2\lambda_3+\lambda_1\mu_2+\lambda_2\mu_1+\lambda_3\mu_2+\mu_1\mu_2} \tag{17}$$

In similar manner, one can expand the formulation to a four state model where the additional state, consistant with avionic reality could represent the interference on the channels. This gives two additional transition probabilities

$\lambda_2\Delta t$ = P{Transition due to interference in t,t+Δt given network in state 0 at t time}

and

$\mu_2\Delta t$ = P{Transition due to recovery from interference in t,t+Δt given network in state 2 at t time}

Clearly, state 2 in earlier formulation would be denoted as state 3 in the four state model. Following identical development which yielded the results of expressions (10) through (17), the new steady state Availability function can be shown to be

$$A_{ss} = \frac{\mu_3(\mu_1+\lambda_5)(\mu_2+\lambda_4)}{(\mu_2+\lambda_4)\{\lambda_1(\mu_3+\lambda_5)+(\mu_3+\lambda_3)(\mu_1+\lambda_5)\}+\lambda_2(\mu_1+\lambda_5)(\mu_3+\lambda_4)} \tag{18}$$

4. SENSITIVITY ANALYSIS OF A_{ss} FUNCTION

The network performance as measured by the steady state Availability can be analyzed for deficiencies or improvement by examining the network characteristics which underlie the transition probabilities. Specifically, the effect of the variation of $\lambda_i,\mu_i, i=1,2,\ldots n$, on the steady state Availability can be expressed in terms of a Sensitivity function denoted by $S^A_{p_{ij}}$. In networks applications, it is a measure of the sensitivity of network response to variations of factors affecting the transition probabilities. In general the sensitivity function is expressed in terms of a ratio of the fractional change in system function to fractional change in system parameters (Tomovic, R., 1963). Here it is defined by

$$S^A_{p_{ij}} = \left\{\frac{p_{ij}}{A}\right\} \cdot \frac{\partial A}{\partial p_{ij}} \tag{19}$$

Thus for the result of the three state model given by the expression of equation (17), taking the partial derivatives of the steady state availability with respect to each p_{ij} gives the following results:

$$S^A_{\lambda_1} = -\lambda_1\{\lambda_3 + \mu_2\}/D \tag{20}$$

where $D = \{\lambda_1\lambda_3 + \lambda_2\lambda_3 + \lambda_1\mu_2 + \lambda_2\mu_1 + \lambda_3\mu_2 + \mu_1\mu_2\}$. Further,

$$S^A_{\lambda_2} = \{-\lambda_2(\lambda_3 + \mu_1)\}/D \tag{21}$$

$$S^A_{\lambda_3} = \{\lambda_1\lambda_3(\mu_2 - \mu_1)\}/(\mu_1 + \lambda_3)D \tag{22}$$

$$S^A_{\mu_1} = \{\mu_1\lambda_1(\lambda_3 + \mu_2)\}/(\mu_1 + \lambda_3)D \tag{23}$$

and

$$S^{A}_{\mu_2} = \{\lambda_1\lambda_3 + \lambda_2\lambda_3 + \lambda_2\mu_1\} / D \qquad (24)$$

Typical computer network Availability requirements range from 0.95 to 0.995. Thus for example the National Library of Medicine specified for its bibliographic retrieval system Availability of no less than 0.95 (National Library of Medicine, 1972). This can be contrasted with Availability requirements of 0.99 for airlines reservation system and 0.995 for priority telephone trunk lines. For any computer network requiring improvement of its availability, the sensitivity analysis can isolate the networks' marginally performing elements. Thus for example let it be assumed that the specified Availability for a given network is 0.95. Let it be further assumed that from the network generated data of failure, overload and recovery times, the following transition probabilities were estimated:

$$\begin{aligned} \lambda_1 &= 0.08 \\ \lambda_2 &= 0.02 \\ \lambda_3 &= 0.02 \\ \mu_1 &= 0.90 \\ \mu_2 &= 0.98 \end{aligned} \qquad (25)$$

Substituting the values of (25) into expression of equation (17) results in $A_{ss} = 0.9016$. Since this is less than the specified availability the expressions given by equations (20) through (24) are evaluated to determine the network's response to variations of transition probilities. This yields

$$\begin{aligned} S^{A}_{\lambda_1} &= -0.08 & S^{A}_{\lambda_2} &= -0.0184 \\ S^{A}_{\lambda_3} &= 0.000128 & S^{A}_{\mu_1} &= 0.072 \end{aligned} \qquad (26)$$

and $$S^{A}_{\mu_2} = 0.02$$

From the absolute magnitudes it is seen that overloads and recovery from overloads have the greatest effect on the network availability. Hence any network modification or design changes which would tend to increase time-to-overload and decrease time-to-recovery from overload would most efficaciously increase the network availability.

5. CONCLUSIONS

To formulate the Availability function for any given computer network consideration must be given not only to network equipment failures but also to other relevant factors such as channel interference, computer equipment overloads and channel overloads.

Markovian model of network Availability function can be formulated from n discrete-states and concommitant number of transition probabilities representing such events as network equipment failure, interference, overloads, and recovery from failure, interference, and overloads. The formulation leads to a set of ordinary-first order differential equations which can be solved directly for the steady state Availability function by the application of LaPlace transforms and its final value theorem. The function so formulated can be subjected to analysis of sensitivity to variations of the functions' parameters.

If meaningful evaluation of computer network Availability function and its Sensitivity is to be attempted, it is necessary to obtain estimates of transition probabilities from the systematically collected network random variables time-to-failure, time-to-overload, time-to-interference, and the time-to-recovery from failure, overload and interference.

6. REFERENCES

1. CRAMER, H. and LEADBETTER, M. R., 1967, Stationary and Related Stochastic Processes, John Wiley & Sons, New York.

2. GRUBB, D. S. and COTTON, I. W., 1975, "Criteria for the Performance Evaluation of Data Communication Services for Computer Networks," Washington, D. C., National Bureau of Standards Tech Note 882.

3. LOCKS, M. O., 1973, Reliability, Maintainability and Availability Assessment, Hayden Book Company, New Jersey.

4. NATIONAL LIBRARY OF MEDICINE, 1972, "Request for Proposal NML-72-102 ADP/Telecommunications Support for a Remote-Access Bibliographic Information Retrieval System."

5. REGULINSKI, T. L., Proceedings of the 1970 Annual Symposium of Reliability, "Systems Maintainability Modeling," New York, IEEE Catalog No. 70C 2-R.

6. TOMOVIC, R., 1963, Sensitivity Analysis of Dynamic Systems, McGraw-Hill Book Company, New York.

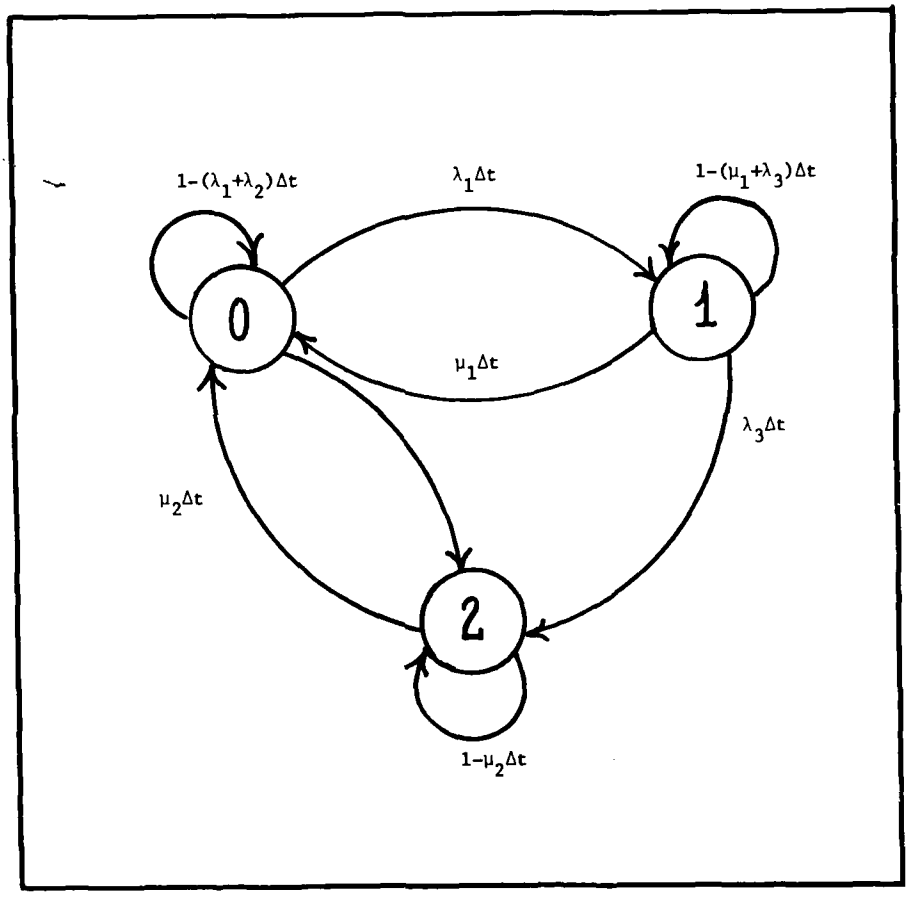


Fig. 1: State Transition Diagram

DISCUSSION

Questioner Unknown

Votre formule d'équilibre synthétique est obtenue portée à l'infini. Est-ce que vous pouvez nous donner un ordre de grandeur du vrai temps pour obtenir cette formule?

Est-ce qu'il s'agit d'une minute, d'une heure, d'un jour?

Author's Reply

Time is really irrelevant. You can collect the data of the time to failure or time to overload simply because you are seeking the probability density function governing the time to overload/interference or repair. From the probability density function governing this process all you want to do is to obtain the transition rates. So whether this is over a given second or hour (e.g. with the microprocessor it would be in seconds, for the time to repair in hours) you are interested in the rate because the transition probability Pij is in fact a rate. The problem is that these rates are required to be constant, this is a constraint of the entire model.

ESTIMATION RAPIDE DES TROIS PARAMETRES D'UNE LOI DE WEIBULL

par

Robert ATTULY et Christian BERTIN
Société Nationale Industrielle AEROSPATIALE
Division des Systèmes Balistiques et Spatiaux
78130 LES MUREAUX - FRANCE

RESUME

L'estimation des trois paramètres de la loi de Weibull est une opération délicate, particulièrement dans le cas d'échantillons de faible taille. La méthode du maximum de vraisemblance (M.L.E.) conduit à un système d'équations dont la résolution nécessite de puissants moyens de calcul ; de plus, la convergence des algorithmes n'est pas garantie en raison de la forme particulière de la fonction de vraisemblance.

La méthode de "saturation" proposée ici utilise la "quasi-exhaustivité" d'un petit nombre de statistiques associées à l'échantillon et donne des résultats voisins de ceux obtenus par la méthode du M.L.E., lorsque cette dernière peut être utilisée, tout en présentant les avantages suivants :

- une mise en oeuvre facile, graphique ou analytique,

- la prise en compte de la taille de l'échantillon,

- seule la statistique associée à l'échantillon doit être conservée en mémoire.

Cette méthode présente enfin l'avantage d'être bien "prévenue" (au sens de E.T. JAYNES) contre toute erreur relative à l'hypothèse de loi. Ce point conduit au concept d'hypothèse paramétrique affaiblie et, en particulier, à la loi tétra-paramétrique généralisant à la fois la loi de Weibull, la loi Gamma et la loi Normale.

1. INTRODUCTION

Malgré l'effort des théoriciens, le problème de l'estimation statistique est loin d'être résolu. L'utilisateur est donc amené à choisir dans l'ensemble des procédures d'estimation proposées par la théorie, celle qui lui semble la mieux adaptée, à partir de critères dont le caractère pratique et parfois subjectif n'est pas à nier. Certaines classes d'estimations douées de propriétés intéressantes se dégagent :

- maximum de vraisemblance (M.L.E.),

- moindres carrés,

- moments,

aucune cependant ne parvient à s'imposer. C'est dans ce cadre que s'est posé le problème de l'estimation des trois paramètres de la loi de Weibull :

$$W(x/a,b,c) = 1 - \exp-\left(\frac{x-a}{b}\right)^c$$

Les difficultés soulevées par la résolution de ce problème, beaucoup plus simple lorsque le paramètre de position a est connu, nous ont conduit à une méthode d'estimation à notre connaissance inédite et présentant certains aspects pratiques et théoriques intéressants :

1°) simplicité et rapidité de mise en oeuvre,

2°) adaptation satisfaisante aux petits échantillons,

3°) aspect non prévenu.

Ces points seront développés plus loin.

Les auteurs ne prétendent nullement que cette méthode est préférable à toutes celles déjà existantes, mais espèrent simplement qu'elle sera de quelque utilité pour d'autres utilisateurs.

Le point 3°) revêt toute son importance si l'on convient d'accepter avec Norman L. JOHNSON et Samuel Kotz que l'utilisation de la loi de Weibull ne repose en fait sur aucune justification théorique, mais sur des considérations pratiques de commodité d'emploi (réf. 3, p. 251). L'utilisateur doit donc savoir dans quelle mesure cette commodité d'emploi affecte, sinon dénature, les résultats statistiques auxquels il parvient à la suite de l'expérimentation (sensibilité à l'hypothèse de loi).

Précisons que la méthode de saturation proposée ici ne concerne, pour l'instant, que les essais complets. Sa généralisation aux cas censurés (types I et II) est envisagée.

Notons, pour terminer, que le concept de saturation développé plus loin nous semble plus important que la méthode retenue, cette dernière devant certainement pouvoir être améliorée par un choix plus judicieux des statistiques utilisées.

2. RAPPELS SUR LA LOI DE WEIBULL

2.1 LA LOI DE WEIBULL

$$W(x/a,b,c) = 1 - \exp -\left(\frac{x-a}{b}\right)^c \qquad x \geq a$$

$$w(x/a,b,c) = \frac{c}{b}\left(\frac{x-a}{b}\right)^{c-1} \exp -\left(\frac{x-a}{b}\right)^c$$

dépend de trois paramètres :

a : paramètre de position,

$0 < b$: paramètre d'échelle,

$0 < c$: paramètre de forme.

Ses moments centrés sur a s'expriment à l'aide de la fonction Gamma :

$$\tilde{M}_n = E\left\{\left(\frac{x-a}{b}\right)^n\right\} = (ns)!$$

où :

$s = 1/c \quad , \quad x! = \Gamma(1+x)$

En particulier pour les premiers moments :

$$M = a + bs!$$

$$\Sigma = b\sqrt{(2s)! - (s!)^2}$$

$$\gamma_1 = \left[(3s)! - 3(2s)!\,s! + 2(s!)^3\right]\Sigma^{-3} \qquad \text{(asymétrie)}$$

$$\gamma_2 = \left[(4s)! - 4(3s)!\,s! + 6(2s)!\,(s!)^2 - 3(s!)^4\right]\Sigma^{-4} - 3 \qquad \text{(applatissement)}$$

Ces différentes fonctions sont représentées sur la planche 1.

2.2 On représente habituellement l'évolution du graphe de $w(x/a,b,c)$ lorsque a et b sont fixés. Il nous a semblé plus instructif (a n'étant pas connu dans le problème d'estimation), d'étudier ces mêmes évolutions sous forme réduite (M et Σ fixés). L'expression de w est sous forme réduite :

$$w(x/a,b,c) = \frac{1}{\Sigma}\,\frac{\rho}{s}\,(s!)^{1/s}\,(1+\rho u)^{1/s-1}\,\exp -\left[s!\,(1+\rho u)^{1/s}\right]$$

avec :

$$u = \frac{x-M}{\Sigma}$$

$$\rho = \frac{\Sigma}{M-a} = \sqrt{\frac{(2s)!}{(s!)^2} - 1}$$

Les courbes obtenues (cf. planche 2) appellent certains commentaires :

2.2.1 pour C < 1 :

la densité n'est pas finie pour x = a

2.2.2 pour C = 1 :

on obtient la distribution de Poisson (exponentielle réduite), seul cas où la densité en a est finie non nulle

2.2.3 pour 1 < C < 2 :

la densité est nulle en a avec une pente infinie

2.2.4 pour C = 2 :

on obtient la distribution de Rayleigh

2.2.5 pour C > 2 :

la densité est nulle et à pente nulle en a. On remarquera que pour C compris entre trois et dix, la densité est très voisine de la densité normale (en pointillé sur la planche 2)

2.2.6 pour C très grand :

la densité tend vers une limite qui est la loi doublement exponentielle ou loi des valeurs extrêmes qui s'écrit, toujours sous forme réduite :

$$f(x) = \frac{\pi}{\Sigma\sqrt{6}}\,\exp\left(\frac{\pi}{\sqrt{6}}u - \gamma\right)\,\exp\left[-\exp\left(\frac{\pi}{\sqrt{6}}u - \gamma\right)\right]$$

où :

$\gamma = .57722$ (constante d'Euler)

On peut déjà dégager un certain nombre de remarques concernant l'estimation des trois paramètres a, b et c de la loi de Weibull à partir d'un échantillon :

$$\mathcal{X} = (X_i) \quad i = 1,n$$

de moyenne M, d'écart type Σ :

- ce n'est que pour $C \geqslant 1$ que la densité est bornée sur $\mathbb{R}$ et pour $C > 2$ que cette densité est continuement dérivable sur $\mathbb{R}$: toutes les méthodes d'estimation reposant sur la "régularité" de la densité w ne seront, à priori, vraiment efficaces que pour $C > 2$ et totalement inopérantes pour $C < 1$ (c'est le cas pour le M.L.E., voir chapitre 3).

- lorsque $C < 2$, le paramètre a de position joue un rôle important dans la forme de la densité, rôle qui devient secondaire lorsque c augmente. De plus, pour un échantillon de taille donnée n, la probabilité de trouver la plus petite réalisation

$$\check{X}_n = \mathrm{Inf}_{i=1,n}(X_i)$$

voisine de a augmente lorsque c diminue. On sait (réf. 3, p. 256) que pour $0 < C < 1$:

$\check{X}_n$ est justement un estimateur "superefficient" de a. On retiendra que l'information contenue dans $\check{X}_n$ est d'autant plus importante que C est faible.

- pour les valeurs moyennes de C, la distribution est proche de la distribution normale, le paramètre a ne joue qu'un rôle théorique ($a \to -\infty$) que seuls des échantillons de grande taille pourront mettre en évidence. L'information semble, en dehors de M et Σ, se rattacher à l'asymétrie γ_1, de la courbe, celle-ci s'annulant pour $C \simeq 3{,}6$.

  Toujours est-il que pour les échantillons de faible taille, il sera illusoire de vouloir déterminer C avec précision.

- pour les fortes valeurs de C ($C > 10$), les densités sont très voisines et on acceptera la loi doublement exponentielle comme une approximation d'autant plus raisonnable que la taille de l'échantillon est faible.

2.3 La fonction de répartition $F_{\check{X}_n}$ de la plus petite réalisation $\check{X}_n$ d'un échantillon $\mathcal{X} = (X_i)_{i=1,n}$ de taille n (réalisations indépendantes et de répartition commune F) est :

$$F_{\check{X}_n} = 1 - (1-F)^n$$

Dans le cas où F est la répartition de Weibull :

$$F = W(x/a,b,c)$$

$$F_{\check{X}_n} = W\left(x/a, \frac{b}{n^s}, C\right) \qquad (s = 1/c)$$

est aussi une répartition de Weibull de même paramètres de position et de forme, celui d'échelle étant divisé par n^s. En particulier, la moyenne $\check{M}_n$ et l'écart type $\check{\Sigma}_n$ de $\check{X}_n$ sont :

$$\check{M}_n = a + \frac{b}{n^s}\,s! \iff \check{M}_n - a = \frac{M-a}{n^s}$$

$$\check{\Sigma}_n = \frac{b}{n^s}\sqrt{(2s)! - (s!)^2} \iff \check{\Sigma}_n = \frac{\Sigma}{n^s}$$

On retrouve mathématiquement le fait que lorsque C est faible ($C < 1$), $\check{X}_n$ est peu dispersé et proche de a dont il est un bon estimateur.

2.4 Les moments généralisés de la loi de Weibull s'expriment à partir des dérivées de la fonction Gamma :

$$\tilde{M}_{n,r,0} = E\left\{\left(\frac{x-a}{b}\right)^n \mathrm{Log}^r\left(\frac{x-a}{b}\right)\right\} = s^r\,\tilde{M}_n\,\frac{1}{(ns)!}\left[(ns)!\right]^r$$

En particulier (n = 0) :

$$M_{\mathrm{Log}(x-a)} = \mathrm{Log}\,b + \Psi(1).s = \mathrm{Log}\,b - \gamma s \qquad (\gamma = .57722 \text{ , C}^{te}\text{ d'Euler})$$

$$\Sigma_{\mathrm{Log}(x-a)} = \sqrt{\Psi'(1)}.s = \frac{\pi}{\sqrt{6}}\,s$$

Ces deux relations (qui utilisent le fait que Log (X - a) est distribué suivant la loi doublement exponentielle) permettent d'estimer, lorsque a est connu, les paramètres b et c de façon satisfaisante (réf. 3, p. 257 et 285).

3. ESTIMATEUR DU MAXIMUM DE VRAISEMBLANCE (M.L.E.)

3.1 RAPPEL SUR LA METHODE

Soient : $f(x/\vec{\theta}) \qquad \vec{\theta} \in (H)$

la densité de probabilité d'une loi dépendant du paramètre $\vec{\theta}$ assujetti au domaine (H) et

$$\mathcal{X} = (X_i)_{i=1,n}$$

un échantillon de n réalisations indépendantes d'une variable aléatoire. La vraisemblance de l'échantillon $\mathcal{X}$ pour la loi f peut être définie par l'expression :

$$\Omega(\mathcal{X}/\vec{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \text{Log } f(X_i/\vec{\theta}) = \overline{\text{Log } f(X_i/\vec{\theta})}$$

On appelle estimateur du maximum de vraisemblance, tout vecteur $\hat{\vec{\theta}}$ vérifiant :

$$\Omega(\mathcal{X}/\vec{\theta}) \leqslant \Omega(\mathcal{X}/\hat{\vec{\theta}}) \qquad \vec{\theta}, \hat{\vec{\theta}} \in (H)$$

Sous réserve d'existence et d'unicité et moyennant certaines hypothèses de régularité de la fonction f sur R x (H), on peut établir les propriétés qui font du M.L.E. un estimateur privilégié. Il est cependant primordial de remarquer que, pour l'essentiel, ces propriétés sont de type asymptotique, c'est-à-dire que le M.L.E. est un estimateur remarquable pour les échantillons de grande taille. Pour les petits échantillons, le M.L.E. conserve certaines propriétés (réf. 5) mais insuffisantes pour lui assurer encore une quelconque suprématie.

Lorsque l'un des paramètres θ_i à estimer délimite en partie le champ (f ≠ 0) de la variable aléatoire, de nombreuses difficultés apparaissent généralement. La fonction de vraisemblance peut déjà ne pas être bornée sur un sous-ensemble θ_∞ de (H) et tout vecteur de θ_∞ maximise de fait Ω et devrait être considéré comme un estimateur du M.L.E. En pratique, on restreint le domaine de définition (H) de $\vec{\theta}$ à un sous-domaine $\overline{(H)}$ de (H) :

$$\overline{(H)} \subset \{\Theta - \Theta_\infty\}$$

sur lequel Ω est majorée. Cette hypothèse sans laquelle le problème n'a aucun sens n'est en pratique presque jamais mentionnée de façon claire et complète ce qui peut conduire à des conclusions inexactes. Ceci est dû au fait que l'on pose le problème du M.L.E. directement en terme dérivé, c'est-à-dire en recherche de maximum local :

$$\overrightarrow{\text{grad}}\,\Omega = 0$$

$[\![\, \partial^2\Omega \,]\!]$ Hessien défini négatif

Le domaine admissible $\overline{(H)}$ étant bien précisé, l'existence et l'unicité de $\hat{\vec{\theta}}$ établie, les propriétés asymptotiques de l'estimateur ne vaudront, en général, que sur une partie seulement de $\overline{(H)}$, ce qui restreint encore le domaine des valeurs du paramètre pour lequel l'estimateur du M.L.E. présente un intérêt particulier.

3.2 CAS DE LA LOI DE WEIBULL

$$W(x/a,b,c) = 1 - \exp - \left(\frac{x-a}{b}\right)^c$$

$$w(x/a,b,c) = \frac{c}{b}\left(\frac{x-a}{b}\right)^{c-1} \exp - \left(\frac{x-a}{b}\right)^c$$

$$\omega(x/a,b,c) = \text{Log } f = \text{Log } c - \text{Log } B + (c-1)\,\text{Log}(x-a) - \frac{(x-a)^c}{B}$$

où : $B = b^c$

(Rappelons que l'estimateur du M.L.E. est un invariant par changement de paramètre). Pour un échantillon $\mathcal{X}$:

$$\mathcal{X} = (X_i)_{i=1,n}$$

$$\Omega(\mathcal{X}/a,b,c) = \frac{1}{n}\sum_{i=1}^{n} \omega(X_i/a,b,c) = \overline{\omega(X_i/a,b,c)}$$

$$= \text{Log } c - \text{Log } B + (c-1)\,\overline{\text{Log}(X_i - a)} - \frac{1}{B}\,\overline{(X_i-a)^c} \qquad (1)$$

On remarque que pour C < 1, la fonction n'est pas bornée :

$$0 < c < 1$$

$$a \to \check{X}_n = \underset{i=1,n}{\text{Inf}}(X_i) \implies \Omega \to +\infty$$

On cherchera donc à résoudre le problème sur :

$\overline{(H)}$: $a \leqslant \check{X}_n$

$b > 0$

$c \geqslant 1$ (et non $C > 0$ comme on le voit très souvent)

AD-A080 301 ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT--ETC F/G 9/5
AVIONICS RELIABILITY, ITS TECHNIQUES AND RELATED DISCIPLINES.(U)
OCT 79 M C JACOBSEN

UNCLASSIFIED AGARD-CP-261 NL

2 OF 8
AD
A080 301

Tout extremum local de Ω sur $\overline{(H)}$ vérifie :

$$\frac{\partial \Omega}{\partial a} = -(c-1)\,\overline{(X_i-a)^{-1}} + \frac{c}{B}\,\overline{(X_i-a)^{c-1}} = 0 \qquad (2\,a)$$

$$\frac{\partial \Omega}{\partial B} = -\frac{1}{B} + \frac{1}{B^2}\,\overline{(X_i-a)^{c}} = 0 \qquad (2\,b)$$

$$\frac{\partial \Omega}{\partial c} = \frac{1}{c} + \overline{\mathrm{Log}\,(X_i-a)} - \frac{1}{B}\,\overline{(X_i-a)^{c}\,\mathrm{Log}\,(X_i-a)} = 0 \qquad (2\,c)$$

Nous examinerons rapidement le cas où a est connu puis le cas plus complexe où a est à estimer.

3.2.1 CAS OU LE PARAMETRE DE POSITION a EST CONNU ($a < \overset{\vee}{X}_n$)

Dans ce cas, la solution du problème existe et est unique. Pour le démontrer, nous utiliserons la propriété suivante :

soit $A(x)$ et $B(x)$ deux fonctions monotones, alors la différence

$$\overline{A(X_i).B(X_i)} - \overline{A(X_i)}\,.\,\overline{B(X_i)}$$

est positive ou négative suivant que A et B varient dans le même sens ou en sens contraire (cette propriété bien connue dans le cas linéaire s'établit simplement pour des distributions continues ou discrètes en remarquant que A et B sont les dérivées de fonctions positivement ou négativement convexes).

3.2.1.1 EXISTENCE ET UNICITE DE LA SOLUTION, METHODE DE RESOLUTION

En posant

$$\lambda = \overline{\mathrm{Log}\,(X_i-a)}$$

$$Y_i = (X_i-a)\exp(-\lambda) \qquad \left(\Longrightarrow \overline{\mathrm{Log}\,Y_i} = 0\right)$$

les équations (2 b) et (2 C) se réduisent au système :

$$B = \overline{(X_i-a)^{c}} \qquad (3.b)$$

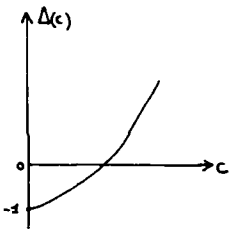
$$\Delta(c) = c\,.\,\overline{Y_i^{c}\,\mathrm{Log}\,Y_i} - \overline{Y_i^{c}} = 0 \qquad (3.c)$$

La fonction $\Delta(C)$ vérifie :

$$\Delta'(c) = c\,.\,\overline{Y_i^{c}.\mathrm{Log}^2 Y_i} \geq 0$$

$$\Delta''(c) = \overline{Y_i^{c}\,\mathrm{Log}^2 Y_i} + c\,.\,\overline{Y_i^{c}\,\mathrm{Log}^3 Y_i} = \overline{Y_i^{c}\,\mathrm{Log}^2 Y_i} + c\,.\,\overline{(Y^{c}.\mathrm{Log}^2 Y_i).\mathrm{Log}\,Y_i} \geq 0$$

$$\Delta(0) = -1$$



c'est-à-dire que Δ est monotone croissante positivement convexe, négative pour C = 0. L'équation (3 C) admet donc toujours une et une seule solution $\hat{C}$. Cette dernière peut être obtenue très rapidement par l'algorithme de Newton-Raphson :

$$c_{k+1} = c_k - \frac{\Delta(c_k)}{\Delta'(c_k)} = c_k - \frac{c_k\,\overline{Y_i^{c_k}\,\mathrm{Log}\,Y_i} - \overline{Y_i^{c_k}}}{c_k\,.\,\overline{Y_i^{c_k}.\mathrm{Log}^2 Y_i}}$$

en prenant Co = 1 comme valeur initiale, ou mieux (cf. § 2.4) :

$$c_o = \frac{\pi}{\sqrt{6}}\,\frac{1}{\sqrt{\overline{\mathrm{Log}^2 Y_i}}}$$

La solution $\hat{B}$ associée est donnée par (3 b).

3.2.1.2 SOLUTION CORRESPONDANT A UN MAXIMUM DE Ω

En effet, les termes diagonaux du Hessien de Ω s'écrivent compte tenu des relations (2 C) et (3 C) :

$$\frac{\partial^2 \Omega}{\partial B^2} = -\frac{1}{\hat{B}^2} < 0$$

$$\frac{\partial^2 \Omega}{\partial c^2} = -\left[\frac{1}{\hat{c}^2} + \frac{\overline{(X_i-a)^{\hat{c}}.\mathrm{Log}^2(X_i-a)}}{\hat{B}}\right] < 0$$

De plus, le déterminant :

$$D = \frac{\partial^2 \Omega}{\partial B^2}\,\frac{\partial^2 \Omega}{\partial c^2} - \left(\frac{\partial^2 \Omega}{\partial B\,\partial c}\right)^2 = \frac{\partial^2 \Omega}{\partial B^2}\,\frac{\partial^2 \Omega}{\partial c^2} - \left[\frac{\overline{(X_i-a)^c \log(X_i-a)}}{B^2}\right]^2$$

s'exprime après simplification par :

$$D = \frac{1}{\hat{B}^3}\;\overline{(X_i-a)^{\hat{c}}}\left[\log(X_i-a) - \overline{\log(X_i-a)}\right]^2 \qquad > 0$$

Le Hessien est donc défini négatif et la valeur de Ω correspond bien à un maximum local. De plus, sur les frontières :

$$c \to +\infty \text{ ou } B \to 0 \implies \Omega \sim -\frac{\overline{(X_i-a)^c}}{B} \to -\infty$$

$$c \to 0 \text{ ou } B \to +\infty \implies \Omega \sim (\log c - \log B) \to -\infty$$

La solution $(\hat{B}, \hat{C})$ est donc un maximum global sur le domaine : $b > 0$ - $C > 0$.

Il est à remarquer que lorsque a est connu, rien n'interdit à C d'être inférieur à 1.

3.2.2 CAS GENERAL (a INCONNU)

Ce cas est plus complexe et l'existence d'une solution au système (2) n'est pas assurée. Soient B^* et C^* les deux fonctions de a solutions de (2 b) et (2 C). Leur existence et leur unicité ont été établies en (2.1). Si une solution à (2) existe, elle se trouve donc nécessairement sur la ligne de crête :

$$B = B^*(a)$$

$$c = c^*(a)$$

où Ω prend les valeurs :

$$\Omega^*(a) = \Omega\left(a, B^*(a), c^*(a)\right)$$

et résoudre (2) revient à résoudre l'équation :

$$\Omega^{*\prime}(a) = -(c^*-1)\,\overline{(X_i-a)^{-1}} + \frac{c^*}{B^*}\,\overline{(X_i-a)^{c^*-1}} = 0$$

Pour $C^* \leqslant 1$, $\Omega^{*\prime} > 0$, on retrouve le fait que s'il existe une solution $(\hat{a}, \hat{b}, \hat{c})$, on aura nécessairement : $\hat{c} > 1$

3.2.2.1 ETUDE DE LA FONCTION C*(a)

C'est la solution de l'équation (3 C). Sa dérivée est, en reprenant les notations du (2.1.1) :

$$c^{*\prime}(a) = -c^*\,\frac{\overline{\log Y_i \cdot Y_i^{c^*-1}\, Y_i'}}{\overline{Y_i^{c^*}\log^2 Y_i}}$$

où :

$$Y_i' = -\exp(-\lambda) + Y_i\,\overline{(X_i-a)^{-1}}$$

- pour $C \geqslant 1$

$\overline{\log Y_i \cdot Y_i^{c-1} \cdot Y_i'} \geqslant 0$ (fonction croissante (de Y) avec $\overline{\log Y_i} = 0$)

- pour $C < 1$

$$\log Y_i \cdot Y_i^{c-1} \cdot Y_i' = \overline{\log Y_i \cdot \left(-\exp(-\lambda)\cdot Y_i^{c-1}\right)} + \overline{(X_i-a)^{-1}} \cdot \overline{Y_i^c \log Y_i}$$

$\geqslant 0$

D'autre part, C* vérifie l'identité :

$$\frac{1}{c} + \overline{\log(X_i-a)} - \frac{\overline{(X_i-a)^c \log(X_i-a)}}{\overline{(X_i-a)^c}} = 0$$

qui peut se mettre sous la forme :

$$\frac{1}{c} + \overline{\log\left(1 - X_i/a\right)} - \frac{\overline{(1-X_i/a)^c \log(1-X_i/a)}}{\overline{(1-X_i/a)^c}} = 0$$

Lorsque $a \longrightarrow \check{X}_n$, C* positive (cf. § 2.1.1) décroissante tend vers une limite qui ne peut être que 0 (identité sous la lère forme) et lorsque $a \longrightarrow -\infty$, C tend vers $+\infty$ (identité sous la 2ème forme).

En résumé, C*(a) est monotone décroissante non majorée et tend vers 0 lorsque $a \longrightarrow \check{X}_n$. L'équation :

$C^*(a) = 1$

admet donc une solution unique a_u que l'on obtiendra facilement en appliquant l'algorithme de Newton-Raphson à la fonction :

$$f(a) = \overline{(X_i - a)\,\mathrm{Log}(X_i - a)} - \overline{(X_i - a)}\cdot\overline{\mathrm{Log}(X_i - a)} - \overline{(X_i - a)}$$

croissante, positivement convexe :

$$f'(a) = \overline{(X_i - a)}\cdot\overline{(X_i - a)^{-1}} \geqslant 0$$

$$f''(a) = -\overline{(X_i - a)^{-2}} + \overline{(X_i - a)}\;\overline{(X_i - a)^{-2}} \geqslant -\overline{(X_i - a)^{-1}} + \overline{(X_i - a)(X_i - a)^{-2}} = 0$$

Remarquons que pour $a = a_u$ $\left(C^*(a_u) = 1\right)$:

$$\Omega^{*\prime}(a_u) = \frac{1}{(X_i - a_u)} > 0$$

$$\Omega^*(a_u) = -1 - \mathrm{Log}\left(\overline{X} - a_u\right)$$

3.2.2.2 ALLURE DE Ω

Plaçons-nous dans le plan (a,c) en supposant vérifié (2 b) :

$$B = \overline{(X_i - a)^c}$$

l'expression (1) de devient :

$$\Omega = \mathrm{Log}\,c - \mathrm{Log}\left[\overline{(X_i - a)^c}\right] + (c-1)\,\overline{\mathrm{Log}(X_i - a)} - 1 \qquad (4)$$

- pour C # 1, lorsque $a \longrightarrow \check{X}_n$

$$\Omega \sim (c-1)\,\mathrm{Log}(\check{X}_n - a) \longrightarrow \begin{cases} +\infty \text{ pour } C < 1 \\ -\infty \text{ pour } C > 1 \end{cases}$$

- pour C = 1, Ω s'écrit :

$$\Omega = -\mathrm{Log}\,B - \frac{\overline{(X_i - a)}}{B}$$

$$\frac{\partial \Omega}{\partial a} = \frac{1}{B} > 0$$

$$\frac{\partial \Omega}{\partial B} = \frac{1}{B} - \frac{\overline{(X_i - a)}}{B^2}$$

Le maximum est donc atteint pour les valeurs :

$$a = \check{X}_n$$

$$B = (\overline{X}_i - \check{X}_n)$$

et vaut :

$$\Omega_1 = -1 - \mathrm{Log}\left(\overline{X}_i - \check{X}\right)$$

Ces propriétés sont résumées sur la planche 3 figure 1, le signe >> indiquant la croissance de Ω .

Dans la recherche du maximum de vraisemblance trois cas peuvent se produire (cf. planche 3 figure 2) :

- <u>cas n° 1</u>

Le système (2) n'admet pas de solution. $\Omega^*(a)$ est monotone croissante jusqu'à Ω^*_u . Cependant, $\Omega_u \leqslant \Omega_1$ de sorte que le maximum de vraisemblance Ω_1 est atteint pour

$a = \check{X}_n$

$B = \overline{X}_i - \check{X}_n$

$C = 1$

- <u>cas n° 2</u>

$\Omega^*(a)$ admet un maximum local ($a \neq a_u$) mais ce maximum est dominé par Ω_u lui-même dominé par Ω_1 . La solution cherchée est encore celle ci-dessus.

- cas n° 3

$\Omega^*(a)$ admet un maximum local ($a \neq a_u$) qui est aussi un maximum global (donc $> \Omega_u$). Là encore, il faut comparer ce maximum à Ω_1. Ce n'est que si $\Omega^*_{max} > \Omega_1$ que le triplet trouvé est l'estimateur du M.L.E. ($\hat{a}$, $\hat{b}$, $\hat{c}$).

L'allure de la surface de vraisemblance $\Omega(a.c)$ dans le cas n° 3 est représentée sur la planche 4.

Une étude plus approfondie éliminerait peut être certaines configurations. Ce qu'il importe de retenir, est que dans la recherche du maximum de vraisemblance de la loi de Weibull à trois paramètres :

1°) la recherche de ce maximum n'a de sens que sur le domaine $\overline{(H)}$: $a < \check{X}_n$ - $0 < b$ - $1 \leqslant C$

2°) le maximum existe toujours sur $\overline{(H)}$ mais n'est pas nécessairement solution du système dérivé :

$$\overrightarrow{\text{grad}}\,\Omega = 0$$

3°) si le système dérivé admet une solution, il faut vérifier :

- qu'il s'agit bien d'un maximum local,

- que ce maximum local est global, en particulier que la valeur de Ω est supérieure à :

$$\Omega_1 = -1 - \text{Log}(\bar{X}_i - \check{X})$$

4°) si la fonction de vraisemblance admet un maximum local, elle admet alors nécessairement un point selle (cas n° 2 et 3). Cette remarque, rappelée par G. BROWN (réf. 1), laisse d'emblée sceptique quant aux performances des algorithmes de résolution du M.L.E.

3.3 APPLICATION

L'algorithme proposé par Dallas R. WINGO (réf. 6 et 7) utilise la méthode de "quasi-linéarisation" dérivée de celle de Newton. Il s'agit, en quelque sorte, de minimiser $\|\overrightarrow{\text{grad}}\,\Omega\|^2$ dans le domaine admissible. Compte tenu de ce qui a été dit plus haut (3.2), on comprendra que malgré ses qualités, l'algorithme pose quelques problèmes d'utilisation, surtout dans le cas d'échantillons de taille faible ($n < 100$) et/ou de faibles valeurs de C ($C < 3$). Le taux de rejet devient prohibitif (> 50 %) et en cas de convergence il faut vérifier que l'on a bien affaire à un maximum (au moins local), car il arrive fréquemment que la solution fournie soit le point selle. Il faut cependant reconnaître la bonne tenue de l'algorithme pour les échantillons de taille plus importante et pour les valeurs de C plus élevées.

HARTER H.L. et MOORE A.H. (réf. 7) font état d'un algorithme dont malheureusement nous n'avons pu prendre connaissance ; ils conviennent cependant de rencontrer les mêmes difficultés que plus haut.

Il est à noter (réf. 3 p. 256), en ce qui concerne les faibles valeurs de C, que l'estimateur du M.L.E. ne possède ses propriétés asymptotiques que pour $C > 2$.

Pratiquement, il convient de remarquer que la mise en oeuvre des algorithmes de recherche du M.L.E. nécessite de puissants moyens de calculs. Le temps C.P.U. sur un processeur puissant varie de quelques secondes à plusieurs minutes suivant les cas. De plus, et par principe, il faut mémoriser la totalité de l'échantillon, ce qui peut dans certains cas poser des problèmes de place mémoire.

3.4 CONCLUSION

Sous les trois aspects, théorique, algorithmique et pratique, la méthode du M.L.E. appliquée à l'estimation des trois paramètres d'une loi de Weibull présente de nombreux inconvénients. Nous ne doutons pas que certains puissent être supprimés (où le soient déjà). Néanmoins, d'autres demeureront, par principe même.

Enfin, il convient de revenir sur l'hypothèse de loi. On a vu plus haut (§ 1) que le choix de la loi de Weibull repose davantage sur une certaine commodité d'emploi que sur des justifications théoriques (sauf il est vrai de rares exceptions qui relèvent du cas d'école). Dès lors, la méthode du M.L.E., étroitement liée à une distribution hypothétique :

$$\Omega = \frac{1}{n} \sum_{i=1}^{n} \text{Log}\, f(x_i / \vec{\theta})$$

est-elle justifiable ? Il est à prévoir que pour les échantillons de très faible taille, la coloration paramétrique sera très forte. Ainsi, par exemple, les deux premiers moments statistiques (M, Σ) de l'échantillon seront remplacés de fait par les moments théoriques de la loi estimée, la différence ne se justifiant que par l'hypothèse de loi.

4. METHODES DES MOMENTS

Soient $\mathcal{X} = (X_i)_{i=1,n}$, un échantillon de taille n et a un réel quelconque. Les moments statistiques centrés sur a sont les quantités :

$$M_k = \frac{1}{n} \sum_{i=1}^{n} (X_i - a)^k = \overline{(X_i - a)^k}$$

Cette définition peut être généralisée en posant

$$M_{k+\tau,a} = \overline{(X_i - a)^k \, \mathrm{Log}^{\tau} (X_i - a)}$$

où k et τ sont des réels ($\tau > 0$). Ces moments généralisés n'existent pas toujours. Pour k non entier et/ou τ non nul, il est nécessaire que :

$$a \leqslant \check{X}_n = \mathrm{Inf}(X_i)_{i=1,n}$$

pour que le moment correspondant soit défini.

Soit $f(x, \vec{\theta})$ la densité de probabilité des X_i (supposés indépendants). On appelle méthode des moments (au sens strict) toute méthode identifiant certains moments statistiques de l'échantillon aux espérances mathématiques respectives pour la loi f, de façon à avoir autant d'équations que de paramètres à estimer. Lorsque le champ de la variable aléatoire est $\mathbb{R}$, on choisit généralement a = 0, lorsque le champ est du type $(a_0, +\infty)$, on choisit naturellement $a = a_0$. La méthode dépend donc du choix (judicieux) des couples de valeurs (k, τ). Nous rappellerons ici deux exemples classiques :

1°) <u>loi normale N (m, σ)</u>

En identifiant les deux premiers moments : $(a = 0)$

$$M_1 = \overline{X_i} = E\{X\}$$

$$M_2 = \overline{X_i^2} = E\{X^2\}$$

on obtient une méthode des moments qui n'est autre que celle du M.L.E. (ce qui historiquement n'est pas un hasard).

2°) <u>loi Gamma</u>

$$f(x) = \frac{1}{b\,\Gamma(\delta)} \cdot \left(\frac{x-a}{b}\right)^{\delta-1} \exp - \left(\frac{x-a}{b}\right)$$

En identifiant les moments suivants :

$$M_1 = \overline{(X_i - a)} = E\{X - a\}$$

$$M_{-1} = \overline{(X_i - a)^{-1}} = E\{(X-a)^{-1}\}$$

$$M_{+0} = \overline{\mathrm{Log}(X_i - a)} = E\{\mathrm{Log}(X-a)\}$$

on obtient ici encore la méthode du M.L.E.

Ces deux exemples mettent en évidence la relation parfois étroite entre deux méthodes en apparence très différentes. Plus généralement, les équations aboutissant au M.L.E. sont du type :

$$\frac{\partial \Omega}{\partial \theta_k} = 0$$

où :

$$\Omega = \overline{\mathrm{Log}\, f(X_i/\vec{\theta})} = \frac{1}{n} \sum_{i=1}^{n} \mathrm{Log}\, f(X_i/\vec{\theta})$$

Calculons (en supposant remplies les hypothèses de régularité nécessaires) :

$$E\left\{\frac{\partial \Omega}{\partial \theta_k}(\mathcal{X}/\vec{\theta})\right\} = E\left\{\frac{\partial}{\partial \theta_k} \overline{\mathrm{Log}\, f(X_i/\vec{\theta})}\right\} = E\left\{\overline{\frac{\partial}{\partial \theta_k} \mathrm{Log}\, f(X_i/\vec{\theta})}\right\} = \overline{E\left\{\frac{\partial}{\partial \theta_k} \mathrm{Log}\, f(X_i/\vec{\theta})\right\}}$$

$$= E\left\{\frac{\partial}{\partial \theta_k} \mathrm{Log}\, f(\mathcal{X}/\vec{\theta})\right\} = \int_{\mathbb{R}} \frac{\partial}{\partial \theta_k} \mathrm{Log}\, f(x/\vec{\theta}) \cdot f(x/\vec{\theta}) \cdot dx$$

$$= \int_{\mathbb{R}} \frac{\partial f(x/\vec{\theta})}{\partial \theta_k} dx = \frac{\partial}{\partial \theta_k} \int_{\mathbb{R}} f(x/\vec{\theta})\, dx = \frac{\partial}{\partial \theta_k}(1) = 0$$

de sorte que :

$$\frac{\partial \Omega}{\partial \theta_k} = 0 \iff \frac{\partial \Omega}{\partial \theta_k}(\mathcal{X}/\vec{\theta}) = E\left\{\frac{\partial \Omega}{\partial \theta_k}(x/\vec{\theta})\right\}$$

Ainsi, d'une façon générale, la méthode du M.L.E. identifie les valeurs statistiques des dérivées partielles de la vraisemblance aux espérances mathématiques respectives. Or, ces dérivées partielles font intervenir des statistiques de l'échantillon qui, pour la plupart des distributions (Normale, log-Normale, Gamma, Weibull, , double exponentielle), sont les moments généralisés définis plus haut. La méthode du M.L.E. apparaît donc comme une méthode "conservant" des fonctions linéaires des moments généralisés :

$$\alpha_1(\vec{\theta})\, M_{k_1+o.n_1} + \cdots\cdots + \alpha_p(\vec{\theta})\, M_{k_p+o.n_p}$$

Par exemple, dans le cas de la distribution Gamma :

$$f(x) = \frac{1}{b\Gamma(\delta)} \left(\frac{x-a}{b}\right)^{\delta-1} \exp - \left(\frac{x-a}{b}\right)$$

$$\frac{\partial \Omega}{\partial a} = \frac{1}{b} - \delta . M_{-1} \qquad M_{-1} = \overline{(X_i - a)^{-1}}$$

$$\frac{\partial \Omega}{\partial b} = -\frac{\delta}{b} + \frac{M_1}{b^2} \qquad M_1 = \overline{(X_i - a)}$$

$$\frac{\partial \Omega}{\partial \delta} = -\Psi(\delta) - \mathrm{Log}\, b + M_{+o} \qquad M_{+o} = \overline{\mathrm{Log}\,(X_i - a)}$$

ce qui explique que le M.L.E. "conserve", M_{-1}, M_{+o}, M_1.

Le cas de la distribution de Weibull est moins simple :

$$f(x) = \frac{c}{b} \left(\frac{x-a}{b}\right)^{c-1} \exp - \left(\frac{x-a}{b}\right)^{c}$$

$$\frac{\partial \Omega}{\partial a} = (1-c) M_{-1} + \frac{c}{B} M_{c-1} \qquad (B = b^c)$$

$$\frac{\partial \Omega}{\partial B} = -\frac{1}{B} + \frac{M_c}{B^2}$$

$$\frac{\partial \Omega}{\partial c} = \frac{1}{c} + M_{+o} - \frac{1}{B} M_{c+o}$$

On peut cependant affirmer que le M.L.E. "conserve" :

$$M_c = \overline{(X-a)^c}$$

(et de façon très approximative M_{-1} et M_{+o}).

(A cet égard, l'avantage de la loi Gamma sur la loi de Weibull, pour a connu, est clair : l'utilisateur de la première pouvant considérer que l'information contenue dans l'échantillon $\mathcal{X}$ se résume exhaustivement dans M_{+o} et M_1 alors que dans le second cas, il faut considérer M_c, M_{+o}, M_{c+o}, C étant inconnu, c'est-à-dire conserver l'échantillon).

5. METHODE DE SATURATION

5.1 PRINCIPE

Il est d'usage en statistique d'accorder une place prépondérante aux deux premiers moments M 1 et M 2 d'un échantillon (cadre non paramétrique), puis aux moments d'ordre plus élevé M 3 et M 4 (dissymétrie, applatissement). Pour une loi tronquée (à gauche) la plus petite valeur $\check{X}$ de l'échantillon revêt également une certaine importance ($\check{X}$ peut être considéré comme le moment d'ordre $-\infty$: $\lim_{\alpha \to -\infty} \left(\overline{X^\alpha}\right)^{1/\alpha} = \check{X}$)

L'utilisateur amené à faire une hypothèse de loi devra résoudre un compromis, puisque comme il a été dit plus haut, chaque loi privilégie certains moments (ou sommes pondérées de moments) spécifiques et introduit donc une coloration paramétrique sur les autres moments.

L'utilisateur qui considère sur des bases qui lui sont propres que les deux premiers moments sont fondamentaux, sera conduit à adopter une loi Normale, aucune coloration paramétrique n'affectera ces deux quantités. C'est le cas le plus fréquent et un tel choix en pratique se fait naturellement. On remarquera que la loi Normale est la moins prévenue pour ces deux quantités (réf. 4 p. 139). (Le même raisonnement s'applique à la loi Gamma, la moins prévenue (réf. 4 p. 152) pour M_{+o} et M_1).

Cependant, d'autres considérations peuvent lui imposer un type de loi déterminé. Ainsi, en fiabilité mécanique, il est d'usage d'utiliser la loi de Weibull même si les premiers moments et l'inf de l'échantillon sont jugés importants. Le compromis qui semble s'imposer consiste :

1°) à adopter une loi de Weibull,

2°) à estimer les paramètres de cette loi par une méthode des moments conservant les moments jugés importants.

Disposant de trois paramètres, il faut choisir trois moments :

1°) M_1, M_2, M_3 (moyenne, variance, assymétrie)

L'inconvénient de cette méthode est que rien n'impose à la valeur estimée â du paramètre de position d'être inférieure à $\check{X}$.

2°) $M_1, M_2, \check{X}$ (moyenne, variance, inf.)

Cette méthode a l'avantage de prendre en compte la taille de l'échantillon et ainsi de rester valable pour les petites tailles. Cependant, elle n'est particulièrement efficace que pour les faibles valeurs de C, ce qui a été expliqué au § 2.

La méthode de saturation est un compromis entre les deux choix précédents dans lequel on essaie de conserver sensiblement les quatre statistiques ($M_1, M_2, M_3, \check{X}$).

5.2 METHODE

Rappelons (voir § 2) que la loi de Weibull :

$$W(x/a,b,c) = 1 - \exp -\left(\frac{x-a}{b}\right)^c$$

a pour premiers moments (s = 1/C) :

$$M = a + b.s! \qquad s! = \Gamma(s+1)$$

$$\Sigma = b\sqrt{(2s)! - (s!)^2}$$

$$\gamma_1 = \frac{(3s)! - 3(2s)! + 2(s!)^3}{[(2s)! - (s!)]^{3/2}}$$

relations qui peuvent se mettre sous la forme :

$$a = M - \Sigma . u(c) \qquad u(c) = \left[\frac{(2s)!}{(s!)^2} - 1\right]^{-1/2}$$

$$b = \Sigma . v(c) \qquad v(c) = \left[(2s)! - (s!)^2\right]^{-1/2}$$

$$\gamma_1 = g(c)$$

et que si $\mathcal{X}$ est un échantillon de taille n et $\check{X}_n$ sa plus petite réalisation, $\check{X}_n$ suit la loi de Weibull :

$$W\left(x/a, \frac{b}{n^s}, c\right)$$

de sorte que :

$$E\{\check{X}_n\} = a + \frac{b\,s!}{n^s}$$

Etant donné l'échantillon $\mathcal{X} = (X_i)\ i = 1,n$:

1°) on calcule :

$$M^* = \overline{X_i}$$

$$\Sigma^* = \sqrt{\overline{(X_i - M^*)^2}}$$

$$\gamma_1^* = \frac{\overline{(X_i - M^*)^3}}{\Sigma^{*3}}$$

$$\check{X} = \underset{i=1,n}{Inf}(X_i)$$

2°) on fait une première estimation de C et b :

$$\tilde{c} = g^{-1}(\gamma_1^*)$$

$$\tilde{b} = \Sigma^* . v(\tilde{c})$$

3°) on estime a en tenant compte de la valeur de $\check{X}^*$:

$$\hat{a} = \check{X}^* - \frac{\tilde{b}}{n^{1/\tilde{c}}} \cdot \left(\frac{1}{\tilde{c}}\right)!$$

4°) on réestime b et C en tenant compte de l'estimation â de a :

$$\hat{c} = u^{-1}\left(\frac{M^* - \hat{a}}{\Sigma^*}\right)$$

$$\hat{b} = \Sigma^* . v(\hat{c})$$

Cette méthode est très facile à mettre en oeuvre, les fonctions g, v, $u_j(1/c)$! pouvant :

- soit être représentées graphiquement,
- soit être approximées avec la précision désirée par des fractions rationnelles ou des polynômes.

5.3 MISE EN OEUVRE

5.3.1 METHODE GRAPHIQUE

Les courbes représentatives des fonctions d'estimation définies ci-dessus peuvent être tracées une fois pour toutes comme sur les planches 5 et 6 :

- courbe (1) $\gamma_1 = g(c)$
- courbe (2) $\Sigma / b = 1/v(c)$
- courbe (3) $\Sigma / (M-a) = 1/u(c)$
- courbe (4) $(1/c)! / (n^{1/c})$ pour différentes valeurs de n.

L'estimation de a.b et C est alors directe, comme on peut s'en rendre compte par l'exemple ci-dessous.

EXEMPLE

Soit l'échantillon de 30 réalisations d'une loi W (0, 1, 2) dont la statistique représentative est :

$M^* = 0{,}82069$, $\Sigma^* = 0{,}39546$, $\gamma_1^* = 0{,}51320$, $\check{X}^* = 0{,}20862$.

Avec $\gamma_1^* = 0{,}513$, on lit la courbe (1) $\tilde{c} = 2{,}2$ et sur la courbe (2) $\frac{\Sigma}{b} = 0{,}425$ d'où $b = 0{,}9305$.

Avec $\tilde{c} = 2{,}2$ et $n = 30$, on obtient sur la courbe (4) $\frac{s!}{n^s} = 0{,}188$, soit $\hat{a} = 0{,}0337$.

Avec $\Sigma^*/(M^*-\hat{a}) = 0{,}5025$, on lit sur la courbe (3) $\hat{c} = 2{,}0$ qui donne sur la courbe (2) $\frac{\Sigma}{b} = 0{,}48$, soit $\hat{b} = 0{,}82$.

L'estimation est donc : $\hat{a} = 0{,}03$ - $\hat{b} = 0{,}82$ - $\hat{c} = 2$.

Cette méthode est très rapide mais ne permet pas toujours d'atteindre une précision suffisante dans l'estimation. On peut alors lui préférer une méthode numérique moins directe mais plus précise.

5.3.2 METHODE POLYNOMINALE

Une méthode permettant d'obtenir la précision désirée consiste à approximer les fonctions d'estimation par des fractions rationnelles.

Nous proposons ici des approximations permettant d'obtenir une précision de 10^{-2} sur les estimateurs, ce qui semble suffisant pour des échantillons de faible taille :

$$\tilde{c} = \exp(-0{,}2\,\gamma_1)\,(3{,}62 - 0{,}266\,\gamma_1 + 0{,}2453\,\gamma_1^2)/(1 + 0{,}871\,\gamma_1)$$

valable sur l'intervale $-0{,}63764 \leqslant \gamma_1 \leqslant 6{,}6188$

$$\tilde{b} = \Sigma^* \cdot \tilde{c}\left[1 - 0{,}406\,(1/\tilde{c} - 1) - 0{,}386\,(1/\tilde{c} - 1)^2 + 0{,}241\,(1/\tilde{c} - 1)^3\right]$$

valable sur $0{,}5 \leqslant C \leqslant 10$

$$\hat{a} = \frac{\check{X}}{n^{1/\tilde{c}}}\left[1 - 0{,}41\,(1/\tilde{c} - 1) - 0{,}248\,(1/\tilde{c} - 1)^2 + 0{,}161\,(1/\tilde{c} - 1)^3\right]^{-1}$$

valable sur $0{,}5 \leqslant C \leqslant 10$

$$\hat{c} = \left[1 + 0{,}529(\alpha - 1) + 0{,}134\,(\alpha - 1)^2\right]/\alpha\left[1 + 0{,}534\,(\alpha - 1)\right]$$

ou $\alpha = \Sigma^*/(M^* - \hat{a})$ et $0{,}12031 \leqslant \alpha \leqslant 2{,}2361$

puis on réutilise, pour calculer $\hat{b}$:

$$\hat{b} = \Sigma^*\,\hat{c}\left[1 - 0{,}406\,(1/\hat{c} - 1) - 0{,}386\,(1/\hat{c} - 1)^2 + 0{,}241\,(1/\tilde{c} - 1)^3\right]$$

En appliquant cette méthode à l'échantillon traité dans l'exemple précédent, on obtient :

$\hat{a} = 0{,}03$, $\hat{b} = 0{,}89$, $\hat{c} = 2{,}10$

alors que les valeurs exactes obtenues par le calcul sont :

$\hat{a} = 0{,}03455$, $\hat{b} = 0{,}88757$, $\hat{c} = 2{,}08721$.

5.4 PERFORMANCES

L'évaluation des moyennes et écarts types des estimateurs a été faite par une méthode de Monte Carlo sur 500 échantillons, de taille n = 10, 30, 50, tirés d'une loi W (0, 1, C) pour C = 0,5, 1, 2, 3, 4.

Les résultats sont rassemblés dans le tableau ci-dessous :

Paramètres	n	C = 0,5	1	2	3	4
$\bar{\hat{a}}$	10	- 0,49	- 0,32	- 0,31	- 0,28	- 0,22
	30	- 0,07	- 0,08	- 0,08	- 0,11	- 0,13
	50	- 0,02	- 0,03	- 0,05	- 0,07 (0,002)	- 0,09 (0,03)
$\bar{\hat{b}}$	10	2,15	1,41	1,34	1,27	1,21
	30	1,36	1,13	1,12	1,11	1,13
	50	1,24	1,06	1,06	1,07 (0,99)	1,09 (1,03)
$\bar{\hat{c}}/c$	10	1,94	1,73	1,65	1,53	1,38
	30	1,28	1,19	1,16	1,18	1,21
	50	1,17	1,10	1,11	1,12 (1,05)	1,15 (1,09)
$\sigma\,\hat{a}$	10	0,44	0,41	0,70	0,80	0,85
	30	0,06	0,11	0,20	0,38	0,56
	50	0,02	0,05	0,14	0,26 (0,28)	0,40 (0,46)
$\sigma_{\hat{b}}$	10	1,41	0,64	0,76	0,83	0,87
	30	0,55	0,27	0,25	0,40	0,57
	50	0,40	0,18	0,18	0,28 (0,29)	0,41 (0,47)
$\sigma\hat{c}/c$	10	0,94	1,13	1,26	1,16	1,04
	30	0,28	0,30	0,35	0,54	0,72
	50	0,19	0,18	0,24	0,36 (0,39)	0,51 (0,58)

Les biais restent stables pour C > 1 et les variances augmentent légèrement avec C.

A titre de comparaison, on a fourni, entre parenthèses, les performances obtenues par la méthode du M.L.E. pour les seuls cas (C et n suffisamment grands) où l'algorithme (réf. 6) converge pour tous les échantillons. On voit que si les biais obtenus par le M.L.E. sont inférieurs à ceux obtenus par la méthode de saturation, les écarts types sont, par contre, légèrement supérieurs.

Ces résultats sont aussi présentés sous forme de courbes sur les planches 7 à 12.

6. HYPOTHESE PARAMETRIQUE AFFAIBLIE, LOI TETRA-PARAMETRIQUE

Nous avons insisté dans ce qui précède sur le fait que l'estimation paramétrique est étroitement liée à la confiance accordée à l'hypothèse de loi et que, en général, l'utilisateur fera un compromis entre la réalité d'une expérimentation statistique et l'hypothèse en question.

La loi Normale représente en ce sens un certain idéal lorsque le champ de la variable est $\mathbb{R}$. Cependant, dans de nombreux cas, on étudie des variables dont le champ est tronqué au moins d'un côté (fiabilité par exemple), ce qui conduit à rejeter la loi Normale (la loi de Weibull souvent utilisée dans ce cas ne repose malheureusement sur aucun modèle théorique).

Le choix d'un type de distribution pour représenter un phénomène aléatoire est une étape décisive en statistique descriptive. Il peut arriver que la connaissance physique du phénomène soit suffisante pour conduire à une modélisation statistique indiscutable. Cette situation reste en pratique assez rare et dans la plupart des cas, c'est par commodité que l'utilisateur décide d'utiliser tel type de loi plutôt qu'un autre, même s'il tente par la suite de justifier ce choix. Une telle démarche, à priori critiquable, conduit cependant à des résultats très souvent raisonnables sinon précis. Il n'y a là aucun paradoxe :

1°) lorsqu'un type de distribution est particulièrement adapté au domaine d'utilisation, on peut généralement le considérer comme également adapté au processus physique à modéliser en raison justement des propriétés sous-jacentes qui font qu'il est d'un maniement aisé dans le domaine en question (loi de Poisson en fiabilité). Une telle distribution doit être regardée comme une approximation au premier ordre de la réalité (analogie avec la linéarisation des systèmes physiques),

2°) il semble que le choix d'une loi paramétrique n'est pas déterminant s'il ne déforme pas de façon flagrante le phénomène étudié. Ainsi en fiabilité, la loi Gamma généralisée à trois paramètres (Pearson type III) est aussi valable que celle de Weibull à trois paramètres (réf. 3 p. 172). Toutes deux admettent la loi de Poisson comme cas particulier. Une démarche plus satisfaisante pour l'esprit consiste à rejeter toute hypothèse de forme.

L'inconvénient de cette approche non paramétrique est qu'elle fait appel à des outils mathématiques puissants mais délicats et dont la signification physique est souvent obscure, ce qui est dangereux au niveau de l'interprétation des résultats. La supériorité de telles méthodes n'est du reste pas toujours établie.

On conçoit donc l'utilité d'une distribution générale affaiblissant suffisamment les hypothèses de modélisation pour être considérée comme une approche du non-paramétrique conservant les avantages du paramétrique.

La loi tétra-paramétrique, ou Gamma généralisée à quatre paramètres, a pour densité de probabilité :

$$f(x/a,b,c,\delta) = \frac{1}{\Gamma(\delta)} \cdot \left|\frac{c}{b}\right| \cdot \left(\frac{x-a}{b}\right)^{\delta c - 1} \exp - \left(\frac{x-a}{b}\right)^{c}$$

pour :

$$\frac{x-a}{b} \geq 0 \qquad \text{avec} \quad b \neq 0 \,,\; c \neq 0 \,,\; \delta > 0$$

Elle généralise la loi Gamma (C = 1), la loi de Weibull (δ = 1) et tend continuement vers le loi normale ($\delta \to +\infty$) suivant l'expression :

$$f(x) = \frac{1}{\Sigma\sqrt{2\pi}}\, e^{-u^2/2} \cdot \left[1 + \frac{\varepsilon_{bc}}{\sqrt{\delta}}\left(1 - \frac{3}{c}\right) u\left(3 - u^2\right) + \mathcal{O}\left(\delta^{-1/2}\right)\right]$$

où :

ε_{bc} = signe de b.c

$u = \frac{x-M}{\Sigma}$

On trouvera en réf. 3 p. 197 des références bibliographiques assez nombreuses. Il sortirait du cadre de cette communication d'étudier les trop nombreuses propriétés de cette loi. On trouvera sur la planche 13 les formes particulières de la loi tétra-paramétrique pour M et Σ fixés. Notons que lorsque $C \to \pm\infty$, la loi (pour M et Σ fixés) tend vers :

$$f_\delta(x) = \frac{1}{\Sigma} \frac{\sqrt{\Psi'(\delta)}}{\Gamma(\delta)} \exp(\delta y) \exp(-\exp y)$$

où :

$$y = \Psi(\delta) + \varepsilon_{bc} \sqrt{\Psi'(\delta)}\, \frac{x-M}{\Sigma}$$

(ε_{bc} = signe de bxC)

On retrouve pour δ = 1 la loi des valeurs extrêmes de type I (valeurs supérieures ou inférieures suivant le signe de ε_{bc}) ou doublement exponentielle.

Cette loi n'est pas tronquée et tend elle aussi vers la loi normale pour $\delta \to +\infty$. Nous l'appelons sur la planche, loi doublement exponentielle généralisée.

L'avantage essentiel de la loi tétra-paramétrique est qu'elle permet d'évaluer sur le plan théorique, des distances par rapport à trois grands types de lois. L'utilisateur sera ainsi guidé, compte tenu de la taille des échantillons dont il dispose et du domaine dans lequel il se situe, dans son choix en faveur de l'un de ces types (s'il ne désire pas utiliser la loi générale à quatre paramètres). A titre d'exemple, on trouvera en planche 14 la "distance" de la loi tétra-paramétrique à la loi Normale. En fait, on utilise la fonction :

$$D_{F/G} = \int_{\mathbb{R}} F \operatorname{Log} \frac{f}{g} \qquad \text{où G est ici la loi Normale}$$

On remarquera le rôle particulier de C = 3 (pour la rapidité de la convergence) et, pour δ = 1, la proximité de la loi de Weibull et de la loi Normale déjà constatée (§ 2) pour les valeurs moyennes de C.

REFERENCES

(1) BROWN G. (1975)
Comments on "M.L.E. of Weibull parameters by quasi-linearization"
IEE Transactions on reliability, vol R-24, n° 2, June 1975, p. 158

(2) HARTER H.L. and MOORE A.H. (1965)
Maximum likelihood estimation of the parameters of Gamma and Weibull populations from complete and from censored samples
Technometrics, vol 7, n° 4, november 1965, p. 639/643

(3) JOHNSON N.L. and KOTZ S. (1970)
Continuous univariate distribution 1
Houghton Mifflin Company - Boston

(4) MYRON TRIBUS (1969)
Décisions rationnelles dans l'incertain
Masson et Cie, Paris, 1972

(5) SHENTON L.R. and BOWMAN K.O. (1977)
Maximum likelihood estimation in small samples
Griffin's statistical manograph series - London

(6) WINGO D.R. (1972)
Maximul likelihood estimation of the parameters of the Weibull distribution by modified quasi-linearization
IEE Transaction on reliability, vol R-21, n° 2, may 1972, p. 89/93

(7) WINGO D.R. (1973)
Solution of the three-parameters Weibull equations by contrained modified quasi-linearization (progressurly censored samples)
IEE Transaction on reliability, vol R-24, n° 2, june 1975, p. 96/102

LISTE DES PLANCHES

(1) Fonctions des moments de la loi de Weibull

(2) Densités de la loi de Weibull réduite

(3) Propriétés de la fonction de vraisemblance

(4) Allure de la surface de vraisemblance

(5) et (6) Estimation graphique pour la méthode de saturation

(7) à (12) Performances de la méthode de saturation

(13) Lois englobées par la loi tétra-paramétrique

(14) Distance de la loi tétra-paramétrique à la loi Normale

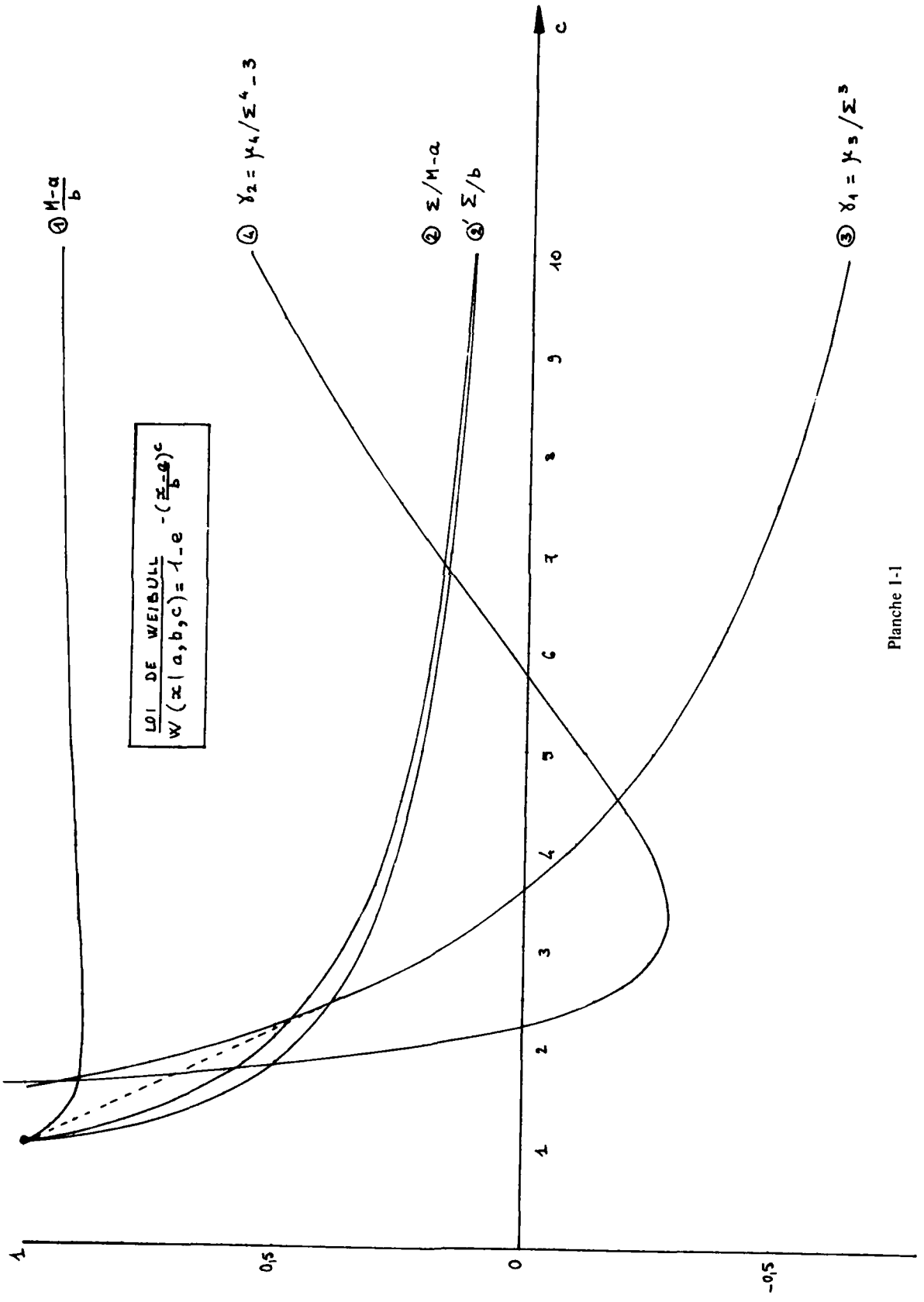

LOI DE WEIBULL
W(x | a,b,c) = 1 - e^-((x-a)/b)^c
① (M-a)/b
② Σ/M-a
②' Σ/b
③ γ1 = μ3/Σ^3
④ γ2 = μ4/Σ^4 - 3
c
1
2
3
4
5
6
7
8
9
10
1
0,5
0
-0,5

Planche 1-1

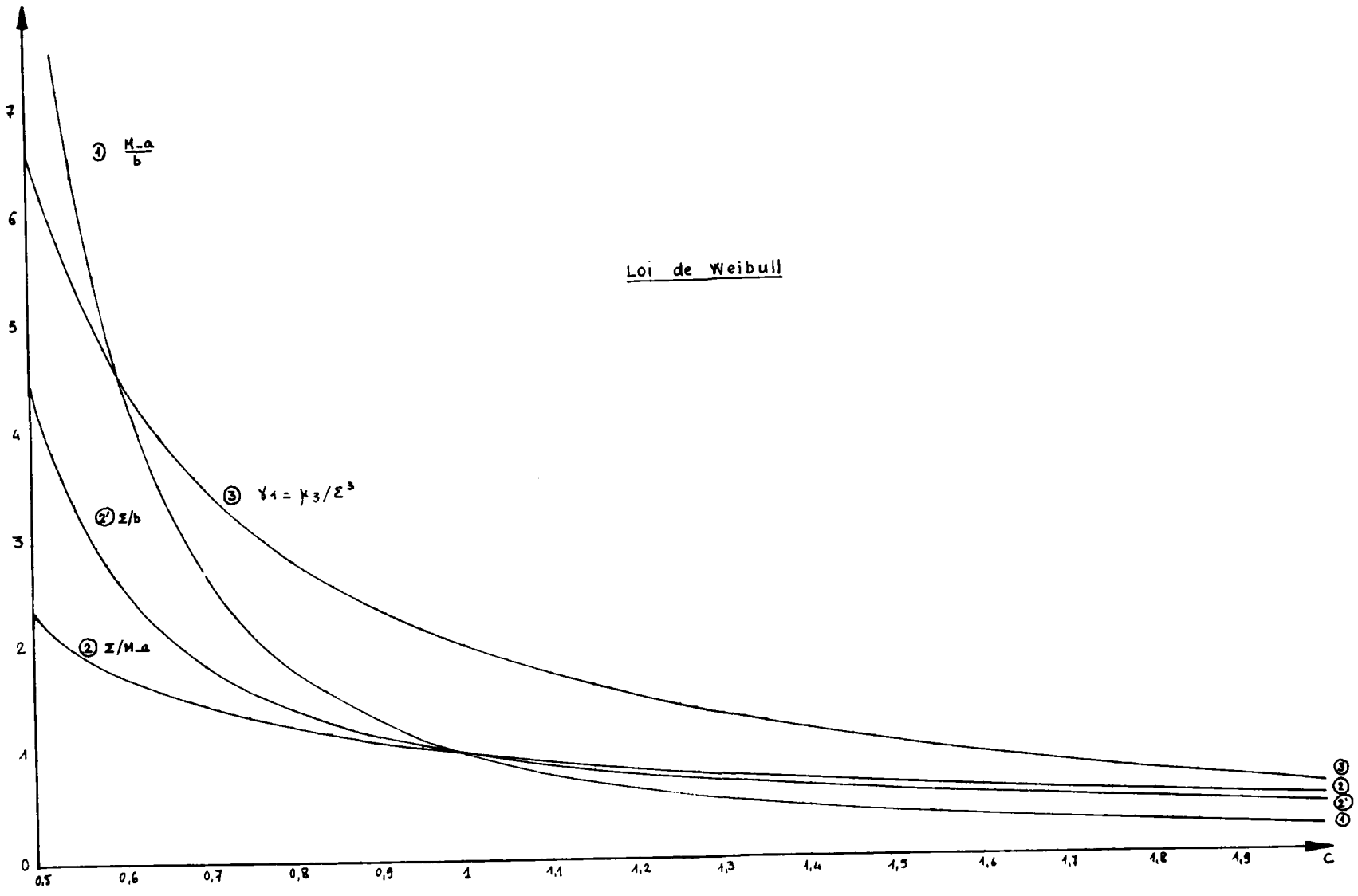

Loi de Weibull
① M-a / b
② Σ/b
② Σ/M-a
③ γ1 = μ3/Σ³
0
1
2
3
4
5
6
7
0,5
0,6
0,7
0,8
0,9
1
1,1
1,2
1,3
1,4
1,5
1,6
1,7
1,8
1,9
C
③
②
②'
④

Planche 1-2

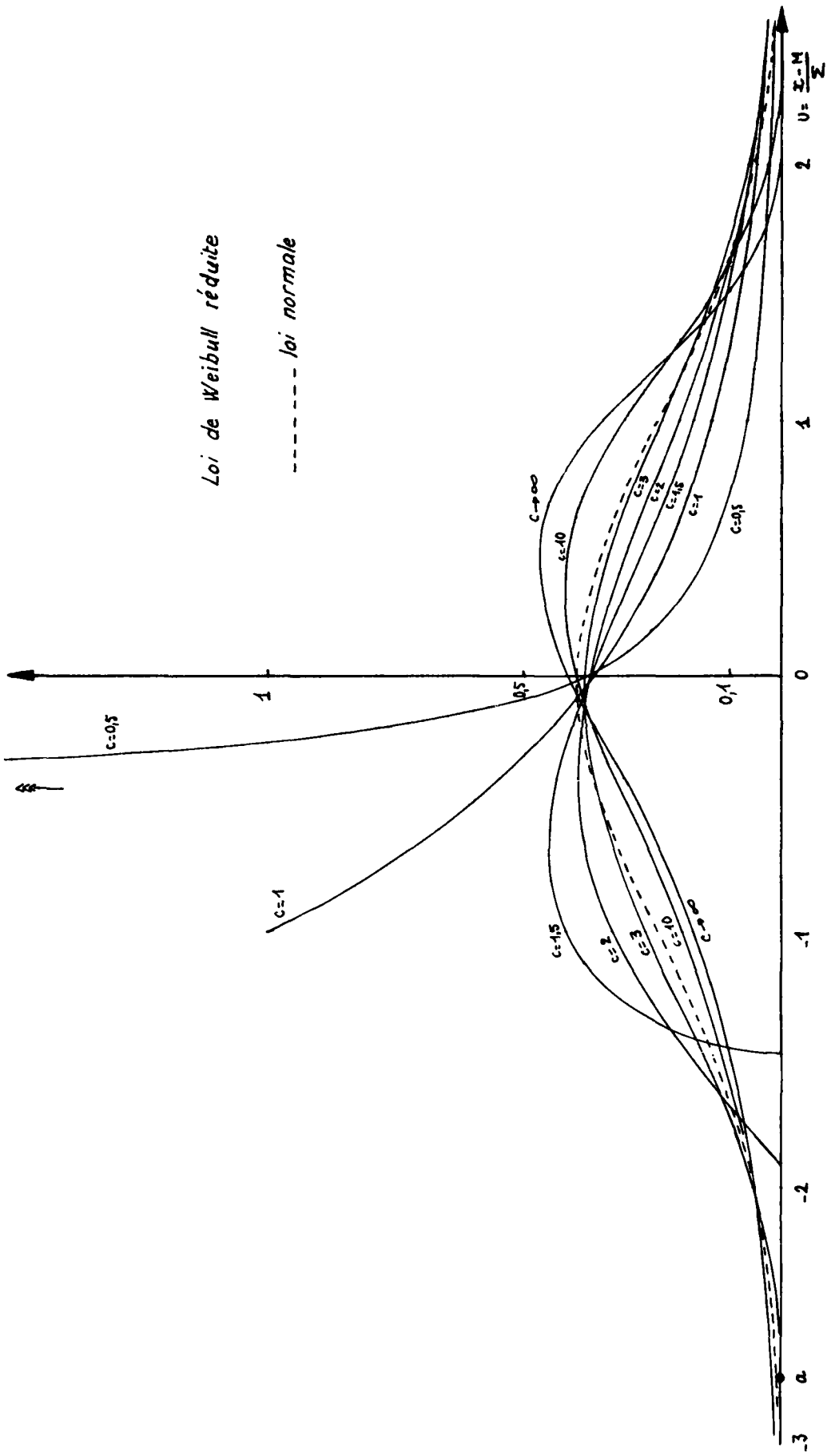

Loi de Weibull réduite
loi normale
c=0,5
1
0,5
0,1
0
1
2
U= x-M/σ
c→∞
c=10
c=5
c=4
c=3
c=2
c=1,5
c=1
c=0,5
c=1
c=1,5
c=2
c=3
c=5
c=10
c→∞
-1
-2
-3

Planche 2

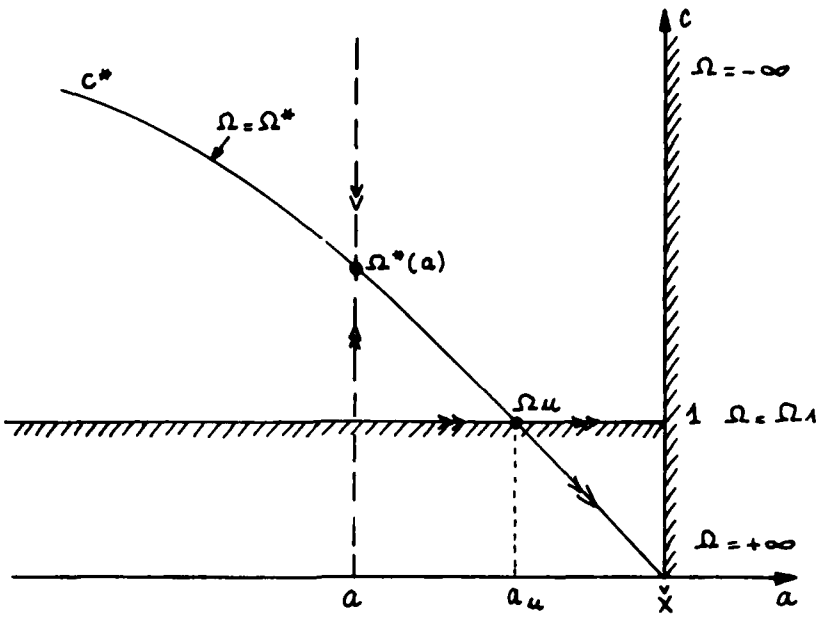


figure 1.

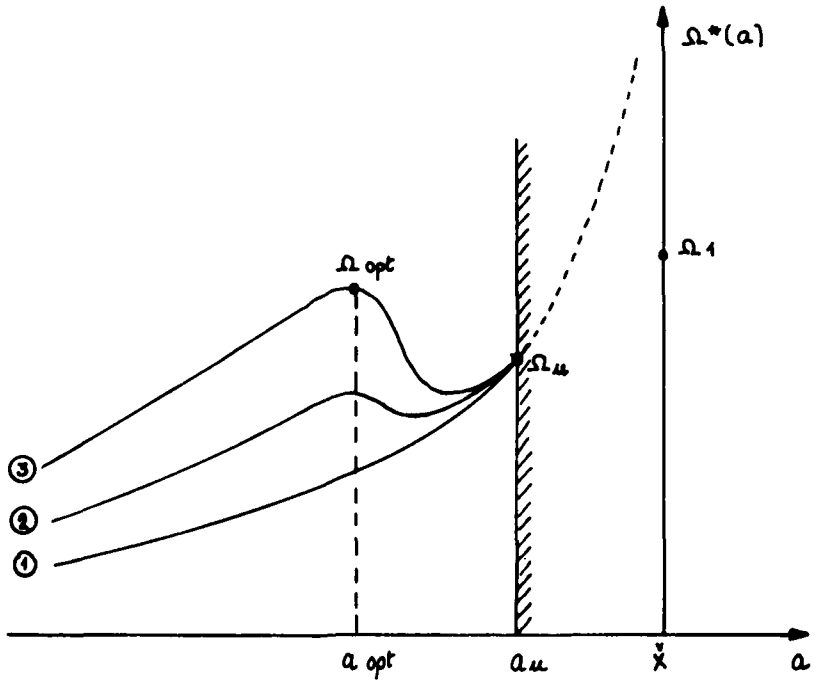


figure 2.

Planche 3

Fonction de vraisemblance $\Omega(a,c)$

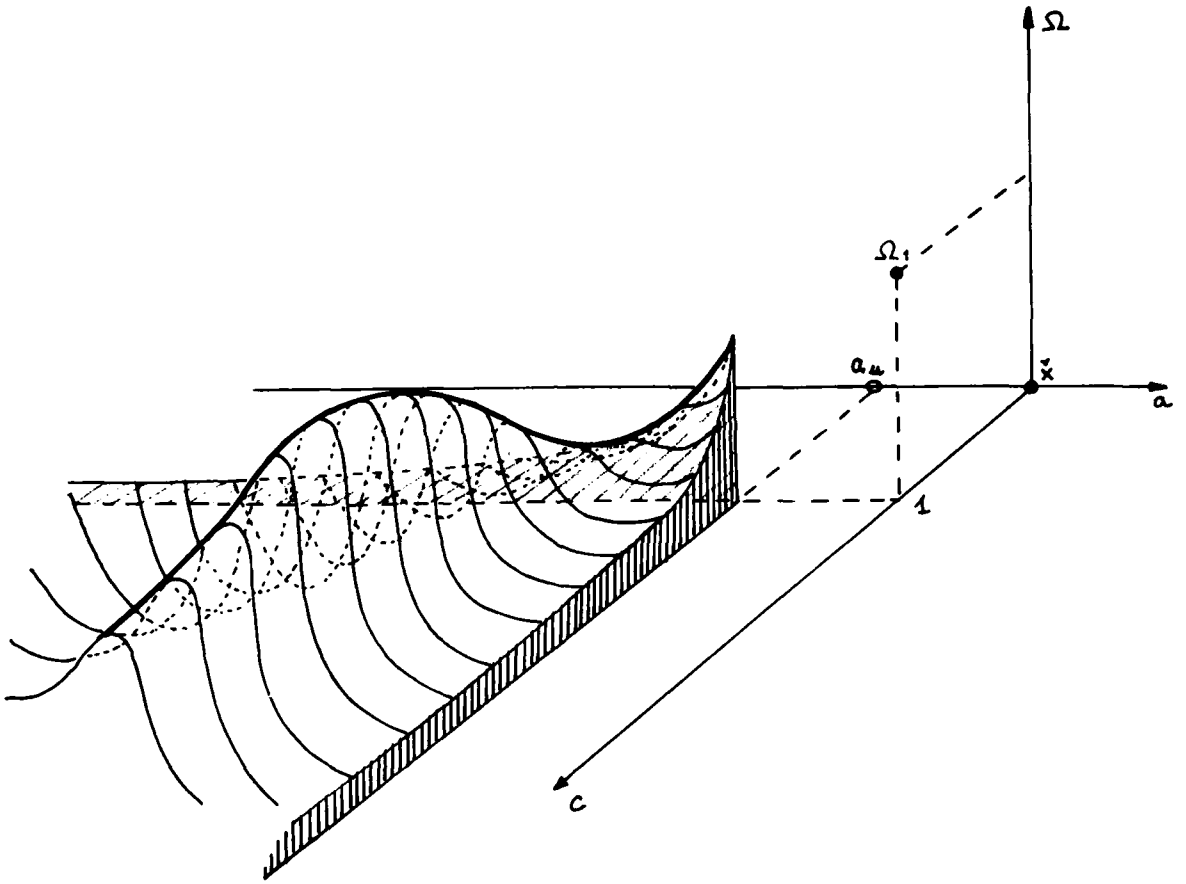


Planche 4

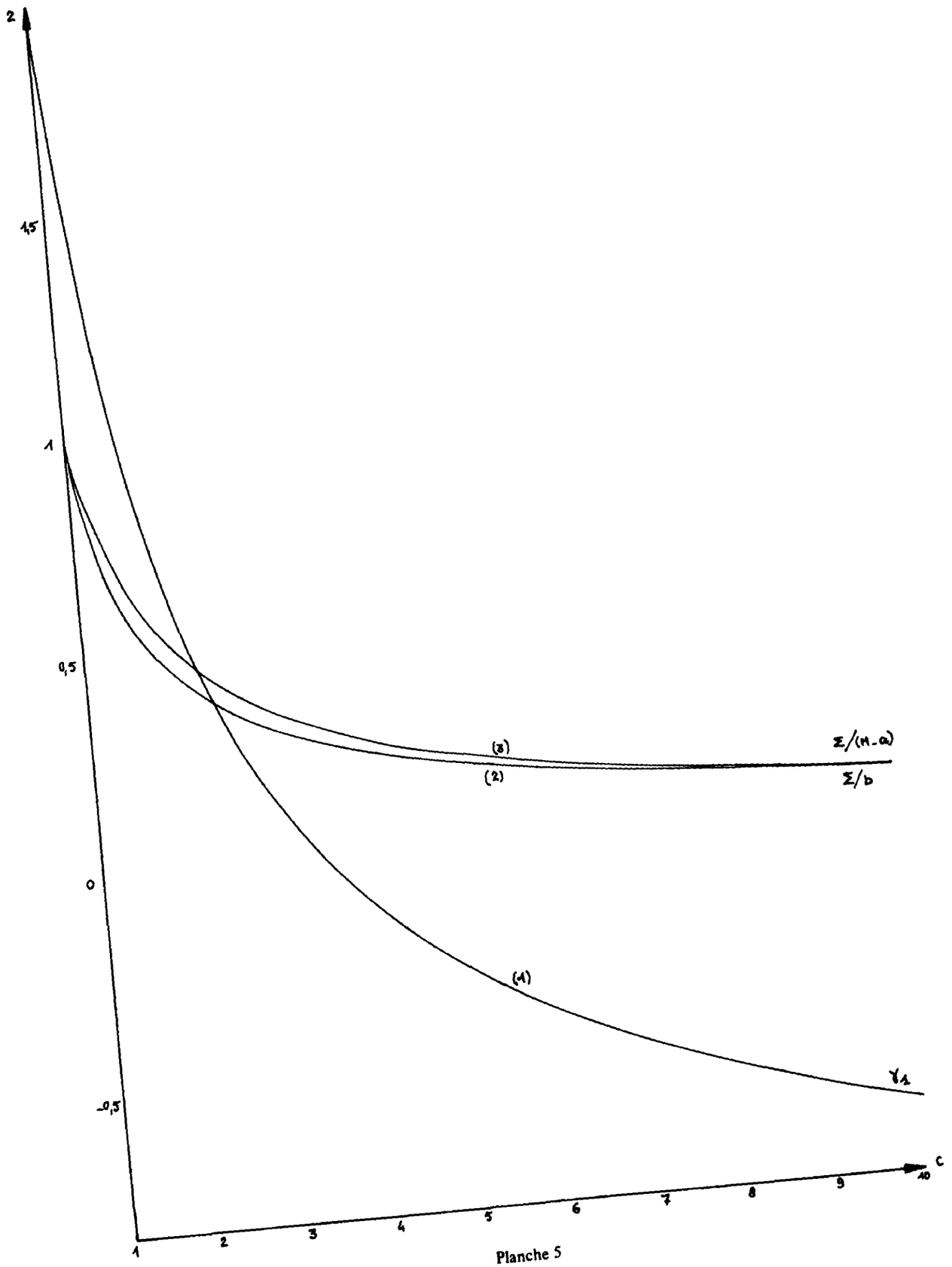


Planche 5

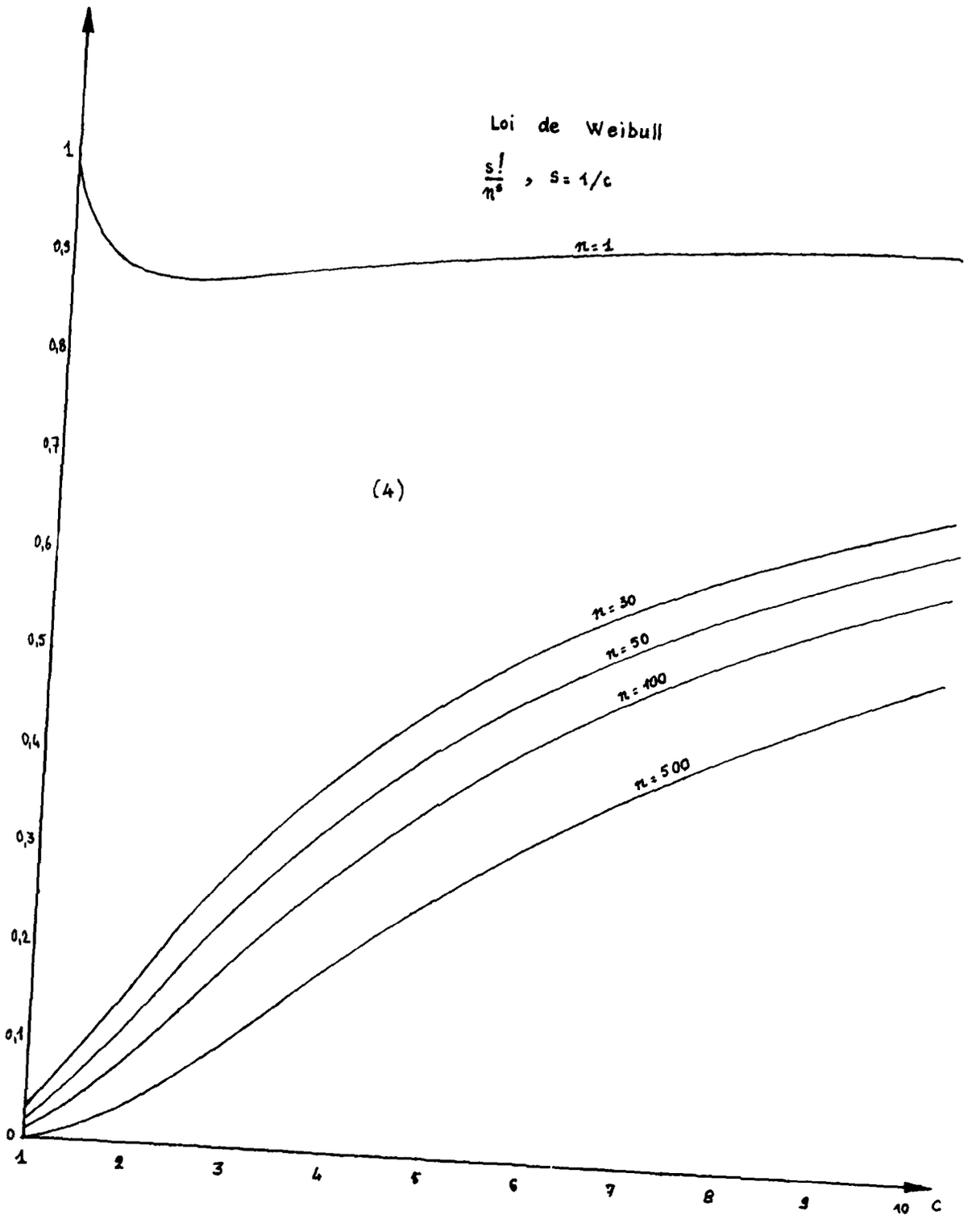

Loi de Weibull
$\frac{s!}{n^s}$, $s = 1/c$
$n = 1$
(4)
$n = 30$
$n = 50$
$n = 100$
$n = 500$
1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
1
2
3
4
5
6
7
8
9
10
C

Planche 6

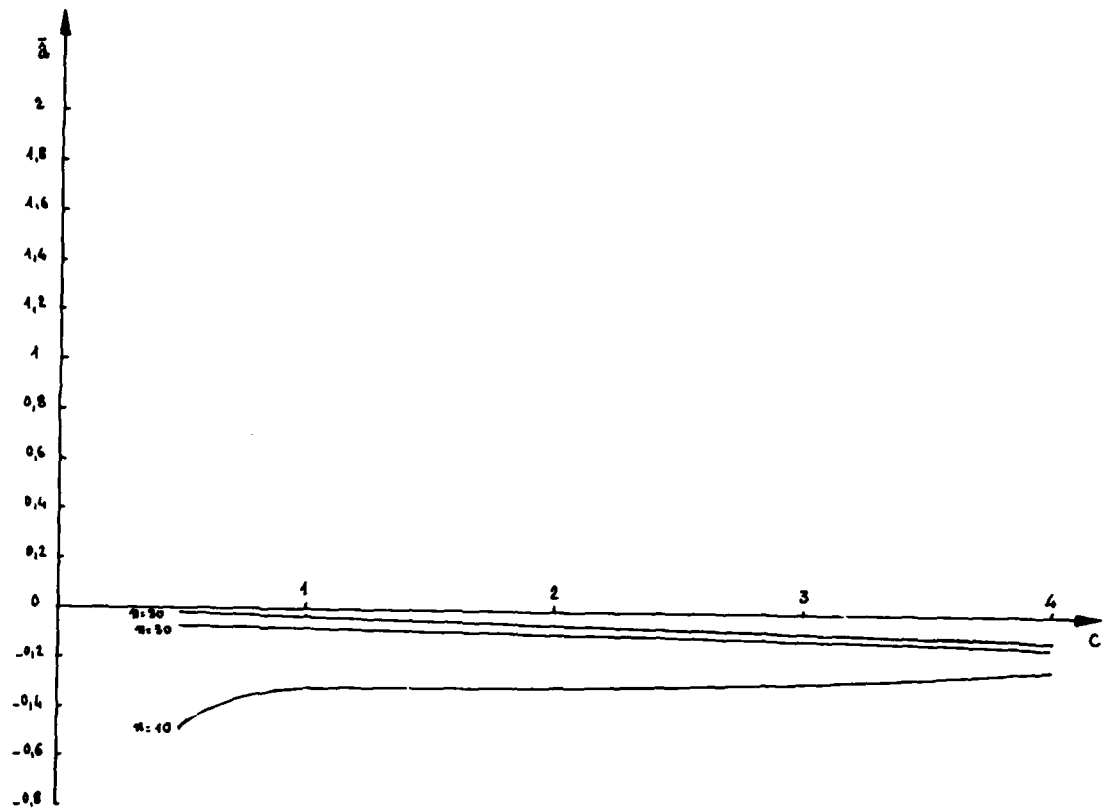


Planche 7

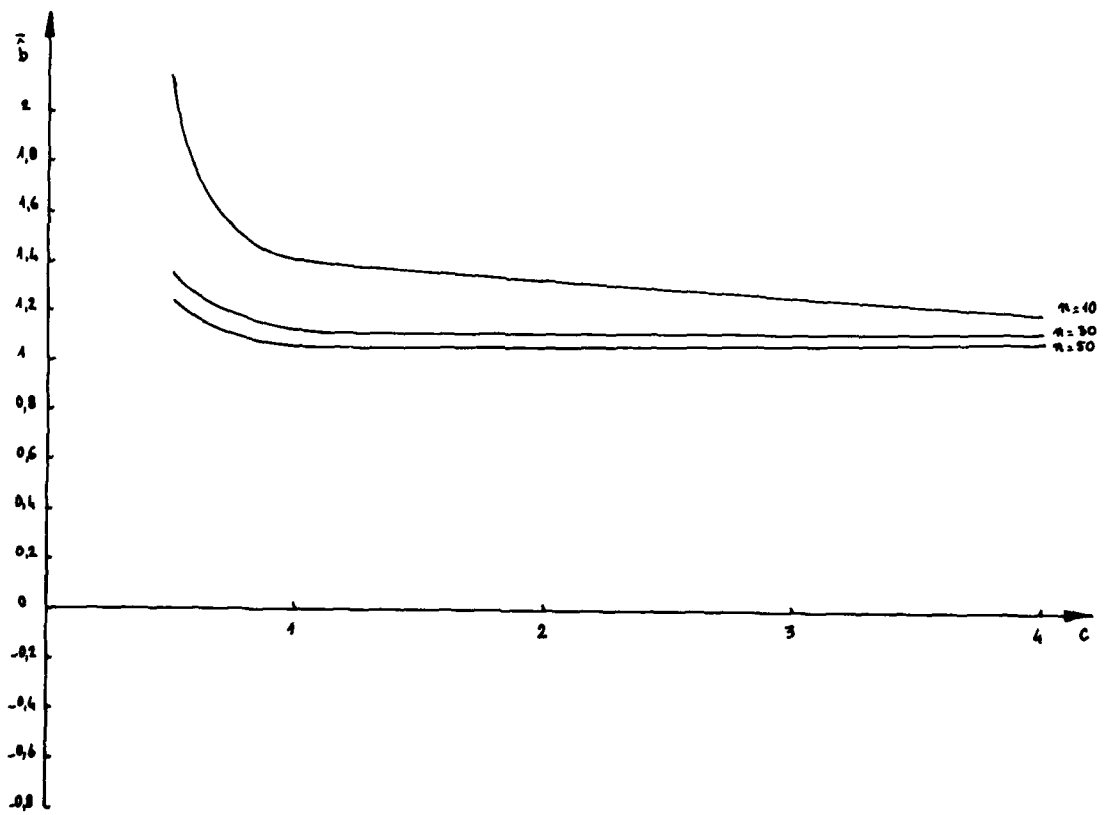


Planche 8

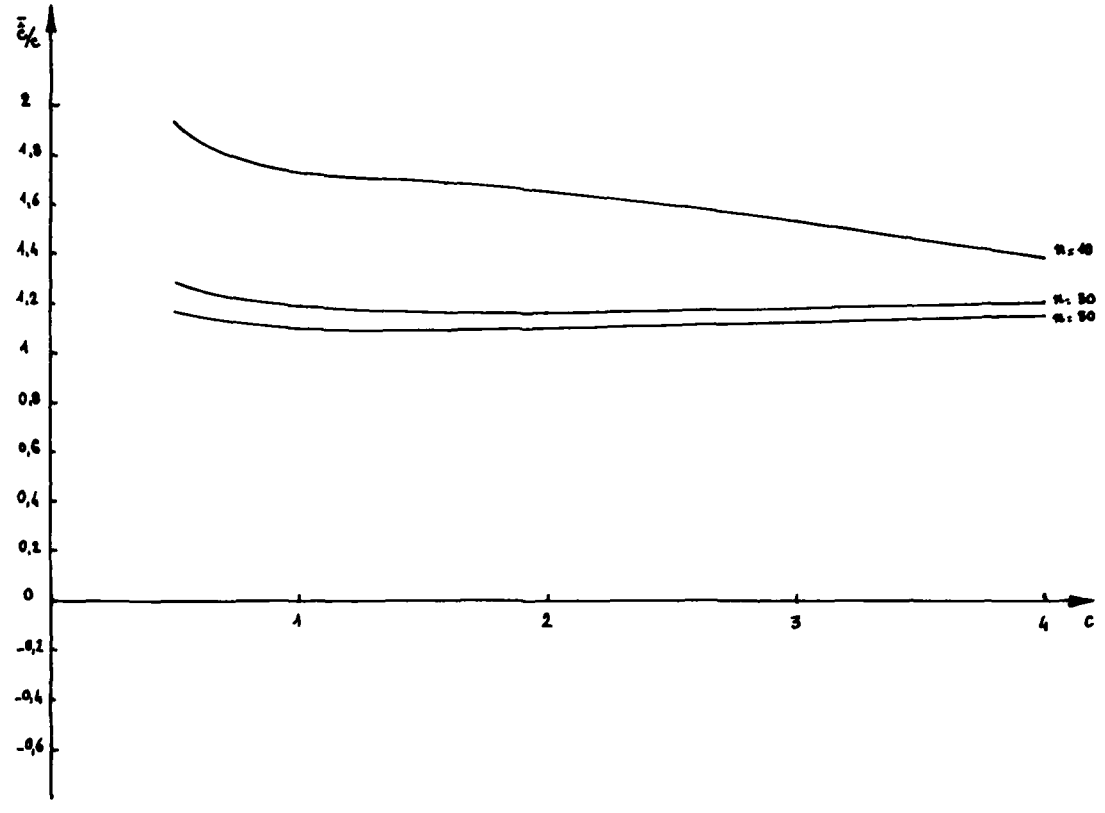


Planche 9

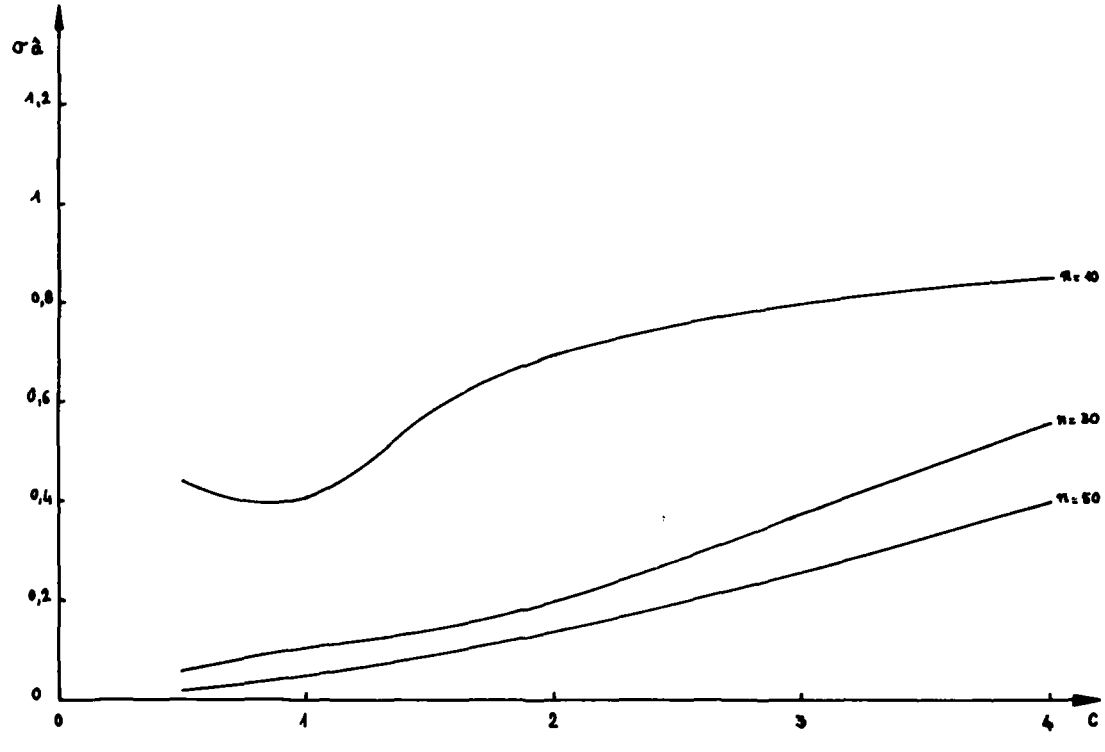


Planche 10

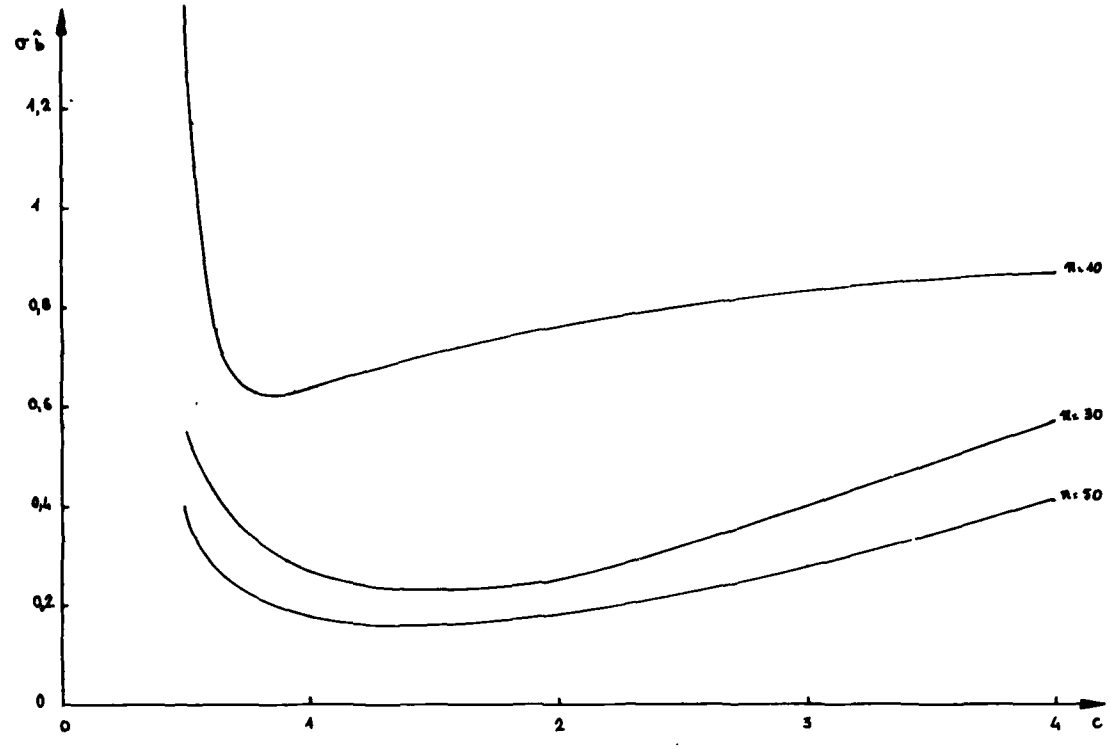


Planche 11

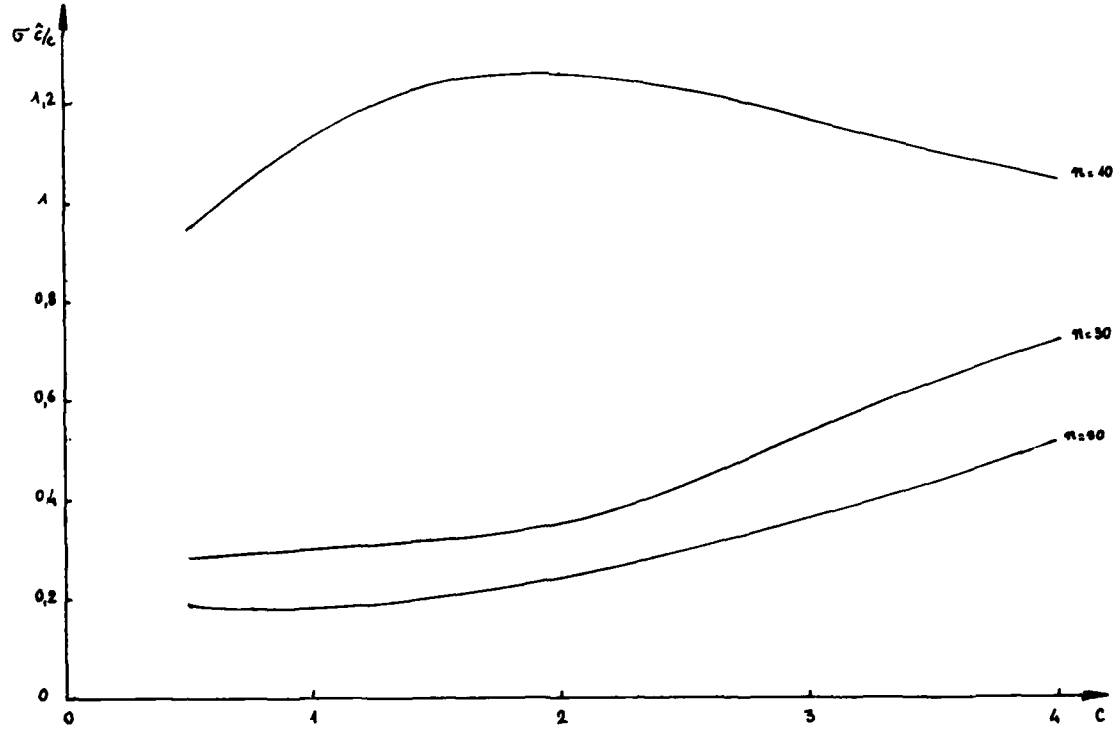


Planche 12

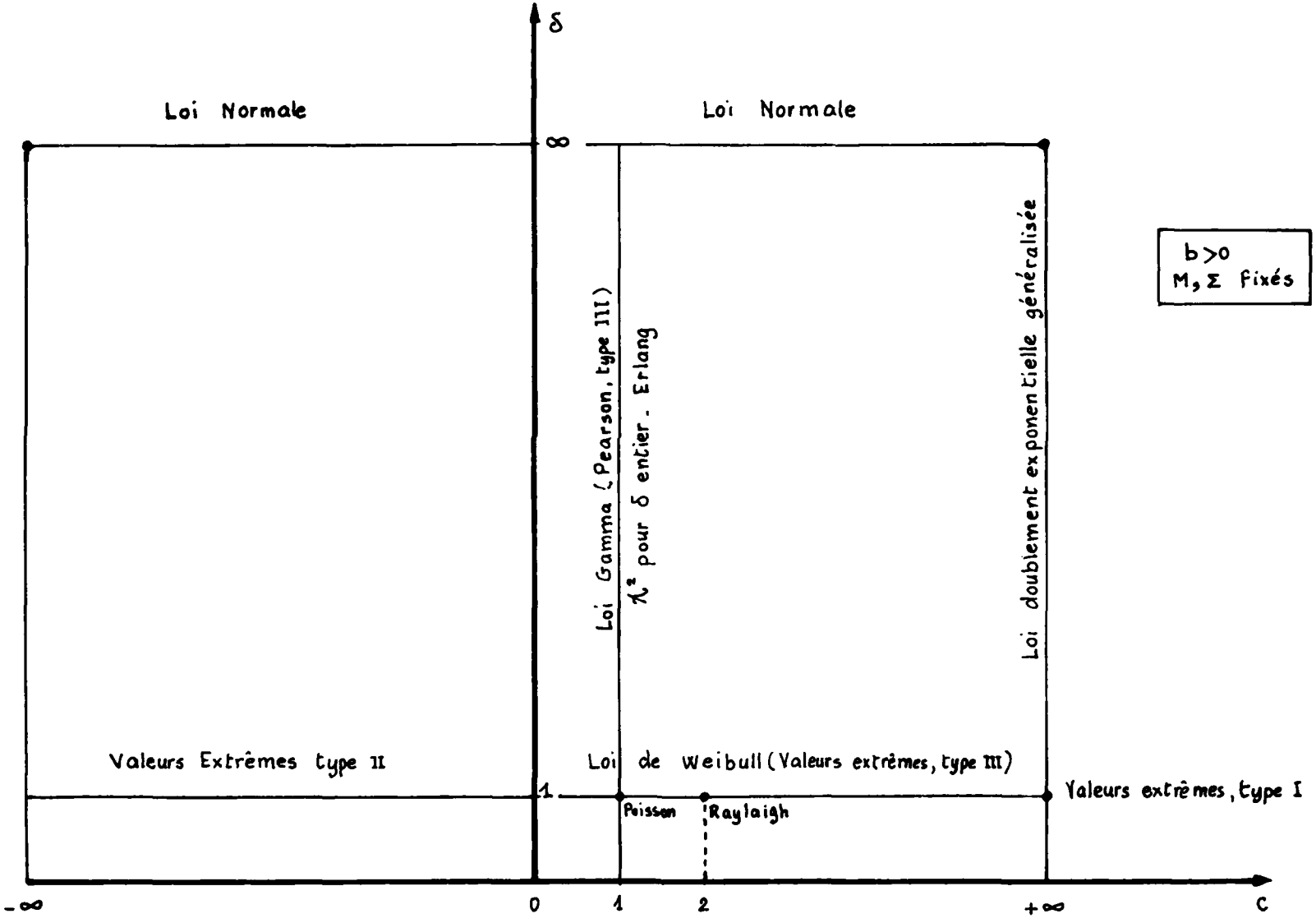

δ
Loi Normale
Loi Normale
∞
b>0
M, Σ Fixés
Loi Gamma (Pearson, type III)
χ² pour δ entier. Erlang
Loi doublement exponentielle généralisée
Valeurs Extrêmes type II
Loi de Weibull (Valeurs extrêmes, type III)
Valeurs extrêmes, type I
1
Poisson
Raylaigh
-∞
0
1
2
+∞
C

Planche 13

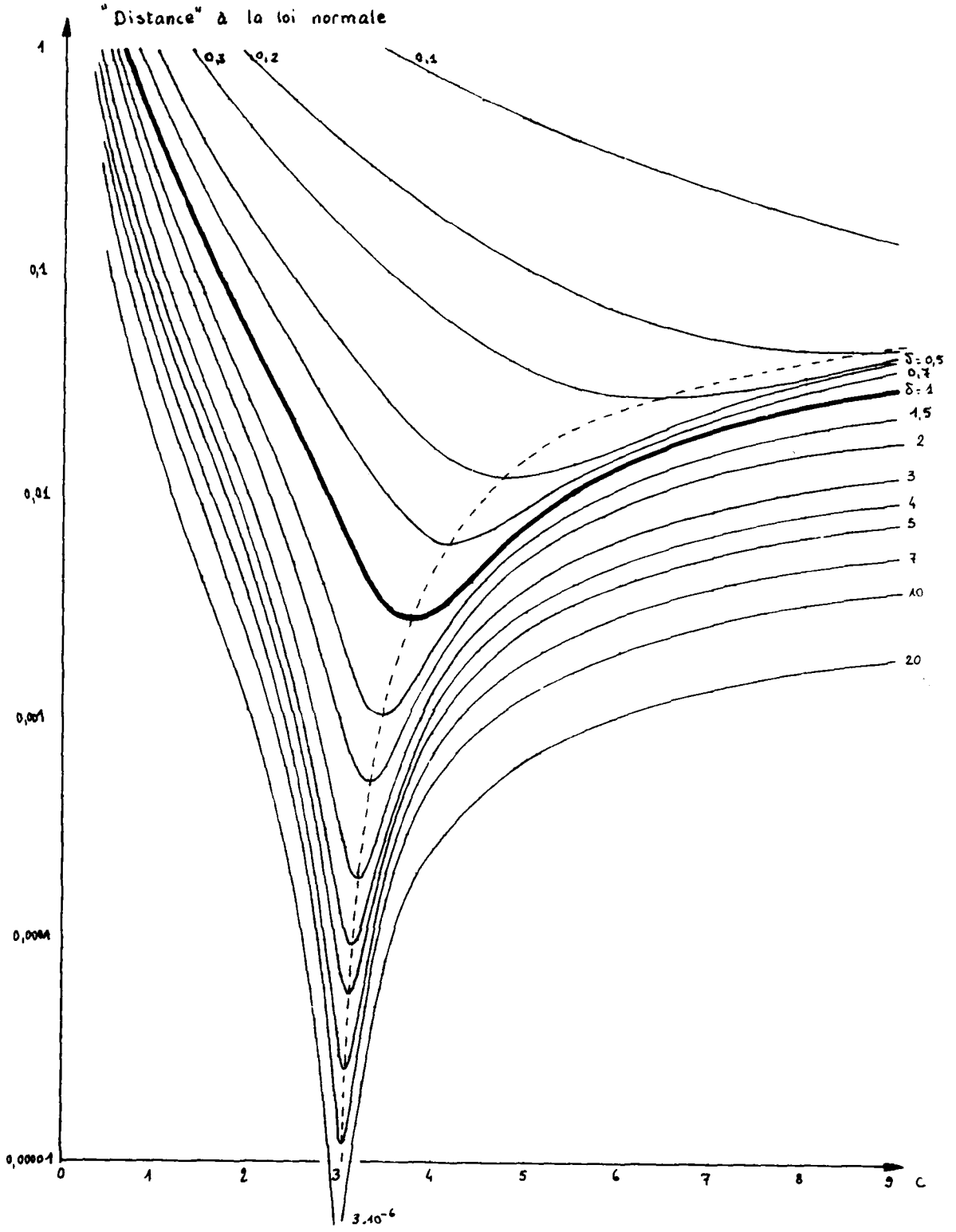

"Distance" à la loi normale
1
0,1
0,01
0,001
0,0001
0,3
0,2
0,1
δ = 0,5
0,7
δ = 1
1,5
2
3
4
5
7
10
20
0
1
2
3
4
5
6
7
8
9
c
3.10⁻⁶

Planche 14

RELIABILITY IMPROVEMENT WARRANTY: AN OVERVIEW

Harold S. Balaban

ARINC Research Corporation
2551 Riva Road
Annapolis, Maryland 21401

SUMMARY

This paper is a review of the concept, genesis, and development of reliability improvement warranty (RIW) in the United States. It is based on the author's participation, since the early 1970s, in a number of RIW research studies and in the application of RIW concepts to military procurements. Today there are RIW programs in all major United States military services as well as in the military services of a number of NATO countries The rapidly increasing use of this procurement/logistic concept strongly suggests that RIW is now a viable approach to securing reliable and maintainable equipment at a reasonable cost.

INTRODUCTION

A reliability improvement warranty is basically a contractual commitment by an equipment supplier to perform depot-type repair services at a pre-established total price for a stated number of years or operating hours. On the surface, it can be viewed as simply a fixed-priced maintenance agreement. While such a view is not entirely incorrect, it is in motivation and implementation that an RIW is differentiated from a repair contract and from a short-term warranty that protects against defective material.

RIWs are normally negotiated in association with the production contract and apply to operational use of the production items. While the major expenditures of a warranty procurement are for repair services, the prime thrust of the approach is to achieve acceptable reliability. The question of whether the contractor can provide depot services at a cost lower than that of military maintenance is secondary to the objective of reliability and maintainability (R&M) achievement and reduced logistics support cost.

Because of the long-term contractor commitment associated with an RIW, prospective contractors are often asked to quote warranty service as a separate line-item option. In this way the government is given the opportunity to evaluate the economies of RIW versus nonwarranty support, with emphasis on the R&M differences between the two alternatives.

The success of early warranty programs provided the impetus for continued development and application of this procurement and logistics support approach. Today, there are a number of RIW programs with variations in coverage designed to meet the particular needs of the using service.

RIW GENESIS

The achievement of satisfactory reliability and maintainability levels in military operational systems has been a challenging problem for a number of years. The potential for improved part and component R&M characteristics offered by technology has been offset to a great extent by the demands for greater sophistication and performance.

During the 1960s, U.S. military agencies expanded considerable effort in developing approaches to achieving satisfactory field R&M performance. The concept of formal R&M programs is now well established, as evidenced by MIL-STD-785 for reliability and MIL-STD-470 for maintainability. Most large military procurements now impose contractual requirements for such programs, including specifications, predictions, design reviews, allocations, parts screening and burn-in, testing, and formal R&M demonstration procedures. A number of military standards, specifications, regulations, and handbooks form a large body of R&M "how to" documentation.

While it is difficult to evaluate the success of the formal R&M program approach quantitatively, continued use suggests that benefits have been realized. However, comparison of field results with predicted and test values shows that reliability achievement has not been completely successful. Hirschberger and Dantowitz, 1976, describe a comprehensive study comparing laboratory-demonstrated and field MTBF values for 95 distinct Navy Weapon Replaceable Assemblies (WRAs). Ground rules were established to provide consistent measurements in the laboratory and in the field; for example, field failures due to identified mishandling were excluded. Figure 1 is a histogram of the MTBF ratio for the 95 WRAs. Eighteen exhibited a field MTBF higher than the MTBF observed in laboratory demonstration, while 77 exhibited a lower field MTBF. By use of a geometric averaging technique, the average ratio of laboratory MTBF to field MTBF was found to be 3.1:1. Similar results were obtained for the ratio of predicted MTBF to field MTBF. A number of MTBF values are generally higher than field MTBF values, even after a common measurement base is established.

This phenomenon has prompted the military services continuously to seek new approaches to assuring timely achievement of satisfactory field reliability. In 1967, the U.S. Navy, through the Aviation Supply Office, contracted with the Lear Siegler Company to provide a Failure Free Warranty (FFW) for the 2171P gyros then in use on A-4 and F-4 aircraft. Lear Siegler provided warranty repair services for five years on 800 of the 2500 gyros in the population. Warranty pricing was based on a 30 percent improvement in MTBF, which was achieved. The Navy's satisfaction with the initial contract resulted in a five-year extension. This FFW contract is considered the prototype of what is now known as RIW.

RIW DEVELOPMENT

In the early 1970s additional small contracts incorporating RIW concepts were awarded -- including an Air Force contract to Lear Siegler for warranting gyros in the F-111 aircraft and a Navy contract to the Abex Corporation for warranting hydraulic pumps on the F-14 aircraft (Aviation Supply Office, 1973; and Markowitz, 1976).

Early in 1973 the U.S. Department of Defense (DoD), through the Defense Advanced Research Projects Agency, contracted with ARINC Research Corporation to explore the potential of applying commercial airline warranty practices to military avionics. It had been widely known that airline avionics of comparable functions in comparable operating environments were achieving reliability far superior to that of military avionics. In the DoD study the airline warranty approach was determined to be one of the significant reasons for this disparity. It was concluded that the military could realize significant avionics reliability and life-cycle-cost benefits from properly constituted and applied warranties. The study was also one of the first to develop a life-cycle-cost model for the quantitative evaluation of warranty.

Following the DoD study, ARINC Research was asked to assist the Electronics System Division of the U.S. Air Force in applying the study recommendations to the forthcoming procurement of the ARN-XXX, the nomenclature given to the standardized Air Force TACAN program. Up to 10,000 TACAN sets were to be purchased under either an RIW with contractor depot maintenance or a logistic-support-cost commitment with Air Force depot maintenance. The procurement was restricted to the two contractors who successfully completed the Full Scale Development phase. After receiving competitive bids, the Air Force chose the RIW alternative and selected Collins Radio as the equipment supplier. That Air Force TACAN RIW program was the first to employ the concept of guaranteed MTBF, together with a number of other innovative features, many of them adopted to balance government and contractor risks (Balaban and Nohmer, 1975).

The DoD's Electronics-X program was also under way during 1973, and a special study category devoted to warranties was established. The Electronics-X report, published early in 1974, concluded that long-term warranties could serve as a competitive alternative to military repair and recommended that they be applied to military electronics.

Late in 1973 the Navy Aviation Supply Office sponsored an FFW seminar that gave further impetus to warranty implementation (Aviation Supply Office, 1973). At the same time, the Army and Marine Corps became interested in applying RIW to commercially available navigation equipment and, under a formal two-step advertised procurement, Bendix Radio was awarded a contract to supply navigation radios for helicopter applications under an RIW and guaranteed MTBF (Mlinarchik, 1977).

Thus during 1974 all United States military services were involved in warranty procurements in an attempt to secure reliable equipment at a reasonable cost. The DoD recognized the potential of this approach as well as the dangers of misapplication and misuse. In mid-1974 the Assistant Secretary of Defense (Installation and Logistics) promulgated a set of guidelines for RIW, including RIW application criteria, special funding requirements, and essential elements of the RIW contractual clause.

In mid-1974 the Air Force's Rome Air Development Center contracted with ARINC Research Corporation to develop a detailed set of guidelines on warranty application for Air Force electronic systems (Balaban and Retterer, 1976). In the same year the Air Force instituted RIW terms and conditions in the request for proposal for major avionic units of the Lightweight Air Combat Fighter aircraft (later to become the F-16). This major step showed industry that the Air Force was firmly in support of the RIW concept. Four NATO countries -- Belgium, Denmark, Norway, and The Netherlands -- also endorsed the RIW concept for the F-16 aircraft they were to purchase.

Industry, on the other hand, had serious reservations about the RIW concept as applied to military systems. The umbrella industry organization known as CODSIA (Council of Defense and Space Industries Association) was the spokesman of industry concern and communicated its views in a number of letters (CODSIA, 1975). CODSIA's principal concern was the inability of contractors to reasonably price an RIW on equipment for which extensive field data were not available. The CODSIA letters established an important communication between government and industry which has continued to the present.*

*A CODSIA Air Force RIW meeting was held as recently as December 1978.

The DoD and the military services certainly recognized the inherent risks of RIW. The DoD established a Tri-Service Reliability and Support Incentives Group to aid in developing and coordinating policy on RIW and other procurement approaches for reliability achievement.

In 1976 ARINC Research conducted a tri-service-sponsored study for DoD on contractor risks associated with RIW. Both qualitative and quantitative aspects were considered, and a number of recommendations for controlling and balancing such risks were developed (Balaban and Retterer, 1977). The military services have acted on many of these recommendations, as evidenced by controls on RIW application and specific risk-limiting terms and conditions. In addition, ARINC Research established a trial RIW data bank during this period. The relatively large number of requests for data during this experimental program was indicative of the recognition of RIW potential (Crum, et al, 1977).

By the mid-1970s, the RIW concept was well established and studies by RAND, IDA, and other government and industry organizations were under way (see Gándara and Rich, 1975; Weimer and Palatt, 1976; Gates, Bortz, and Bicknell, 1977).

Today more than 30 RIW procurements are in progress or under serious consideration. Despite this relatively large number (as compared with only one RIW program ten years ago), only a few programs have accumulated enough data to evaluate the success of the RIW concept: the Navy and Air Force gyros, the Navy hydraulic pump, and the Air Force TACAN. In three of these programs the expected reliability levels have been met or bettered. In the smallest of the four programs, the Air Force gyro, the expected MTBF has not been achieved, but the reliability level is still considered satisfactory. While this small sample cannot be considered representative of the broad spectrum of avionics procurements, the results are encouraging in comparison with field reliability realized in non-RIW procurements.

THE RIW PLAN

A reliability improvement warranty is a long-term fixed-price commitment for contractor depot repair. A typical RIW agreement includes the following terms and conditions:

- Statement of contractor warranty -- the basic agreement, requirements for corrective action, exclusions and limitations, extent of warranty coverage, requirements for maintenance facilities, and warranty price-related information.
- Contractor obligations -- collateral terms and conditions regarding Engineering Change Proposals (ECPs), warranty marking and seals, turnaround time and penalties, and data requirements.
- Government obligations -- requirements for government warranty administration, timely approval of ECPs, and provision of data.

THE MTBF GUARANTEE PLAN

The MTBF guarantee, pioneered by the airlines, requires the contractor to guarantee that a stated mean time between failures (MTBF) will be experienced by the equipment in the operating environment. If a guaranteed level is not met, the contractor must institute corrective action and provide consignment spares until the MTBF improves.* The MTBF guarantee is usually used in conjunction with an RIW, and the term RIW/MTBF is used to denote such a contractual arrangement.

An RIW plan provides indirect incentive for MTBF achievement through the contractor's maintenance-support commitment. The MTBF guarantee provides an even stronger incentive because the contractor is obligated to provide consignment spares to relieve pipeline shortages that may develop as a result of low MTBF. The MTBF guarantee may also include requirements for improving the MTBF to stated values. The RIW and MTBF-guarantee plans are considered totally compatible.

Because of the importance of contractor involvement in failure determination and correction, the MTBF plan is considered feasible only where the contractor either performs the maintenance (RIW or contract maintenance) or has the opportunity to monitor the maintenance process.

The MTBF provisions address the following elements:

- Basic guarantee -- a schedule of required MTBFs to be met by the equipment in the field for various time periods.
- MTBF definition -- countable failures and the time base for computing MTBF.
- Compliance determination -- frequency of MTBF measurement and a formula for computing consignment-spare requirements in the event the unit does not meet MTBF requirements.
- Contractor corrective-action requirements -- the additional action to be taken by the contractor to achieve the required MTBF levels.

*Consignment spares are units loaned to the government by the contractor in accordance with the terms of the guarantee clause. Obligations other than consignment spares may be substituted.

- Consignment-spares administration -- spares obligation, delivery, government return, and ownership conversion.
- Data Requirements -- data to be developed by the contractor in support of the MTBF guarantee.

LOGISTICS FLOW UNDER WARRANTY

The typical logistics flow process for a warranty repair is as follows (see Figure 2):

1. A suspected failure of a warranted unit is tested by military personnel at the using activity to verify the failure.
2. If the unit tests "good", it is put back into service or sent to supply as a ready-for-issue spare.
3. If the unit tests "bad", it is shipped with appropriate data to the contractor for repair.
4. The contractor receives the unit and verifies the failure and warranty coverage.
5. If the failure is not verified or is not covered by the warranty, this is corroborated by a government representative.
6. Covered failures are corrected at no additional cost to the government, and necessary data records are prepared.
7. The repaired unit is shipped back to the using activity, placed in a bonded storeroom maintained by the contractor, or sent to a central military supply depot.

In many respects this logistics flow process is similar to military depot repairs or to contractor repair under a service contract. However, there are significant differences. Under military maintenance, modules rather than black boxes (e.g., LRUs or WRAs) are sent back to a central depot facility; under warranty, the opposite is generally true. Return of black boxes can have a significant spares impact unless pipeline times are carefully controlled. In addition, RIW and RIW/MTBF contracts often have price adjustments or other controls associated with such factors as unverified failures ("test good"), field MTBF, utilization, lost units, ECPs, and batching of returns. If the government is to receive full value for the warranty and meet its contractual obligations, some new or modified administrative activities may be necessary. Generally these have not been burdensome; however, to date there has been no major transition from warranty to organic maintenance for a large population of equipment. Transition must be carefully planned to ensure an orderly process and to preclude any reduction in operational availability.

INCENTIVES AND RISKS

The incentive feature of an RIW is clear. If a contractor can provide equipment that either fails less often than initially anticipated or can be repaired at lower cost than anticipated, his profit is increased. Therefore, with an RIW, the contractor has a direct profit incentive to provide equipment with good R&M characteristics. Under a normal procurement scenario, it might be said that R&M levels above minimum acceptable are not in the contractor's long-range profit interests (see Balaban, 1978). While it is true that reliability must be designed into the system, it must also be recognized that an RIW provides a contractual framework in which the test-analyze-fix process can be extended to initial system operation. Such a process is basic for reliability growth. It has been shown that providing the contractor with near-real-time information on field failures and depot maintenance facilitates R&M problem identification, correction, and growth.

The risks associated with an RIW procurement must also be recognized. Government risks are listed as follows:

- RIW price. The government may pay too much for the warranty coverage.

  Reduced self-sufficiency. Long-term dependence on contractor support will reduce military self-sufficiency, especially if strikes or natural disasters occur at the contractor's facilities.
- Administrative complexity. The warranty concept introduces greater complexity into the military logistics system.
- Transition. The transition from RIW coverage to military maintenance introduces a number of administrative and logistics problems.
- Equipment design. The contractor may use the design that is most amenable to his warranty maintenance but is not the most appropriate for military repair following transition.
- Contractor performance. The contractor cannot or does not perform because of high repair costs, large losses, contract interpretation or loopholes, strikes, or natural disasters.

The following are contractor risks:

- Operational stresses. Equipment may be subjected to unforeseen operational stresses.
- Mishandling and tampering. Military maintenance personnel may cause failures beyond contractor control.
- Usage rate. Increased equipment use will increase failure exposure.
- Processing of engineering change proposals. Slow government processing of R&M engineering change proposals will hamper the improvement process.
- RIW price. The contractor may bid too low a price because of competitive pressures, optimistic R&M estimates, or misinterpretation of provisions.

It is a formidable challenge to the government and industry to ensure that the overall risk associated with a particular RIW application is acceptable to both parties. Fortunately, there are actions that can be taken to keep the risks within reasonable bounds:

- Develop and use criteria for determining the applicability of RIW.
- Structure the procurement contract terms and conditions and implementation procedures to address high-risk factors.
- Perform economic analyses in evaluating warranty potential and developing contractual and implementation procedures.

Balaban and Retterer, 1977, present a number of approaches for risk reduction.

THE FUTURE OF RIW

The number of programs containing long-term warranty commitments has increased approximately tenfold in the last five years. The RIW experiment must be viewed as generally successful to date. The reliability/life-cycle-cost achievement of equipments under RIW or RIW/MTBF has exceeded that usually attainable under standard procurements. While some contractual and implementation difficulties have been experienced, industry and the government appear capable of positively responding to this procurement approach.

It cannot be concluded at this time that introducing the RIW concept in a program will permit discontinuing current reliability and maintainability program controls. Some adjustment may be in order -- perhaps one that will give the contractor more freedom in allocating funds for such control. Adequate funding and time to obtain relevant R&M test data in the development phase are critical to risk control on new technology equipment and can be the key to a successful RIW program.

The future of RIW is promising if continued efforts are made to ensure that the concept is properly applied and implemented. It is also necessary for the military services to continue to support research in RIW and allied areas as technology, resources, and military demands change. The RIW concept that embodies the suitable form of contractor incentive for R&M achievement will also be flexible enough to encompass most forseeable changes provided the appropriate effort is made.

REFERENCES

Aviation Supply Office, 1973, Proceedings of the Failure Free Warranty Seminar, Philadelphia, PA.

Balaban, H. and Nohmer, F., 1975, "Warranty Procurement: A Case History", Proceedings of the 1975 Annual R&M Symposium, 543-548.

Balaban, H. and Retterer, B., 1976, Guidelines for Application of Warranties to Air Force Electronic Systems, ARINC Research Corporation, RADC Report TR-76-32.

Balaban, H. and Retterer, B., 1977, An Investigation of Contractor Risk Associated With Reliability Improvement Warranty, ARINC Research Publication 1619.

Balaban, H., 1978, "Reliability Improvement By Profit Incentive", Quality, Vol. 17, No. 11, 22-28.

CODSIA, 1975, Comments on Warranties and RIW to the Honorable M. R. Currie, DDR&E, 18 July 1975; to Dr. Paul Arvis, U.S. Army Procurement Research Office, Fort Lee, VA., 2 July 1975; to Mr. M. D. Bruns, Chairman, Tri-Service Reliability and Support Incentives Group, OASD (I&L), 30 December 1975.

Crum, F., et al., 1977, Establishment and Operation of a Pilot Warranty Information Center, ARINC Research Publication 1612.

Gándara, A., and Rich, M., 1975, Reliability Improvement Warranties for Military Procurement, Rand Report R-7505.

Gates, R., et al., 1977, "Quantitative Models Used in the RIW Decision Process", Proceedings of the 1977 R&M Symposium, The Analytic Sciences Corporation, 229-236.

Hirschberger, G., and Dantowitz, A., 1976, Evaluation of Environmental Profiles for Reliability Demonstration, Grumman Aerospace Corporation RADC Report TR-76-32.

Markowitz, O., 1976, "Aviation Supply Office FFW/RIW Case History #2, Abex Pump", Proceedings of the 1976 Annual R&M Symposium, 357-362.

Mlinarchik, R., 1977, "RIW Experiences at ECOM", 1977 Annual R&M Symposium, 257-260.

Weimer, C. D., and Palatt, P. E., 1976, The Impact of Reliability Guarantees and Warranties on Electronics Subsystem Design and Development Programs, Institute for Defense Analysis, IDA Study S-482.

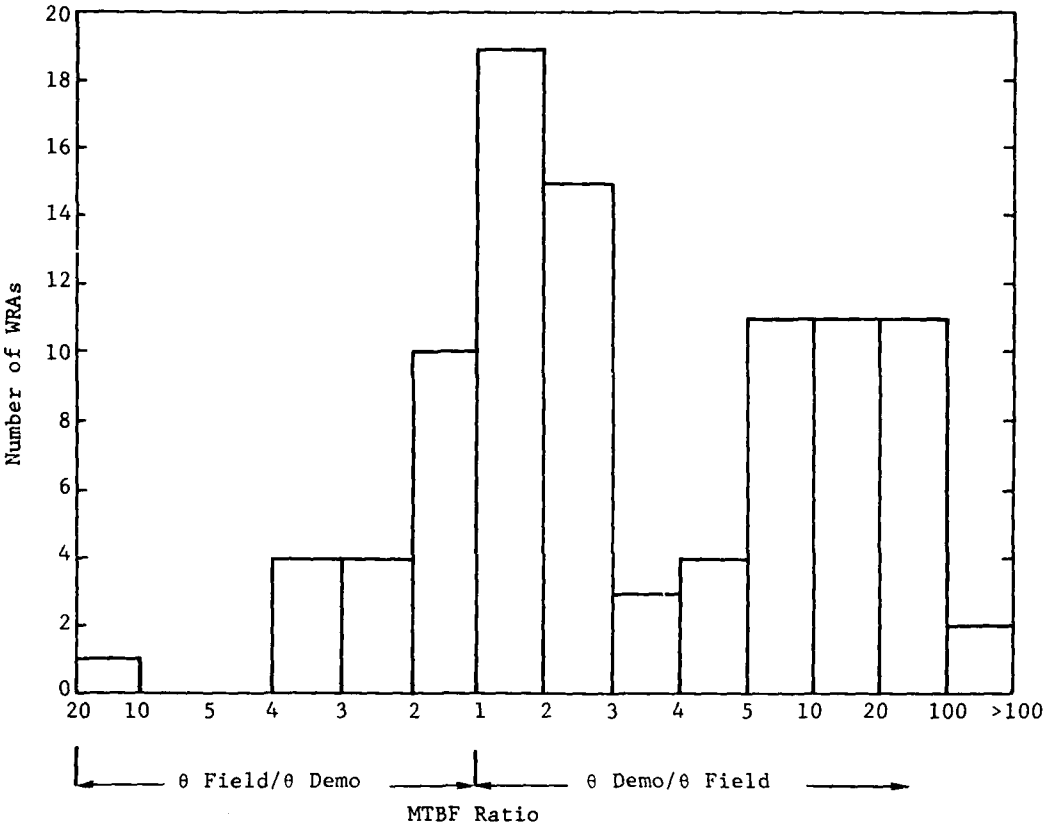


Figure 1. RATIOS OF FIELD MTBF TO DEMONSTRATION MTBF

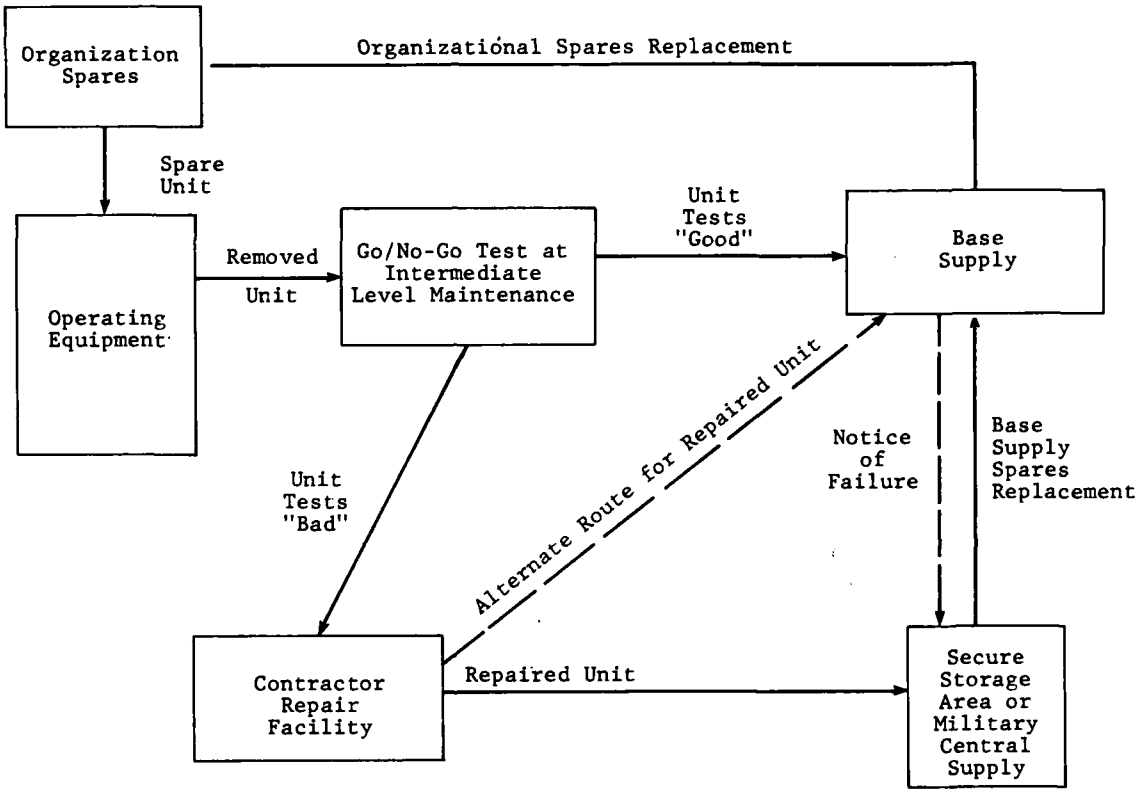


Figure 2. LOGISTICS FLOW UNDER RIW

DISCUSSION

B.G.Peyret, Fr

Existe-t-il des contrats RIW pour des équipements électroniques de technologie complètement nouvelle?

Author's Reply

RIW as I have described it is primarily applicable to evolutionary-type equipment such as the ARN-118 TACAN.

For completely new technology, I would say that RIW is not applicable since both the customer and manufacturer would have little basis for structuring and pricing a long-term commitment. Some form of cost and risk sharing may be possible for new technology equipments for which some applicable experience and data exist.

R.Voles, UK

You have shown that the ratio of measured MTBF in the field to the MTBF demonstrated at the contractor's plant can vary over a range of 2000 : 1. But if a contractor is to enter into a RIW contract, he must have a much more accurate estimate of the field MTBF than this.

How are these two points reconciled?

Author's Reply

The data illustrates the poor relationship between test and field reliability under the usual procurement procedures for which contractor incentives for good R & M are, at best, somewhat limited. With RIW, the contractor's incentives are changed – better reliability means greater profit. The contractor will have the interest and should have the opportunity to become familiar with the expected field environment early in the design stages. The test-analyse-fix process inherent in RIW repair provides a rapid and effective means for correcting design and initial production/QC problems. These factors should yield a better relationship between test and field reliability for warranted equipment and the limited data now available to us supports this contention.

F.S.Stringer, UK

How do the Military ensure that their repair and maintenance skills and abilities are not eroded by RIW?

Author's Reply

The question is relevant with regard to the impact of RIW on military self-sufficiency. In many cases the erosion of military maintenance skills due to RIW is minimal. For many equipments under RIW, contractor depot repair for some initial period would be the normal course of events. In such cases RIW basically extends the contractor repair period.

On the other hand, the RIW permits a more orderly introduction of military maintenance which can benefit from the lessons learned during the contractor repair period. Examples include starting with debugged and up-to-date procedures manuals and test equipment.

If the military essentiality of the equipment is such that complete reliance on contractor depot repair is undesirable, the user may elect to initially purchase applicable test equipment and train a cadre of maintenance personnel.

LES CLAUSES DE FIABILITE DANS LES CONTRATS

J.P. PLANTARD

THOMSON - C.S.F.

S.C.T.F.
Domaine de Corbeville
B.P n° 10
91401 ORSAY

FRANCE

-:-

R E S U M E

L'auteur présente les résultats des travaux accomplis au sein d'un Groupe de Travail interministériel du Comité de Coordination des Télécommunications (C.C.T.); ceux-ci se sont soldés par les deux documents suivants :

190 A/CCT

ETCA/NOR - X68-09/1

Fiabilité des équipements et des systèmes

- Recommandations pour l'introduction des clauses de fiabilité,

- Guide pour l'établissement d'un plan de fiabilité.

191/CCT

ETCA/NOR - X68-09/2

Fiabilité des équipements et des systèmes.

- Annexes techniques

L'objet de la présente communication est d'exposer le contenu de ces deux documents en en expliquant les motivations, les points techniques les plus marquants, ainsi que les modalités d'application.

Ce travail a été réalisé alors que l'auteur était responsable du Centre de Fiabilité au Centre National d'Etudes des Télécommunications (C.N.E.T.)

1 - INTRODUCTION

L'inclusion des clauses de fiabilité dans les contrats d'équipement n'est pas un sujet neuf. Il a fait et fait encore l'objet de nombreuses discussions au sein des Administrations et de leurs contractants. Le Comité de Coordination des Télécommunications (C.C.T.) s'en était déjà préoccupé dans la passé et avait publié en 1969 une spécification générale 190/CCT "Fiabilité des équipements électroniques". Cependant cette spécification s'est avérée très incomplète et difficilement utilisable dans un certain nombre de cas; aussi s'est-il trouvé nécessaire de la réactualiser afin de servir véritablement de guide aussi bien à ceux qui rédigent des clauses de fiabilité qu'à ceux qui sont chargés d'y répondre.

Pour ce faire, la collaboration des différentes administrations et services publics concernés, de même qu'une participation des syndicats professionnels était indispensable.

2 - HISTORIQUE

Afin d'expliquer les raisons qui ont conduit à l'activation d'un nouveau Groupe de Travail sur ce sujet, il parait judicieux de revenir sur l'ancienne spécification 190/CCT.

Outre des considérations à caractère général sur la conduite d'une étude de fiabilité, elle traitait surtout l'aspect vérification des objectifs de fiabilité à l'aide d'essais progressifs dont les plans étaient calqués sur la spécification américaine MIL STD 781 A "Reliability tests : exponential distribution".

Du fait, du moins en partie, de son caractère restrictif, cette spécification a été utilisée mis à part quelques incitations de l'Administration à l'appliquer dans certains marchés d'études.

En égard à cette situation, une enquête publique a été lancée par le Comité de Coordination des Télécommunications qui a confirmé, non seulement que le document 190/CCT n'était pas appliqué mais qu'il était nécessaire de la modifier pour le rendre utilisable. Trois axes d'études ont été jugés indispensables, ils peuvent se résumer dans les trois titres suivants :

- rédaction d'un plan de fiabilité,
- fiabilité prévisionnelle,
- fiabilité en exploitation.

A partir de ces trois idées maitresses, trois Groupes de Travail ont été créés :

- SG 1 Plan de Fiabilité,
- SG 2 Fiabilité Prévisionnelle,
- SG 3 Fiabilité en exploitation,

sous-groupes chargés de rédiger les chapitres correspondants.

Un Groupe d'experts s'est chargé de la coordination et de la surveillance des travaux.

L'objectif était, comme nous l'avons dit, de rassembler une représentation suffisante pour que les documents résultant de ces travaux aient la plus large audience possible.

Les principaux organismes représentés ont été les suivants :

- Le Centre National d'Etudes des Télécommunications (C.N.E.T.)
- La Délégation Générale pour l'Armement (D.G.A.) par l'intermédiaire de ses services techniques.
- Le Ministère de l'Industrie.
- Le Centre National d'Etudes Spatiales (C.N.E.S.).
- Le Commissariat à l'Energie Atomique (C.E.A.)
- L'Electricité de France (E d F)
- L'Union Technique de l'Electricité
- Les Syndicats Professionnels concernés (fabricants de composants et constructeurs d'équipements civils et militaires).

De même, à l'intérieur de chaque Groupe de Travail, chaque organisme y a délégué un représentant.

Les travaux entrepris par le SG 2 et le SG 3 qui avaient un caractère purement technique, n'ont pas présenté de difficultés majeures et ceux-ci ont été terminés dans un temps relativement court.

Deux chapitres ont été ainsi rédigés et inclus dans un document intitulé "Annexes Techniques", nous en examinerons plus loin les détails. L'essentiel du travail a été celui entrepris par le SG 1, nous allons nous y arrêter quelques instants.

ORIENTATIONS DES TRAVAUX DU SG 1 "ETABLISSEMENT D'UN PLAN DE FIABILITE"

En effet, le travail essentiel devait être réalisé par ce sous-groupe puisque le plan de fiabilité regroupe toutes les tâches qui sont effectuées au titre de la fiabilité. D'autre part, une certaine évolution est apparue dans le concept "Clauses de Fiabilité" et il s'est avéré essentiel pour les membres du groupe de se pencher d'une manière plus précise sur les notions d'objectifs, les notions de clauses et leur introduction dans un appel d'offres et dans un contrat, ce qui avait été fait d'une manière succincte dans le document 190/CCT (le nombre de contrats avec clauses de fiabilité passés avant 1969 étant assez limité).

Ceci impliquait donc beaucoup plus que l'élaboration de chapitres supplémentaires au document existant, mais une refonte quasi-complète de l'ancienne spécification, même si certains chapitres ou paragraphes ont pu être repris dans leur quasi-totalité.

Cette solution nous a semblé susceptible de favoriser l'utilisation d'un tel document. La question du statut de ce document s'est naturellement posée immédiatement.

Il est apparu très vite qu'un statut de "spécification" serait un obstacle majeur à l'application du document et donc serait tout à fait en sens opposé au but recherché.

D'autre part, il est non moins certain que chaque contrat avec clause de fiabilité, se présente comme un cas particulier et ce pour diverses raisons : particularité de l'équipement ou du système, utilisation spécifique, savoir-faire du constructeur, organisation spécifique etc... Il apparaît donc tout à fait impossible dans un tel document, de rassembler tous les cas de figures; d'autre part, un statut de spécification ou de norme, impose l'application complète de celle-ci chaque fois qu'elle est citée dans un contrat, ce qui ne saurait être acceptable.

Il convenait donc de donner à ce document un statut moins contraignant et tel qu'il procurerait une incitation à son utilisation. Nous lui avons donc donné le caractère de "GUIDE" pour souligner que le document est un support à la rédaction de la spécification particulière d'un contrat.

Certaines parties du document (ceci sera explicité ensuite) peuvent servir d'aide aux constructeurs pour la mise en place et la réalisation de certaines tâches particulières à la fiabilité.

Ce document se présente donc comme un "GUIDE" :

1 - pour le client, afin de l'aider à rédiger les clauses de fiabilité à introduire dans l'appel d'offres et dans le contrat.

2 - pour le fournisseur afin de l'aider à répondre aux exigences du client au moment de la réponse à l'appel d'offres et au moment du contrat, et à concevoir et réaliser certaines tâches spécifiques.

3 - ARTICULATION DES DOCUMENTS

Il est à noter en premier, que certaines parties ne sont applicables qu'en matériel électronique, en particulier les chapitres concernant :

- les prévisions de fiabilité,
- les essais de fiabilité.

Les recueils de données sur les composants ou pièces détachées non-électroniques, sont pratiquement inexistants. Quant aux essais de fiabilité, les plans d'essais utilisables ne sont valables que lorsque la distribution est esponentielle, ce qui ne peut donc s'appliquer aux pièces mécaniques.

Or, beaucoup d'administrations ou servuces publics, telle la Délégation Générale pour l'Armement (D.G.A.), utilisent des équipements ou systèmes à caractères non-électroniques et mécaniques pour lesquels les chapitres précités ne sont pas applicables.

Il a donc été décidé de séparer ces chapitres à caractère technique et plus spécifique du document principal qui lui est applicable à tout type d'équipement ou système.

Nous avons donc deux documents distincts dont les titres sont les suivants :

190 A/CCT
ETCA/NOR - X68-09/1
Fiabilité des équipements et des systèmes

- Recommandations pour l'introduction des clauses de fiabilité,
- Guide pour l'établissement d'un plan de fiabilité.

191/CCT
ETCA/NOR - X68-09/2
Fiabilité des équipements et des systèmes

- Annexes techniques.

4 - DOCUMENT PRINCIPAL 190A/CCT

4.1. - Recommandations pour l'introduction des clauses de fiabilité

Il convient, en premier lieu, de définir ce que sont les clauses de fiabilité.

4.1.1. Clauses portant sur des objectifs quantitatifs

Elles sont constituées par les objectifs quantitatifs eux-mêmes et par les modalités d'évaluation des paramètres associés.

Les objectifs (chiffrés en Moyenne des Temps de Bon Fonctionnement par exemple) sont fixés :

- soit à priori par le client lui-même,
- soit après consultation des industriels en réponse à un appel d'offres.

Ces objectifs doivent naturellement tenir compte :

- de la mission,
- de la technologie et de la complexité de l'équipement ou du système,
- de l'expérience préalablement acquise.

4.1.1.1. Comment déterminer les objectifs quantitatifs de fiabilité ?

C'est bien là que réside la difficulté.

La solution actuelle est d'effectuer un calcul prévisionnel de fiabilité sur la base d'une conception plus ou moins fine de l'équipement ou du système à réaliser (les modalités techniques de ce calcul sont détaillées dans le document 191/CCT ch. I). Cependant, une certaine marge d'incertitude, difficile à évaluer, existe. Celle-ci peut conduire à des objectifs sur-dimensionnés qui ne pourront être atteints dans la réalité sans des dépenses supplémentaires non prévues au budget initial; accroissement des coûts dû à la nécessité d'améliorer la conception et/ou la fabrication du produit ou provoqué par des dépenses d'exploitation plus élevées (maintenances préventive et corrective plus importantes).

Pour semble-t-il pallier cet inconvénient, une nouvelle optique est apparue aux Etats-Unis à l'initiative de l'U.S. Navy. Afin d'éviter des déboires dus à des objectifs mal calibrés, une corrélation entre le calcul prévisionnel et des essais de fiabilité (basés sur les spécifications MIL STD 781 B et probablement sur la nouvelle spécification MIL STD 781 C) en laboratoire, sur les équipements qui ne sont pas encore de série, est faite avant de fixer des objectifs contractuels pour les équipements de série.

Ceci est-il applicable en France ? La question reste posée et mérite certainement que l'on s'y arrête.

4.1.1.2. Comment vérifier les objectifs de fiabilité ?

Deux possibilités sont offertes :

- Essais de fiabilité en laboratoire

  Les modalités en sont détaillées dans le document 191/CCT chapitre II. Nous dirons simplement que les plans d'essais utilisables sont basés sur la distribution exponentielle. Les conditions d'essais sont calquées sur les spécifications existantes (MIL STD 781 B et sa dernière édition MIL STD 781 C).

  Actuellement cette modalité d'évaluation n'est pratiquement pas utilisée en France.

- Evaluation de la fiabilité en exploitation.

  Les modalités de celle-ci sont détaillées dans le document 191/CCT ch. III.

  C'est en pratique la méthode utilisée. Elle nécessite une période d'observation suffisamment longue pour que l'évaluation soit statistiquement valable. Le plan d'essai (ou d'évaluation) à utiliser peut être déterminé en utilisant les courbes du ch. II compte tenu des objectifs à vérifier.

  Remarque : Certains cas particuliers méritent notre attention car ils ne permettent en général, ni une évaluation en laboratoire, ni une évaluation en exploitation à proprement parler. C'est en effet le cas des équipements pour l'espace où l'on ne peut se contenter que d'une évaluation à l'aide d'un calcul prévisionnel de fiabilité.

On compense l'incertitude engendrée par des tâches de fiabilité supplémentaires par rapport à ce qui serait entrepris sur un équipement non destiné à l'espace, ces opérations visant à obtenir une assurance adéquate que les objectifs seront atteints (on en aura la certitude qu'une fois la mission terminée).

4.1.2. Clauses portant sur une assurance de fiabilité.

Il nous paraît essentiel d'examiner ce cas qui concerne le (ou les) équipement(s) pour lesquels il est difficile de préciser un objectif quantitatif réaliste. Ce peut être le cas de nouveaux systèmes que l'on étudie, de nouvelles technologies, etc... Dans cette optique, on peut être amené à définir une liste de tâches à effectuer au titre de la fiabilité, le choix de celles-ci peut être fait à l'aide du chapitre II du document 190 A CCT, intitulé "GUIDE POUR L'ETABLISSEMENT D'UN PLAN DE FIABILITE".

Son titre indique clairement qu'il ne saurait être question de l'appliquer "in-extenso". Pour chaque cas, suivant les besoins, on sera amené à y choisir certains éléments qui figureront (après accord entre client et fournisseur) dans la spécification particulière du contrat. Il peut se faire que le fournisseur choisi par le client soit à même de faire la preuve qu'il dispose d'ores et déjà d'une organisation et de moyens, lui permettant de répondre aux exigences du client sans qu'il ait à se référer au document en question. La seule contrainte étant qu'il doit, dans ce cas, justifier auprès du client ce qu'il compte faire au titre de la fiabilité.

Ce qui signifie que nous ne saurions être exhaustifs et que la liste des tâches explicitées dans le document n'a aucun caractère impératif et qu'elle est seulement une aide pour répondre à des exigences.

L'ensemble des tâches ainsi retenues constitue ce que l'on appelle LE PLAN DE FIABILITE.

4.1.3. Introduction des clauses de fiabilité dans l'appel d'offres

Les clauses de fiabilité ne doivent pas cohabiter seules. Il est indispensable d'y associer les coûts qui s'y rapportent. Il serait en effet absurbe, qu'un client exige un niveau de fiabilité pour un équipement ou un système qu'il compte utiliser sans qu'il se préoccupe de ce que cela va lui coûter.

On ne saurait donc s'étonner que, lors de l'appel d'offres, il soit du simple bon sens que le client demande aux industriels de répondre à ses exigences de fiabilité en y incluant les coûts correspondants.

On peut distinguer deux cas principaux :

- le client demande aux éventuels fournisseurs de répondre sur des exigences précises en y associant les coûts correspondants;

- le client demande aux éventuels fournisseurs de préciser les éléments de fiabilité qu'ils comptent mettre en oeuvre, en y adjoignant les coûts qui leur sont propres. Ceci afin de permettre un choix au client des clauses de fiabilité qu'il compte introduire dans le futur contrat.

D'une façon plus précise, le client peut demander :

. la présentation, sous forme d'un plan de fiabilité (même succinct), des tâches et moyens que le fournisseur compte mettre en oeuvre au titre de la fiabilité;

. un calcul prévisionnel de fiabilité.

A ce propos, dans le but de disposer d'éléments de comparaison entre les différentes réponses à l'appel d'offres, le client peut demander que celles-ci soient faites sur une base commune pour certains points tels que : prévision de fiabilité, source de données, etc...

4.1.4. Introduction des clauses de fiabilité dans le contrat

Il convient de distinguer entre les divers types de contrat possibles, que l'on peut classer en trois grandes catégories :

- contrat d'étude,
- contrat de prototype ou de développement,
- contrat de fourniture de série,

Lorsqu'on se trouve dans une phase d'étude, il paraît difficile de se fixer un objectif de fiabilité précis dans la mesure où un matériel "en étude" évolue au cours de celle-ci.

A fortiori, il parait impossible de vérifier quelque paramètre que ce soit, dans la mesure où le matériel ne peut exister que sous forme de schémas, documents....
La clause de fiabilité ne pourra donc, en tout état de cause, que se résumer pratiquement à une étude théorique de fiabilité débouchant sur un calcul prévisionnel, servant à fixer des objectifs. Ceux-ci ne pouvant être vérifiés que dans une phase ultérieure.

Lorsqu'il s'agit d'un contrat de développement, on peut admettre de démarrer sur un objectif de fiabilité issu d'un calcul prévisionnel effectué au cours de l'étude. Durant cette phase, la clause de fiabilité pourrait porter sur l'élaboration d'essais de fiabilité permettant de contrôler le bien-fondé du niveau de fiabilité requis. A la fin de cette phase, on pourrait être amené à modifier les objectifs de fiabilité à spécifier pour les équipements ou systèmes de série. Devant l'impossibilité éventuelle de réaliser des essais de fiabilité statistiquement probants, il pourrait être admis un nouveau calcul prévisionnel relatif aux équipements ou systèmes issus de la phase développement.

Dans le cas d'un contrat de fourniture de série, la clause de fiabilité va consister essentiellement à vérifier les objectifs de fiabilité fixés lors des phases précédentes. S'il en est ainsi, dans la majorité des cas, cette vérification se fera en exploitation. Les modalités de celle-ci seront spécifiées au contrat sur la base, par exemple, du chapitre III du document 191/CCT "Annexes techniques". Les éventuelles incidences financières ou techniques, attachées aux écarts entre les objectifs fixés et les résultats d'évaluation seront spécifiées au contrat.

Il est à remarquer que ceci ne doit pas être à sens unique et que la rédaction définitive des clauses de fiabilité introduites dans le contrat ne doit se faire qu'après négociation entre les parties concernées.

Nous n'avons évoqué jusqu'ici que le cas où des objectifs quantitatifs sont fixés. Dans certains cas, les clauses de fiabilité peuvent ne se rapporter qu'à une assurance de fiabilité i.e. que client et fournisseur définissent ensemble une liste de tâches à remplir au titre de la fiabilité. Un guide pour l'établissement et la réalisation de celles-ci est constitué par le chapitre III du document général 190 A/CCT. Ce statut même de guide signifie que la spécification particulière du contrat pourra comporter des paragraphes ou sous-paragraphes ne figurant pas dans le document précité.

4.2 - Guide pour l'établissement d'un plan de fiabilité.

Le but de ce chapitre (chapitre III document A/CCT) est d'apporter une aide :

- au client, pour spécifier les tâches à remplir au titre de la fiabilité dans le cas où il souhaite imposer des actions qu'il juge utiles (cas, par exemple de clauses portant sur une assurance de fiabilité);

- au fournisseur, pour mettre en oeuvre les tâches spécifiées au contrat ou éventuellement mettre en place les actions et moyens qu'il juge nécessaires pour parvenir à des objectifs quantitatifs spécifiés au contrat.

L'ensemble de ces tâches constitue un éventail de possibilités parmi lesquelles client et fournisseur ont naturellement toute liberté de choisir et/ou d'en ajouter d'autre à leur convenance.

Nous ne nous étendrons pas plus sur cette partie, dont l'essentiel s'est inspiré de documents officiels déjà existants.

Il nous a semblé utile cependant de compléter certains aspects du plan de fiabilité à caractère technique. Comme cela a déjà été dit précédemment, ces développements ont été introduits dans un document séparé, nous nous contenterons de faire état des points essentiels.

5 - ANNEXES TECHNIQUES AU DOCUMENT PRINCIPAL 190A/CCT : DOCUMENT 191/CCT

Celui-ci a été divisé en trois grands chapitres :

- Annexe technique T1 : Prévision de fiabilité;
- Annexe technique T2 : Essais de fiabilité;
- Annexe technique T3 : Evaluation de la fiabilité en exploitation.

5.1. - Prévision de fiabilité

Cette première annexe définit les méthodes particulières applicables aux prévisions de fiabilité. Elle constitue un support au développement des techniques de fiabilité prévisionnelle et complète certaines dispositions générales du plan de fiabilité.

Les méthodes de prévision qui y sont brièvement évoquées sont maintenant bien connues et ne sont, bien sûr, pas uniques. Un des points essentiels cependant du chapitre, évoque le "Choix des sources de données". Dans le cadre d'un appel d'offres par exemple, et dans le but d'aider au choix de différentes solutions possibles, il nous paraît essentiel qu'une source de données unique soit utilisée.

En matière de composants électroniques, deux sources sont actuellement à notre disposition : le Recueil de données de fiabilité du CNET (Centre National d'Etudes des Télécommunications) et le document américain MIL HDBK 217 B "Reliability Prediction of Electronic Equipment" dont on peut raisonnablement penser que les mises à jour fréquentes en feront des références suffisamment valables.

Nous sommes tout à fait conscients que, dans certains cas, de meilleures sources d'informations peuvent être utilisées : par exemple, données d'exploitation sur des équipements similaires, résultats d'essais, etc.... Si telle est la situation, il est indispensable que l'utilisateur de celles-ci puisse les justifier.

5.2. - Essais de fiabilité.

Ce chapitre est entièrement issu de la Spécification générale 190/CCT "Fiabilité des équipements électroniques". Seuls quelques aménagements de forme et de présentation ont été apportés. Les plans d'essais qui y sont exposés ne sont applicables que pour les équipements dont le taux de défaillance est constant.

5.3. - Evaluation de la fiabilité en exploitation.

Ce chapitre est un guide pour la mise en place d'un système d'évaluation de la fiabilité en exploitation. Il en précise les règles, les moyens et les méthodes à utiliser.

Ce qui y est consigné est, dans son ensemble, le fruit de l'expérience des administrations et services publics, dans le suivi des équipements en service.

6 - CONCLUSIONS

Ces deux documents dont je viens de parler, ne traitent en fait qu'une partie du problème qui intéresse le client et bien sûr le constructeur.

En effet le client, qui est souvent aussi l'utilisateur, demande aux équipements ou systèmes qu'il achète, une qualité de service la meilleure possible pour un coût donné. Les deux documents évoqués ici ne s'intéressent qu'à l'aspect fiabilité, alors que la qualité de service est un problème de disponibilité. Elle inclut bien sûr la fiabilité, mais aussi la maintenabilité i.e. l'aptitude à la maintenance de l'équipement. Un document a d'ailleurs été aussi édité sur ce sujet par le Comité de Coordination des Télécommunications.

DISCUSSION

M.Giraud, Fr

Envisagez-vous dans le document à paraître l'application de plans d'essais Bayesius?

Réponse d'auteur

Non, pas actuellement. Ces plans sont similaires à ceux de la norme américaine MIL STD 781 B.

ETUDE DE LA CROISSANCE DE LA FIABILITE D'UN EQUIPEMENT ELECTRONIQUE SOUMIS A DES CLAUSES DE FIABILITE

J.C. CHABIN

S.A. CROUZET

B.P. 1014 - VALENCE - FRANCE

R E S U M E

L'apparition, depuis plusieurs années, des contrats de fiabilité a été un des facteurs prépondérants de l'intérêt manifesté par les équipementiers pour les techniques de fiabilité prévisionnelle.

Les résultats constatés en début d'exploitation du matériel sont souvent très éloignés de la prévision, mais les différentes tâches associées aux programmes de fiabilité appliqués aux équipements permettent d'en améliorer le comportement.

La connaissance de la loi de croissance de la fiabilité en fonction du temps de fonctionnement cumulé d'un équipement faisant l'objet d'efforts permanents d'amélioration doit aider l'équipementier à mieux cerner ses prévisions de fiabilité en vue de la négociation d'une garantie de fiabilité.

I. LES CONTRATS DE FIABILITE

La généralisation des contrats de garantie de fiabilité appliqués aux équipements a sensibilisé profondément les équipes de fiabilité sur les difficultés à prévoir le comportement du matériel en vue de le garantir. Ces difficultés sont d'autant plus grandes qu'un matériel aéronautique militaire présente généralement un faible taux d'utilisation.

Les contrats de garantie de fiabilité existant en France présentent différents types de clauses, selon que le but recherché est la fiabilité ou la disponibilité du matériel, qui se traduisent soit par des incidences financières directes (pénalités et/ou bonifications) liées ou non au coût de maintenance, soit par le prêt d'équipements supplémentaires.

Lors des négociations de ces contrats, la question se pose donc de savoir quelle valeur garantir et sur quelles références techniques. En laissant volontairement de côté la stratégie commerciale liée à ce type de garantie, la base d'établissement d'une valeur garantie de fiabilité dépend de la connaissance que l'équipementier a du produit au moment où cette garantie lui est demandée. Le cas le plus critique étant celui où la demande d'engagement est jointe à l'appel d'offres traitant d'un nouvel équipement.

2. PREVISION DE FIABILITE

L'établissement d'une garantie de fiabilité pour un matériel nouveau en phase de définition ou de prédéfinition impose donc l'utilisation d'une méthode de prévision suffisamment fiable.
Une des méthodes les plus couramment utilisée pour du matériel électronique reste le MIL-HDBK 217.
En fait, les différentes révisions du MIL-HDBK complètées par les différents articles ou communications sur les modèles de prévision de fiabilité des composants démontrent la difficulté d'établir une prévision réaliste.

Les deux principales causes en sont :

- L'évolution rapide des technologies.
- Des observations statistiquement insuffisantes pour suivre cette évolution.

Ce dernier point est particulièrement important en aéronautique militaire où les activités ne dépassent guère 500 heures de vol par an et par avion.

L'expérience montre qu'une prévision de fiabilité du type RADC ou MIL-HDBK 217 est optimiste vis-à-vis du comportement opérationnel d'un matériel en début d'exploitation et pessimiste pour un matériel en utilisation depuis plusieurs années.

Pour un matériel d'avionique militaire, la valeur prévisionnelle est parfois atteinte au bout de quelques années, ce qui correspond à un âge moyen de 1.000 à 1.500 heures de vol (Cf figure 1).

3. FACTEURS INFLUENCANT LA FIABILITE EN EXPLOITATION

Différents facteurs influencent la fiabilité en exploitation d'un équipement. Ils sont liés soit à l'équipement lui-même, soit au support (l'avion), soit enfin à l'utilisation. Les principaux facteurs avec les centres de responsabilités associés sont rappelés sur la figure 2. Cette liste non exhaustive ,car fonction de l'application, permet cependant de dégager les trois modes d'actions majeurs pour améliorer le comportement en exploitation d'un équipement :

- Modification de l'équipement (équipementier)
- Modification de l'interface Avion-Equipement (Avionneur ou Equipementier)
- Modification des conditions d'utilisation (utilisateur)

Il n'est pas inutile de rappeler ces éléments montrant bien que l'équipementier, seul garant contractuellement de la fiabilité de son matériel, n'est pas seul impliqué, et qu'à défaut de partager les risques, il souhaite que les contraintes liées à un programme efficace d'amélioration de la fiabilité soient partagées avec les autres parties.

4. IMPORTANCE DU PROGRAMME DE FIABILITE

Le but du programme de fiabilité est de prendre en compte l'ensemble de ces facteurs afin de contrôler et d'orienter l'évolution de l'équipement pendant les phases de conception,développement, réalisation et exploitation.

Sans récapituler les différentes étapes d'un tel programme, il est important de souligner sa caractéristique principale, à savoir, être un système bouclé, basé sur le recueil et le traitement des anomalies et couvrant l'ensemble des phases d'élaboration d'un produit, et pouvant être représenté par la figure 3.

Les différentes sources d'information utilisées pour cette construction de la fiabilité sont principalement :

Phase de conception	- Etude de fiabilité prévisionnelle - FMEA - Etude des conditions d'utilisation des composants - Etude de sécurité - Etude des nomenclatures - ----------
Phase de développement	- Suivi des incidents en cours de développement - Suivi de la mise au point et d'essais divers - ----------
Phase de fabrication	- Suivi des incidents en réception - Suivi des incidents en fabrication - Suivi des incidents en déverminage
Phase d'exploitation	- Suivi des incidents en exploitation

L'analyse de ces informations conduit généralement à un certain nombre d'actions correctives, telles que :

- Sélection et choix des composants et technologies
- Modifications de conception
- Modifications de gammes de fabrication
- Modifications de la politique de maintenance
- Introduction d'essais de déverminage
- ----------

Il est donc naturel de penser qu'au fur et à mesure que des informations concernant des incidents sont appréhendées, les actions correctives en découlant auront pour conséquence une amélioration de la fiabilité.

5. ETUDE DE LA CROISSANCE DE LA FIABILITE

Il faut considérer deux types de croissance de la fiabilité en fonction du temps de fonctionnement :

- L'un, que certains auteurs (BEZAT,A.,1975) appellent "déverminage permanent" (ou Endless Burn-in), est relatif au phénomène de remplacement des composants à l'occasion des défaillances. En effet, la plupart des composants électroniques ont une telle espérance de vie que toute défaillance révélée dans une population améliore la fiabilité du reste.

- L'autre est lié à l'efficacité des actions correctives décidées après l'étude du comportement.

Seul, ce deuxième phénomène a fait l'objet de nombreux travaux et de modélisations associées, sans doute parce qu'en début d'exploitation la part du taux de défaillances induit au cours de l'élaboration du produit représente une part importante du taux de pannes global. La figure 4 montre le processus d'amélioration de la fiabilité.

Cette croissance de la fiabilité est une fonction directe du temps de fonctionnement cumulé par l'équipement concerné, puisque ce fonctionnement permet de révéler les anomalies.

L'étude de cette croissance de fiabilité peut donc être faite à toutes les phases d'élaboration d'un matériel, lorsque celles-ci comportent des périodes de fonctionnement, par exemple :

- Essais de développement,
- Essais de déverminage sur équipement,
- Exploitation.

Ceci explique qu'un matériel de conception nouvelle, issu d'un programme de développement trop court, a souvent en exploitation un comportement très éloigné des prévisions de fiabilité et reste susceptible d'être amélioré.

La présence d'une garantie de fiabilité rend donc indispensable la recherche de la modélisation de l'évolution du comportement consécutive aux efforts d'amélioration consentis.

Plusieurs auteurs ont présenté des modèles de croissance de la fiabilité et, parmi ceux-là, J.T.DUANE de General Electric a défini en 1962 un modèle qui s'est révélé à l'usage souvent très apte à décrire le comportement d'un équipement électronique ou électromécanique caractérisé par un taux de pannes constant.

RAPPELS SUR LA THEORIE DE DUANE

$$\lambda_\Sigma = \frac{N}{H} = K.H^{-\alpha}$$

- λ_Σ = taux de défaillances cumulé
- H = Nombre d'heures de fonctionnement cumulé
- α = Coefficient de croissance de la fiabilité (en principe > 0)
- K = Constante
- N = Nombre de défaillances cumulé

Le taux de défaillances instantané est obtenu par différentiation du taux de défaillances cumulé.

$$\lambda(t) = \lim_{\Delta H \to 0}\left(\frac{\Delta N}{\Delta H}\right) = (1-\alpha).K.H^{-\alpha}$$

$$\lambda_i = (1-\alpha).\lambda_\Sigma$$

Ceci peut se représenter simplement sous forme d'un diagramme comme le montre la figure 5.

OBSERVATIONS FAITES SUR α

- α varie généralement de 0,15 à 0,5
- Les plus grandes valeurs de α sont le fait de matériels analogiques par opposition aux matériels numériques.
- α est d'autant plus grand que le matériel présente un caractère de nouveauté.
- α tend à être plus grand pour les matériels soumis à de sévères conditions d'environnement.
- α est d'autant plus grand que l'effort de correction des anomalies est important.

Trois variables principales vont donc conditionner l'amélioration de fiabilité (LILIUS,W-A,1978):

- Nombre de sources de défauts identifiées
- Efficacité des actions correctives
- Temps nécessaire aux différentes phases du processus

L'importance de ce troisième facteur peut-être représentée par la figure 6 - Les différentes durées de phase indiquées sur cette figure sont assez représentatives de ce qui peut se passer réellement en aéronautique militaire.

Un délai d'environ deux ans s'écoule entre l'identification d'une source de défauts et l'observation de l'amélioration de la fiabilité du matériel soumis à une action corrective.

Il est ainsi possible de mettre en évidence un certain nombre de limitations qui doivent inciter à la prudence dans l'utilisation à des fins prévisionnelles de tels modèles.

- L'efficacité du système repose sur la durée de la boucle de modification vis-à-vis de la durée du programme d'amélioration de la fiabilité ainsi que sur le *flux d'informations à traiter.*
- Même en cas de nombreuses défaillances, encore faut-il que leurs mécanismes soient faciles à détecter, à identifier et à corriger.

Ce sont quelques unes des raisons qui font que la croissance de la fiabilité est surtout observée en phase de développement et en début d'exploitation.

6. PRINCIPE D'APPLICATION A RETENIR

L'utilisation d'un modèle de croissance de fiabilité doit conduire à une meilleure prévision du comportement du matériel pendant les premières années d'exploitation, afin d'en tenir compte au mieux lors de la négociation des contrats de garantie de fiabilité.

La méthode à retenir, schématisée par la figure 7, peut s'énoncer de la manière suivante :

a/- Observation, par familles de matériels et pour un même contexte de réalisation et d'utilisation, des courbes de croissance de la fiabilité obtenues avec la situation de la valeur prévisionnelle.

b/- Au titre d'une étude nouvelle, extrapolation, d'après les résultats précédents, d'une valeur "a priori" du coefficient de croissance.

c/- Prévision de fiabilité par une méthode classique (ex MIL HDBK 217 B).

d/- Construction de la courbe prévisible pour l'évolution de la fiabilité en exploitation de l'équipement considéré.

Cette méthode permet ainsi de répondre au mieux aux exigences d'un contrat de fiabilité en présentant les résultats sous différentes formes :

- Courbe continue du MTBF en fonction du temps
- Valeurs du MTBF par paliers en fonction du temps
- Valeur initiale + valeur finale

Bien que cette méthode soit un moyen efficace de limiter l'incertitude d'une prévision faite très en amont de l'expérimentation, il faut conserver en mémoire le fait que la validité des prévisions, en l'absence d'essais représentatifs, est entièrement basée sur la confiance accordée aux tables de taux de défaillances utilisées.

Un développement suffisant avec un programme d'essais complet, permet de s'affranchir de cette variable et de retrouver l'utilisation classique des lois de croissance de la fiabilité appliquées aux résultats d'essais.

Ce n'est pas toujours le cas, et pour un équipement de conception nouvelle, l'application d'une telle méthode permet d'aborder les problèmes de garantie de fiabilité avec un peu plus de réalisme qu'au moyen des techniques classiques de prévision de fiabilité.

BIBLIOGRAPHIE

- BEZAT, A. et al., 1975, "Growth modelling improves reliability predictions", Annual R and M Symposium.
- BIRD , G.T. et al., 1976, "Avionics reliability control during development", AGARD LS 81.
- CODIER , E.O.,1968, "Reliability growth in real life", Annual R and M Symposium.
- CODIER, E.O., 1969, "Reliability predictions - Help or Hoax ?", Annual R and M Symposium.
- COX, T.D. et al., 1976, "Reliability growth management of SATCOM terminals", Annual R and M Symposium.
- GREEN, J.E., 1973, "the problems of reliability growth and demonstration with military electronics", Microelectronics and Reliability, Vol 12.
- GREEN, J.E., 1976, "Reliability growth modelling for avionics", AGARD LS 81.
- LILIUS, W.A.,1978, "Reliability growth planning and control", Annual R and M Symposium.
- MEAD, P.H.,1975, "Reliability growth of electronic equipment", Microelectronics and Reliability, Vol 14.
- MEAD, C. et al., 1975, " Reliability growth management in USAMC", Annual R and M Symposium.
- SUMMERLIN, W.T., 1976, "Illusory reliability growth", AGARD LS 81.

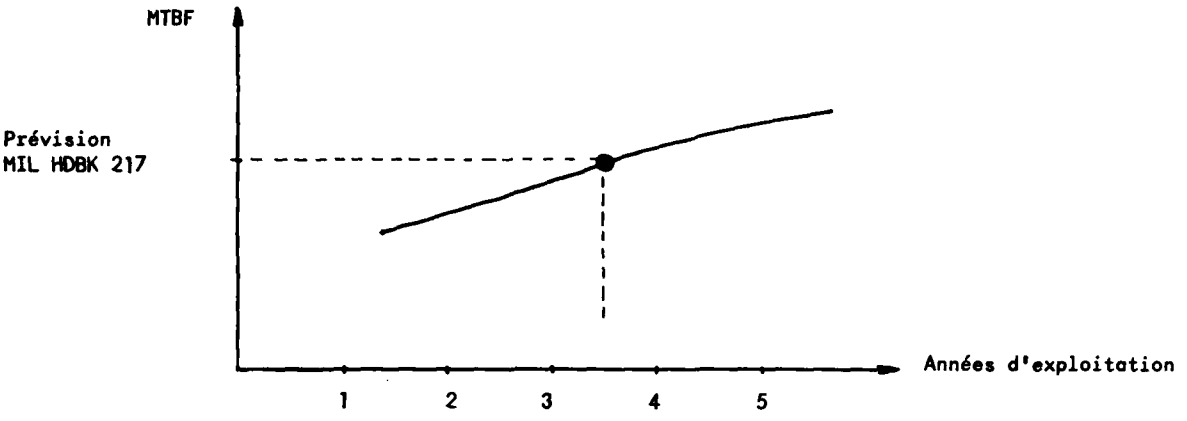


Figure 1

	Client	Equipementier	Avionneur	Utilisateur
- Spécifications	x			
- Qualité de la Conception		x		
- Qualité du développement	x	x		
- Qualité des composants et technologies		x		
- Qualité de la fabrication		x		
- Politique de Maintenance				
. Définition		x		
. Application				x
- Facteurs liés à l'avion (type, mission, environnements opérationnels...)			x	x
- Efficacité du système de Recueil et de traitement des anomalies ainsi que des actions correctives en résultant		x	x	x

Figure 2

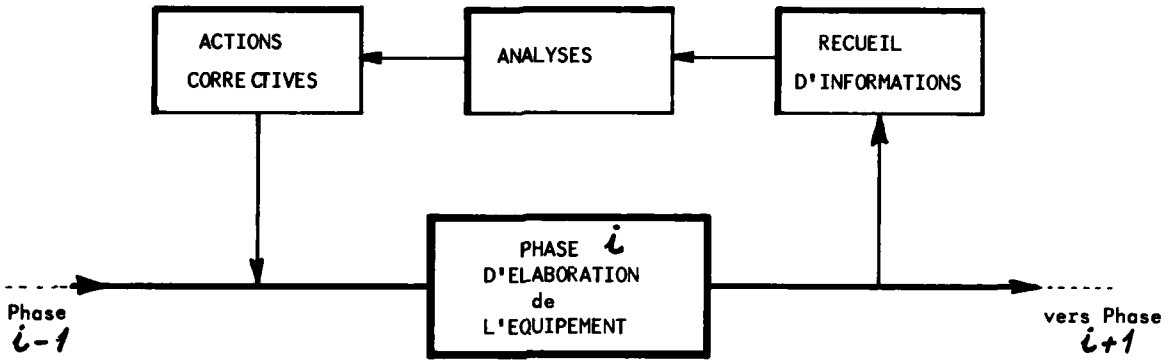


Figure 3

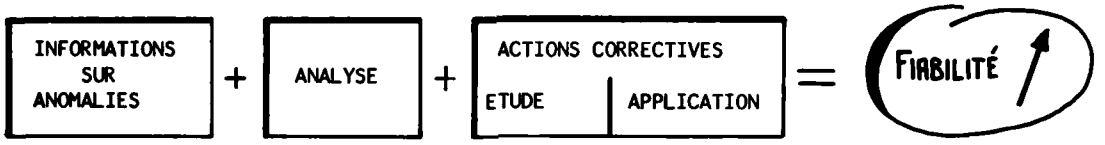


Figure 4

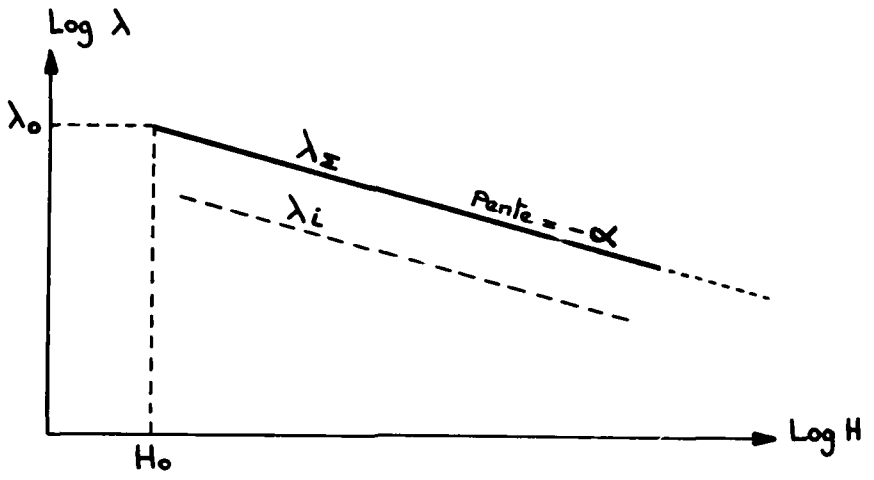


Figure 5

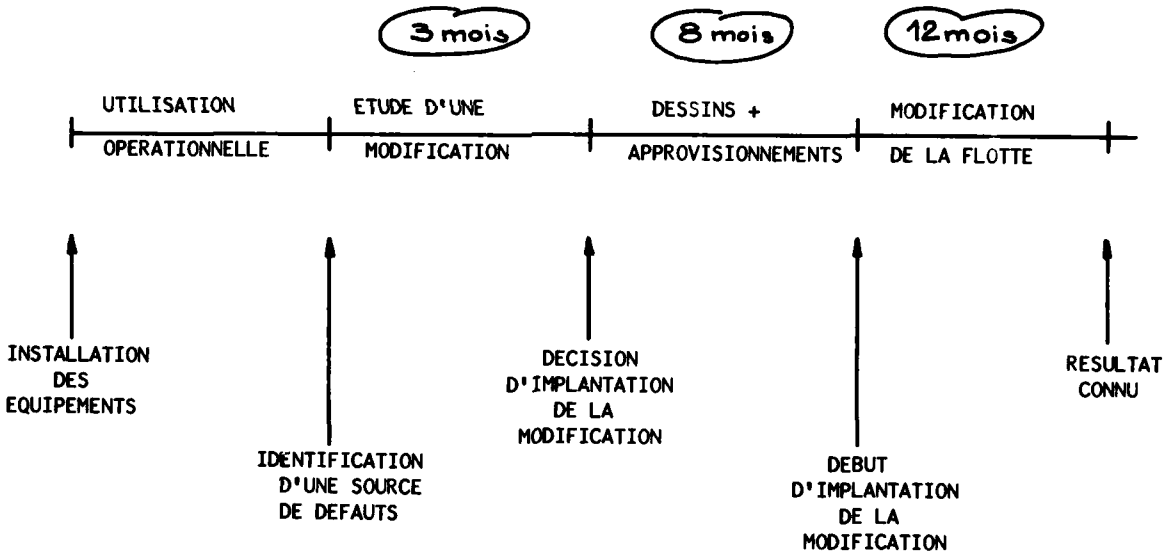


Figure 6

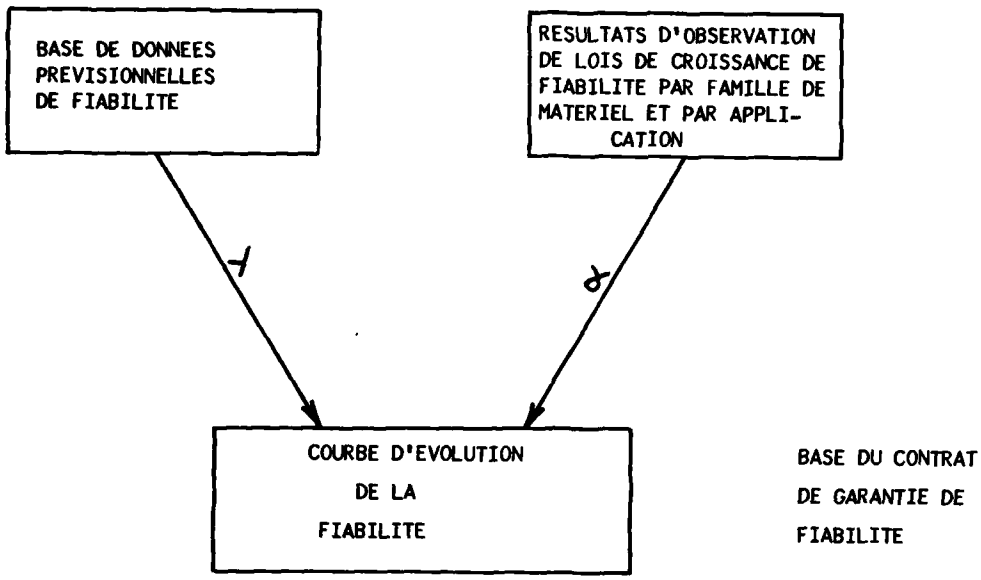

BASE DE DONNEES PREVISIONNELLES DE FIABILITE
RESULTATS D'OBSERVATION DE LOIS DE CROISSANCE DE FIABILITE PAR FAMILLE DE MATERIEL ET PAR APPLI-CATION
γ
α
COURBE D'EVOLUTION DE LA FIABILITE
BASE DU CONTRAT DE GARANTIE DE FIABILITE

Figure 7

AMELIORATIONS DE FIABILITE DUES A L'APPLICATION DES CLAUSES DE FIABILITE OPERATIONELLE

par

Ingénieur Principal de L'Armement J.Laurensou
Service Technique des Télécommunications de l'Air
Paris, France

INTRODUCTION

Une bonne fiabilité de ses divers constituants est nécessaire pour obtenir une bonne disponibilité opérationnelle d'un système d'armes quelconque. Aussi tout organisme chargé de l'approvisionnement ou de la maintenance des équipements a-t-il le souci d'obtenir la fiabilité optimum compte tenu des contraintes financières imposées. La fiabilité est déterminée par les choix faits lors de la conception d'un équipement mais elle peut être améliorée en observant son fonctionnement réel et en lui apportant des modifications.

Le présent exposé va donc décrire le système mis en oeuvre pour la connaissance et l'amélioration de la fiabilité d'un certain nombre d'équipements utilisés sur différents aéronefs des Armées françaises et donnera les résultats obtenus sur un cas concret.

Pour pouvoir améliorer la fiabilité d'un équipement il est nécessaire de:

- recueillir les données concernant les pannes (faits techniques)
- déterminer la fiabilité réelle en utilisation
- définir les modifications à apporter
- inciter les constructeurs à améliorer la fiabilité de leurs équipements en leur apportant des modifications
- contrôler l'incidence de ces améliorations sur la fiabilité.

Le recueil des données est fait à l'aide du Système Automatisé d'Informations Techniques (S.A.I.T.), la fiabilité est déterminée par un groupe de travail et l'incitation à l'amélioration est fournie par l'application des clauses de fiabilité.

1. RECUEIL DES DONNEES CONCERNANT LES PANNES (Faits techniques)

Dans l'Armée de l'Air française, il a été mis en place un Système Automatisé d'Informations Techniques (SAIT) dont le document de base, la FIT (Fiche d'Intervention Technique), est établi chaque fois qu'une panne ou une anomalie quelconque apparait lors de l'utilisation d'un équipement.

1.1 Les Faits Techniques

Les informations concernant l'exploitation des équipements électroniques en service dans l'Armée de l'Air sont de plus en plus prises en compte par un Système Automatisé d'Information Technique (SAIT). Ce système regroupe tous les faits techniques intervenant pendant la période d'exploitation opérationnelle des systèmes mis en oeuvre par l'Armée de l'Air. C'est donc le système d'information permettant de connaître les défaillances des équipements électroniques.

On dit qu'il y a "Fait Technique" chaque fois qu'une intervention a lieu sur un matériel, à titre correctif ou préventif. Les informations correspondantes sont recueillies sur un document de base, la Fiche d'Intervention Technique (FIT).

C'est donc grâce à ce document de base qu'il est possible au technicien de connaître les informations relatives à la fiabilité réelle de l'équipement auquel il s'intéresse. Ces informations sont indispensables pour l'amélioration de la fiabilité de l'équipement considéré; ils donnent également des informations indispensables pour de nouvelles générations d'équipement. Nous allons donc examiner successivement comment s'effectuent le recueil, le traitement et l'exploitation des faits techniques, permettant l'étude des pannes en fonctionnement réel.

1.2 Recueil des Faits Techniques

Le recueil des faits techniques est la fonction qui consiste en la saisie des informations techniques élémentaires au niveau de l'utilisateur. Ce recueil a une grande importance car, de lui, dépendent l'aboutissement et la qualité des études techniques, en particulier en matière de fiabilité, par l'analyse qualitative et statistique des défaillances. En particulier, il doit être systématique et précis.

Ensuite, par exploitation et traitement ultérieurs, on établira des informations plus élaborées (états d'analyse et de synthèse) à partir de ces faits techniques élémentaires.

Le document de recueil des faits techniques est la FIT (Fiche d'Intervention Technique). C'est un imprimé format standard (21 x 29,7 cm) présenté sous forme de liasse de 4 feuillets autocopiants, qui auront chacun un destinataire différent; en particulier un de ces feuillets sera adressé au Centre de Calcul responsable du traitement automatique.

1.3 Traitement et Exploitation des Faits Technique

Le traitement des faits techniques est effectué de façon automatique à l'aide de calculateur. Il a pour objet la centralisation, la mise en mémoire et le tri des informations techniques recueillies grâce aux FIT et d'en déduire des informations plus élaborées aux plans techniques, opérationnels et logistiques. Ces informations sont regroupées dans différents documents, qui peuvent être périodiques ou établis à la suite d'une demande particulière.

Pour ce qui concerne la fiabilité, deux documents (ou "états") sont principalement utilisés:

Etats de synthèse

Ils sont établis selon une périodicité trimestrielle ou annuelle. Ils donnent une vision globale de la fiabilité au niveau d'un système d'armes (avion par exemple). Pour chaque équipement ou sous-ensemble important, on trouve: le nombre d'heures de vol, le nombre d'équipements en fonctionnement, le nombre de FIT émises et de défaillances confirmées, le MTBF "apparent" (relatif aux heures de vol) et le taux de défaillance.

$$\text{MTBF apparent} = \frac{\text{nombre d'heures de vol X quantité par avion}}{\text{nombre de défaillances confirmées}}$$

Ces états sont diffusés à l'Etat-Major et aux Services Techniques, pour connaître la fiabilité des équipements sur les différents aéronefs. Ils permettent:

- de comparer la fiabilité (MTBF) des différents équipements d'un avion,
- de connaître l'évolution avec le temps du MTBF de chaque équipement.

Etats d'analyse

Pour la fiabilité et l'analyse des pannes affectant les équipements, l'état le plus communément utilisé est un document intitulé "CLASDEF" ou "état de classement des défaillances par cause de panne". Cet état regroupe dans un seul document les différentes FIT émises pour un équipement donné. Pour chaque équipement, on trouve donc des renseignements sur les constatations faites et la (ou les) causes de la panne; ces informations ont été tirées des FIT que l'on a décrites précédemment.

L'état de classement des défaillances par cause de panne est utilisé par les Services Techniques, en liaison avec les industriels, pour examiner les circonstances des avaries, les causes de ces avaries et déterminer les éléments critiques d'un équipement. C'est donc le document essentiel permettant l'analyse des pannes en exploitation des équipements électroniques de bord. C'est aussi le document de base pour l'application des clauses de fiabilité garantie.

En effet, à l'aide de cet état d'analyse, une Commission composée de représentants de l'Etat-Major, des Services Techniques et de l'Industiel fabricant l'équipement, examine les interventions effectuées et détermine en particulier si les pannes sont ou non imputables à l'équipement. Connaissant par ailleurs le nombre d'heures de vol des avions sur lesquels l'équipement est monté et compte tenu d'un coefficient heures de vol/heures de fonctionnement, il est aisé de déterminer le MTBF du matériel.

2. CLAUSE CONTRACTUELLE DE FIABILITE

2.1 Principe des Clauses de Fiabilité Garantie

Par les clauses de fiabilité, l'industriel fournisseur de l'équipement s'engage à obtenir une valeur N de MTBF en exploitation réelle (par exemple N = 800 h pour un TACAN). Cette valeur contractuelle est fixée après discussion entre fournisseur et client. Parfois, elle est fixée après des essais de fiabilité en laboratoire.

Dans le cas où la valeur contractuelle N n'est pas obtenue en fonctionnement réel l'industriel doit effectuer gratuitement toutes les corrections et modifications nécessaires pour atteindre l'objectif, sur les matériels à livrer et également sur ceux déjà livrés; ces modifications ont pour but de remédier aux pannes systématiques.

D'autre part, si la valeur obtenue réellement est inférieure à N 20%, l'industriel s'engage à effectuer toutes les réparations gratuitement (pièces de rechanges; main d'oeuvre; déplacement. . .)

Cette garantie de MTBF est assurée en principe pour 5 ans à partir de la livraison.

2.2 Conditions d'Application

La fiabilité de l'équipement est représentée par son MTBF opérationnel. Ce MTBF opérationnel est établi périodiquement (tous les 3 mois en principe) à partir des états d'analyse et de synthèse de FIT, sur tous les équipements en service du type considéré. Une commission de spécialistes composée de représentants des utilisateurs, du Service Technique client et de l'industriel fournisseur, retient les pannes directement imputables au matériel et établit le MTBF opérationnel pour la période considérée.

L'établissement du MTBF demande la connaissance du nombre de pannes et du nombre d'heures de fonctionnement.

Le nombre d'heures de fonctionnement est celui réalisé par tous les équipements en service. Il est lu sur des compteurs horaires; dans le cas où il n'y a pas de compteur horaire, on détermine les heures de fonctionnement à partir des heures de vol, par application d'un coefficient fixé à priori, et qui dépend du type d'avion et du type d'équipement considéré.

Les pannes prises en considération sont celles qui sont imputables au matériel; elles englobent toutes les anomalies et fonctionnements défectueux qui empêchent l'équipement d'avoir des performances opérationnelles satisfaisantes et de remplir sa mission.

Les pannes imputables comprennent donc:

- les défaillances techniques de pièces ou composants, même si les pièces ou composants satisfont aux exigences qui leur sont imposées par le dossier de fabrication.
- un mauvais fonctionnement quelconque décelable au banc d'essai, dans les conditions extrêmes d'utilisation prévues aux clauses techniques et dont la cause directe est intermittente ou inconnue.
- la défaillance imputable à plusieurs pièces de types différents qui doit être considérée comme constituant plusieurs défaillances, si chaque pièce considérée séparément empêche d'atteindre les performances satisfaisantes. En revanche, elle doit être considérée comme une défaillance unique si chaque pièce ne peut à elle seule, provoquer la défaillance de l'équipement.
- les défaillances entrainées par une mauvaise conception.
- les défaillances entrainées par une fabrication défectueuse.
- les défaillances imputables à tout défaut des réglages effectués en usine.

En revanche, on ne retient pas pour la détermination du MTBF:

- les fonctionnements défectuex ou anomalies dûs à des erreurs de manipulation, à des procédés de contrôle, de réglage, d'installation non conformes.
- les défaillances résultant directement d'une autre défaillance déjà décomptée, si elles apparaissent avant un délai de 50 heures après la première défaillance.

Le MTBF est garanti dans les conditions d'utilisation suivantes:

- l'équipement ne doit pas être utilisé de façon continue dans les conditions climatiques extrêmes.
- les tensions d'alimentation et les transitoires ne doivent pas excéder des limites autorisées.
- la compétence du personnel effectuant les opérations de détection de panne et d'échange de sous-ensemble doit être suffisante pour qu'il n'y ait ni fausse manoeuvre ni insuffisance dans le travail effectué.
- les composants dont la durée de vie arrive à expiration doivent être remplacée.

3. DETERMINATION DU MTBF REEL EN EXPLOITATION

Les renseignement issus du S.A.I.T., en particulier les "états de classement des défaillances par cause de panne" permettent de déterminer:

le MTBF global au niveau de l'équipement entier. C'est à ce niveau qu'il est déterminé si le constructeur satisfait ses obligations contractuelles Si le MTBF constaté est inférieur au MTBF garanti, le fournisseur devra modifier à ses frais les équipements. Le suivi de ce MTBF global permet de constater les améliorations ou les détériorations de fiabilité qui pourraient survenir.

le taux de défaillance de chacun des sous-ensembles (ou modules) constituant l'équipement. La détermination de ce taux au niveau des sous-ensembles permet d'isoler les éléments les moins fiables et permet d'éviter que des variations contraires sur des sous-ensembles soient invisibles au niveau de l'ensemble.

4. APPORT DES AMELIORATIONS

4.1 Choix des Améliorations à Apporter

Après détermination du MTBF, pour déterminer l'endroit où faire porter l'effort d'amélioration, à l'aide des renseignements fournis par le S.A.I.T., on établit un état de répartition des pannes par types de défaillance et par types de modules.

Un exemple de cet état est donne en figure 1, à titre d'exemple. Cet état prend en compte toutes les défaillances survenues au cours d'une période donnée et les ventile par type de défaillance et par module. Ceci permet de voir apparaître les défaillances systématiques dûes à certains composants et met aussi en relief les pourcentages de défaillance par module.

En fonction de ces éléments, une attention particulière sera apportée au composant défaillant ou au module ayant le plus de panne et une action corrective sera entreprise.

4.2 Réalistion des Améliorations

En fonction des éléments ci-dessus, le fournisseur déterminera les améliorations à apporter aux équipments et après accord du Service de Surveillance Industrielle, ces améliorations seront apportées en série sur les chaines en cours de fabrication et aussi en rattrapage soit par déplacements des spécialistes sur les bases soit lors du retour des équipements en usine pour réparation.

En application de ce processus, sur un émetteur-récepteur, il a été appliqué 8 modifications principales tant en série qu'en rattrapage, et sur un autre équipement, un sous-ensemble a été complètement réétudié. Un suivi très précis est fait, appareil par appareil, pour l'application en rattrapage de ces modifications.

5. RESULTATS OBTENUS

Les résultats suivants ont été obtenus sur un émetteur-récepteur utilisé sur un grand nombre d'aéronefs.

5.1 Au Niveau des Modules

Les résultats sont schématisés dans les tableaux en figure 2. Sur ce tableau, on a porté, à titre d'exemple, pour 2 modules parmi les plus importants, l'émetteur et le synthétiseur, le taux de défaillance de chaque module par rapport à l'ensemble complet.

Une pente négative traduit l'amélioration du module. On peut, sur ces tableaux, remarquer des améliorations très notables survenant, pour l'émetteur, après une modification, et pour le synthètiseur après une modification et une amélioration de la fabrication.

5.2 Au Niveau du MTBF de l'Equipement Complet

Le tableau en figure 3 donne l'évolution du MTBF dans le temps. On peut y constater que de 1974 à 1978, la fiabilité opérationnelle de l'équipement, tous utilisateurs confondus, s'est accrue de 65%.

Pour les 3 utilisateurs principaux, ces accroissements ont été de:

- 65% pour l'utilisateur n° 1
- 136% pour l'utilisateur n°2
- 97% pour l'utilisateur n° 3

Remarque

L'auteur n'ignore pas les incidences bénéfiques que peuvent avoir, sur la fiabilité, une meilleure accoutumance des services de maintenance ainsi que des taux de fonctionnement plus élevés mais il faut reconnaitre que l'essentiel des accroissements spectaculaires constatés est dû aux améliorations apportées.

6. CONCLUSION

Les résultats ci-dessus, confirmés par des expériences indentiques sur d'autres équipements, démontrent que les clauses de fiabilité permettent, par le suivi qu'elles imposent une meilleure connaissance du comportement en utilisation de l'équipement et par les mesures prises, une amélioration de la fiabilité de cet équipement.

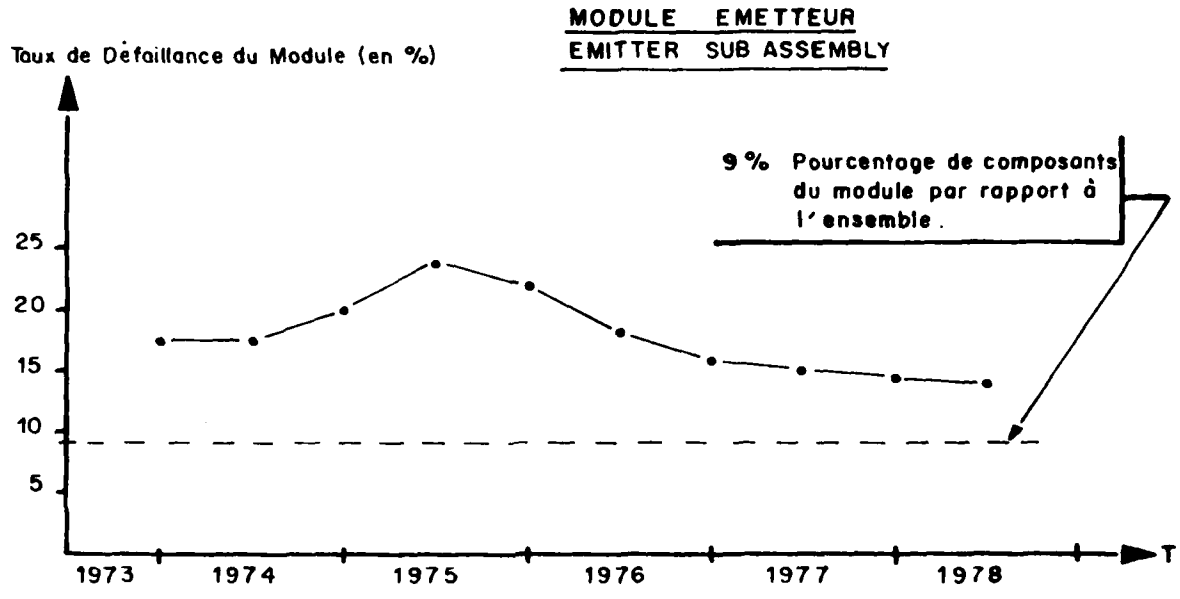
Ainsi, ces clauses de fiabilité permettent d'assurer aux utilisateurs une meilleure disponibilité opérationnelle de leurs équipements en un coût de maintenance global réduit.

Répartition des Pannes par Types de Défaillance et par Types de Modules

Types de Défaillance		Module Concerné					Totaux		%
		Nº 1	Nº 2	Nº 3	Nº 4	-----			
I COMPOSANTS									
- Circuits Intégrés	Q1	1		1			2		
	Q2		4		1		5	9	12,85
	Q3	1		1			2		
- Transistors	Q4	2	2		1		5		
	Q5		2	1	3		6	11	15,70
- Diodes	CR1	1	3	2			6		
	CR2		2	1	3		6	12	17,10
- Condensateurs	C1	3	2	2	1		8		
	C2	1		2	2		5	13	18,60
- Resistances	R1	1		2			3		
	R2		2		1		3	6	8,60
TOTAL COMPOSANTS		10	17	12	12		51		72,85%
II FABRICATION									
- Contrats Défectueux		1		1			2		
- Courts Circuits			1				1		
- Défaut de Montage				1	1		2		
- Fil Cassé		1					1		
- Soudure Défectueuse			1		3		4		
- Divers		3		2	2		7		
TOTAL FABRICATION		5	2	4	6		17		24,30%
III REGLAGES		1		1			2		
TOTAL REGLAGES		1		1			2		2,85%
TOTAUX GENERAUX DES DEFAILLANCES		16	19	17	18		70		
POURCENTAGE PAR MODULE		22,85%	27,10%	24,30%	25,70%				

Figure 1

– Evolution du Taux de Panne par Module –



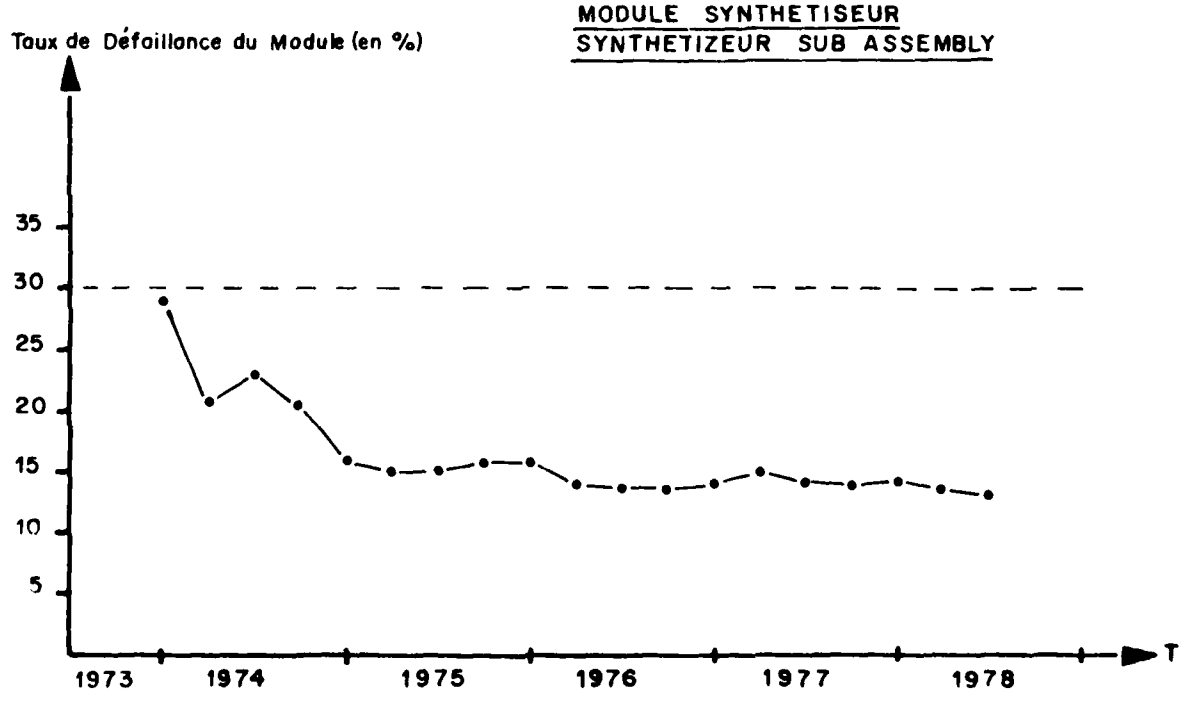


Figure 2

- EMETTEUR RECEPTEUR - - TRANSCEIVER -

- Amelioration de la Fiabilité - - Reliability Improvement -

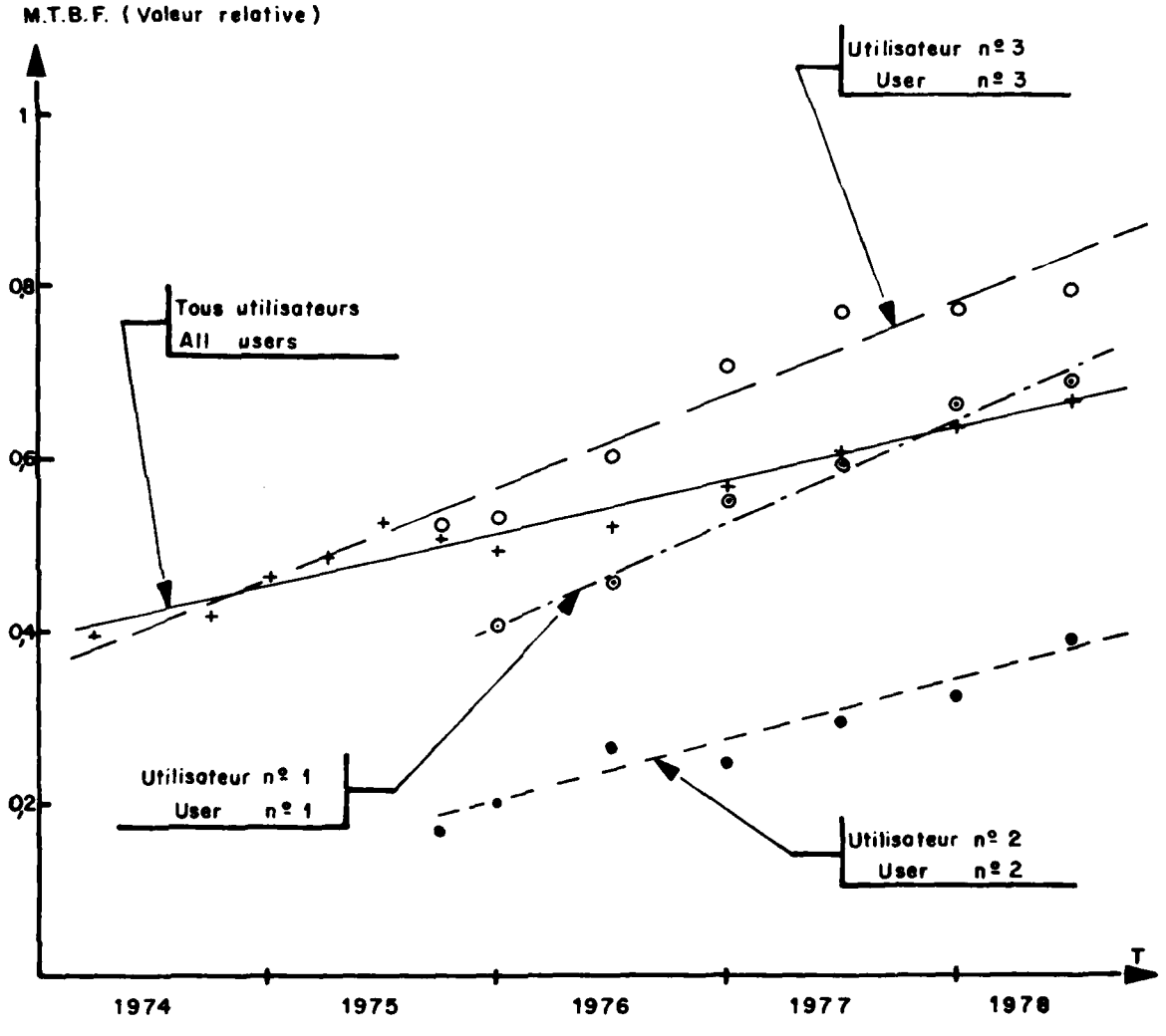


Figure 3

DISCUSSION

M.Jacobsen, Ge

Do French RIW contracts specify the maximum turn-around time for repair items?

Réponse d'auteur

Le temps de réparation des sous-ensembles défectueux est fixé par contrat, que la réparation soit gratuite ou non. Ainsi, en cas de retard, l'industriel supporte des pénalités de la même façon que lors d'un retard pour une réparation non gratuite.

J.N.Basmaison, Fr

Avez-vous pû, parmi les 30% de pannes non imputables à l'équipement chiffré la part des défauts générés par la génération de bord (alimentation système)?

Réponse d'auteur

Effectivement, parmi les environs 30% de pannes observées non imputables à l'équipment, certaines on été provoquées par des défectuosités de la génération électrique. Mais leur nombre n'est pas suffisant pour établir une statistique significative.

PRODUCTION RELIABILITY ASSURANCE (PRA) - TESTING

Artfried Weihe, MESSERSCHMITT-BÖLKOW-BLOHM GMBH
OTF , Systemunterstützung FE 07
Postfach 80 11 60
D - 8000 München 80

SUMMARY

Production Reliability Assurance Tests are all-equipment-reliability tests which are applied for assurance purposes during the production phase of major equipments once the formal qualification status concerning reliability is achieved. A fundamental principle of the tests is the liability of the producer for corrective actions if the test requirements are not met: i.e. if pattern defects are present and/or if the Corrective Action Required (CAR) line is exceeded by the plot of accumulated failures. Suitable test conditions for both producer and customer are obtained by selection of an appropriate CAR line and by restart of the plot after a specific test experience is gained. CAR lines can be derived from Mil-Std-781 test plans but are not limited to these.
Newly developed lines have proved to be more appropriate for short test experience and high specified MTBF. A comparison with fixed - length tests shows that shorter time to decisions is obtained in the case of PRA-testing.

1. INTRODUCTION

Production Reliability Assurance Testing (PRA-Testing) is part of a comprehensive Reliability Programme for major equipments. It is a test which is intended to be applied to every equipment produced during series production and shall assure that the inherent level of reliability achieved by development activities is maintained during production. Hence a full reliability programme which includes PRA-Testing normally embraces the following tests:

Development Phase:

1) Burn-in test on every equipment
2) Reliability Improvement Test
3) Reliability Demonstration Test as part of the equipment qualification

Early Production Phase

1) Burn-in test on every equipment
2) Early Production Reliability Demonstration Test

Series Production Phase

1) Burn-in test on every equipment
2) PRA-test on every equipment

This programme can be modified and adapted to individual cases by deletion of Improvement Tests or Development or Early Production Reliability Demonstration Tests. The deletion of the latter would almost invariably lead to the application of PRA-testing from the first series production unit onwards.

2. TECHNICAL DETAILS OF PRA-TESTING

The principle of PRA-testing, i.e. to test every equipment produced, necessitates a relatively short test time per equipment in order to keep the test costs and life consumption low and to maintain the required rate of delivery of the equipments produced. A test time (equipment on-time) of 30 hrs. per equipment is considered reasonable. The PRA - test is performed after the burn-in test and prior to the final acceptance test and is, therefore, part of the production process. The test conditions applied shall simulate, as far as possible, those expected during in service operation. The recent discussions about Mil-Std-781 B test conditions not being representative of inservice operation, which lead to revised Mil-Std-781 C test conditions are also valid for PRA-testing. A better simulation of operational loads will result in a higher efficacy of the PRA-test.

In order to obtain a relation of the test results and the contractually stated MTBF requirement, the test hours and failures of the equipments tested are accumulated, hence resulting in a staircase chart. This chart is compared with a straight line (see Fig. 1) which is called "Corrective Action Required line" (CAR-line). The slope of the CAR-line is dependent on the specified MTBF. The selection of a special type of CAR-line has to be based on the overall number of test hours expected and the specified MTBF such that the CAR-line is an indicator of the specified MTBF probably being met by the equipment.

Hence no corrective action is required as long as the plot of failures vs time remains below the CAR-line, failures are remedied and testing continued as production units become available. However, if the CAR-line is exceeded by the staircase or if pattern defects exist it is very probable that the equipment does not meet the MTBF requirement. In such a case corrective actions (and not solely repair actions) which improve reliability are mandatory to be applied to all equipments produced thereafter e.g. changes of the production process, of quality control, selection of components, environmental and operational conditions, design changes or change of burn-in length.

A fundamental point of PRA-Testing is the contractual coverage of the corrective actions. The principle is that all expenditures for corrective actions are covered by the production contract. An early knowledge of this fact already in the development phase should stimulate the suppliers to design a high degree of reliability into the equipments thus obviating, in advance, possible faults and the consequent expenditure for corrective actions which these faults would have engendered, and thereby increasing the suppliers profit. Unfortunately, this idea cannot be realised in all cases due to the application of only cost-plus contracts by some public customers.
After description of the general idea of PRAT some further details are presented in the following paragraphs.

3. SELECTION OF AN APPROPRIATE CORRECTIVE ACTION REQUIRED LINE (CAR - LINE)

One of the most important steps during the definition and negotiation of PRA-test requirements is the finding of an appropriate Corrective Action Required Line. The starting point in doing this is the definition of acceptable risks for user and supplier of the equipment. As usual the purchaser's risk is defined to be the probability of accepting a low MTBF equipment (the risk is the probability that the staircase chart of a low MTBF equipment stays below the CAR-line). Correspondingly the supplier's risk is the probability of a high MTBF equipment being rejected (the risk is the probability that the staircase chart of a high MTBF equipment exceeds the CAR-line). Preferably these risks are selected to be 10 or 20 % as known from Mil-Std-781 testing.

In order to enable the selection of suitable risks for all MTBF levels in a reasonable PRA-test time, sets of different CAR-lines have been considered which are identical to the reject lines of Mil-Std-781 sequential test plans. Test plan XXIX of Mil-Std-781 B is one of these CAR-lines. However, it turned out that, in particular equipments with a required MTBF between 2000 and 4000 hrs. could not be PRA-tested in a reasonable time with reasonable risks. Therefore, in addition, more suitable CAR-lines independent of Mil-Std-781 lines have been developed and the risks determined.
The calculation of risks was performed by the usage of the Poisson distribution. Looking at the CAR-line a) of Fig. 1 (which is the reject line of sequential test plan III of Mil-Std-781) it is obvious that the probability of continuing testing between O and T_1 is given by the probabilities of having 0, 1 or 2 defects in this period. The probability to continue between T_1 and T_2 is given by the probabilities of the following failure combinations

- O failure between 0 and T_1 and either 0, 1, 2 or 3 failures between T_1 and T_2 or
- 1 failure between 0 and T_1 and either 0, 1 or 2 failures between T_1 and T_2
- 2 failures between 0 and T_1 and either 0 or 1 failure between T_1 and T_2

The probabilities for continuing testing between T_2 and T_3 are calculated correspondingly, et seq. This determination of probabilities by progressively proceeding to higher test times was resolved with the aid of a computer programme. The outcome of this calculation is shown in Fig. 2 where the accept probability is drawn as a function of the test time accumulated by individual equipment PRA-tests. The ratio of the true MTBF to the specified MTBF serves as a parameter. As a first important result it is obvious that the accept probabilities are dependent on the accumulated test times. With small accumulated test times the accept probabilities are relatively high and decrease with increasing accumulated test time. This feature is quite suitable for testing equipments where some lack of confidence in the reliability still exists at the start of testing and also at the time of negotiating PRA-testing. It can help in persuading equipment suppliers to abandon their objections to PRA-testing as, at the beginning, even not fully satisfactory equipments get a chance of being accepted. Only during the further run of the test, when the production process is well established, the test criteria become more stringent.
That these accept probabilities change with accumulated test time means also, that the PRA-test probabilities are different, in general, from corresponding test plans of Mil-Std-781.
As another example a CAR-line which is not derived from a Mil-Std-781 test plan is presented in Fig. 1 b). This CAR-line suits high MTBF equipments where the accumulated test time to MTBF ratio remains small.
With the above derived knowledge in mind we return to the original task of selecting a suitable CAR-line.

1st example : Specified MTBF = 1000 hrs. Testtime per equipment = 30 hrs.
No. of equipments = 100 (at least)
The suppliers and the purchasers risks shall be limited to 10 %
where a "bad equipment" is defined to have a MTBF of less than
0,3 times specified MTBF.

Using the CAR-line of Fig. 1a) it can be found that the length of the test must be about 3000 test hours or more which means that 100 equipments must be tested to get the above specified risks. It does not mean that a decision is only possible after achievement of 3000 hours because there is a good chance of much earlier rejection of bad equipment.

2nd example: Specified MTBF = 4000 hrs. Testtime per equipment = 30 hrs. No of equipments = 100 (at least). The suppliers risk shall be about 20 %. The test time shall be in the range of the above example, i.e. 3000 test hours. In this case, because of the test time limitation, it is impossible to define a purchasers risk as low as above. However, where the selection of a CAR-line according to Fig. 1a) yields a purchasers risk of 70 % and a suppliers risk of less than 2 %, the selection of the CAR-line of Fig. 1b) results in a much more suitable ratio of risks: The purchaser's risk is then 26 % and the supplier's risk is 23 % which is an acceptable test criteria for a test which has a length of only 0,75 times the specified MTBF.

By these two examples it is shown, that, by selection of appropriate test plans, test conditions which are acceptable to both, purchaser and supplier, can be developed for different MTBF values.

Other CAR-lines which start with a horizontal line and then, after a specific test time, continue with the usual upward sloping line have been developed and the probabilities determined. These lines allow for reliability growth of the equipments under test during the initial test time, starting with a MTBF lower than specified and reaching the specified MTBF during the further run of the test.

4. REPETITION OF TEST PLOTS

When the number of produced equipments and the test time accumulated by PRA-tests is high such that a PRA-test plot of a usual length of 3 to 4 times specified MTBF is exceeded, the question arises whether to continue the plot or to start further plots. Both alternatives are considered and compared in the following.
Obvious advantages for dividing the overall possible test time into several plots can be determined without any special mathematical consideration. These advantages are:

1.) The possibility for the purchaser and the supplier to negotiate only one limited testplot with well established test conditions, which are defined during the negotiations in advance of the actual test performance. Later, during the run of the test and based on the test experience it could be decided whether to stop testing after the first plot or to continue testing by starting a new test plot.

2.) The case has to be considered that the reliability of the equipments is fairly good, so that the staircase chart is well within the allowed limits for a long time. However, at a certain time the reliability of the equipments could decrease significantly for some reason. Then this fact would result in a reject decision far more quickly if the test criteria were given by more than one plot (see also Fig. 3).
Also in Mil-Std-781 C this possible behaviour of equipment reliability trends is taken into account by the introduction of "boundary lines" into the all equipments test plans IC to VIII C of Mil-Std-781 C.

A mathematical assessment of the PRAT accept probabilities with different numbers of plots for a specific number of test hours is fairly simple. The results are presented in Fig. 4. It is shown that the accept probabilities achieve a minimum value for a certain number of plots assuming that the overall number of equipments / test hours is the same for all different numbers of test plots. The limit of accept probabilities for all different ratios of true MTBF to specified is 1, if the number of plots is going to infinity. From these curves the most appropriate number of plots can be easily obtained. The selection of the exact minimum would mean a low accept probability for low MTBF equipment but also a rather low accept probability for high MTBF equipment, the latter resulting in a high producers risk. Therefore it is better to select a number which is smaller than the minimum, in our example a figure of less than 5. In this case both of the overall risks can be kept within acceptable limits.

5. TIME TO REJECT IN THE CASE OF LOW MTBF.

Another criterion for the cost-effectiveness of a reliability test is to get a reject decision as early as possible during the test and during the production run in case the MTBF of the equipments is too low. Then corrective actions can be implemented before a great deal of the equipments is produced and delivered. Therefore the concept of PRA-testing was compared, in this respect, with a possible alternative to PRA-tests, namely fixed length tests.

A prerequisite of such a comparison is that the test conditions and the ultimate probabilities of reject for a full PRA-test plot and the alternative fixed length test are made the same. Taking again a PRA-test according to the reject line of sequential test plan III of Mil-Std-781 as an example with a plot time of 2,43 x MTBF, fixed length tests allowing either 4 or 5 defects are determined as possible alternatives, whereby the fixed length test allowing 4 defects yields accept probabilities which are a little lower than the PRA test and the fixed length test allowing 5 defects yields accept probabilities which are a little higher than the PRA test (see the following table).
However, the expected time to achieve a reject decision in the case that the true MTBF equals 0,3 x specified MTBF is at least 17 % shorter in the case of PRA-testing.
For further values see the following table

PRAT, Reject line TP III T = 2,43			Fixed Length Test Plan			
			≤4 defect allowed		≤5 defects allowed	
True MTBF / Spec. MTBF	Prob. of No Corr. Action	Mean Time to Decision	Prob. of No Corr. Action	Mean Time to Decision	Prob. of No Corr. Action	Mean Time to Decision
1,0	0,942	2,360	0,900	2,377	0,962	2,416
0,5	0,591	1,949	0,468	2,025	0,645	2,206
0,3	0,159	1,245	0,093	1,456	0,180	1,701

This result can be easily verified by the following qualitative considerations:
Because of the slope of the CAR-line PRA Testing is marked by rather stringent test conditions from early testing onwards whereas in the case of fixed length tests (horizontal test line) the test conditions become stringent only near the end of the test. Hence, in the latter case, poor reliability equipment will only be detected rather late.

6. REPEATED EARLY REJECTS AS A CLEAR INDICATOR OF POOR RELIABILITY

One rule of the PRA-Testing as presently applied is that the plot is restarted after a reject event and the incorporation of corrective actions. However, the case may be that the corrective action is not effective resulting in a second reject event and eventually further reject events within a short test time.
The probability of this occurring was determined with the true MTBF as a parameter. If N is the number of reject events and restarts and n is the number of allowed defects at the start of the plot, the probability of N restarts was derived to be the probability of any number of defects between and including (N + 1) n + N and Nn + N (using the Poisson distribution). The probability of at least N restarts (or more) is given by the probability of any number of defects of more than and including N (n + 1).
Corresponding curves are presented in Fig. 5. for one, two and three restarts at least. The curves are showing a slope increasing with increasing number of restarts. The conclusion drawn from this is that the supplier's risk of a good MTBF equipment being rejected is decreasing significantly if more than one reject event occurs. Therefore if the fact of more than one reject event is given this is an almost foolproof indicator that the MTBF of the equipment is too low and therefore corrective actions are necessary.

7. APPLICATION OF PRA - TESTING

PRA-Tests as described above are presently being applied in the MRCA Tornado Series Production Programme . A number of main electronic and avionic equipments the MTBF values of which are in the range of 100 to 2000 have been selected for PRAT application. In general, these equipments were subject to Reliability Demonstration Tests during the Development Phase. Several of these equipments were modified after performance of the Reliability Test so that the repetition of a Demonstration Test in form of an Early Production Reliability Demonstration Test was preferred before actually starting PRA-Testing. In these cases PRA-Testing will start on the second production batch. Furthermore some equipments contain lifed items. Also in these cases an Early Production Reliability Demonstration Test was selected for the first production batch (and PRA-testing from the second batch onwards) due to long test times per test unit enabling a better observation of reliability changes with operating time.
The experience gained up to date is limited as the production of series equipments started only recently. Hence PRA-testing is being performed on only about one quarter of the ultimate number of equipments. At time of writing there is no case where a reject decision was achieved with the need for corrective actions.
This result was expected as the probability of rejection is low at the beginning of testing as explained above.

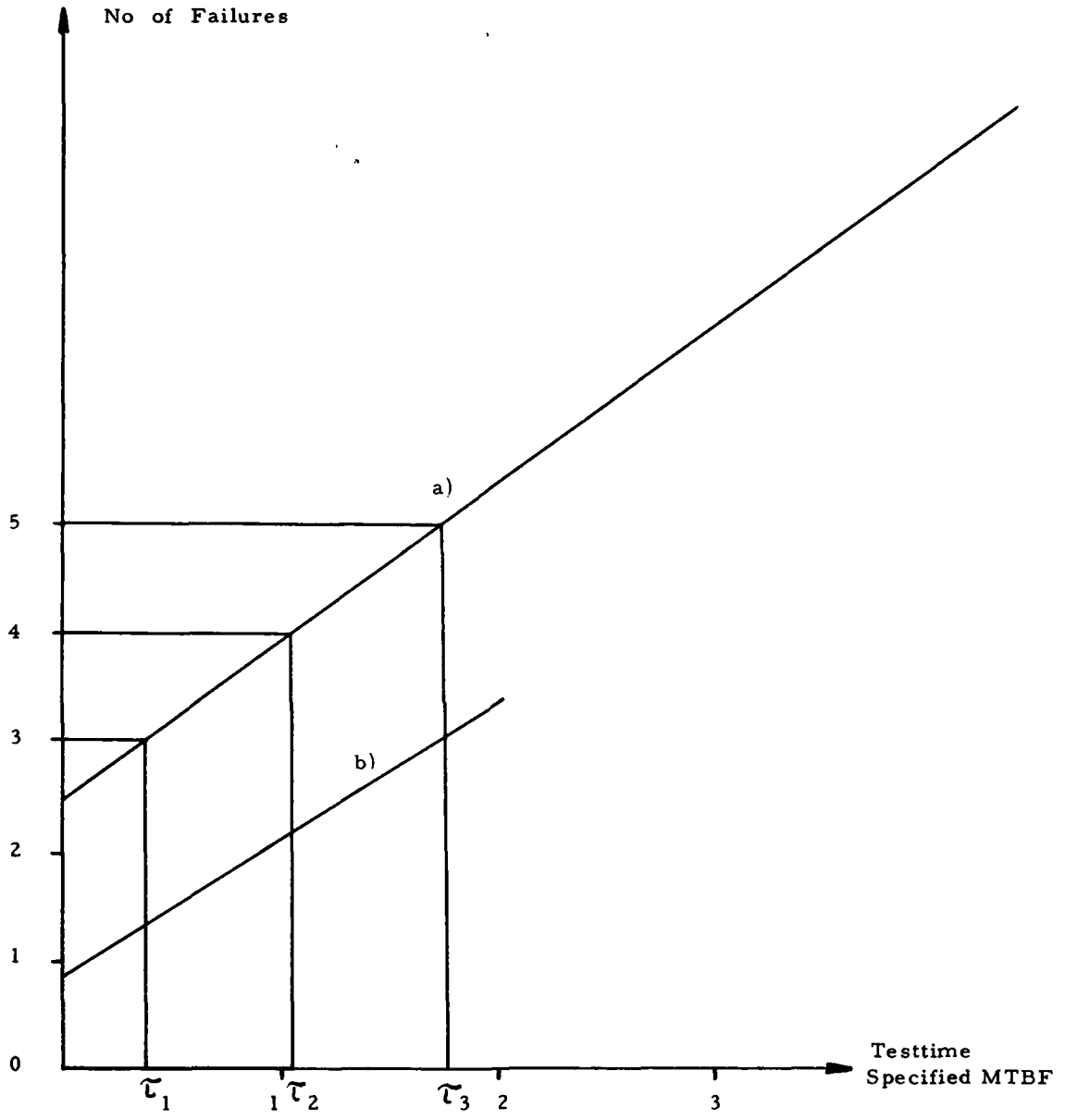


Fig. 1: Corrective Action Required Lines (Examples)
a) according to the Reject Line of Testplan III of Mil-Std-781 (Sequential Testplan)
b) for Short Tests

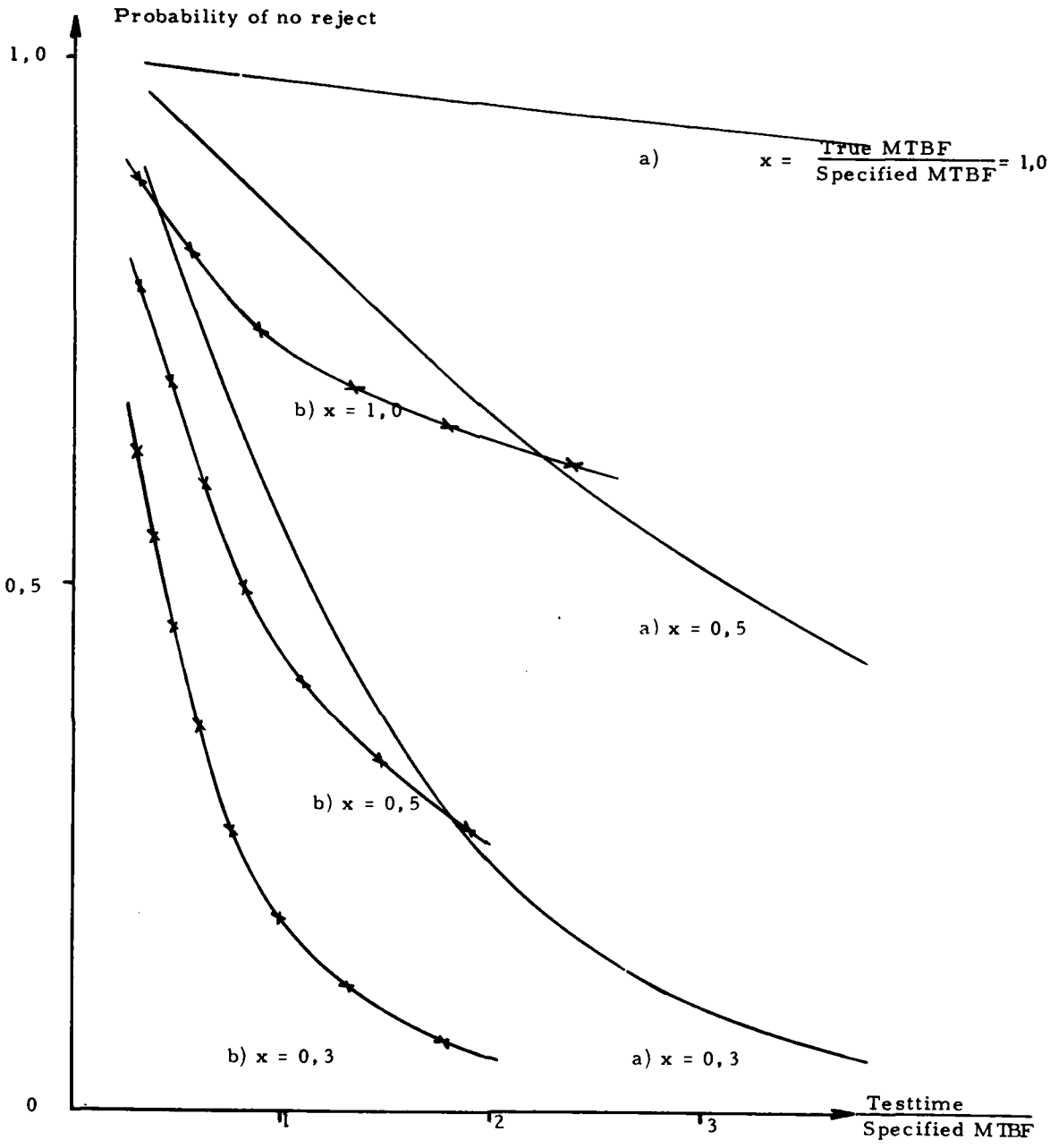


Fig. 2: Probability of no reject for
a) a CAR line according to the Reject Line of Testplan III of Mil-Std-781 (sequent testplan)
b) a CAR line for Short Tests

Parameter $x = \frac{\text{True MTBF}}{\text{Specified MTBF}}$

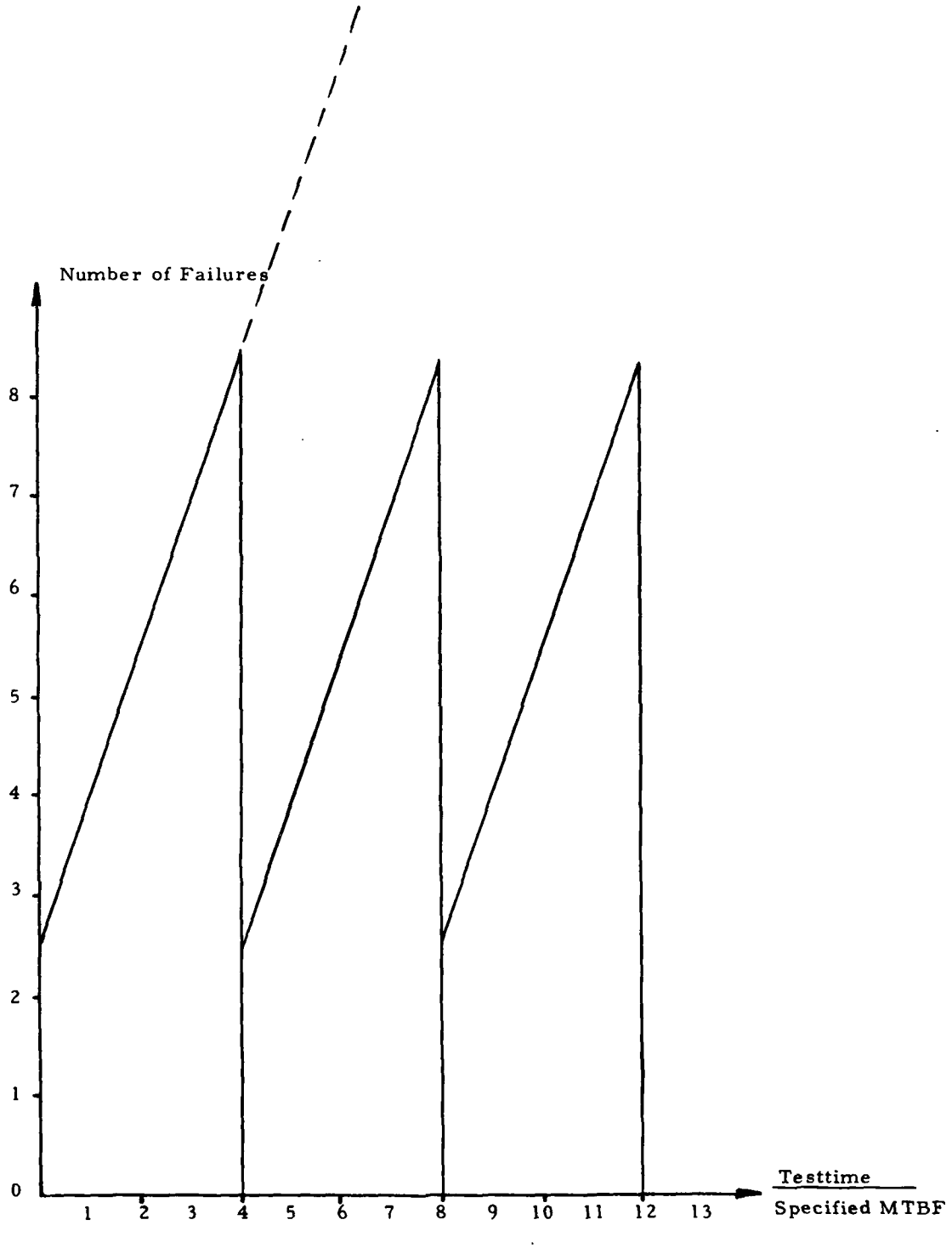


Fig. 3 : Repetition of Testplots during PRAT

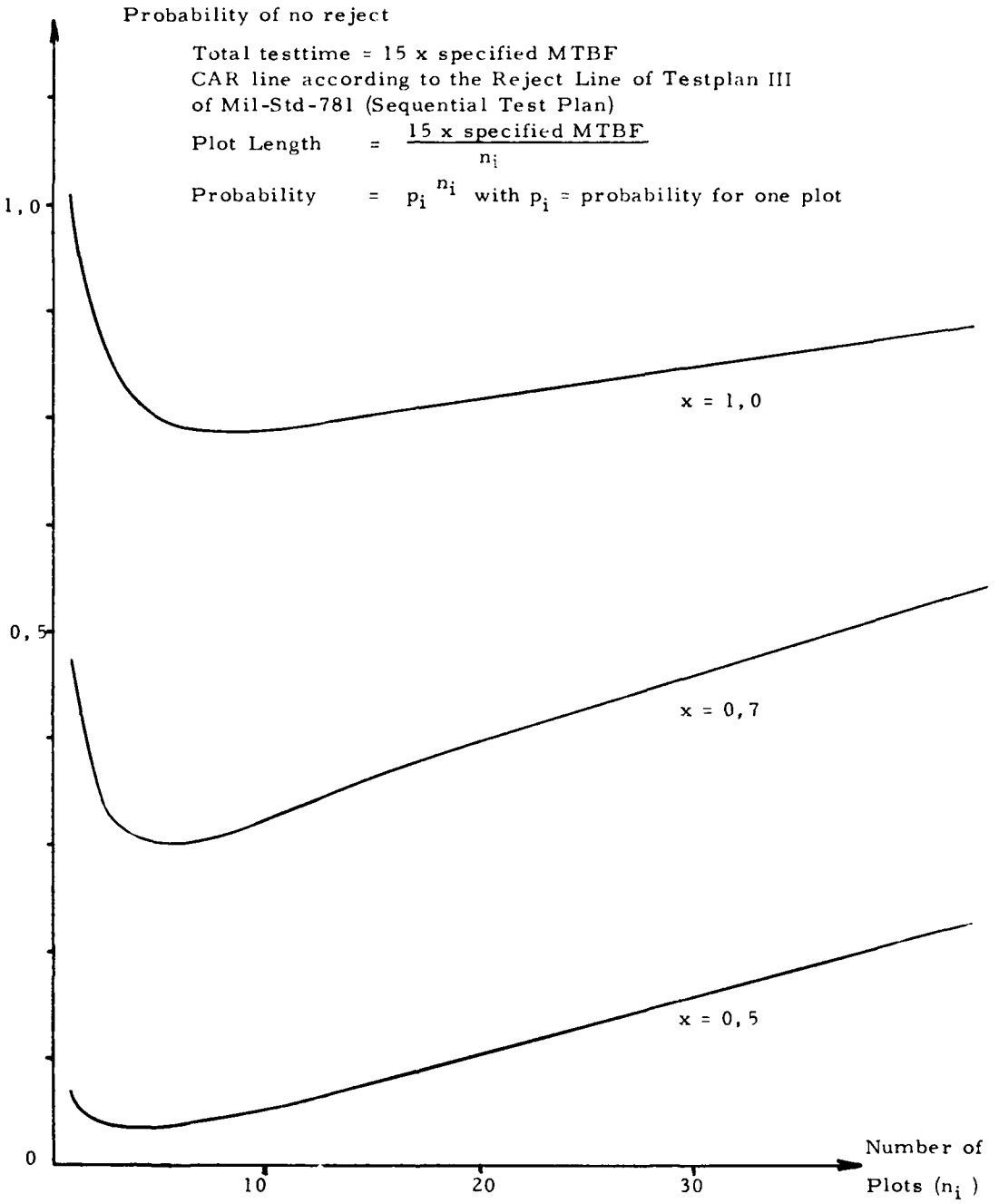
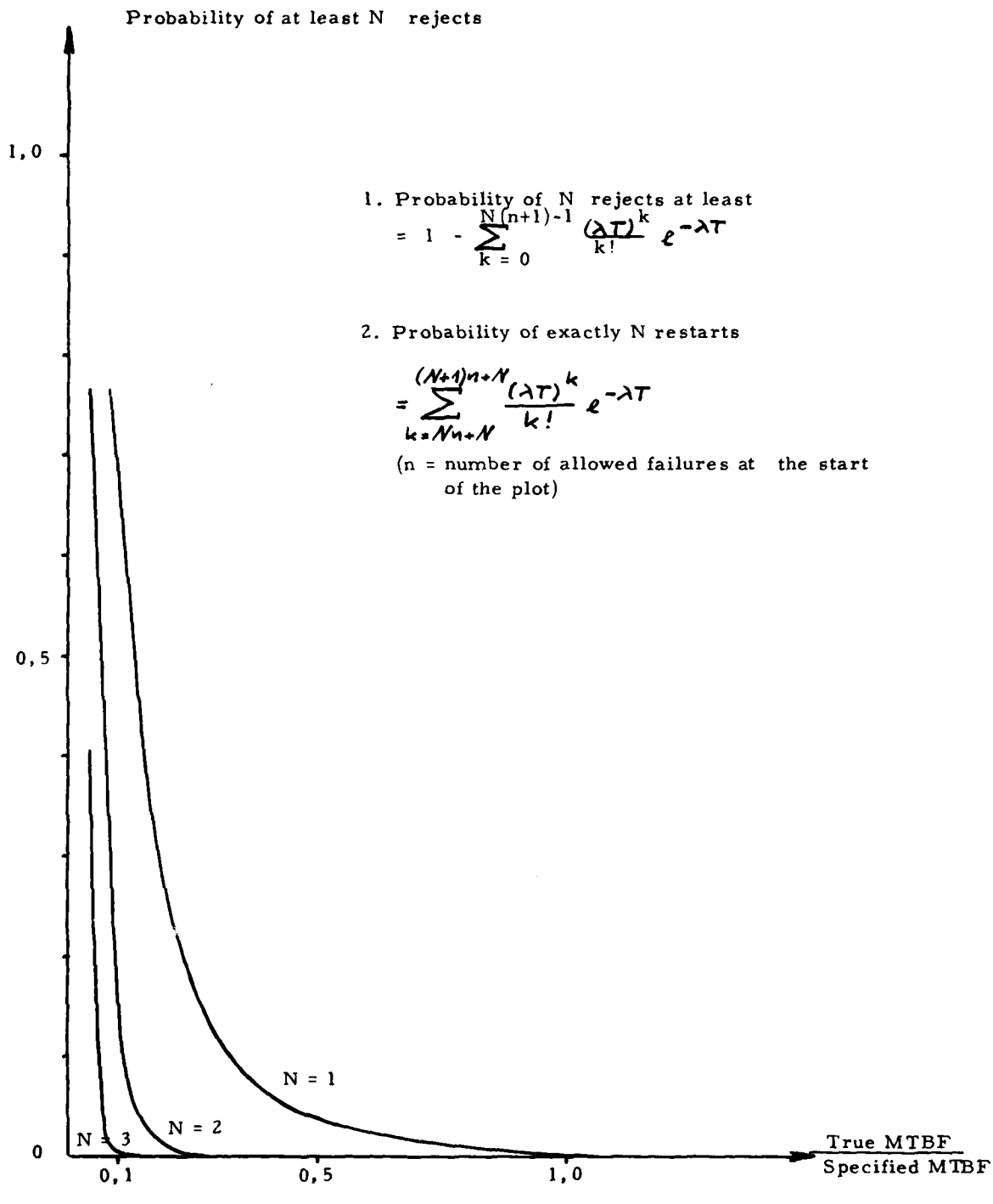


Fig. 4: Overall Probabiiity of no reject as a Function of the Number of Plots
Parameter x = True MTBF / Specified MTBF



1. Probability of N rejects at least

$$= 1 - \sum_{k=0}^{N(n+1)-1} \frac{(\lambda T)^k}{k!} e^{-\lambda T}$$

2. Probability of exactly N restarts

$$= \sum_{k=Nn+N}^{(N+1)n+N} \frac{(\lambda T)^k}{k!} e^{-\lambda T}$$

(n = number of allowed failures at the start of the plot)

Fig. 5: Probability of at least N Rejects within a testtime of 0,35 x specified MTBF for a CAR-Line according to the Reject Line of Testplan III of Mil-Std-781 (Sequential Testplan)

DISCUSSION

M.Jacobsen, Ge

Are the Corrective Action Requirement (CAR) lines modified from batch to batch because of reliability growth in the production process?

Author's Reply

Yes, they can be modified from batch to batch if so required, i.e. if it is shown that the accepted probabilities for lower MTBF values at the beginning of the plot (which are higher anyway than later on) are not high enough to provide acceptable supplier's risks.

W.Ehrenberger, Ge

Concerning Figure 5: Since you use the Poisson distribution one would intuitively assume that you would get step functions instead of smooth ones.

Why did you obtain smooth curves?

Author's Reply

The variable on which the curves are dependent is the ratio of the true MTBF to specified MTBF. When this variable changes, the probability, as shown on the vertical axis, will also change smoothly. The step function character is contained within the different curves shown, i.e. the transition from the curve $\mu = 1$ to the curve $\mu = 2$ etc. is a step function.

J.N.Basmaison, Fr

Précédant l'essai PRA d'une durée de 30^h, vc is avez précisé qu'entre autres un essai de déverminage était nécessaire.

Quelle était la durée de cette essai de déverminage?

Réponse d'auteur

La durée de déverminage était d'environ 48^h avec aucune défaillance au cours des 10 dernières heures.

Pour certains équipements, le déverminage dure 70^h avec les 30 dernières heures sans défaillances.

Cette dernière période sans défaillance est très importante avant de pouvoir procéder à un essai PRA.

METHODES UTILISEES POUR CONNAITRE LA FIABILITE D'UN RADAR D'AVION D'ARMES

J.C. CHARLOT - THOMSON-CSF

178, boulevard Gabriel Péri 92240 MALAKOFF - FRANCE

- : - : -

SOMMAIRE :

Dans le but de connaître la fiabilité d'un radar équipant un avion d'armes THOMSON-CSF a mis en place une organisation de collecte et du traitement des informations nécessaires à mesure de la moyenne des temps du bon fonctionnement.

Au cours de l'exposé, on examine successivement :

- la collecte des informations
- l'organisation existante en 1979
- les traitements effectués
- les perspectives d'avenir (extension à d'autres équipements et information des fabricants de composants)

INTRODUCTION

Le présent exposé se propose de vous faire part d'une expérience de THOMSON-CSF (département radar, Contre-Mesures, missile) en matière d'acquisition de données de fiabilité dans les conditions réélles d'emploi et de traitements de telles informations.

Dans ce but, nous verrons successivement :

1 - Les objectifs poursuivis

2 - La mise en place de l'organisation

3 - L'organisation en 1979

4 - Les traitements

5 - Les perspectives d'avenir

6 - Conclusions

1 - LES OBJECTIFS POURSUIVIS

Le besoin d'informations permettant :

- de prédire la fiabilité d'un équipement

- de modifier éventuellement sa conception dans le sens d'une réduction des taux de pannes,

nous a obligé à poser la question :

Les tables de données, type MIL HDBK 217 B, et les méthodes de calcul associées sont-elles valables pour nos équipements ?

Pour répondre à cette question la méthode la plus directe consiste à mesurer les taux de défaillance des composants et des équipements utilisés dans les conditions d'environnement adéquates et à comparer les résultats à ceux donnés par la MIL HDBK 217 B.

En raison des contraintes qui lui sont propres, (coût et délais en particulier), l'industrie électronique professionnelle produisant des équipements à usage militaire ne peut s'engager dans une mesure systématique des taux de défaillance au niveau des composants.

Un petit exemple simple permettra de s'en convaincre :

Pour la plupart des composants utilisés le taux de défaillance par heure, dans les conditions "avion hors cabine" par exemple, est du 10^{-6} environ ; cela conduit en usine en reconstituant imparfaitement l'environnement (table vibrante et étuve), si on veut obtenir en 3 mois un résultat presque sûr du point de vue statistique (soit 10 pannes environ), à essayer simultanément environ 5 000 composants, ce qui est généralement inaccessible en raison des coûts et de la grande variété de composants utilisés.

Pour tourner cette difficulté en sacrifiant l'exigence sur le délai mais en améliorant la représentativité de l'environnement, il semble intéressant d'organiser une collecte efficace des informations relatives d'une part aux pannes se produisant en exploitation et d'autre part aux durées de fonctionnement ; cette action permet en effet de déterminer les taux de défaillance au niveau des composants et aux niveaux des équipements complets.

2 - MISE EN PLACE DE L'ORGANISATION

En 1973, pour atteindre les objectifs indiqués précédemment et éventuellement faire apparaître des possibilités d'amélioration d'un radar équipant un avion d'armes, une équipe THOMSON-CSF fut chargée de mettre en place, en France, une organisation de collecte des informations nécessaires à une exploitation du type fiabilité et d'effectuer les premiers dépouillements.

Cette équipe fut successivement conduite à :

- identifier les moyens et organisations qui pouvaient contribuer à la collecte des informations

- définir les adaptations nécessaires de façon à donner naissance à un système efficace

- roder les dispositifs mis en place

- convaincre les différents participants de la chaîne de saisie d'informations d'une part de la nécessité d'effectuer de façon aussi précise que possible le travail demandé et d'autre part de l'absence d'arrière pensée chez l'industriel dans la curiosité ainsi manifestée.

- effectuer les premiers dépouillements

La planche 3 montre les flux d'équipements produits et les flux d'informations en retour.

2.1. Les moyens et organisations existants

Ils sont de deux natures :

- les moyens humains
- les moyens documentaires

Les moyens humains comprennent les personnels utilisateurs, les personnels chargés de la maintenance et les personnels THOMSON-CSF chargés de l'assistance technique.

Les moyens documentaires existants sont les rapports de vol écrits par les pilotes (forme 11) et les fiches d'interventions techniques émises par les différents échelons de maintenance.

2.2. Adaptations nécessaires

Les documents existants peuvent sous certaines conditions être utilisés pour atteindre l'objectif fixé dans la première partie de l'exposé.

Pour cela, il a fallu faire parvenir et centraliser ces informations à THOMSON-CSF ; dans ce but, l'équipe qui avait en charge le problème, a demandé à avoir connaissance mensuellement des fiches d'interventions techniques et du temps de vol de la flotte concernée.

2.3. Rodâge du dispositif mis en place

L'analyse des documents reçus a montré qu'une collaboration avec les équipes d'assistance technique THOMSON-CSF était souhaitable pour :

- faire prendre conscience, du point de vue panne, aux pilotes de l'existence des équipements à bord de l'avion en effet, le rapport du pilote signalait les anomalies relatives à la sécurité de l'avion mais omettait quelque fois les anomalies radar.
- expliquer aux différents personnels d'utilisation et de maintenance, l'intérêt de porter le maximum d'indications sur les fiches d'interventions techniques.

La collaboration avec les équipes d'assistance technique permet aussi de vaincre les réticences plus ou moins rationnelles qui se manifestaient devant la "curiosité" du fournisseur.

2.4. Obtenir la coopération du personnel utilisateur

La première condition est de faire admettre le personnel d'assistance technique THOMSON-CSF par tous les échelons hierarchiques du personnel "clients" : pilote, mécanicien, officiers, etc.... Le personnel THOMSON-CSF a dû être choisi avec beaucoup de soins de façon à créer les relations humaines les meilleurs possibles, ce problème est très complexe car le profil du personnel "Assistance technique" le mieux adapté n'est pas le même selon les pays en raison d'usages locaux et de la culture propre à chaque "client". Ce personnel satisfait d'autre part aux exigences de la sécurité militaire.

Lorsque les conditions précédantes sont remplies il faut expliquer clairement à chacun quel est le rôle du personnel d'Assistance technique ; en effet, le recueil des informations relatives aux anomalies et a fortiori aux heures de vol est ressenti comme une intrusion dans les affaires du client.

Il faut montrer l'intérêt qu'il y a à gérer de façon rigoureuse les anomalies ; cela permet en effet de :

- maîtriser correctement les opérations de maintenance
- de déceler les anomalies à caractère répétitif qui peuvent être provoquées par une utilisation, une maintenance mal adaptées.

Pour protéger le secret sur les missions accomplies, le client fournit une indication globale du temps de vol. Après tout ce travail de mise en place et de préparation il devient possible de centraliser les informations souhaitées et de procéder aux analyses de dépouillements.

2.5 Analyses et dépouillements

Les premiers traitements des informations reçues ont eu pour but de :

- remplacer le sentiment des utilisateurs par des données objectives
- faire apparaître les anomalies pour lesquelles une rétro action était possible au niveau de l'utilisateur et de la maintenance
- de déceler les faiblesses dans le dispositif de collecte des informations.

Nous étudierons plus à fond ces traitements dans la quatrième partie de l'exposé ; nous allons avant donner un aperçu plus précis de l'organisation existante aujourd'hui.

3 - L'ORGANISATION EN 1979

Nous décrirons ici une organisation typique certaines variantes peuvent exister selon le client en effet, 10 pays sur 4 continents sont concernés.

Nous verrons successivement :

- les documents de base (forme 11 et fiche intervention technique)
- le chaînage qui lie ces documents entre eux
- les documents du "tableau de bord" à l'échelon central THOMSON-CSF/RCM.

3.1. Les documents de base

La forme 11 et les fiches d'interventions techniques sont les documents de base.

Pour des raisons de secret, les informations détaillées portées sur la forme 11 (voir planche 4) ne sont pas communiquées à THOMSON-CSF.

Une synthèse est faite par le personnel du client qui communique le nombre d'heures globale de vol relatives à la période pendant laquelle les fiches d'interventions techniques ont été émises.

Les fiches d'intervention techniques (voir planche 5) sont établies par le personnel chargé de la maintenance et transmises mensuellement a THOMSON-CSF. On notera que cette fiche d'intervention technique porte les informations relatives.

- à la constatation de l'anomalie (lieu, avion, niveau maintenance, etc....).

- à la nature de l'intervention

- à la cause de l'anomalie

Les fiches d'interventions techniques pouvant être multipliées pour une même anomalie, nous allons voir le chaînage entre chacun de ces documents.

3.2. Le chaînage des documents

La planche 6 illustre cette chaîne et montre le mode de transmission vers l'échelon central THOMSON-CSF.

On notera l'action de l'assistance technique qui s'assure avant transmission à l'usine THOMSON-CSF Malakoff que les informations portées pourront être exploitées (signature des fiches).

3.3. Le tableau de bord

Ce tableau de bord comporte deux éléments :

- la fiche A (voir planche 7) qui remet pour un radar donne les informations relatives aux diverses anomalies.

Ces fiches résument radar par radar, toutes les anomalies signalées et les interventions effectuées dans la période considérée.

Elles ne sont donc à remplir que pour les radars ayant donné lieu à observations (une fiche par radar).

1ère ligne : Elle indique le numémo de série de la pointe avant, la base, la période d'utilisation et le nombre de vols effectués pendant cette période.

Tableau

Date : Jour où a eu lieu l'évènement

Support : Position du radar au moment de l'évènement (n° de l'avion, ou Banc global (BG) ou Banc partiel (BP)).

Vol-Sol : La case appropriée est numérotée selon que l'évènement a eu lieu en vol (observations) ou au sol (observations ou interventions).

Observations : Résumé très succinct de toute anomalie signalée soit en vol soit au sol (sur avion au BG ou au BP).

Interventions : Résumé très succinct de toute intervention effectuée soit sur avion soit au BG, soit au BP suivie ou non de la rédaction d'une fiche d'intervention technique par l'utilisateur.

L'anomalie et l'intervention correspondante seront inscrites sur la même ligne.

Fiche d'intervention technique : Inscrire son numéro sur la même ligne que l'intervention correspondante.

Cas particulier : Intervention non accompagnée de fiche ou du document équivalent : dans ce cas, la case correspondante est barrée.

Observations sans réponse	Mentions dans colonne "Interventions"
Anomalie non confirmée au sol	NC
Anomalie en cours d'investigation	En cours
Anomalie en attente de contrôle	En attente
Anomalie suivie de remise en service sans contrôle	Néant

La fiche B (voir planche 8) qui permet de connaître la situation des divers radars pour lesquels aucune anomalie ne s'est produite.

Dans la période considérée cette fiche :

- récapitule le nombre de vols et le temps de fonctionnement (en vol et au BG) de l'ensemble radar.
- résume le nombre de vols, mouvements, état et situation des radars n'ayant fait l'objet d'aucune observation.

Elle concerne donc seulement les radars non mentionnés sur les fiches A mais tous les radars en dotation à la base doivent apparaître, soit sur les fiches A soit sur les fiches B.

Ce sont ces deux documents qui serviront de base aux traitements, bilans et synthèses diverses.

4 - LES TRAITEMENTS

Ce sont :

- les bilans globaux qui indiquent :
  - . les durées d'exploitations, les anomalies pour chaque période considérée,
  - . la répartition des anomalies par fonction du radar
- les bilans par module et par type de composants

Ces bilans sont édités périodiquement et peuvent être consultés auprès des équipes d'Assistance Technique.

4.1. Les bilans globaux

Ils sont de trois sortes :

- le bilan global proprement dit,
- la répartition des anomalies signalées en vol,
- la répartition des anomalies signalées en vol et au sol.

Nous voyons planches 9 et 10 la présentation de ces bilans.

4.1.1. Bilan global

Le tableau donne le nombre de :

- Avions et radars en service à l'Escadre,
- Vols effectués pendant la période considérée,
- Heures de fonctionnement du radar
- Anomalies signalées en vol ou au sol par les pilotes ou l'équipe de maintenance,
- Pointes radar déposées.

4.1.2. Répartition des anomalies signalées en vol

Dans un tableau à 2 entrées, les anomalies signalées par les pilotes sont réparties par :

- type d'anomalies (colonnes A et G)
- fonction ou sous-ensemble (lignes 1 à 14).

Ce tableau fait apparaître le taux d'imputation affecté à chaque type d'anomalie et à chaque fonction ou sous-ensemble.

4.1.2.1. Types d'anomalies

4.1.2.1.1. Imputées au radar (colonnes A et E)

A - Pannes : Anomalies résorbées par remplacement de composants défectueux, ou réparations diverses (câblage, mécanique, etc...)

B - En cours : Anomalies faisant l'objet d'investigations et dont la cause est incomplètement définie à la date d'établissement du bilan.

C - Réglages : Reprises de réglages consécutives à un déréglage ou à un réglage mal réalisé.

D - Modifications : Anomalies consécutives à la non-exécution de modifications officiellement adoptées.

E - Non confirmées : Anomalies fugitives non décelées au cours de contrôles au sol.

4.1.2.1.2. Non imputées au radar (colonnes F et G)

F - Répétées : Anomalies donnant lieu à répétition de remarques identiques sans intervention de l'équipe de maintenance.

G - Divers : - défectuosités d'équipements extérieurs au radar se traduisant par une anomalie de fonctionnement du radar.

- erreurs de diagnostic : interventions sans effets sur le défaut signalé.
- maintenance non conforme aux directives du manuel d'utilisation
- fausses manoeuvres entraînant un défaut de fonctionnement avec ou sans détérioration d'éléments.

4.1.2.2. Fonctions ou sous-ensembles

1 - PROGRAMME : Manche, boites programme G et H

2 - INDICATEUR : Boite de commande, coffret circuits et coffret tube

3 - ALIMENTATIONS : Blocs alimentations à l'exclusion des platines servomécanismes modèle et antenne.

4 - EMISSIONS : Bloc émetteur y compris bague articulée

5 - SERVO-MODELE : Platine servo-modèle et modèle analogique

6 - SERVO ANTENNE : Platine servo antenne et mécanismes antenne

7 - RECEPTEUR HYPER : Partie du récepteur comprise entre la source antenne et l'entrée des préamplificateurs FI

8 - AMPLI PARA : Corps et alimentation de l'amplificateur paramètrique et sécurités intégrées dans l'ensemble ampli para

9 - RECEPTEUR FI : Châssis CAF, pré-amplificateur, opérateur, amplificateurs FI, détecteur Liaison FI.

10 - POURSUITE VIDEO : Circuits synchronisation, vidéo, télémétrie et écartométrie

11 - CALCULATEUR : Circuits calculateur d'intégration et domaine de tir et recopie distance

12 - SERVITUDES : Structure pointe avant (mécanique et câblage), circuit de climatisation, circuits de pressurisation, liaisons guide, arête commutation, boîtier démarrage et circuits de commutation répartis dans la structure pointe avant.

13 - ANNEXES : Coques, accéléromètre, filtre réseau et équipements extérieurs du radar.

14 - NON IDENTIFIEES : Anomalies, généralement non confirmées, dont la cause peut être imputée à plusieurs sous-ensembles différents.

4.1.3. Répartition de l'ensemble des anomalies signalées (Planche 10)

Un tableau identique à celui du § 2, donne la répartition de l'ensemble des anomalies signalées en vol et au sol, sur avion ou aux bancs de maintenance global ou partiels.

4.2. Les bilans par modules et par types de composants

Dans ces bilans, les fonctions et sous-ensembles qui apparaissent dans les bilans globaux sont éclatés d'une part, en module de base, la planche 11 reproduit un tel éclatement pour les sous-ensembles alimentation et émetteur, et d'autre part en groupe type de composants la planche 12 en donne une illustration.

4.3. Exploitation des bilans

Comme cela apparaît sur les exemples que nous venons de voir, il est possible à partir de ces documents :

- d'évaluer les moyennes des temps de bon fonctionnement dans l'ambiance composite (vol, sol, maintenance) qui représente les conditions opérationnelles réelles. La planche 13 donne une idée de l'évolution des MTBF

- d'identifier les fonctions sous-ensembles, modules et composants responsables des anomalies et d'adapter les stocks pour améliorer la disponibilité.

- d'entreprendre les actions correctrices qui pourraient être nécessaires au niveau de la réalisation des radars et éventuellement de l'Après Vente.

- de mettre en évidence les écarts de MTBF entre différentes bases d'utilisation. La planche 14 montre des écarts qui peuvent être expliqués par :

  . des conditions d'emploi différentes (proportion des tirs canon par rapport au durée de vol, haute ou basse altitude etc...).

. les écarts de maintenance (formation du personnel, temps disponible entre deux vols pour l'entretien etc...)

. les écarts climatiques entre bases

- de mettre à jour les tables de données de fiabilité pour les ambiances "vol porté en cabine et hors cabine". La planche 15 donne un exemple.

- d'améliorer la conception des radars en étude.

5 - LES PERSPECTIVES D'AVENIR

Elles sont de plusieurs natures, il semble aujourd'hui souhaitable de :

- mettre en place une organisation semblable pour d'autres équipements

- de compléter l'organisation existante en transmettant les informations aux fournisseurs de composants

- d'étudier avec les utilisateurs à travers les bilans par base l'influence des conditions d'emploi sur la fiabilité des équipements.

En effet des spécifications décrivant mieux l'utilisation de l'équipement permettront pour des matériels en étude de mieux maîtriser la fiabilité.

5.1. Organisation semblable pour d'autre équipement

Nous pensons que ce phénomène va se développer d'une part en raison de l'importance des problèmes de fiabilité liés aux équipements complexes en cours d'étude et d'autre part en raison de la demande permanente visant à connaître les taux de défaillance des composants de technologie nouvelle.

5.2. Extension de l'organisation vers les fournisseurs de composants

Il est aujourd'hui très difficile pour un fabricant de composants (tubes hyperfréquence, semi-conducteurs, etc...) de donner une prévision précise du taux de défaillance de ses nouveaux produits dans les conditons réelles d'utilisation ceci en raison d'une part de l'effet d'écran que crée le constructeur d'équipement (l'environnement est en général décrit par ce dernier en moyen des enveloppes de température, de vibrations... qui cumulent des simplifications et des coefficients de sécurité) et d'autre part par le petit nombre d'actions élémentaires nécessaires à la réalisation du composant, ce petit nombre ne permettant pas d'effet de moyenne (c'est-à-dire ce qui est surestimé est mal compensé par ce qui est sousestimé).

Il est donc nécessaire d'aider ces fabricants à se former une expérience en leur communiquant pour les produits utilisés aujourd'hui les informations relatives aux anomalies liées à leurs produits et en leur donnant l'occasion de faire en grand nombre d'analyses de défaillance.

Pour ces raisons, nous pensons qu'il faut intéresser les fabricants de composants à ces problèmes et leur fournir l'occasion, à travers une organisation à mettre en place avec nos clients, de connaître la fiabilité de leurs produits dans les conditions réelles d'utilisation.

5.3. Analyse des bilans avec les utilisateurs base par base

Comme nous l'avons vu précédemment des écarts de fiabilité existent d'une part entre les différents lieux d'utilisation et d'autre part en un endroit donné au cours du temps l'analyse détaillée par l'utilisateur seul ou aidé par le constructeur d'équipement permettra probablement de définir :

- les types de mission les plus contraignantes

- les points faibles de l'équipement face à ces missions

- les points faibles de la maintenance

- etc...

Ce type de travail devra être précédé d'une action de motivation pour éviter par exemple une réaction négative du personnel de maintenance qui peut se sentir critiqué et mis en compétition avec celui d'une autre base.

6 - CONCLUSION

Comme nous avons pu le voir au cours de cet exposé la mise en place d'un système de collecte des anomalies et d'informations des parties concernées (utilisateur, équipementiers, fabricants de composants), permet de se donner les moyens de connaître la fiabilité de façon objective et de définir les actions et règles à observer pour améliorer cette performance.

C'est un complément indispensable aux actions entreprises lors de la conception et la réalisation des équipements pour maîtriser qualité et fiabilité de ceux ci.

Remerciements : Je remercie les divers services de la THOMSON-CSF/RCM qui ont participé à l'élaboration de ce document et en particulier MM. SOUBRA, DUHAIN, PHILLIPON et GIGOUX.

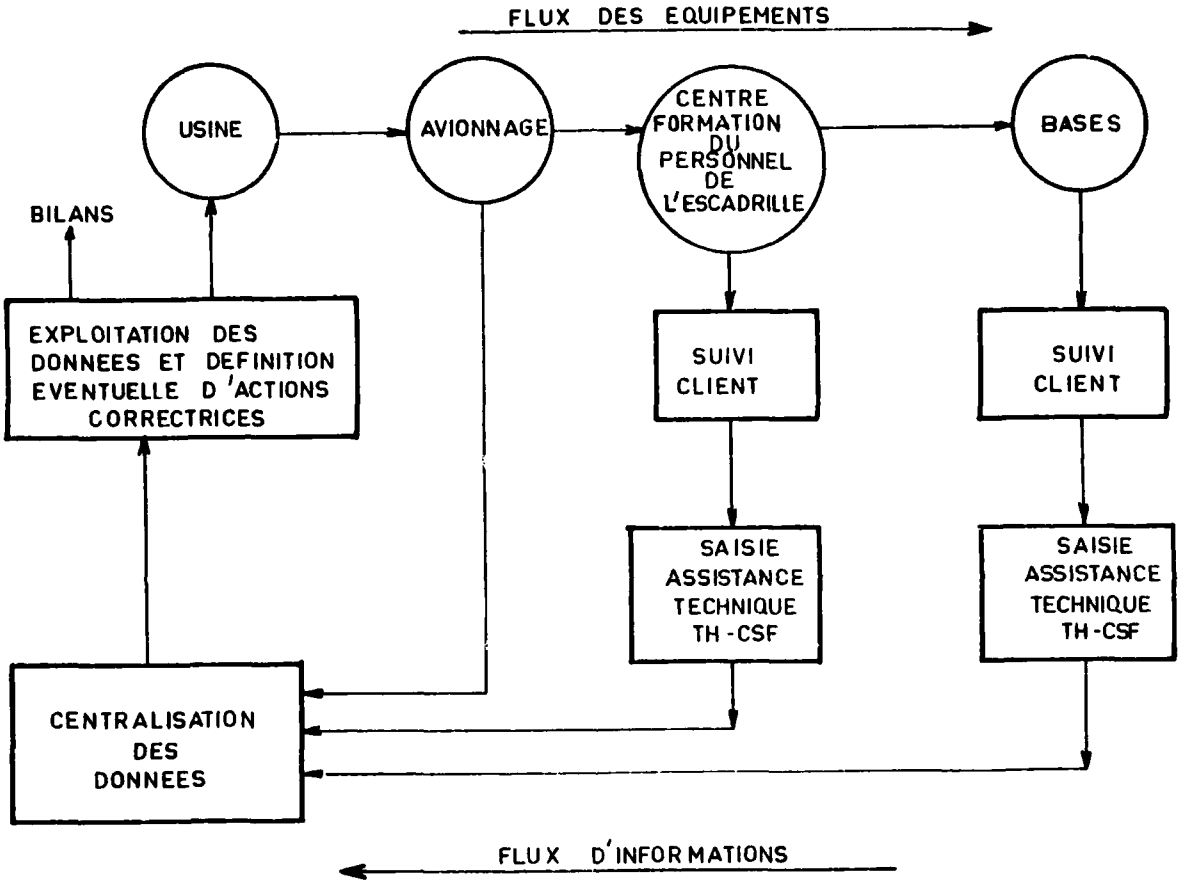

FLUX DES EQUIPEMENTS
USINE
AVIONNAGE
CENTRE FORMATION DU PERSONNEL DE L'ESCADRILLE
BASES
BILANS
EXPLOITATION DES DONNEES ET DEFINITION EVENTUELLE D'ACTIONS CORRECTRICES
SUIVI CLIENT
SUIVI CLIENT
SAISIE ASSISTANCE TECHNIQUE TH-CSF
SAISIE ASSISTANCE TECHNIQUE TH-CSF
CENTRALISATION DES DONNEES
FLUX D'INFORMATIONS

Planche 3

FICHE D'INTERVENTION TECHNIQUE : 1 ouverte par : Piste 2 BG 3 BP 4 An Numéro 5 Indice

ORIGINE - Base 6 Escad 7 Date 8

Avion 9 N° 10 Hr Fonct 11

SUPPORT N° 13 Hr Fonct 14

Equipement 12 N° Remonté 15 Hr Fonct 16

A OBSERVATIONS PILOTE :

B CONSTATATIONS TECHNICIEN DE PISTE :

C CONSTATATIONS TECHNICIEN D'ATELIER :

D CONSTATATIONS CONSTRUCTEUR :

RENSEIGNEMENTS RELATIFS A LA REPARATION : Date remise en état 39

Défaut non constaté 40 Non imputable au racer 41 Réglage 42 Modification 43

DIRMAT-CSF
66, Av. Pierre Brossolette
92240 MALAKOFF

EXEMPLAIRE N°

DESTINATAIRE :

CIRCONSTANCE AVARIE
Indisponibilité avion 17 En Vol 18 Au Sol 19 Au BG 20 Au BP 21

SOUS-ENSEMBLE DEFAILLANT
Désignation 22
23 Référence Fabricant 24 N° Série 25 Hr Fonct
26 N° Remonté 27 Hr Fonct

ELEMENT REMPLACE
Désignation 28
29 Référence Fabricant 30 N° Série 31 Hr Fonct
32 N° Remonté 33 Hr Fonct

PROPOSITIONS
Remise en état atelier 35 Date 34 Envoi au réparateur 36 Rebut 37

COMPOSANTS DEFAILLANTS 38

Repère	Désignation et Référence	N° Série	Hr Fonct

SERVICES				
VISAS				

Planche 5

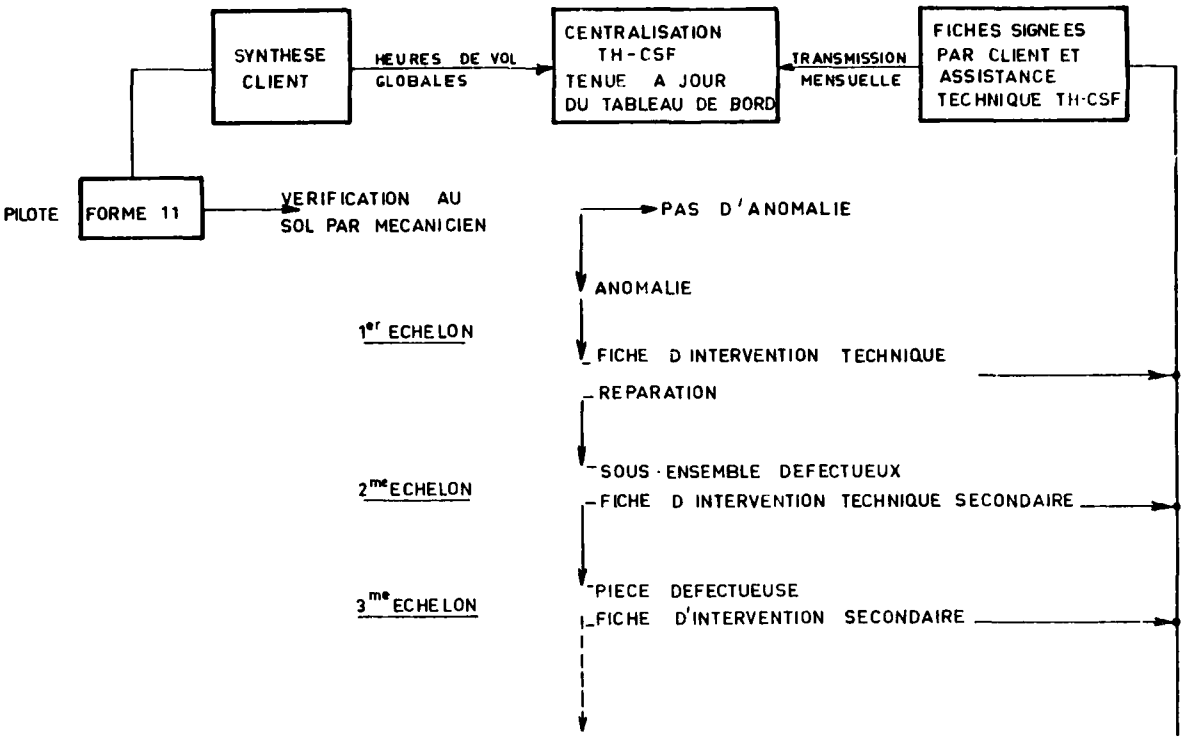


Planche 6

FICHE A

RADAR N°: BASE: PERIODE: Nbre DE VOLS:

DATE	SUPPORT	VOL	SOL	OBSERVATIONS	INTERVENTIONS	FICHE D'INTERVENTION TECHNIQUE N°

Planche 7

FICHE B

BASE : PERIODE :

NOMBRE DE VOLS : (TOTAL)

TEMPS DE FONCTIONNEMENT - EN VOL : H.
- AU BG : H.

SITUATION DES RADARS N'AYANT DONNE LIEU A AUCUNE OBSERVATION

RADAR N°	DATES	SUPPORT	Nombre de vols ou état	OBSERVATIONS

BASE : PERIODE :

1 - BILAN GLOBAL

DOTATION		VOLS	Heures de fonctionnement		ANOMALIES SIGNALEES -		DEPOSES POINTES AVANT	
AVIONS	RADARS		En vol	TOTAL	En vol	TOTAL	Causes Radar	TOTAL

2 - REPARTITION DES ANOMALIES SIGNALEES EN VOL

ANOMALIES			IMPUTEES AU RADAR						Causes extérieures			TOTAL
			Pannes	en cours	Règl.	Modifs	N.C.	TOTAL	Repet.	Divers	TOTAL	
			A	B	C	D	E		F	G		
		%										
PROGRAMME	1											
INDICATEUR	2											
ALIMENTATIONS	3											
EMISSION	4											
SERVO-MODELE	5											
SERVO-ANTENNE	6											
RECEPTEUR HYPER	7											
AMPLI PARA	8											
RECEPTEUR FI	9											
POURSUITE VIDEO	10											
CALCULATEUR	11											
SERVITUDES	12											
ANNEXES	13											
NON IDENTIFIEES	14											
T O T A L												

Planche 9

3 - REPARTITION DE L'ENSEMBLE DES ANOMALIES (VOL + SOL)

ANOMALIES			IMPUTEES AU RADAR						Causes extérieures			TOTAL
			Pannes	en cours	Regl.	Modifs	N.C.	TOTAL	Repet.	Divers	TOTAL	
			A	B	C	D	E		F	G		
		%										
PROGRAMME	1											
INDICATEUR	2											
ALIMENTATIONS	3											
EMISSION	4											
SERVO-MODELE	5											
SERVO-ANTENNE	6											
RECEPTEUR HYPER	7											
AMPLI PARA	8											
RECEPTEUR FI	9											
POURSUITE VIDEO	10											
CALCULATEUR	11											
SERVITUDES	12											
ANNEXES	13											
NON IDENTIFIEES	14											
T O T A L												

Planche 10

DESIGNATION	REPERE	A Pannes	B En cours	C Réglages	D Modifs	E N.C.	TOTAUX
3. ALIMENTATIONS	C251						
Structure							
Redressement	C253						
Régulation 12 volts	C255						
Régulation 30 volts	C257						
Régulation 60 volts	C259						
Régulation 270 volts	C261						
Bloc sécurités	C263						
Alimentation sécurités	C265						
Relais démarrage	C267						
Sécurité de phases	C273						
Oscillateur 500 Hz	C269						
Amplificateur 500 Hz	C271						
Non déterminé							

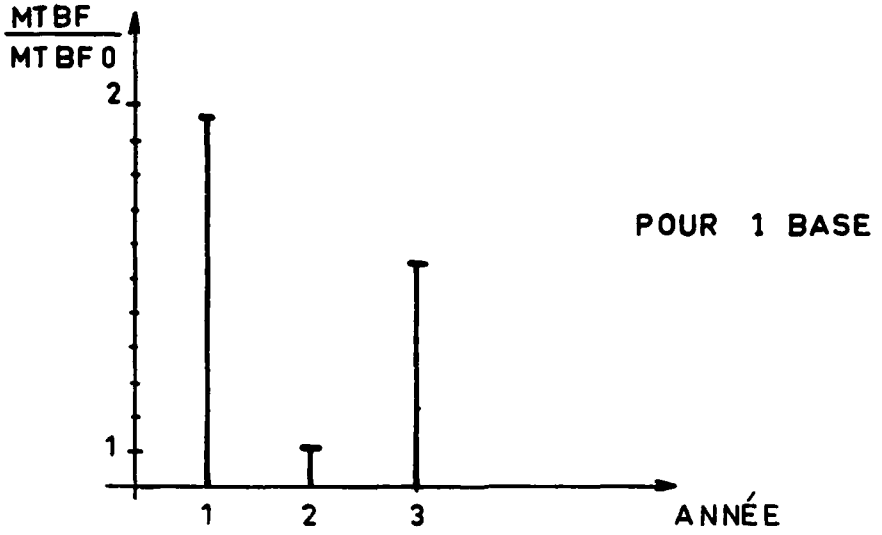
4. EMETTEUR	C203						
Structure et fagot Z10							
Bande modulateur	C207						
Commutateur phases	C208 Z1						
Alimentation THT	Z3 Z4 Z5						
Bleeder THT	Z6						
Self de charge	L2						
Bloc à diodes	Z7						
Résistances backswing	Z8						
Ligne à retard	DL1						
Relais Jennings	K2.K3						
Bleeders L à R	Z2.Z11						
Filtre thyratron	Z9						
Thyratron	V1						
Transfo d'impulsions	T2						
Magnétron	V2						
Bague articulée							
Non déterminé							

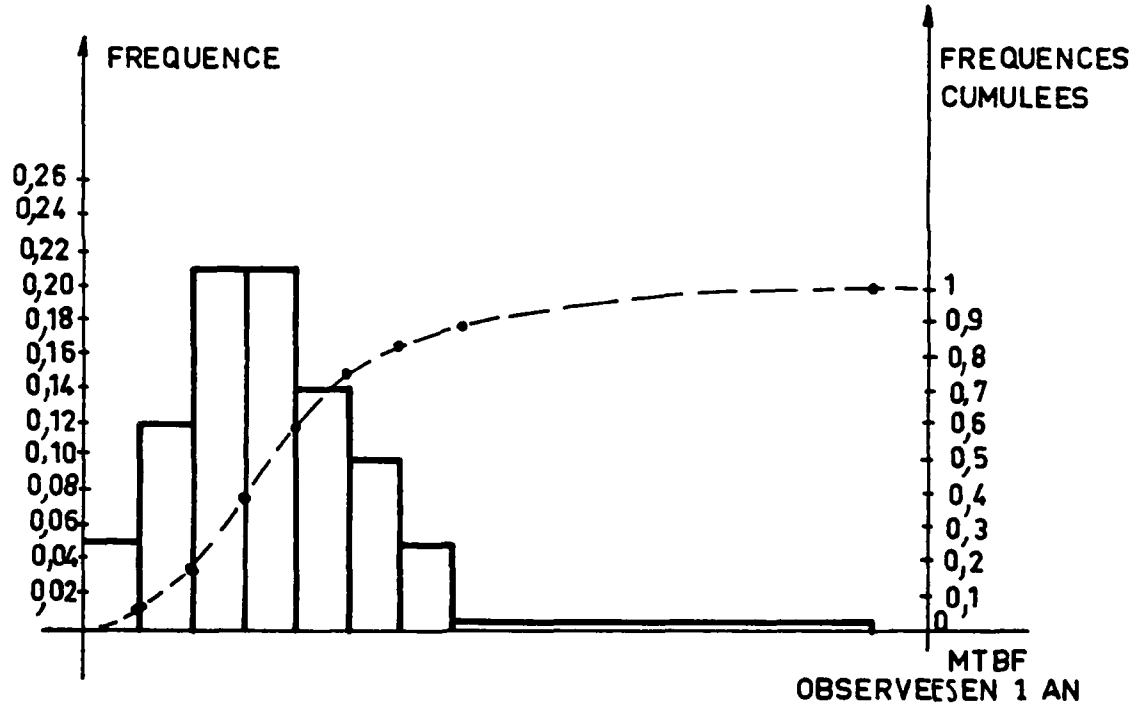
Planche 11

5 – RÉPARTITION DES PANNES CONFIRMÉES (A)

Éléments / Sous-ensembles					Machines					Composants passifs						Semi-conducteurs						Hybrides		Divers				Totaux	
					Moteurs	Résolvers	Gyromètres			Résistances	Condensateurs	Selfs .	Potentiomètre	Transformateurs	Divers	Diodes	Thyristors	Transistors	Micrologiques			Blocs moulés	Couches minces	Lampes et fusibles					
1	Programmes	Manche																											
		Boîtes																											
2	Indicateur	Boîtier C^ts et Coffret tube																											
		C^t circuits																											
3	Alimentations	Structure																											
		Fagots																											
4	Émission																												
6	Servo antenne	Platine	Châssis																										
			Fagots																										
		Antenne																											
7	Récepteur hyper	Oscillateur local																											
		Divers																											
8	Ampli para	Alim. sécurités																											
		Corps ampli par																											
9	Récepteur FI																												
10	Poursuite Vidéo	Châssis																											
		Fagots																											
11	Calculateurs	Arêtes	Châssis																										
			Fagots																										
		Recopie distance																											
12	Servitudes	Structure																											
		Climatisation																											
		Pressurisation																											
		Boîtier démar																											
		Arête 307																											
Totaux																													

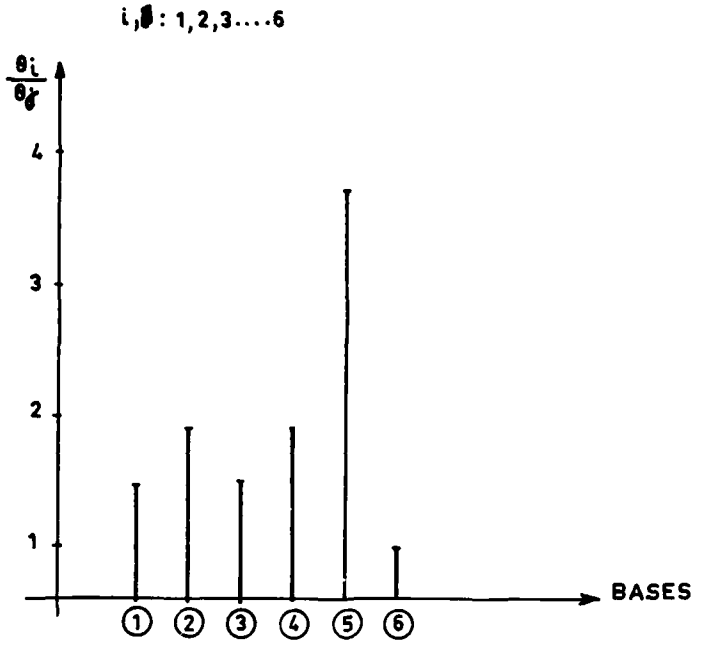
Planche 12





EVOLUTION DES MTBF (HISTOGRAMME ET REPARTITION)
(TOUTES BASES CONFONDUES)

Planche 13



DISPERTION ENTRE DIFFERENTES BASES

Planche 14

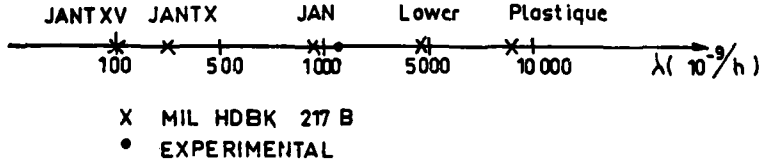
TAUX DE DEFAILLANCE EXPERIMENTAUX

ENVIRONNEMENT : AEROPORTE INHABITE

TEMPERATURE = 60° C

NIVEAU DE QUALITE "MILITAIRE"

_ DIODES SIGNAL ET COMMUTATION SILICIUM



_ TRANSISTORS FAIBLE PUISSANCE SILICIUM

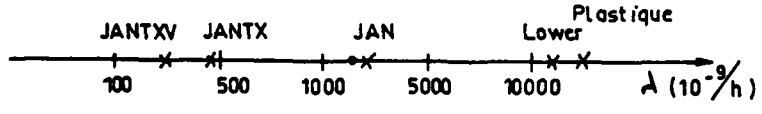


Planche 15

A Fault Tolerant Architecture Approach to Avionics Reliability Improvement

Donald C. Fraser and John J. Deyst

C.S. Draper Laboratory

Cambridge, Mass. 02139

SUMMARY

A difficult technology challenge in the reliability of avionics systems for advanced aircraft is identified. Three architectures are compared on the basis of a number of criteria which together constitute the issues which must be examined when considering the overall reliability, maintenance, and support problem. It is concluded from these comparisons and the limitations identified in contemporary approaches that the only effective and practical solution to the reliability challenge is through architecture. An advanced integrated, distributed fault and damage tolerant digital avionics system architecture is summarized which shows promise of meeting this challenge.

INTRODUCTION

The practical development of the microprocessor has led to the explosion of diverse digital computer applications with which we are all familiar. Innovative uses of this new device can already be found in aircraft avionics - particularly in the highly competitive business aviation field. The revolution for airline and military avionics has begun. Future generation aircraft are, because of the power of this new technology, very likely to nearly all be digital, fly-by-wire designs.

The digital avionics system offers many attractive advantages over currently available designs. Key among these are greater design flexibility and commonality of parts, enhanced maintenence through simpler common modules and system self check, and lower cost of ownership - not only for the aforementioned reasons but because the avionics application can capitalize upon the developments in the far larger commercial market. All this is now possible, but the use of digital technology in place of current analog electronic components will not substantially improve avionics reliability. It is the hypothesis of this paper that substantial improvements in this latter characteristic are possible by capitalizing upon recent developments (DEYST, J.J., et al,..., 1978) in another new and developing field - that of digital system architecture. Indeed, based upon technology already established, it should be technically possible to configure an avionics system that as far as the pilot is concerned will never loose a key function. This system is likely to be a fully integrated, digital, dispersed configuration which simultaneously will provide considerable lower life cycle costs than any design which provides similar reliability employing the currently popular approach of subsystem isolation.

A fundamental premise of an advanced integrated architecture for future aircraft is that the stability augmentation and flight control systems be integrated within the avionics system. With today's level of avionics reliability this notion is totally unacceptable to most pilots. It becomes practical, however, if the overall avionics system can indeed be configured to never lose a key function such as aircraft stability augmentation. The potential from the availability of such a system is in fact more exciting than the basic functional reliability itself. The existence of such levels of failure tolerance permit leverage in the aircraft design that never before could be considered. Aircraft would no longer need to be designed to inherently possess good basic stability and handling characteristics. They could be configured to be totally dependent upon the electronic flight control system. These control configured aircraft can potentially offer considerably more performance than their conventionally designed counterparts. As an example, the aircraft shown in Figure 1, which was the result of a Boeing study (WALKER, S.A..., 1974), would have the properties shown in Table 1. This tanker aircraft employed flutter mode control and negative static stability to decrease the structural weight. The result is an aircraft that can perform the same mission as the conventially designed aircraft it is compared to in Table 1 for almost 20% less total weight. Similar and even greater leverage is available depending upon the aircraft mission.

Avionics reliability, when viewed from the overall system point of view which an integrated system necessitates, is a complex entity to evaluate. It consists of many facets, spanning the spectrum from basic design issues to repair and logistics considerations. This paper first sets the reliability requirements which a control configured aircraft application will demand and then addresses the reliability issue on a point by point basis by comparing an advanced integrated avionics architecture to more contemporary approaches.

FUNDAMENTAL GOALS AND LIMITATIONS

The flight control function in an advanced control configured aircraft such as the one used in the example above will be as critical to survival as major aircraft structural elements such as the wings. This function must then be at least as reliable as the primary structure itself. Figure 2 is an attempt to place this challenge in the context of two more familiar generations of avionics systems.

There are three elements considered in each of the lines of Fig. 2: exposure time, vulnerability, and component failure rate. This last item is the area of focal interest to this paper: it represents the probability of functional failure (per hour) of the avionics function considered. Exposure time is that fraction of the total operating system time during which the aircraft is exposed to a failure of the avionics function indicated. Vulnerability, in the context of Fig. 2, is the probability of a catastrophic event being caused by a failure of the avionics function.

The first line of Fig. 2 considers a pilot relief cruising flight autopilot - an avionics item that has been in service for many years. Because it is likely to be used almost continuously on any long flight the exposure time fraction here is set to unity. On the other hand it can be quickly disconnected by the pilot who can manually fly the aircraft; thus the probability that a failure will lead to a catastrophe is remote - it is assigned here to be one in a million. Typical functional reliabilities of this type of installation are of the order of 1000 hours. The resulting probability of catastrophe is then the product of these three quantities or a probability of 10^{-9} per hour. This is a commonly accepted level for civil applications for the probability of catastrophe due to functional failure of a major aircraft element (UK CAA..., 1976).

The second line of Fig. 2 represents a system which is at least a generation newer - the category 3A Autoland system found in several of the wide body transports (FLAPPER, J.A...., 1977). The same 10^{-9} failure per hour level of safety is required during automatic landing. Due to the greater vulnerability, this places a 10^{-6} failure per hour requirement on the autoland function. This three order of magnitude increased burden on the equipment is the driving reason behind the use of redundant configurations in automatic landing systems as opposed to the simplex installations commonly found in cruise autopilots.

The final line of Fig. 2 is for a control configured, fly by wire aircraft. In this case the system is always in use and its loss will ensure a catastrophe. Because of these two facts the entire burden of the 10^{-9} target is placed upon the flight control system function. This must be the reliability goal of future generation flight control systems - a frightening three orders of magnitude beyond the most advanced in service today.

This reliability goal is even more awesome when considered in the context of the architectural limitation identified by Osder (OSDER, S.S...., 1977). His point, repeated here in simplified form as Fig. 3, is that to build and provide for the management of a redundant avionics configuration leads to a rapidly increasing level of complexity in the electronics required as more redundancy is desired. The dual-dual data point in Fig. 3 is typical of the autoland installations represented by the second line of Fig. 2. The redundancy required to obtain the additional three orders of magnitude required in a control configured aircraft would be well beyond the dual-dual point. The conclusion which one must reach is that even conventional advanced architectural approaches will not lead to a practical solution to the reliability requirement imposed by a control configured aircraft design.

ARCHITECTURAL ALTERNATIVES

The previous sections have identified some of the opportunities, goals and limitations associated with an advanced avionics architecture. The purpose of this section is to define three classes of architecture to serve as the basis of the comparisons which will be made in the next section.

The first of the three avionics architectural classes is commonplace today - it is the use of separate and often dissimilar primary and secondary resources. An example of this is the cruise autopilot discussed in the previous section. In that case the automatic system is backed up manually by the pilot. The Saab-Viggen digital flight control is a contemporary example of this class. A single primary system with monitoring is used in this case with a simpler backup reversion. In all systems of this class either the pilot or a highly dependable system monitor must control the switchover to the backup system. For convenience in the next sections this class architecture will be referred to as "dissimilar"

The second architectural class may be found in some of the most advanced flight vehicles which have been tested to date - the Space Shuttle, NASA's F8 Digital Flv by Wire aircraft (SZALAI, K.J et al,..., 1976) the USAF F-16 and the F-18. In all of these cases the primary avionics resource is replicated in various ways and failures are identified and isolated via conventional voting techniques. Figure 4 illustrates one example of this type architecture - the F8 Digital Fly by Wire installation. In the following sections this type of architecture will be referred to as the "replicated" approach.

The third class and the one which is presented in this paper as a possible solution to the challenges described in the previous sections, is organized as an ensemble of dynamically allocated, pooled resources (DEYST, J.C., et al,..., 1978)(HOPKINS, A.L.,et al,..., 1978). Unlike the other two classes it has not yet been flight tested. The basic philosophy behind this approach is that no distinction is drawn between elements in terms of dedication to specific functions. Instead, any element can serve any function for which it can provide useful capability.

The pooled elements are the line replaceable units within the avionics system. All internal fault isolation is carried to the level of these units and maintenance and logistics procedures are designed around these as the basic system elements. In addition, the pooled units are basic building blocks which are interconnected, in hierarchical fashion, to perform the necessary avionics system functions.

The interconnections between units are not static but can be altered by the system, in real time, to respond to changing requirements and loss of capabilities due to failures and damage. Figure 5 illustrates a possible configuration for an advanced integrated avionics system using this type of architecture. The lighter lines indicate the interconnections between resources which would be available in network fashion through either electrical or fibre optic links. The darker line illustrates a possible instantaneous connection of the resources.

Embodied within the architecture of this class is a comprehensive redundancy management function which identifies faults to the level of the line replaceable units and reconfigures the system to isolate failed units. Each resource pool of units is represented at a level of redundancy reflecting the possible functions that can be served by the elements of that pool. In many instances elements can serve multiple

functions. The prime example here is a small computer or microprocessor which can serve any function for which it is programmed. By loading appropriate code in real-time and providing the necessary input/output interfaces, the computer can serve any function for which its capabilities are sufficient. Thus, for example, a processor is not dedicated to flight control but can serve navigation, display or communications functions as well.

More details on this architectural class may be found in (DEYST, J.J., et al..., 1978) (SMITH, T.B., ..., 1978). In the following sections this architectural type will be referred to as the "pooled" approach.

AVIONICS ARCHITECTURAL COMPARISONS

In this section the "pooled" architectural class is compared to the other two on the basis of a number of criteria which are key to the life cycle cost and reliability associated with deploying each. The purpose is to demonstrate that by most criteria the pooled architectural alternative offers distinct advantages over the more contemporary approaches. It will be seen that benefits may be obtained from this advanced class of architecture whether or not the control configured aircraft design leverage is used.

The comparisons which follow are all based upon the observation that all three architectural classes can be configured to meet the normal performance requirements associated with an advanced avionics system such as throughput, memory capacity, bandwidth etc. All are equally capable in this regard. No order of importance is inferred in the order of presentation of the criteria. The sum total of these advantages is what should enable the "pooled" architecture to provide significantly greater reliability than the more contemporary approaches.

Automation Potential:

This criteria is the ability of an architecture to provide sufficient reliability for a function to permit that function to become completely automated in a life critical function. The "dissimilar" class is totally inadequate here since it simply does not provide sufficient coverage. Both the "replicated" and "pooled" architectures can be configured with sufficient redundancy to satisfy this criterion. Due to its ability to freely assign elements the "pooled" class can provide the necessary coverage more economically than the "replicated" class. Figure 6 illustrates this point for an integrated flight control and navigation system on a 1000 aircraft lot. In generating this figure, which compares the replicated and pooled classes flight control is configured to be fail operational - fail operational - fail safe while navigation is fail safe.

Utilization Flexibility:

There are two types of flexibility that can be addressed. The first is real time flexibility to adapt to a changing situation. The concept of pools of units, capable of being assigned to perform tasks as appropriate, provides in-depth flexibility. Since no unit is specifically identified with a particular task or function, all units are available to support any required function, and the maximum flexibility is afforded within the constraints of numbers of available units. The size of the units is an important aspect here because typically the unit size is smaller in the "pooled" system than in the other architectures.

The second type of flexibility is the ease with which the system can be modified for growth and change. With the pooled approach, modification means changing the constituents of pools, sizes of pools, or altering by adding or subtracting pools. To the extend that communications system flexibility permits, this can be a very straightforward process. A rigidly string organized system (i.e. triplex or quad redundant) can be amenable to changing particular types of elements, but is not amenable to adding additional redundancy of particular elements. A non-string organized "replicated" system can have change flexibility equivalent to a "pooled" system. It is important to realize, however, that this configuration begins to approach the "pooled" architecture approach.

Availability:

Because of its ability to flexibly configure its resources, the "pooled" system has the potential to provide the highest level of availability or operational readiness. Since no element is specifically assigned to a function, all elements can serve any function to which their capabilities are useful. Thus, for a specified number of faults within the system, the "pooled" architecture, with elements serving multiple roles and flexibility to reconfigure around faults, has the greatest chance to successfully complete a mission. This is a key reason for the savings indicated in Fig. 6 and is the driving mechanism behind the earlier claim that a system could be configured so as to never loose a key function. Whereas individual components may fail at the "thousand hour" rate discussed earlier, there will be enough of them incorporated into a sufficiently flexible "pooled" architecture to guarantee the existence of key functions to even the 10^{-9} level required by a control configured aircraft.

Modularity:

In terms of procurement, maintenance, and logistics it is desirable that the avionics system be configured from a small set of unique modules. The very nature of the "pooled" architecture lends itself readily to minimizing the number of different modules. Pools can serve multiple tasks and, by appropriate choice of elements in a pool, the array of functions served by a pool can be maximized and the number of pools minimized. The "pooled" class is, in effect, inherently modular whereas the other two classes are not.

Complexity, Weight and Volume:

The total set of functions served by the avionics system overlap in the sense that more than one function can be served by a single unit within the system. By purposely choosing units to serve multiple roles, both performance and reliability can be achieved without excessive complexity and its associated weight and volume penalties. As faults occur, the system can be reconfigured, gracefully dispensing with less critical mission functions so that remaining resources can support flight critical tasks. The "pooled" architecture, with its dynamic reconfiguration capability and pooled resources, provides the flexibility to make maximum use of all elements, and hence has the greatest potential to minimize complexity, weight and volume. This is also evident from the comparison provided in Fig. 6.

Maintainability:

The fault isolation procedures of the "pooled" architecture are specifically designed to identify failures in real time to the level of line replaceable units. The fault isolation process occurs in the operational environment and hence a high level of validity of fault isolation is achieved, as compared to after-the-fact test procedures. Line replaceable units are sized to facilitate ease of maintenance and the entire maintenance and logistics scenario is based on the pooled units as basic system elements. This, plus the fact that the number of different LRU's is minimized, greatly simplifies the overall maintenance problem. The "dissimilar" and "replicated" architectures could also be designed for greater ease of maintenance than has been customary but not to the degree possible with the "pooled" architecture.

Diagnosability:

This aspect or criterion for an avionics system reflects the ease with which faults can be identified for maintenance purposes. It is more a design aspect than it is an inherent property of particular architectures. Whereas in the past fault isolation and built in tests were often an after thought, appended to the design; in future systems they must be an integral part of the design process from the outset. All three architectures lend themselves to fault diagnosis; however the "pooled" design specifically incorporates this aspect as a significant part of the entire design process.

Programmability:

Architectures can have a significant impact on the orderly development of software, its verification and validation. The most important aspect of a software development effort is the systematic partitioning or modularization of the job. A modular architecture imposes a natural partitioning on the software that greatly enhances this process. Since the "pooled" approach is by its very nature the most modular of the three architectures, it has the greatest potential for advantages in this area.

Producibility:

In terms of the problem of long-term procurement of an avionics system for a large fleet of aircraft, producibility is in large measure the ability to establish a highly competitive procurement of the various elements of the system. The "pooled" architecture, consisting of a relatively few pools, containing large numbers of identical elements, can provide the basis for establishing this competition. Large numbers of units will characterize each buy, attracting many potential supplies. Parts can be easily added or replaced in the "pooled" architecture since unlike the others it is configured to be highly modular. Fewer different types of units will be required, resulting in a reduced inventory of spares. The procuring agency can thus take maximum advantage of the innovative cost reduction methods that are inevitably stimulated by competion. The procuring agency owns the architecture and will be able to supply the parts for it on a piecewise competetive basis.

Design Risk:

Of the three architectures described, the "pooled" approach is the newest and hence has the smallest base of experience. Both the "dissimilar" and "replicated" approaches have been used in prototype and operational aircraft. While extensive analyses, simulations and prototype experiments have demonstrated the potential and feasibility of the "pooled" approach, it has not been brought to the flight test stage of development. This is its most important limitation.

Multiple Fault Tolerance:

Both the "replicated" and the "pooled" architectures can be provided with sufficient levels of redundancy to tolerate multiple faults. Similarly, both architectures can embody redundancy management procedures to identify failures and reconfigure the system to isolate faulty elements. The pooled architecture, however, with no dedications of units to specific functions and flexibility to allocate resources on a priority basis; can provide a higher level of fault tolerance at a given level of complexity or reduced complexity for a given level of fault tolerance. Indeed, in many cases it will be possible to isolate faults to the simplex level through the use of analytic redundancy algorithms (DECKERT, J.C., et al..., 1978)

Damage Tolerance:

The most effective means of attaining damage tolerance is through physical separation of avionics system elements. This requirement, if applied literally to a "replicated" architecture, would result in a considerable overhead penalty due to the additional elements which would be required to obtain the required dispersion. Since the "pooled" architecture generally contains smaller units, as compared to the other two architectures, greater freedom in location of elements within the aircraft is afforded by this approach.

Malfunction Correlation:

Malfunction correlation poses the greatest threat to fault-tolerant architectures. The very basis of fault tolerance is the assumption that the system can be designed so that failures are independent events. A correlated failure that affects all redundant copies of a particular type of unit immediately thwarts the purpose of the redundancy.

The "dissimilar" architecture is, by its very nature, highly immune to correlated malfunctions. The repetitive and pooled architectures must be designed with special care to eliminate all correlated malfunction mechanisms. Considerable effort has been dedicated to design and manage the "pooled" approach in a way which protects it from this type failure (SMITH, T.B..., May 1975), (SMITH, T.B...., April 1975).

Fault Latency:

Fault latency manifests itself in a fashion that is similar to a correlated malfunction. A latent fault occurs, for example, when a unit which is designated as a spare fails and is not detected because it has not been exercised for a period of time. The latent fault poses a special threat because when the spare is brought on line to replace a detected fault, the spare proves inoperative.

The "dissimilar" architecture typically operates with its backup system inoperative for long periods of time. Latent faults can accumulate and defeat the redundant strategy when a failure occurs in the prime system. The "replicated" system tends to use its redundant units in parallel fashion, all performing identical tasks. If the particular task does not exercise a certain facet of these units, so that a failure is not observed, latent failures may accumulate. The "pooled" architecture routinely reconfigures itself to uncover latent faults and hence it has the best chance of purging them.

Intermittent Fault Identification:

Intermittent faults are the most difficult to diagnose. Rapid detection and diagnosis and special demerit procedures or other record keeping methods must be used to identify faults in this class. In some sense the intermittent fault is a type of latent fault and for the same reasons as those given above, the "pooled" architecture has the best potential for effectively handling these failures.

This concludes the section on comparison of architectural candidates. Each of the individual elements listed in the comparison contribute to the overall issue of avionics reliability and life cycle costs in some way. It can be seen that in almost every case design is of primary importance to the ensuing reliability and that architecture of the overall avionics suite can be a controlling factor in achieving greater avionics reliability.

CONCLUSIONS

Some fundamental limitations in the contemporary approaches to avionics reliability have been identified. Key among these is a complexity divergence which will demand a new outlook on the design of future generations of avionics suites. This becomes even more necessary if it is desired to fully realize the potential available from control configured aircraft design. Functional failure rates of less than 10^{-9} per hour will be required in that case.

Based upon the limitations identified in the contemporary approach to avionics reliability and in the comparisons of the three architectural candidates it must be concluded that the challenge of the avionics reliability required for advanced aircraft can only be met by careful design of the overall system architecture. An integrated digital avionics architecture based upon dynamically allocated pooled resources appears to be capable of meeting this technology challenge.

As is so often the case, however, the path of greatest potential gain also represents the largest risk. Although it has received considerable attention in terms of analyses, simulations and experimental prototypes, the "pooled" approach has not been flight tested and has the smallest base of experience. However, results to date indicate that this avionics architecture is practical and has the potential to satisfy the emerging requirements of the 1990's while taking maximum advantage of the technology advances that are likely to occur in that time frame.

REFERENCES

1. Deyst, J.J. and Hopkins, A.L., Jr., "Highly Survivable Integrated Avionics", Astronautics and Aeronautics, September 1978

2. Walker, S.A., "Design of a Control Configured Tanker Aircraft", NASA Advanced Control Technology and its Impact on Future Transport Aircraft Symposium, Los Angeles, July 1974

3. UK Civil Aviation Authority, "The Safety Assessment of Systems", British Civil Airworthiness Requirements, Paper No. 670, September 1976

4. Flapper, J.A., "L-1011 Flight Control System", Agardograph 224, "Integrity in Electronic Flight Control Systems", April 1977

5. Osder, S.S., "Chronological Overview of Post Avionic Flight Control System Reliability in Military and Commerical Operations, Agardograph 224, "Integrity in Electronic Flight Control Systems", April 1977

6. Szalai, K.J., Felleman, P.G., Gera, J., and Glover, R.D., "Design and Test Experience with a Triply Redundant Digital Fly-By-Wire Control System", AIAA Guidance and Control Conference, San Diego, August 1976

7. Hopkins, A.L., Smith, T.B., Lala, J.H., "FTMP-A Highly Reliable Fault Tolerant Multiprocessors for Aircraft", Proceedings of the IEEE, October 1978

8. Deckert, J.C., Desai, M.N., Deyst, J.J., Willsky, A.S., "Reliable Dual Redundant Sensor Failure Detection and Identification for the NASA F-8 DFBW Aircraft" NASA CR2944, February 1978

9. Smith, T.B., "A Damage and Fault Tolerant Input/Output Network", IEEE Transactions on Computers, May 1975

10. Smith, T.B., "Damage Control Mechanisms in Digital Communication Networks for Distributed Real Time Control Systems", IEEE Intercon 75 Conference, New York, April 1975

TABLE 1

Advanced Tanker Comparison

	Conventional Design	Control Configured Design
Wing Area (ft^2)	10,640	8,984.
Wing Span (ft)	275	251.4
Fuselage Length (ft)	197	125.
Horizontal Tail (ft^2)	2,310	0
Vertical Tail (ft^2)	1,173	571.2
Design Weight (lb)	1,000,000	835,900.

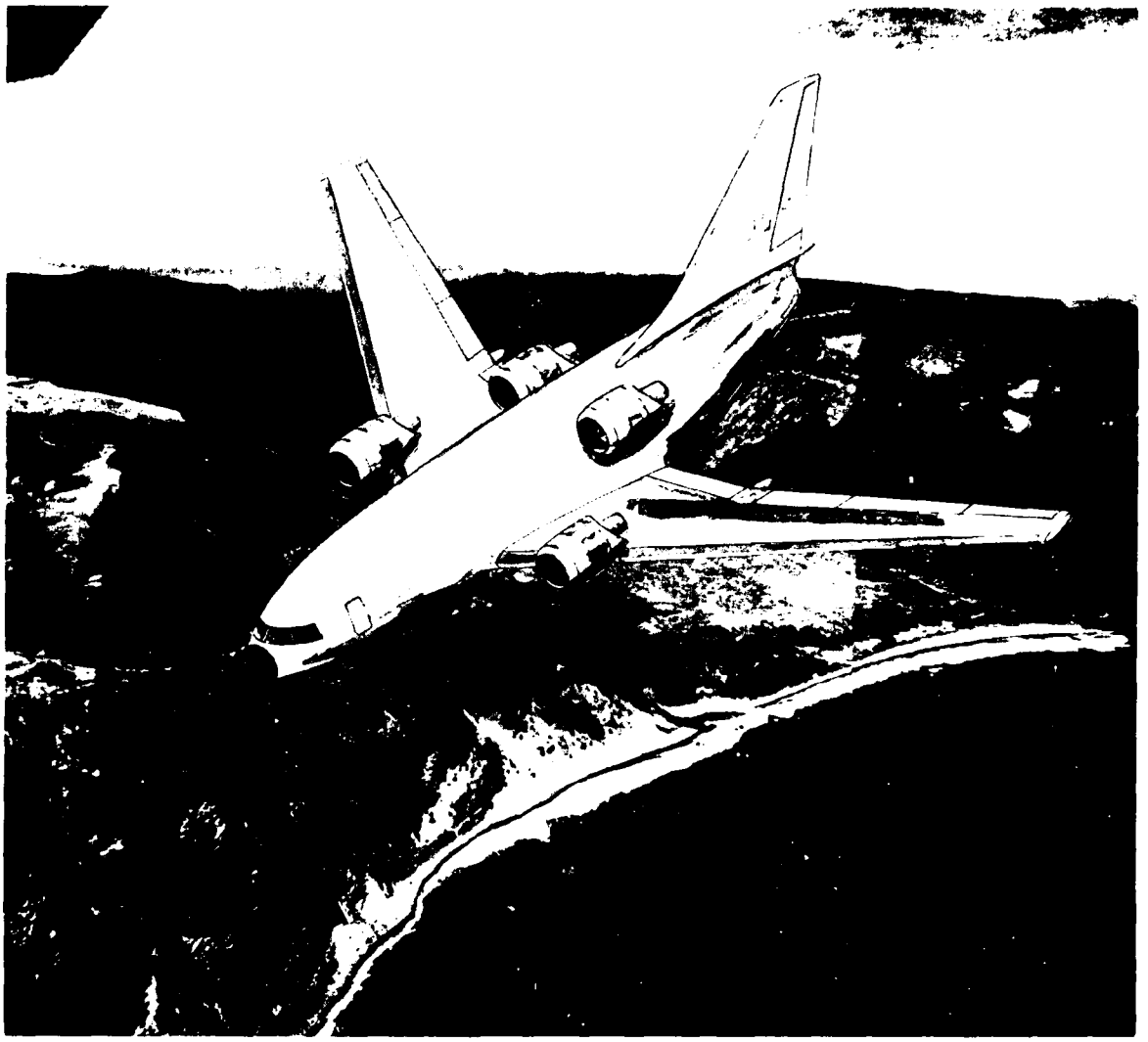


Fig.1 Control configured tanker

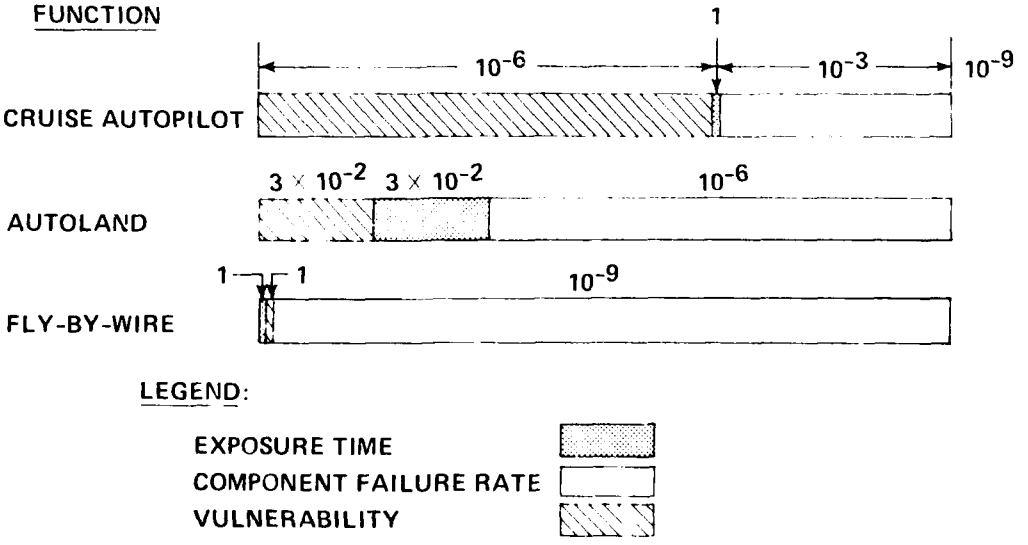


Fig.2 Reliability requirements for flight control

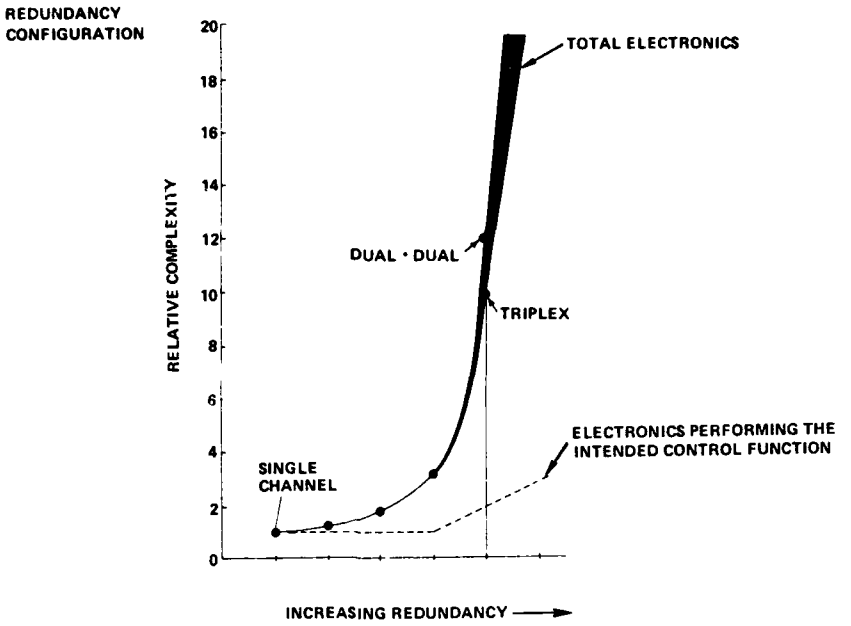


Fig.3 Redundancy management complexity overhead

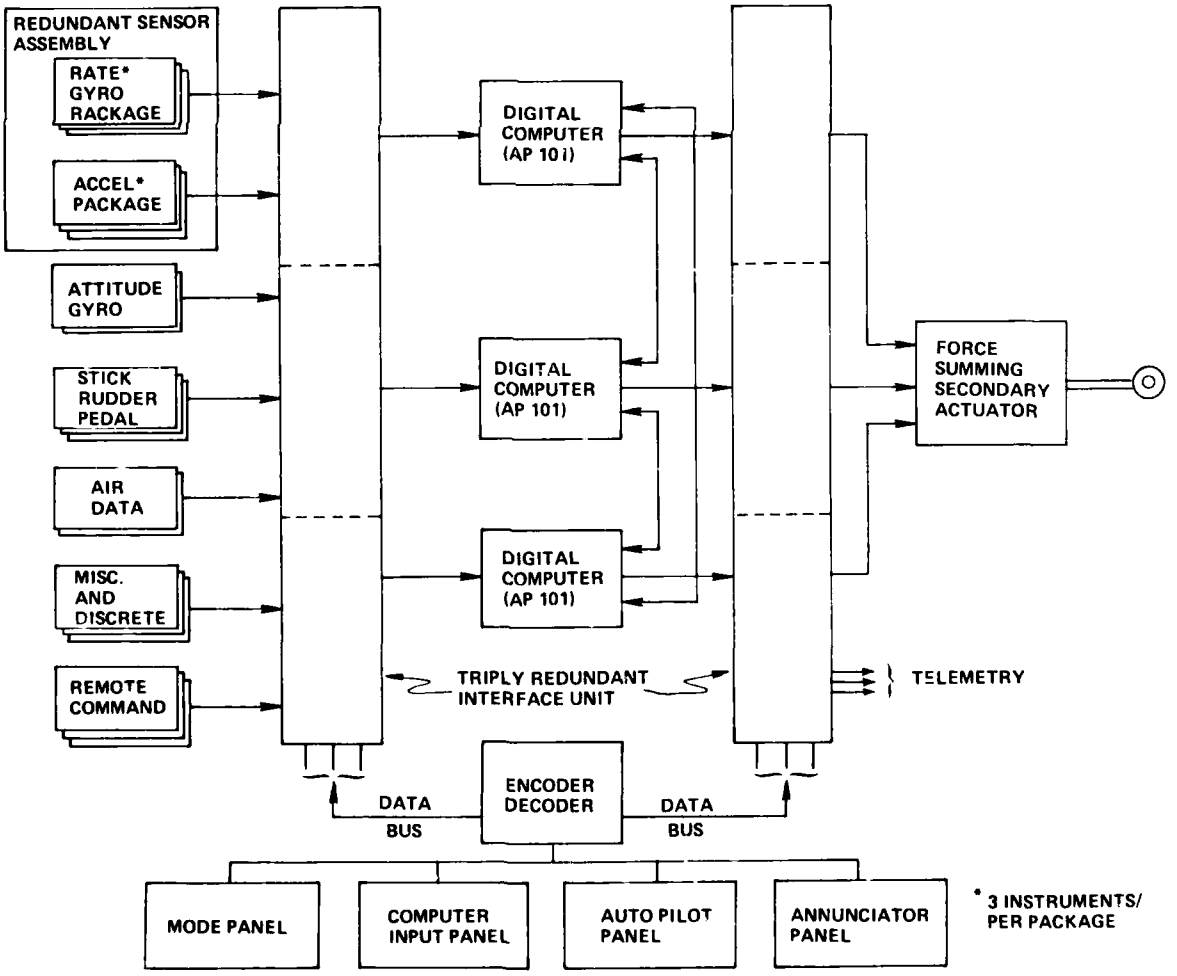
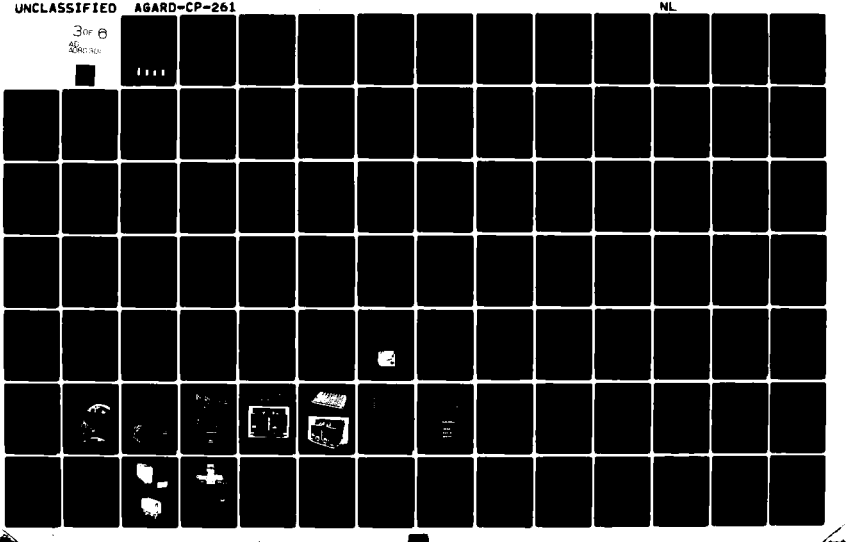


Fig.4 F-8 digital fly-by-wire system architecture

AD-A080 301 ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT--ETC F/G 9/5
AVIONICS RELIABILITY, ITS TECHNIQUES AND RELATED DISCIPLINES.(U)
OCT 79 M C JACOBSEN

UNCLASSIFIED AGARD-CP-261 NL

3 OF 8
AD A080301



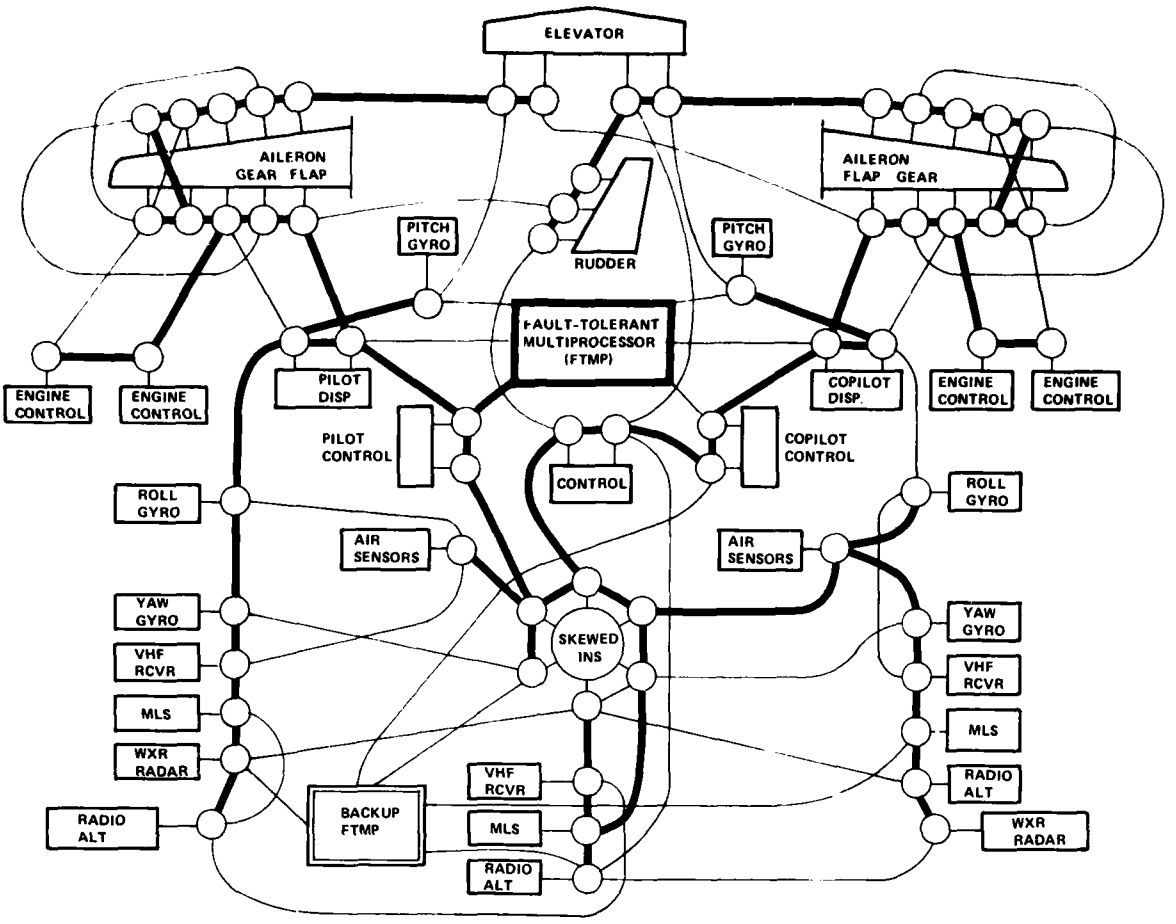


Fig.5 Pooled resource network architecture

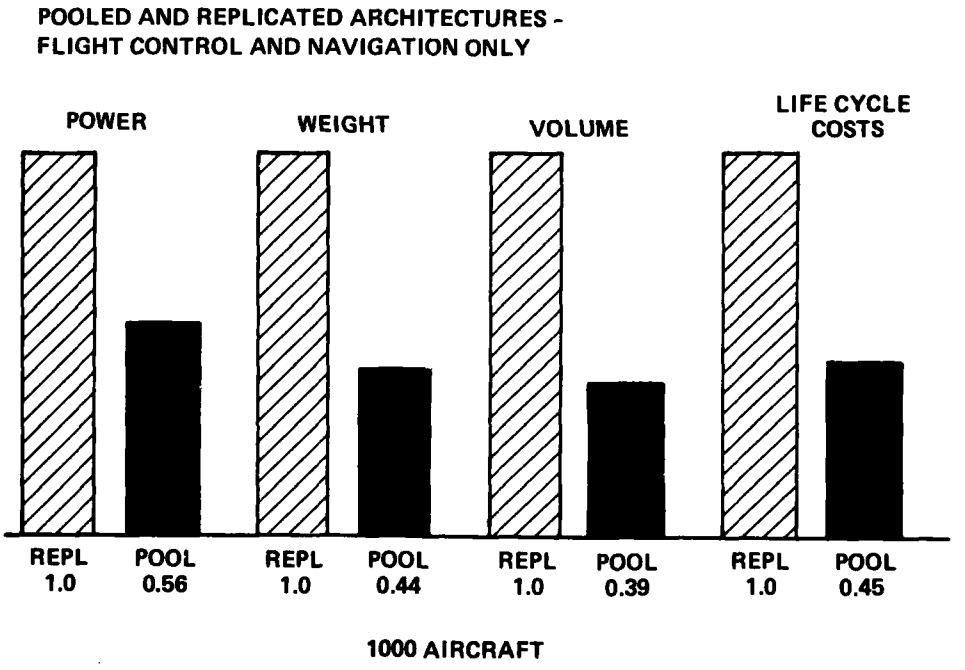


Fig.6 Pooled resource architecture advantages

TRENDS IN RELIABILITY MODELING TECHNOLOGY FOR FAULT TOLERANT SYSTEMS

Salvatore J. Bavuso
NASA Langley Research Center
Hampton, Virginia

SUMMARY

Recent developments in reliability modeling for fault tolerant avionic computing systems are presented. Emphasis is placed on the modeling of large systems where issues of state size and complexity, fault coverage, and practical computation are addressed.

A two-fold (analytical modeling) developmental effort is described based on the "structural modeling" and "fault coverage modeling" approaches. With regard to the structural modeling effort, two techniques under study are examined. One technique which was successfully applied to a 865 state pure death stationary Markov model is presented. The modeling technique is applied to a fault tolerant multiprocessor currently under development. Of particular interest is a short computer program which executes very quickly to produce reliability results of a large-state space model. Also, this model incorporates fault coverage states for processor, memory, and bus LRU's (Line Replaceable Unit).

A second structural reliability modeling scheme which is aimed at solving nonstationary Markov models is discussed. This technique which is under development will provide the tool required for studying the reliability of systems with nonconstant failure rates and includes intermittent/transient faults, electronic hardware which exhibits decreasing failure rates, and hydromechanical devices which typically have wearout failure mechanisms.

A general discussion of fault coverage and how it impacts system design is presented together with a historical account of the research which led to the current fault coverage developmental program. Several aspects of fault coverage including modeling and data measurement of intermittent/transient faults and latent faults are elucidated and illustrated. The CARE II (Computer-Aided Reliability Estimation) coverage is presented and shortcomings to be eliminated in the future CARE III are discussed.

The emergence of the so-called latent fault as a significant factor in reliability assessment is gaining increased attention from a modeling viewpoint; therefore, nuances of latent faults, models for such, and a method for latent fault measurement are depicted.

1. INTRODUCTION

The importance of achieving a faithful reliability assessment capability for avionic fault tolerant systems cannot be overstressed. Reliability issues involve virtually every aspect of design, packaging, and field operations, with regard to safety, maintainability, and invariably profits. Successful implementation of digital fault tolerant computers for critical flight functions in commercial aircraft cannot be realized without rigorous and credible analytical and simulative demonstrations of system reliability and fault tolerance. This conviction is fostered by the observation and supported by analysis that life testing to demonstrate the ultrareliability of these systems will be impractical, and because of the safety aspect, the full potential of such systems will not be realized until system reliability and fault tolerance are substantiated.

The task of producing a credible reliability assessment capability is indeed a formidable one. The root of the problem is embodied in the very essence that makes the digital computer such an attractive device for use in a host of applications, namely its adaptability to changing requirements, computational power, and ability to test itself.

Among the many factors to be considered in the design of fault tolerant systems are those which can have a direct impact on reliability. These factors must be accurately accounted for in a faithful reliability assessment. Figure 1 depicts some of the more important elements delineated into four categories: (1) Type and Manifestation, (2) Cause, (3) System Effect, and (4) Defense. Every digital avionic fault tolerant system must be designed to effectively cope with a myriad of hardware and software anomalies which are classified in categories 1 and 2. Categories 3 and 4 typify the effect of anomalies and some techniques for coping with them. Figure 2 portrays the combinations of categories 1 and 2. For example, a hardware anomaly could be a permanent random failure. On considering the number of devices in a digital system that are susceptible to failure in the ways depicted in figure 2 and combining software anomalies in a similar manner, one quickly begins to appreciate the designer's and the reliability analyst's tasks in accounting for these factors in reliability assessments. A rigorous discussion regarding some of these factors is given in McCluskey and Losq, 1978.

From a reliability assessment viewpoint, it was not until recently that analysts began to account for these factors (Roth et al., 1967) with the probabilistic concept of fault coverage. Since then, numerous reports have appeared on the effects of fault coverage accountability (Ultra-Systems, Inc., 1974; Bavuso, 1975; and Bjurman et al., 1976).

2. RELIABILITY MODELING APPROACH

Reliability modeling research at the NASA Langley Research Center has been strongly influenced by our fault tolerant computer architectural research program which commenced circa 1971 with the initiation of a study on the Design of a Fault Tolerant Airborne Digital Computer (Wensley et al., 1973, and Ratner et al., 1973). This study identified two potentially viable computer architectures for aircraft flight control

applications. They are the SIFT (Software Implemented Fault Tolerance) and the FTMP (Fault Tolerant Multiprocessor) (Wensley et al., 1978, and Hopkins and Smith, 1975 and 1978). Both architectural concepts utilize multiple LSI (Large Scale Integration) processor and memory devices, resulting in a large number of SRU's (Smallest Reconfigurable Units). From a reliability modeling point of view, this scheme contributes heavily to the modeling complexity by increasing the number of possible operational hardware states. This state of affairs has focused our research in the direction of developing modeling techniques that are applicable to large-state models. For convenience, this modeling thrust will be referred to as the structural analytic approach. A parallel effort to the structural analytic approach was initiated by a study in 1973 which produced the Computer-Aided Reliability Estimation (CARE II) computer program. To date, the CARE II fault coverage model represents the most advanced generalized model published in the open literature. It was this study which launched the Langley fault coverage modeling approach.

Because it is anticipated that viable fly-by-wire digital fault tolerant systems for aircraft flight control will be required to meet unreliability requirements of (less than or equal to) 10^{-9} per flight and to be practical (less than or equal to) 10^{-9} at 10 hours, reliability models must be implemented in analytic form in lieu of simulation models; however, the use of very high speed emulators and/or parallel computers may at some future time diminish the analytic approach's dominance. This is not to say that simulative techniques are not presently applicable in reliability modeling. On the contrary, simulation plays a major role in determining vital reliability parameters associated with fault coverage modeling.

3. STATE-OF-THE-ART MODELING PROBLEMS

The state-of-the-art of structural analytic modeling of large systems is typified by the reliability analysis method employed in the ARCS (Airborne Advanced Reconfigurable Computer System) study (Bjurman et al., 1976). The solution technique is matrix oriented and is based on constructing a similarity relation such that the transition matrix is similar to a diagonal matrix containing the eigenvalues along the diagonal. For pure-death Markov processes with distinct eigenvalues, this solution method is extremely fast in a general purpose digital computer and, thus, very attractive for use in large-state space models. With some minimal care in assigning failure rate data so that, for all practical purposes, the system eigenvalues are mathematically distinct, this solution scheme is applicable to a large class of computer architectures of practical interest. Such a system is the FTMP which was analyzed at Langley using the described method. An abbreviated state transition diagram for the FTMP appears in figure 3 where a system state is defined as the 6-tuple vector, (a,b,d,c,e,f), where

a = number of working processor modules
b = number of processor modules in a recovery state
c = number of working memory modules
d = number of memory modules in a recovery state
e = number of working bus modules
f = number of bus modules in recovery

and the SRU's are the processor, memory, and bus modules. Initially the system is in state (10, 0, 10, 0, 5, 0) and the final state is (5, 0, 2, 0, 2, 0). Further loss of hardware is considered system failure since crucial flight functions cannot be effected. Elements, b, d, and f describe states involving recovery. In addition to system loss resulting from hardware depletion, system failure occurs (in this model) when a second fault occurs within a recovery interval. This condition was imposed because the FTMP's primary fault detection and isolation mechanisms are based on a functional level software majority voting scheme. In actuality, the FTMP can recover from many double failures; however, the double failure constraint was necessary to reduce the state size of the reliability model; fortunately, it also produces a conservative reliability estimate. Several other necessary conservative assumptions were required to bring the state size down to a manageable level; in this case, a 865-state model resulted. Although 865 states for a reliability model is considered very large by industry standards, this analysis presented no problem for our Control Data Corporation CYBER 175 computer. In fact, a mission time of 10 hours required only 74 CPU seconds.

Aside from the surprising low CPU time of such a complex analysis, another unexpected outcome resulted and is shown in figure 4. The probability of system failure in 10 hours is plotted against processor failure rate per hour for the 865-state model with 10 processors, 10 memories, and 5 buses; and for a 673-state model with 10 processors, 8 memories, and 5 buses. The data show that the addition of 2 memory modules increased the system probability of failure. This trend also applies if in lieu of "processor" appearing in figure 4, "memory" or "bus" is plotted. One explanation for this unexpected data is the sensitivity of the reliability model to the occurrence of a second fault during recovery. Beyond a particular hardware complement, increasing hardware redundancy diminishes system reliability because of the increased likelihood of additional faults. If the constraint that a second fault occurring in a recovery interval fails the system were relaxed, the results will change in favor of increasing redundancy. The penalty for increased realism is a considerable increase in the model state size. To date, a practical upper bound on the state size for the matrix solution technique previously discussed has not been explored. On the pessimistic side, it is sobering to realize that the 865-state model was reduced from approximately 10 million states through the imposition of certain conservative constraints on the model.

The state-of-the-art of reliability modeling of large systems has progressed one step beyond that already described to include transient faults. This amounts to adding the transient failure rate (transition rate) to hardware failure rates to account for persistent transient faults that behave like permanent faults (Bjurman et al., 1976, and Ng, 1976). The reliability contribution due to the time the machine spends in the recovery state because of a transient is not accurately modeled: As most analyses assume constant transient transition rates, one can ignore the recovery state and combine the transient transition rate with the permanent fault transition rate.

This scenario of the state-of-the-art of reliability modeling for fault tolerant systems surely must convey the notion of modeling inaccuracies, not to mention the conspicuous absence of any discussion of software anomalies and other anomalies portrayed in figure 2. Even though the reliability analyst makes

every attempt to be conservative when he cannot be accurate, more often than not he is forced into a compromising position that raises doubt and diminishes confidence in the analysis.

4. A NOVEL APPROACH FOR RELIABILITY ASSESSMENT

In the aforementioned analysis of the FTMP, irrespective of software considerations, the major suspects which challenge both the accuracy and the conservatism of the analysis are the transient and fault recovery treatment. In both cases, it was assumed that the state transition rates are constant and values for these were determined by educated guesses. Also, it was assumed that the latency time is zero (Shedletsky and McCluskey, 1976). Trends in reliability modeling technology for fault tolerant systems are being driven by the need for analytic techniques capable of modeling fault tolerant systems with state sizes on the order of 1000, to include sensors, actuators, and their computer interfaces. There is mounting evidence that certain electronic devices exhibit nonconstant hazard rates (Timming, 1975, and Shooman, 1974); and mechanical and hydraulic devices commonly exhibit wearout, i.e., increasing hazard rates with time. These observations coupled with the need to accurately account for fault latency, intermittent/transient faults, and software failures present a strong case for an analytic technique capable of modeling nonconstant hazard rates.

The development of such a technique is currently under study and will result in the development of a General Computer-Aided Reliability Estimation (CARE III) computer program. The desire to reduce the large state sizes for Markov processes vis-a-vis CARSRA (Computer-Aided Redundant System Reliability Analysis, Bjurman et al., 1976) and the need to treat nonconstant hazard rates directed the study toward a generalized Markov process concept, namely the processes in which the Chapman-Kolmogorov equation holds:

$$P_{\ell i}(t,\tau) = \sum_{\nu} P_{\nu i}(s,\tau) P_{\ell \nu}(t,s)$$

for all $\tau < s < t$, where $P_{\ell i}(t,\tau)$ is the probability that the system is in state ℓ at time t given that it was in state i at time τ (Feller, 1957). By judicially defining system states to satisfy the Chapman-Kolmogorov equation, the forward Kolmogorov equation can be satisfied under some very general conditions:

$$\frac{\partial P_{\ell i}(t,\tau)}{\partial t} = -P_{\ell i}(t,\tau)\,\lambda_{\ell i}(t,\tau) + \sum_{j \neq \ell} P_{ji}(t,\tau)\, C_{j\ell i}(t,\tau)\lambda_{j\ell i}(t,\tau)$$

If the notation indicating the condition that the system be in state i at time τ be suppressed, the following recursive equation results:

$$P_{\ell}(t) = e^{-\int_0^t \lambda_{\ell}(\tau)d\tau} \int_0^t \frac{\sum_j P_j(\tau) c_{j\ell}(\tau)\lambda_{j\ell}(\tau)}{e^{-\int_0^\tau \lambda_{\ell}(\eta)d\eta}} d\tau$$

where

$P_{\ell}(t)$ = probability of being in state ℓ at time t

$\lambda_{j\ell}(t)$ = transfer rate from state j to state ℓ

$\lambda_{\ell}(t) = \sum_j \lambda_{\ell j}(t)$

$c_{j\ell}(t)$ = coverage associated with a failure which, if coverage were perfect, would cause a transfer from state j to state ℓ

The system reliability is given by

$$R(t) = \sum_{\ell \in L} P_{\ell}(t)$$

for the set L of allowable states.

From a computational point of view, a more accurate form is obtained by letting

$$Q_{\ell}(t) = P^{*}_{\ell}(t) - P_{\ell}(t)$$

where $P^{*}_{\ell}(t) = P_{\ell}(t)$ given perfect coverage. The system unreliability $Q(t)$ is given by

$$Q(t) = 1 - R(t) = \sum_{\ell \in \bar{L}} Q_\ell(t) + \sum_{\ell \in L} P^*_\ell(t)$$

with $\bar{L} \cup L$ being the set of all possible states. And $\bar{c}_{j\ell}(t) = 1 - c_{j\ell}(t)$ so that

$$Q_\ell(t) = e^{-\int_0^t \lambda_\ell(\tau)d\tau} \int_0^t \frac{\left[\sum_j Q_j(\tau) + P_j(\tau)\bar{c}_{j\ell}(\tau)\right]\lambda_{j\ell}(\tau)}{e^{-\int_0^\tau \lambda_\ell(\eta)d\eta}} d\tau$$

The virtues of this scheme are that the hazard rate $\lambda_{j\ell}(t)$ and coverage $c_{j\ell}(t)$ are time dependent; also the contribution to system unreliability due to perfect and imperfect coverage is decoupled. The need for the $\lambda_{j\ell}(t)$ was previously discussed, but the importance of $c_{j\ell}(t)$ was not presented.

In avionic systems which utilize dynamic resource allocation schemes such as is possible with the FTMP and SIFT systems, the proportion of hardware and software resources is dependent on the aircraft flight phase and/or flight envelope. Flight critical phases require greater hardware redundancy and fault monitoring. The latter factor appears in reliability models as time-varying coverage $c_{j\ell}(t)$. A more subtle need for $c_{j\ell}(t)$ is to account for fault latency. The probability of system failure due to insufficient coverage is a function of the number of existing failures embedded in the system. That is, the probability of a second SRU (processor, bus, memory) failure occurring during the τ second recovery time is a function of the number of SRU's functioning at that time.

Preliminary studies of the Kolmogorov technique are encouraging from an accuracy viewpoint and computer run time. Figures 5 and 6 compare FTMP reliability data generated with the Kolmogorov technique against data generated with other conventional techniques. To make a meaningful comparison, $c_{j\ell}(t)$ and $\lambda_{j\ell}(t)$ were constrained as constants in the Kolmogorov technique. It is suspected that the discrepancies depicted in figure 6 are attributed to simplifying assumptions required to keep the conventional analysis technique tractable.

Current work on CARE III is directed toward developing a coverage model compatible with the Kolmogorov technique and is based to a large extent on the CARE II coverage model (Raytheon Company, 1974 and 1976). Improvements to be sought are modification for coverage time dependency $(c_{j\ell}(t))$ to model latent faults and of greater difficulty, to reduce the burden placed on the user in defining input data for the modified CARE II coverage model. A third improvement is to include a more sophisticated intermittent/transient fault coverage model and if possible a software failure model.

The CARE II coverage model is a powerful basis upon which to build the Kolmogorov coverage model (KCM). In its completed form, the KCM will determine coefficients for the Kolmogorov reliability model (KREL-M). Coverage is conceived as consisting of three fundamental processes, system fault detection, fault isolation to the SRU, and recovery, which may require hardware replacement and/or software correction. Failure to properly effect one of these processes constitutes a coverage failure which is usually modeled as a system failure. A faithful coverage model must provide the mechanisms by which the reliability analyst can relate the coverage coefficients to the system factors that affect coverage. These factors include the fault classes (permanent/intermittent hardware/software faults), the system fault detection mechanisms (software/hardware voting, software self-monitoring, BITE (Built In Test Equipment), etc.), SRU fault isolation mechanisms (similar to detection), and recovery procedures (hardware replacement, instruction retry, etc.). Detectors are modeled as competitors in the detection process. Every detector has some chance of discovering a fault; however, most detectors usually are specialized for a particular class of faults. In CARE II, this modeling process is under user control. It is assumed in the coverage model that the detector which discovers a fault is most capable of defining fault isolation and recovery strategies. These strategies are user defined.

The CARE II coverage model takes the following form:

$$c_x(i,j) = P_i P_{sx}{}^{j-1} P_i' \int_0^\infty \int_0^\infty g_i(\tau) h_i(\tau' - j\tau_{sx}) r_i(\tau,\tau') d\tau \, d\tau'$$

where

$c_x(i,j)$ = conditional probability system can recover from a fault in stage x given the fault belongs to fault class j and is detected by detector i *

τ = detection time

τ' = isolation time

P_{sx} = defective spare detection

τ_{sx} = spare unit test time

*A stage is defined as a set of identical devices.

P_i = noncompetitive detection probability

P_i' = isolation probability associated with P_i

h_i = isolation rate

r_i = recovery probability

g_i = competitive detection rate

Of all of these parameters, $g_i(\tau)$ is the most difficult to obtain because it is a function of detector i and the entire ensemble of detectors and their interrelationships.

5. ACQUISITION OF COVERAGE DATA

Assuming success in modifying the CARE II coverage model for the KREL-M, some difficulty in using this capability still remains. Eventually the analyst must obtain coverage data peculiar to the system of interest. Three types of data are urgently required: intermittent hazard rate data including duration densities, fault detection densities for various classes of faults and detectors, and software hazard rate data. There is some encouraging news on the first two; a discussion on the third is beyond the scope of this paper and will not be addressed further.

A source of intermittent arrival data has been identified and work has recently commenced to generate a data base of intermittent field hardware failure data in digital electronics*. The long-term aim of this endeavor is to produce intermittent hazard rate data for a variety of digital devices using different parts technology but applicable to avionics.

Beyond the pressing issues surrounding software reliability, validity and/or validation, characterization of the latent fault ranks in equal importance to the eventual success of utilizing digital systems for flight critical functions. Because of the near infinite number of possible machine states that a digital computer can obtain as a result of failures, it is impossible to exhaustively test such a device to determine its health. Therefore the presence of undetected faults is always a possibility, and for systems designed to obtain system probabilities of failure of less than 10^{-9} in 10 hours of flight, even small probabilities of latent faults occurring can have a large effect on system reliability. It is certainly with these thoughts in mind that designers incorporate redundancy; however, the cost of constructing machines which tolerate more than three coexisting manifested faults becomes prohibitive. An acceptable solution is to constantly search for faults and eliminate their effects so that the machine is never presented with two coexisting manifested faults, i.e., only one at a time. To insure that this goal is satisfied, the designer must have a priori knowledge of fault occurrence and manifestion rates so that adequate fault detection and recovery mechanisms can be incorporated.

There are a number of detection schemes; the most obvious is comparison/voting and can be implemented in at least one of two ways: by executing a special software test and comparing expected results with computed results (self-monitoring) or two or more uniprocessors can compare functional level outputs during normal computation where both processors are executing the same code. The time between fault occurrence and its detection is the latency time. If this time is short compared with the failure rate of SRU's, then the machine will essentially see single failures and have sufficient time to cope with them. Long latency times are conducive to system failure.

In an attempt to determine methods of acquiring latency data, a study entitled, "Modeling of a Latent Fault Detector in a Digital System" was conducted (Nagel, 1978). A very simple computer (VSC) modeled at the gate level was designed and simulated to execute on a CDC CYBER 175 host computer. Six simple programs were written using the VSC that consisted primarily of the following instructions:

Fetch and store
Add and subtract
Shift right and shift left
AND and OR
Indirect addressing
Overflow indicator
Branch
Copy to and from temporary storage

While the VSC executed each of the six programs, single faults were induced random uniformly over the gate list. Input, output, stuck-at-one, and stuck-at-zero faults were equally likely occurrences. Initially the number of runs manifesting faulty output was recorded and produced the following results:

PROGRAM	SAMPLE SIZE	DETECTIONS	ESTIMATED DETECTION PROBABILITY	ESTIMATED STANDARD DEVIATION
Fibonacci (FIB)	211	98	0.464	0.034
Fetch and Store (F&S)	118	42	.356	.044
Add and Subtract (A&S)	208	117	.563	.034
Search and Compute	118	64	.542	.046
Linear Convergence	133	78	.586	.043
Quadratic	97	55	.577	.050

*NASA Contract Number, NAS1-15574 with Sperry Univac.

Extensive data analysis was performed to explain the observed differences in terms of the number of executed instructions, the number of different instructions used in computation, the degree of branching, the fault mode (stuck-at-one or zero, input or output), and number size. The results of the statistical analyses indicate that latency time, or equivalently, detection capability, depends primarily on the instruction subset used during computation and the frequency of its use. Moreover, little direct dependence was observed for such factors as fault mode, number size, degree of branching, and program length. An exponential model was proposed and applied to the data from three programs (Add and Subtract, Fibonacci, and Fetch and Store)

The exponential model is based on the density function of $y = \min(t, T)$, where t is the detection time measured in repetitions and T is the truncation time of test, and is given by:

$$f(y) = \begin{cases} P_0 \lambda e^{-\lambda y} & y < T \\ P_0 e^{-\lambda T} + Q_0 & y = T \qquad (Q_0 = 1 - P_0) \\ 0 & \text{Elsewhere} \end{cases}$$

where

P_0 = the detection probability

Q_0 = the probability of nondetection for all time

$P_0 e^{-\lambda T}$ = the probability of nondetection due to insufficient test time

Values for P_0 and λ were obtained using maximum likelihood estimators, enabling the following data to be generated:

Program	P_0	λ	λ/P_0
A&S	0.568	0.577	1.02
FIB	.474	.491	1.04
F&S	.371	.398	1.07

A pictorial representation of this model is shown in figure 7 superimposed on the raw data in histogram form.

If after careful testing, this method of measuring and modeling fault latency proves to be acceptable, an important set of coverage parameters will become available for reliability modeling. As an aside, this scheme also provides a method for synthesizing test programs both for pre-flight and in-flight monitoring.

6. CONCLUDING REMARKS

Testing digital systems which perform flight critical functions is not a feasible method for estimating system reliability. Analytic modeling of system reliability in conjunction with simulative techniques for coverage measurement appears to be the only alternative on the horizon. Accurate reliability estimates which account for such factors as latent faults, intermittent/transient faults, and software errors require sophisticated techniques which are currently being developed and will result in the KREL-M reliability assessment capability embodied in the CARE III computer program. The effects and significance of these factors on the reliability of fault tolerant digital systems are yet to be determined, and the potential of increased complexity brought about by the inclusion of these factors in an assessment capability such as KREL-M is a major concern. It is anticipated that after extensive trade-off analyses, KREL-M will be simplified and take on more of the characteristics of a production tool in lieu of its initial experimental character.

In a parallel effort, methods for acquiring indispensable coverage data required by KREL-M are now becoming available.

REFERENCES

Bavuso, S. J., 1975, Impact of Coverage on the Reliability of a Fault Tolerant Computer, NASA TN D-7938.

Bjurman, B. E., Jenkins, G. M., Masreliez, C. J., McClellan, K. L., and Templeman, J. E., 1976, Airborne Advanced Reconfigurable Computer System (ARCS), The Boeing Commercial Airplane Company, NASA CR-145024.

Feller, W., 1957, An Introduction to Probability Theory and Its Applications, John Wiley & Sons, Inc.

Hopkins, A. L., and Smith, T. B., 1975, The Architectural Elements of a Symmetric Fault-Tolerant Multiprocessor, IEEE Trans. on Computers, vol. C-24, no. 5.

Hopkins, A. L., and Smith, T. B., 1978, A Fault Tolerant Multiprocessor Architecture for Aircraft, Vol. I, The Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts, NASA CR-3010.

McCluskey, E. J., and Losq, J., 1978, Critical Fault Patterns Determination in Fault-Tolerant Computer Systems, Stanford University, NASA CR-145352.

Nagel, P. M., 1978, Modeling of a Latent Fault Detector in a Digital System, NASA CR-145371.

Ng, Y., 1976, Reliability Modeling and Analysis for Fault-Tolerant Computers, PhD Dissertation, UCLA - EWG-7698.

Ratner, R. S., et al., 1973, Design of a Fault Tolerant Airborne Digital Computer, Volume II - Computational Requirements and Technology, Stanford Research Institute, Menlo Park, California, NASA CR-132253.

Raytheon Company, Sudbury, Massachusetts, 1974, Reliability Model Derivation of a Fault-Tolerant, Dual, Spare-Switching Digital Computer System, NASA CR-132441.

Raytheon Company, Sudbury, Massachusetts, 1976, An Engineering Treatise on the CARE II Dual Mode and Coverage Models, NASA CR-144993.

Roth, J. P., Bouricius, W. G., Carter, W. C., and Schneider, P. R., 1967, Phase II of an Architectural Study for a Self-Repairing Computer, SAMSO TR-67-106, United States Air Force. (Available from DDC as AD 825460.)

Shedletsky, J. J., and McCluskey, E. J., 1976, The Error Latency of a Fault in a Sequential Digital Circuit, IEEE Trans. on Computers, vol. C-25, no. 6.

Shooman, M. L., 1974, Hazard Function Monitoring of Airline Components, Proceedings of Annual Reliability and Maintainability Symposium.

Timming, A. R., 1975, A Study of Total Space Life Performance of GSFC Spacecraft, NASA TN D-8017.

Ultra-Systems, Inc., Newport Beach, California, 1974, Reconfigurable Computer Systems Study, NASA CR-132537.

Wensley, J. H., et al., 1973, Design of a Fault Tolerant Airborne Digital Computer, Volume I - Architecture, Stanford Research Institute, Menlo Park, California, NASA CR-132252.

Wensley, J. H., et al., 1978, Design Study of Software - Implemented Fault Tolerance (SIFT) Computer, SRI International, Menlo Park, California, NASA CR-3011.

CATEGORY		
1. TYPE & MANIFESTATION:	- HARDWARE ANOMALY ● PERMANENT ● TRANSIENT ● INTERMITTENT	- SOFTWARE ANOMALY ● PERMANENT ● TRANSIENT ● INTERMITTENT
2. CAUSE:	- DESIGN ERROR - FABRICATION ERROR - RANDOM FAILURE - EXTERNALLY INDUCED ● SIGNAL ERROR ● POWER FAILURE ● PHYSICAL FAILURE ● EMI	- DESIGN ERROR - CODING ERROR - EXTERNALLY INDUCED ● DATA PATTERN ERROR ● PROCEDURE ERROR
3. SYSTEM EFFECT:	- COMPUTER SYSTEM CONTROL LOSS - APPLICATION COMPUTATION ERROR - NONE	- COMPUTER SYSTEM CONTROL LOSS - APPLICATION COMPUTATION ERROR - NONE
4. DEFENSE	- HARDWARE REDUNDANCY ● SPATIAL - ALTERNATE HARDWARE ● TEMPORAL - RETRY (TRANSIENT)	- SOFTWARE REDUNDANCY ● SPATIAL - ALTERNATE CODE ● TEMPORAL - RETRY

Figure 1. Factors Affecting Coverage.

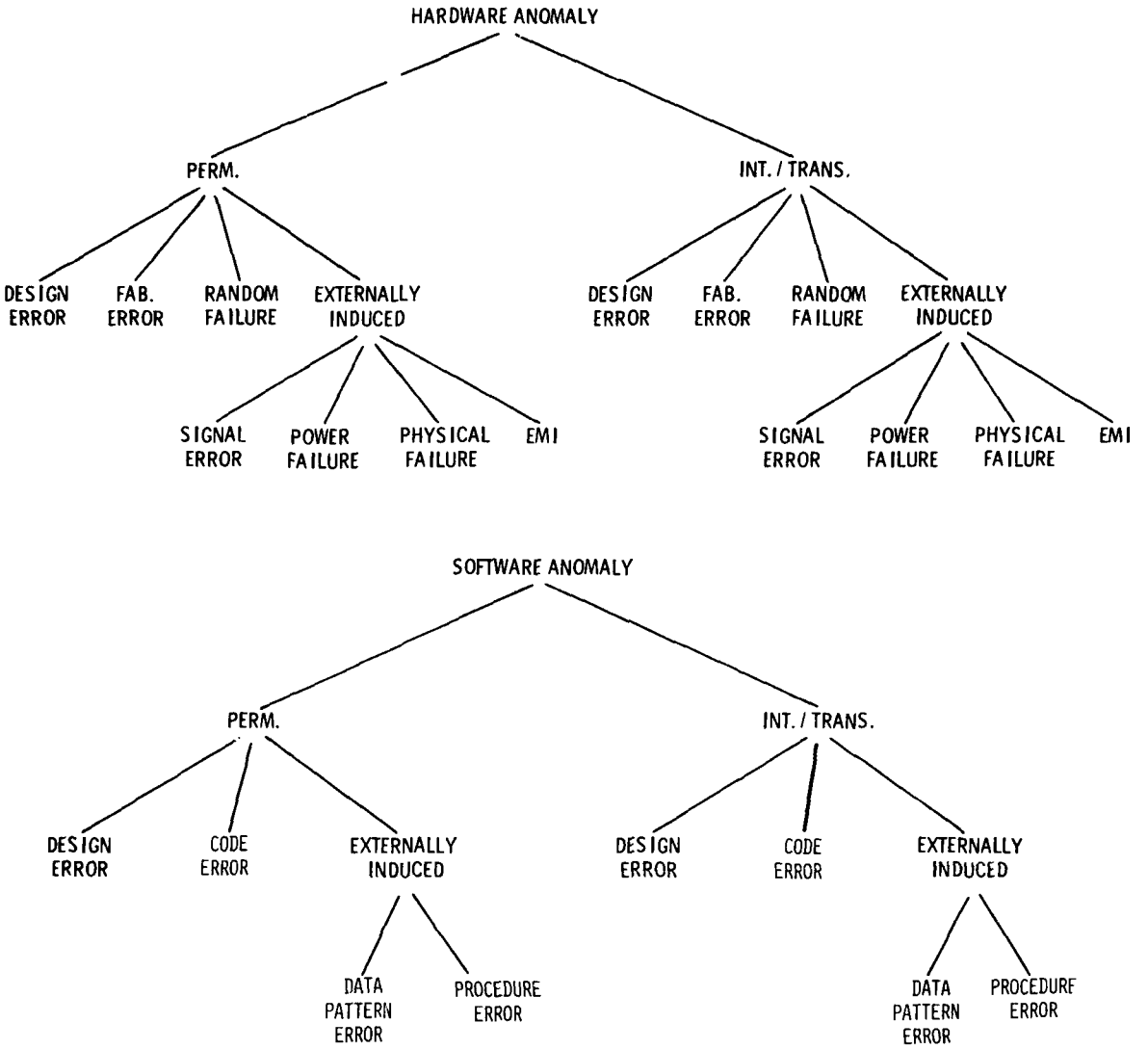


Figure 2. Delineation of Hardware and Software Anomalies.

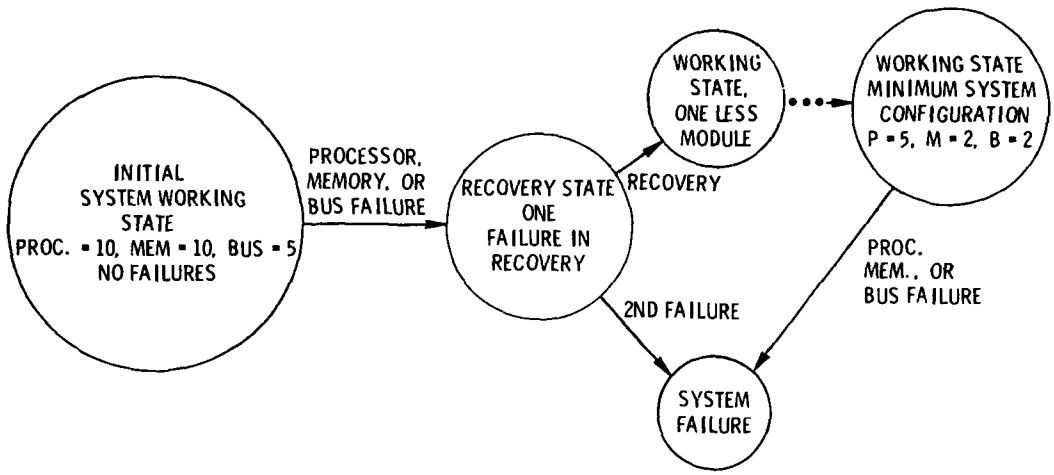


Figure 3. FTMP State Transition Diagram.

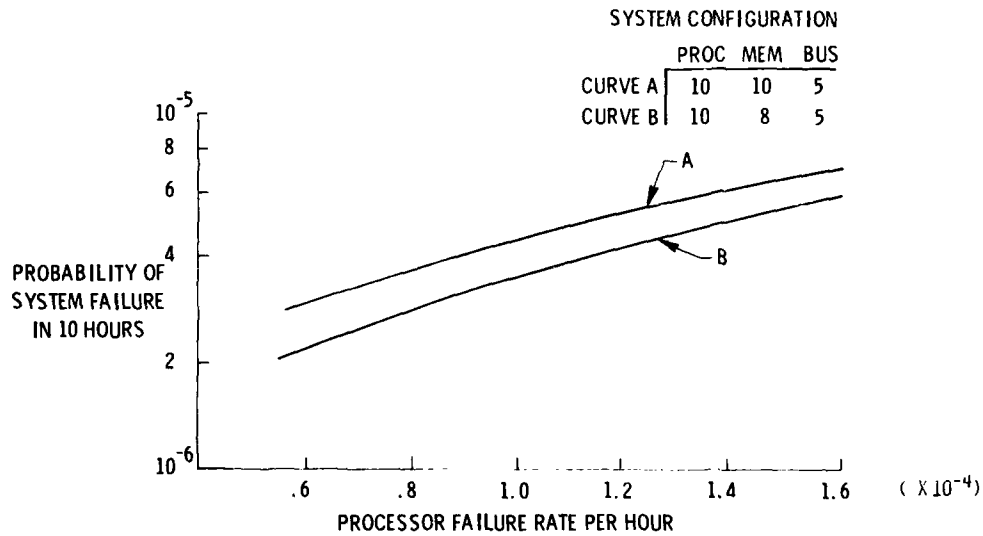


Figure 4. Probability of System Failure Versus Processor Failure Rate for the FTMP.

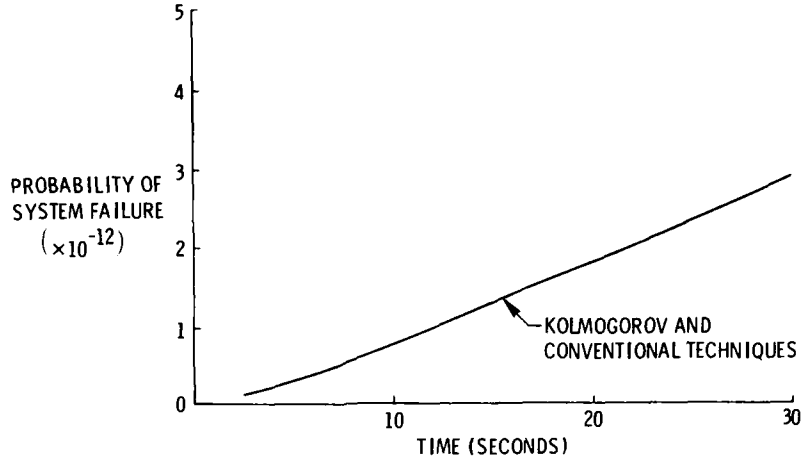


Figure 5. Probability of System Failure Versus Time for FTMP.

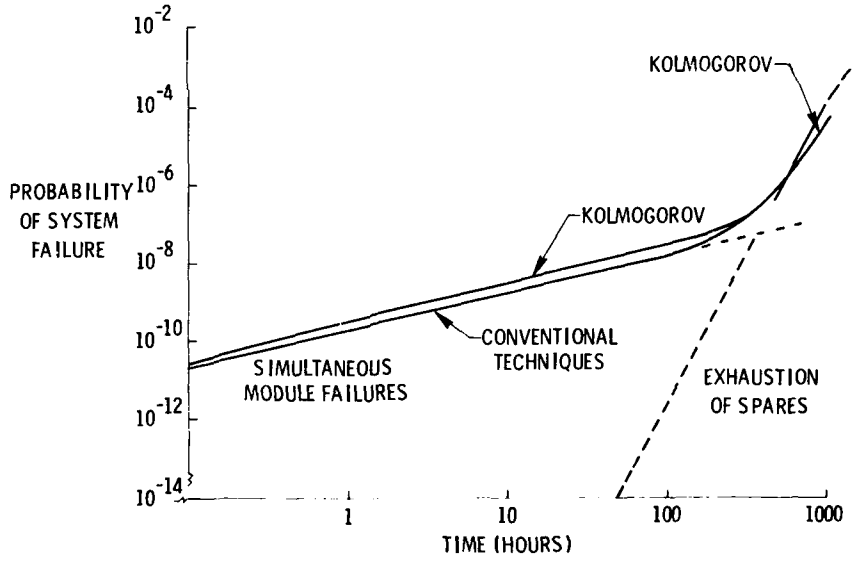


Figure 6. Probability of System Failure Versus Operating Time for the FTMP.

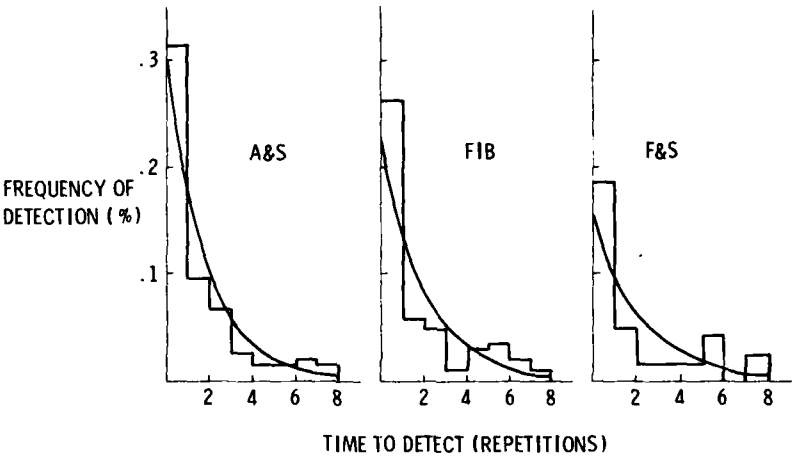


Figure 7. Exponential Model of Fault Latency (Detection).

NON-ELECTRONIC ASPECTS OF AVIONIC SYSTEM RELIABILITY

by

C. V. KENMIR

R. G. HILTON

H. H. DIXON

Dowty Boulton Paul Limited,
Wolverhampton, WV 9 5 EW, England

SUMMARY

The actuation system is an important link in the control chain which transforms the electrical demands from the avionics system into large forces capable of positioning the control surfaces. In the past such actuation systems, usually hydraulically powered, have been mechanically signalled by the pilot but may have had electrical inputs added. Protection against malfunction has been by limiting authority in the case of autostabilisers and by limiting rate in the case of autopilots. In new aircraft with electrical signalling as the main, and perhaps the only, mode of control, protection against malfunction is being obtained by means of redundancy.

This paper examines methods of obtaining system integrity by means of redundancy in the actuation and how the chosen solution is affected by constraints such as control surface layout, numbers of power supplies and the form of the avionics. It also looks at the effect of this redundancy on the defect rates and considers developments which improve reliability and also remove some of the constraints.

1. INTRODUCTION

After the Avionics system has taken note of all the flight conditions from various sensors and has considered any instructions it may have received from the pilot or elsewhere, it very rapidly transforms these into a command to maintain or to change the flight path of the aircraft. To do this, usually by means of control surfaces, requires a large amount of power and this is usually hydraulic power. This non-electronic link in the control chain, however, is just as vital as any other link and therefore the problem of reliability is just as important.

In most aircraft flying today, and in many aircraft being projected, the primary input to the hydraulic actuators which position the control surfaces is a mechanical system of rods and/or cables. In these cases the avionics input is usually in two parts, that is the stability augmentation input (S.A.S) which is high speed and low authority, and the automatic pilot input which is high authority and low speed. By separating these functions, a measure of protection against malfunction is automatically obtained by the limited authority in one case and limited rate in the other.

Some aircraft already flying, for example Concorde, use electrical signalling as the normal operating mode with the mechanical input as a stand-by and many future designs will be completely fly-by-wire. In such cases the avionic system is controlling a high speed, high authority actuator. Because of this, the results of a malfunction can be disastrous and so protection against such an event must be provided. It is about this type of system, the type which will be used more and more in the future, that we are concerned with in this paper.

2. RELIABILITY AND INTEGRITY

We say a system is unreliable if it goes wrong frequently and so when we talk about reliability we usually mean the time between defects. When defects can lose the aircraft we must protect against this by use of redundancy, this improves the integrity but, because we now have more components, it usually makes the system less reliable.

Using round numbers, if we required the integrity of a system to be such that the probability of a malfunction was 10^{-9} per hour and if we could design a simplex system to meet this requirement then the mean time between defects would be 10^{9} hours which would be excellent.

However, if we can only make a signalling lane with a mean time between defects of about 1500 hours then we would have to design a system which would operate until a third failure and so we would require a quadruplex system. To meet the target of 10^{-9}/hour the quadruplex system would need to have a lane mean time between defects of 1585 hours giving a total for four lanes of 396 hours.

Table 1, which is only approximate, shows the relationship between redundancy and defect rates for majority voting systems meeting an overall failure rate of approximately 10^{-9} per hour.

Type of System	No. of failures to lose control	Required Lane Reliability	Mean Time Between Defects
Simplex	1	10^{-9} /hr	10^{9} hrs
Triplex	2	1.82×10^{-5}/hr	18315 hrs
Quadruplex	3	6.3×10^{-4}/hr	396 hrs

Table 1

This table also shows the great improvement in component reliability which is required before the amount of redundancy can be reduced. For example, in a quadruplex system, a tenfold increase in component reliability would raise the mean time between defects from 396 to 3960 hours. An increase in reliability of 34.6 times however, is required before the system can be simplified from quadruplex to triplex. At the present state of the art, fly-by-wire actuation systems for military aircraft are being designed for two failure survival, it will be seen that appreciable improvements in reliability will be required before that amount of redundancy can be reduced.

3. EFFECT OF AIRCRAFT CONFIGURATION ON ACTUATION REQUIREMENTS

When it is said that a system for military aircraft should be capable of two failure survival the control surfaces must be considered as part of the control chain. Thus in the pitch axis, if there is a single control surface, then the actuation system itself must have two failure survival characteristics. On the other hand, if the aircraft roll control were to consist of two ailerons plus two spoilers, a lower level of actuation redundancy would be acceptable. Possible levels of actuation redundancy are given in Table 2 below.

Axis	Surfaces	1st Failure	2nd Failure
Pitch	Tailplane	Fail operate	Fail operate
Yaw	Rudder	Fail operate	Fail central*
Roll	2 Ailerons	Fail operate	Fail central
Roll	4 Ailerons	Fail central	

Table 2

* If the aircraft is such that it can be landed without rudder in a reasonable cross wind.

The V.C.10, a large transport aircraft, has 4 ailerons plus spoilers, 3 rudders and 4 elevators, and the actuation systems are each fail neutral. A fail neutral signalling system could be used with such an aircraft with acceptable integrity.

Further consideration of integrity problems will be confined to single surface cases where the actuation system must be fail operate, fail operate, because this is the most severe case.

4. CONSTRAINTS

If we are considering an actuation system which must operate after two failures we must consider any two failures of any of the following :-

a) An aircraft hydraulic system

b) An aircraft electrical system

c) An electrical signal from a computer

d) An actuation lane.

The layout of the system will depend upon the following :-

a) Number of hydraulic systems available.

b) Number of non-guaranteed signals from the computers.

or c) Number of guaranteed computer signals.

According to the general constraints it may be decided to use :

a) An actuation system where monitoring is by comparison of individual lanes. (Majority voting)

b) A system where each lane is monitored individually.

At the present state of the art, secondary actuator systems are almost universally electro-hydraulic consisting of servovalves driving small hydraulic actuators. Thus the number of hydraulic systems available on an aircraft has a powerful effect on the layout of the secondary actuator system. For example, a quadruplex actuator system with a separate hydraulic supply to each lane will fail as follows.

1st failure - 3 lanes vote against 1
Remove faulty lane.

2nd failure - 2 lanes vote against 1
Remove faulty lane.

3rd failure - 1 lane against 1.
System uncontrolled.

Thus we have a 2 failure survival system.

However, suppose our aircraft has only 2 hydraulic supplies and each system supplies 2 lanes of our quadruplex actuator.

Then we may have :

1st failure - hydraulic supply failure - 2 lanes become inoperative

2nd failure - a signalling lane - 1 lane against 1
System uncontrolled.

Thus if we have quadruplex electronics and dual hydraulics some better method than the simple quadruplex actuator must be used.

5. ELECTROHYDRAULIC SECONDARY ACTUATOR SYSTEMS.

The number of options open to the designer when choosing a secondary actuator system is bewildering. For example, in 1974-5, Dowty Boulton Paul working in conjunction with Smiths Industries, and sponsored by the British Ministry of Defence, carried out a study of the problems of interfacing of computers and actuators. In this work 144 layouts were considered and eventually reduced to 7. Out of these 7 only 2 are being seriously considered for aircraft but in the meantime, due to changes in technology, we are aware of at least 5 other systems which did not appear in the 144. In this section we do not intend to deal with vast numbers but only to describe some of today's more promising contenders. We list these according to the number of hydraulic systems on the aircraft.

5.1 Four hydraulic systems.

Military aircraft are not likely to have 4 hydraulic systems unless they are large transports. However, on smaller machines it may be that there could be 2 supplies for the main flying controls and 2 back-up systems. These back-up systems could be variable delivery pumps driven by electric motors so that in normal operation they supplied only the secondary actuators but in emergency they could meet the demands of the main system. As shown in fig.1A four hydraulic supplies allows the use of quadruplex avionics and quadruplex actuation. Fault finding can be carried out by force comparison at the summing point which is at the level of the control valves. With careful design there need be no connections between lanes until the summing point.

5.2 Three hydraulic systems.

Again the third hydraulic system may be obtained from an electrically driven pump and used to back up the main system where necessary. It should be noted that in all layouts, no interconnection of the main hydraulic supplies has been allowed. Possible solutions using three hydraulic supplies are shown in figures 1B to 1G. It will be seen however, that only one of these, namely 1D, can be operated with only 4 computer signals; all other solutions require 5 or 6 signals. In this respect, 3 guaranteed signals from 3 self monitored computers is classed as 6 signals.

There are two solutions using 5 signals. One of these, shown in 1E, is very similar in layout but is a hybrid system in that actuator lanes 1, 2 and 3 operate on a majority voting principle, whereas, 4, 5 represent a monitored system. This layout may appeal to the advocates of dissimilar redundancy. It is also interesting to note that it is also a failure absorption system in that it can survive two failures without pilot action but only if actuator 4, 5 is more powerful than 1, 2 or 3. System 1C, also using 5 signals is a more clear cut failure absorption system. Figures 1B, 1F and 1G show layouts using 6 signals (or 3 monitored signals) differing largely in the way in which the hydraulic supplies are run. 1B and 1F are shown using monitored actuators whilst 1G shows a majority voting system. 1G can be used with monitored actuators if desired.

5.3 Two hydraulic systems.

Figures 2A and 2B show how 4 non guaranteed signals can be used with two hydraulic supplies to produce a two failure survival actuation system. In each case, however, the price is a large number of servovalves. In figure 2A, by using two quadruplex actuators, one to each hydraulic supply, we have a system which will survive 2 electrical failures plus 1 hydraulic failure and thus is a case of overdesign. Figure 2B shows 4 non guaranteed signals operating into 6 monitored actuators. A typical monitored actuator, say 1, 2 could be one in which the actuator is driven by signal 1 and signal 2 is passed through a model and then compared with the output of the actuator. Alternatively, signals 1 and 2 could both drive servovalves. The sum of the outputs would then be used to drive the actuator while the individual outputs would be used for monitoring.

In either case the philosophy of actuator 1, 2, is that if signals 1 and 2 agree the actuator operates, if 1 and 2 disagree it is switched off. Examination of figure 2B will now show that the system will continue to operate after any two failures.

Figures 2C and 2D indicate what can be done if 5 signals are available. These layouts can be compared 2C with 2A and 2D with 2B. In the first case the extra signal has reduced the number of secondary actuators by 2: in the second case 1 less monitored actuator is required.

Three solutions are shown using 6 signals. Figure 2E uses 6 unmonitored actuators on a majority voting layout, figure 2G uses 4 monitored actuators and figure 2F shows a hybrid. At first sight 2E seems to offer no advantages over 2C but closer examination will show that 2E can be used as a failure absorption system whereas 2C can not. To illustrate this, if there were to be a failure of signals 1 and 2 in figure 2C, three actuators out of five would fail and unless these were switched out at the instant of failure the whole channel would be out of control. In 2E however, loss of 2 signals still leaves 4 correct lanes which will dominate the system. Systems 2F and 2G just meet the stipulated requirements. It should be noted however, that in 2F it is essential that the monitored actuators are more powerful than the unmonitored ones.

Figures 2H and 2J show methods of using 4 signals with not more than 4 servovalves. It should be noted, however, that although in both cases the solutions will survive 2 signalling failures or 1 hydraulic plus one signalling failure they will not survive 2 servovalve failures.

In 2H the technique is to pass each electrical signal through one coil of each servovalve and to monitor the coil current. In this way 2 electrical signalling failures can be tolerated quite independently of any hydraulic failures. The secondary actuator outputs are then compared in pairs and any disagreement then results in a pair of actuators being bypassed.

In 2J a slightly different philosophy is used, the four electrical signals are compared and three good ones are taken to drive 3 servovalves. The outputs from the three servovalves are compared by a majority voting technique so that one servovalve can be switched out after a first failure but all are switched off on a second failure. This is not to say, however, that these systems are not acceptable.

5.4 Other factors.

In the previous paragraphs we have looked at many solutions which could be used, all based on a minimum requirement of surviving two failures. The selection of a system philosophy however, depends upon many other factors such as lane matching, weight, cost, power consumption (hydraulic and electrical) and general reliability. There is, in fact, no one solution to suit all aircraft.

6. RELIABILITY

Apart from obtaining integrity through redundancy as described in section 5 the actuator designer, like the avionics designer, is concerned with providing equipment which requires a minimum of unscheduled maintenance. The flying control actuator manufacturer collects information on defects occurring in service and breaks these down in to categories such as dynamic seals, static seals, pipes etc. From this information he is not only able to predict the defect rates for proposed new designs but is able to pick out black spots for remedial action. One of the major defects in simple hydraulic rams is leakage and here the designer may have to improve the sealing characteristics by changing seal materials, changing geometry, or improving the surface finish of sliding parts. When a highly redundant secondary actuator system is introduced, because of the redundancy, there is an adverse effect on defect rates. To illustrate this the table below indicates the percentage contribution to defect rates of various parts of an electrically signalled actuator.

Tandem ram, control valves, bypass valves and stabilisation system	-	32%
Servovalve & solenoid valves	-	45%
Secondary actuators	-	3%
Transducers & connectors	-	20%
		100%

This indicates that in order to obtain sufficient redundancy of the secondary actuation system we have increased the defect rate by 200%. The bulk of this increase is in solenoid and servovalves and so elimination of those items would allow a major increase in reliability. This is becoming feasible and is discussed in section 7.

7. FLY BY WIRE USING LARGE TORQUE MOTORS

7.1 Historical.

Shortly after World War II one of the methods used for controlling hydraulic power by means of electrical signals was to use a torque motor to operate a miniature hydraulic valve. A typical torque-motor would have a peak consumption of about 2 watts of electrical power and would weigh about 300 gms. Torque motors of the Law's relay type were used in missiles for autostabilisation in aircraft and on the Tay engined Viscount which in 1957 was the first aircraft to fly controlled by electric signals. At the present time the Buccaneer, V.C.10 and Jaguar are using torque motors in this way.

The electrohydraulic servovalve had the advantage of only using about 50 milliwatts of electrical power which was an important factor at the time bearing in mind the state of the art of amplifier design. The servovalve was competitive in size and weight, the whole valve being smaller than the torque motor alone, but it did consume an amount of hydraulic power, say the equivalent of 350 watts in a 3000 psi system.

The use of torque motors, or force motors, in fly by wire actuation systems is being considered with renewed interest because of the following factors

a) With increasing numbers of servovalves being used, hydraulic power losses can be 10 kilowatts per aircraft.

b) With the advent of new magnetic materials, torque motor efficiency can be improved.

c) Developments in electronics mean that high output amplifiers are now feasible.

d) Because of the basic simplicity, torque motors offer a chance to improve reliability.

e) Increased redundancy can be obtained by increasing the number of windings which involves a very small weight penalty.

Studies at present indicate that for a simplex system a torque motor/valve system is not competitive with the equivalent servovalve system. But, in say a quadruplex layout, a large torque motor can be competitive with four servovalves and their actuators especially when the hydraulic power and cost is taken into consideration.

7.2 Torque motor designs.

At the present time there are three designs of motor being considered,

a) The moving iron torque motor.

b) The moving iron torque/force motor.

c) A moving coil force motor.

Consideration is also being given to other designs and also to stepper motors but it is felt that these require more development.

7.2.1 The moving iron torque motor.

A section through a typical torque motor is shown in fig.3A. The magnetic circuit shown by solid lines and arrows acts as a magnetic centring spring to give basic stiffness. Passing current through the coils gives rise to an asymmetric magnetic flux as shown by the dotted lines and arrows. As can be seen, this second magnetic flux neutralizes that from the permanent magnet on one side and reinforces that on the other side thus applying a torque to the armature.

7.2.2 The moving iron torque/force motor.

This is shown in fig. 3B and it will be seen that this is a large version of the electromagnetic part of a flapper nozzle servovalve. Although the armature effectively rotates, the mechanical output is linear which is useful for driving a linear valve.

7.2.3 The moving coil force motor.

This is shown in fig. 3C. It will be seen that a coil moves axially in a samarium-cobalt magnet and again the output is suitable for driving a linear valve. This motor has the advantage that there is no iron circuit to saturate and so very high outputs can be obtained for a short period of time.

7.3 The effect of torque motors on system layout.

The use of a large torque motor allows the designer to choose his system philosophy without considering the number of hydraulic supplies available. The main consideration becomes the number of signals available from the computers and whether these are monitored and if it is intended to detect and remove failed signals.

7.3.1 Five computer system.

Five is not normally a popular number in aircraft but if the state of the art is such that we wish to survive two failures and we also wish to avoid the complication of selecting out failed signals then five is the minimum number of non-guaranteed signals acceptable. That is ;

1st failure - four signals overcome one

2nd failure - three signals overcome two.

The worst case is when the two failed signals are hardovers in the same direction. If they are in opposite directions, or if one is a null failure (e.g. due to loss of an electrical supply) then the system may well survive a further fault.

Figure 4A indicates such a system. Although this philosophy requires five independent signals which may mean five computers there is now no need for any switch-out mechanisms and, in fact, no need for any form of failure detection in flight unless a pilot warning is required. If failure indication is a requirement this can be done by comparison of torque motor coil currents and this could be used in a ground checkout procedure to detect dormant failures.

7.3.2 Four computer system.

With a four computer system it is necessary to remove failed lanes as they occur and so it is essential to compare coil currents and to introduce some form of disconnect mechanism. It should be noted that coil current comparison can be used to identify overall failures, be they in torque motor, computer or sensors. A typical system is indicated in fig. 4B.

7.3.3 Three computer systems.

In a three computer system the first fault can be isolated by cross comparison of lanes (majority voting). To isolate the second failure, however, requires each of the two remaining lanes to be checked independently. This amounts to each lane having a good self checking facility. It is for the computer designer to decide how this can be achieved; in the ultimate it could amount to a 6 computer system but certainly serious consideration has to be given as to when 3 self checking computers become more complex than a quadruplex majority voting system. A typical layout is shown in fig. 4C.

7.4 Monitoring of torque motor systems.

Although the diagrams 4A, 4B and 4C show a single torque motor it may well be that for layout convenience, two motors are used. In either case it may be desired to feed all coils individually as shown in fig. 4D. In this case monitoring is by comparison of all eight coil currents but the coils are disconnected in pairs.

8. SERVOVALVES VERSUS TORQUEMOTORS.

In section 5 it was shown how, using electrohydraulic servovalves and secondary actuators, the system was very dependent upon the number of hydraulic supplies available. In section 6 it was indicated how the electrohydraulic valves made a large contribution to the defect rate. As shown in section 7, the use of one or two large torque or force motors frees the designer from the constraint of hydraulic supplies and allows an easy interface between electronics and hydraulics. It also is a rugged device with a low defect rate.

On the face of this it could be asked why servovalves should be considered but as usual there are two sides to the story, servovalves require less drive current, have a better frequency response and also, with a servovalve - secondary actuator system there are much larger reserves of power to overcome any tendency to seizing in the main control valves.

To summarize:-

Advantages of torque or force motors.

- Simple
- No waste of hydraulic power
- Interface with electronic systems simplified
- Relatively cheap
- Reliable

Advantages of servovalve - secondary actuator systems.

- Low electrical power required
- High response
- Ample force for main valve drive
- Compact

Studies so far show the torque motor once again becoming a serious competitor to the servovalve. But there is no doubt that the designer of actuation systems will be faced with many 'trade off' decisions for a long time to come.

9. TERMS.

The following terms have been used in the preceding paragraphs.

a) Channel.

A channel is the complete path of signal communication for a control.

b) Lane.

A lane is a sub-channel. For example, a triplex channel would contain 3 lanes.

c) Primary actuator.

This is the power actuator which drives the control surface.

d) Secondary actuator.

This is a small actuation system designed to drive the primary actuator control valve.

e) Failure absorption system.

A multiplex system which will continue operating after the stipulated number of failures without removing any failed lanes.

f) Failure rejection.

A multiplex system in which failed lanes must be detected and switched out before a further failure occurs.

g) Comparison monitored.

A system where the outputs of the different lanes are compared (majority voting).

h) Lane monitored.

A system where each lane is monitored independently of other lanes.

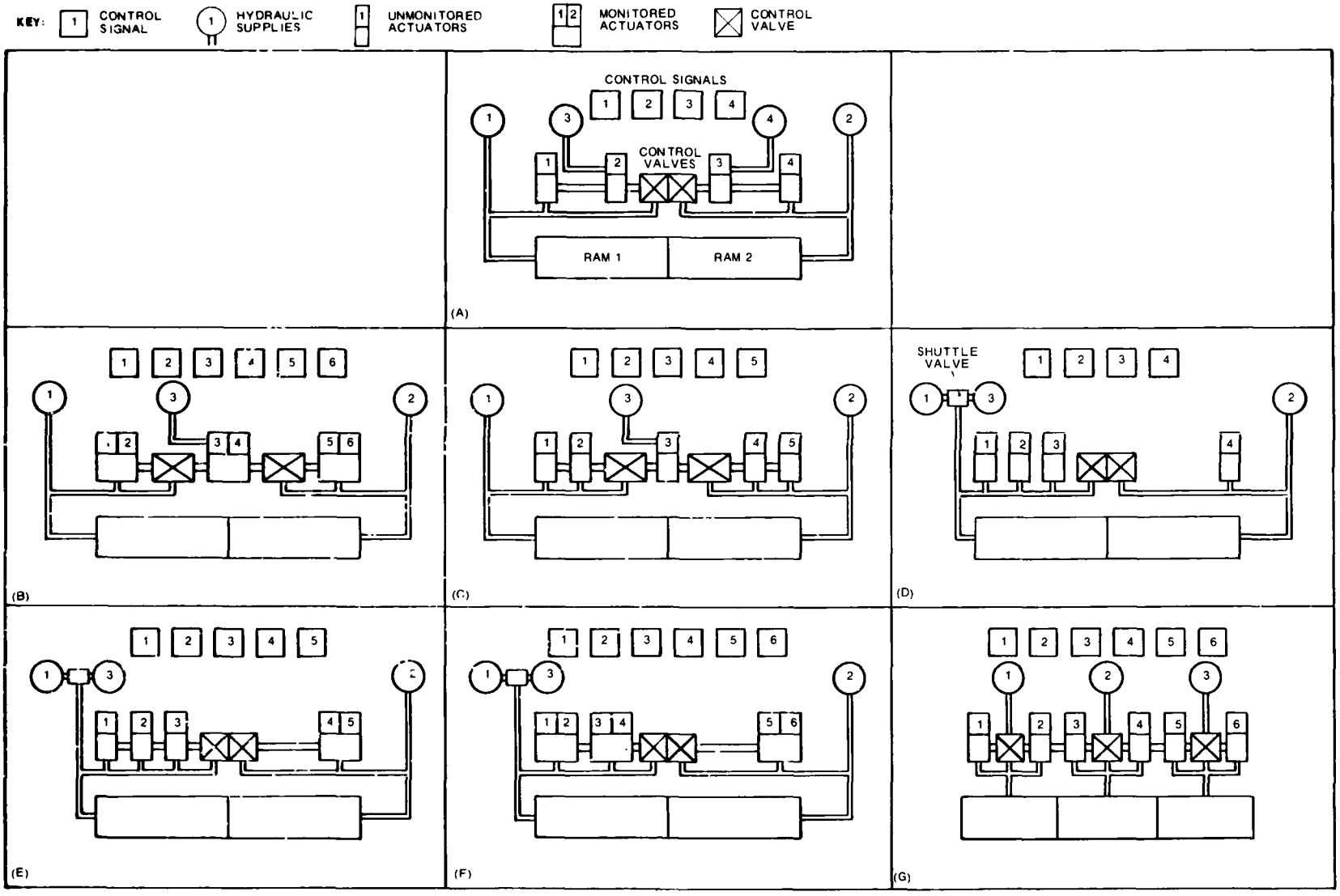


FIGURE 1. CONTROL SYSTEMS USING 3 OR 4 HYDRAULIC SUPPLIES

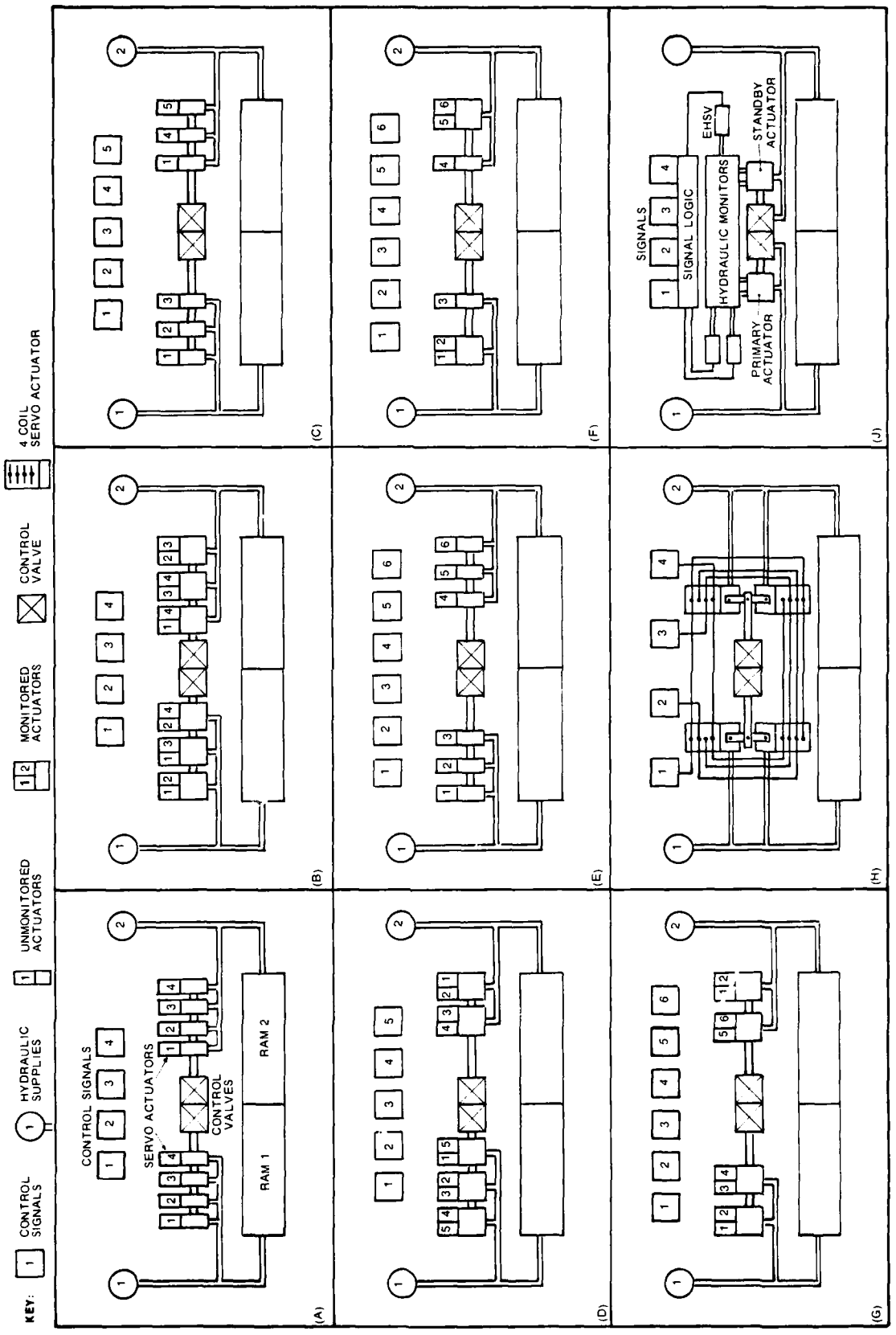


FIGURE 2. CONTROL SYSTEMS USING 2 HYDRAULIC SUPPLIES

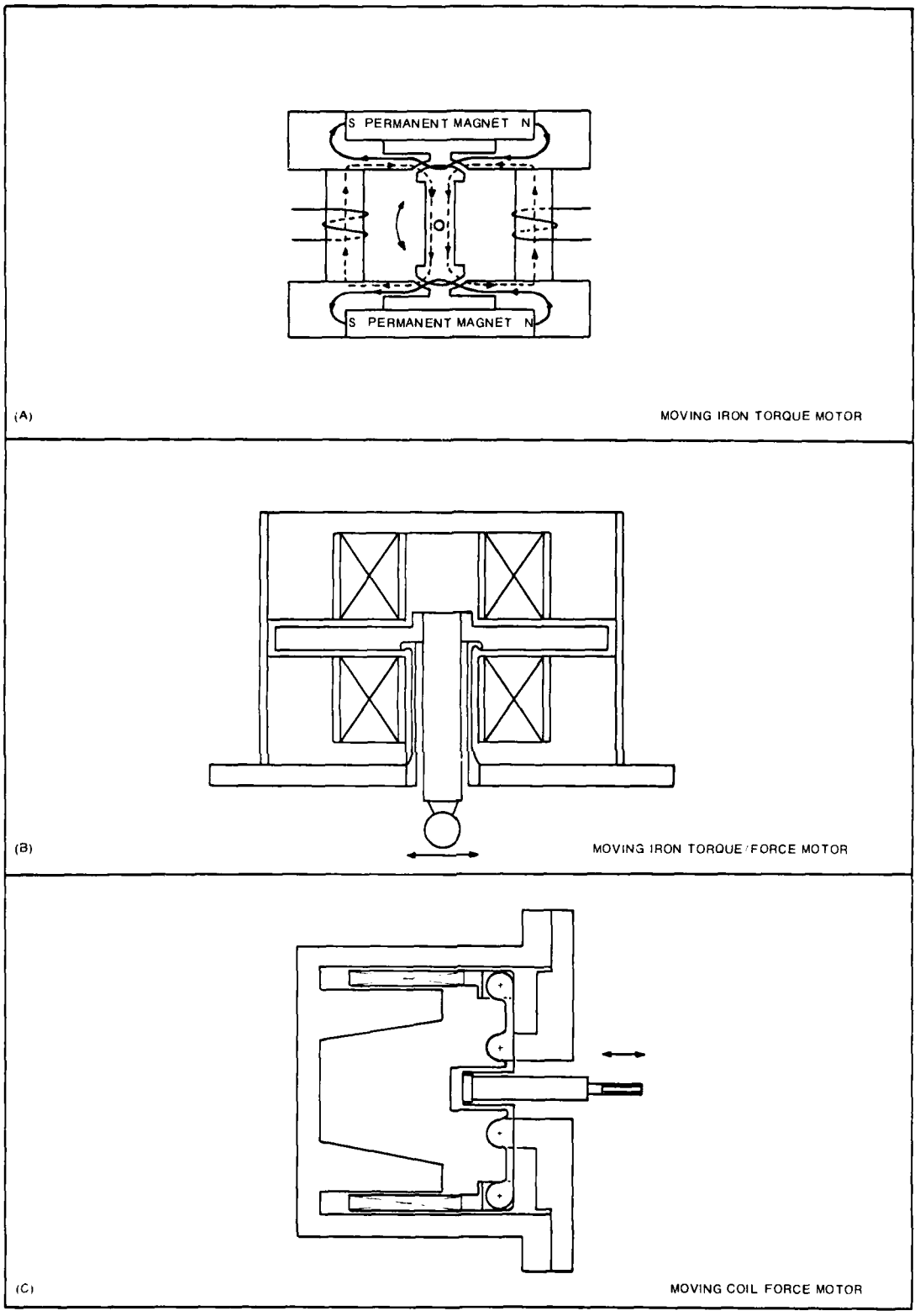


FIGURE 3. TORQUE MOTORS

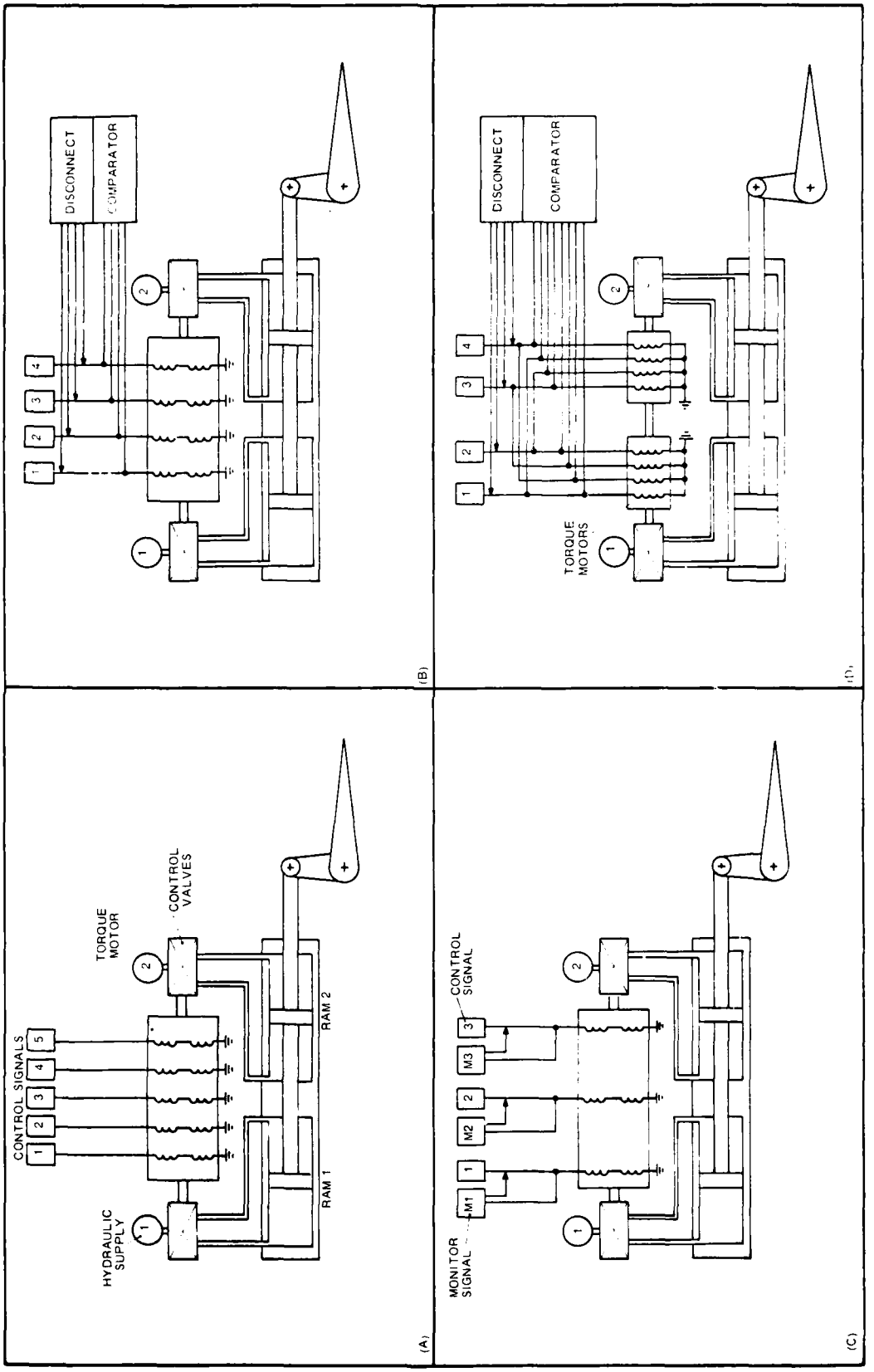


FIGURE 4. CONTROL SYSTEMS USING TORQUE MOTORS

IMPACTS OF TECHNOLOGIES SELECTED ON THE RELIABILITY AND OPERATIONAL AVAILABILITY OF EQUIPMENTS. COST CONSIDERATIONS

J.M. GIRARD
M. GIRAUD
Electronique Marcel DASSAULT
55, Quai Carnot - SAINT CLOUD
FRANCE

SUMMARY

This paper aims to propose a simple way allowing the manufacturer to appreciate the merits of technological variants through a single criterion "V", once an equipment baseline version has been designed and quoted.

To do that, submodels and transfer coefficients are needed to approximate globally the evolution of effectiveness parameters (like reliability, maintainability) and those of acquisition cost. Also to have an idea about how unit cost of hybrids and custom designed LSI versions compare, according to serial length and in relation to a conventional (D.I.L.) one.

This exercise was undertaken at Electronique Marcel DASSAULT and centered around objective computation of "V" factor for three different types of avionics (airborne digital computer, Doppler navigational Radar and search and rescue beacon) each one being considered in three versions. Results are given hereafter for appraisal.

1. INTRODUCTION

At proposal stage, the equipment's conformation to a required need, induces careful trade-offs selections between the - often antagonistic-constraints of the specifications such as : weight, volume, performance levels, power consumption, reliability on mission profile, safety factors, operational maintainability, time delays for development and so on. Once settled upon moreover, the design should fit into a minimum financial envelope, relative to other bidders or to a fixed budget.

Equipment's adequation restricted to the single matching "unit cost-performance" is no longer acceptable if it is intended for a long period of use, because such obtained reliability is no more than a mere statement of fact - making life bitter, later on, for the customer.

Obviously it is in the initial technological choice that concordance of the design to the spec's constraints shall (or not) find a solution in a concrete form. Therefore it seems necessary to provide the decision maker with an evaluation "tool" which permits the most objective selection among possible technologies for his particular problem. That means :

- setting a synthetic criterion basis on which all parameters have been accounted for,
- predetermining the transfer's consequences from a conventional technological reference version (I.C., discrete components) towards more sophisticated ones - like hybrids or custom designed LSI - merely with reliability, maintainability and cost submodels.

We shall examine :

- the methodology used for "V" factor quantification,
- approximated transfer formulations of the parameters,
- some examples of application,
- and finally what results one can expect of this approach, as well as the present limits and what work remains to be done for improvement.

2. METHODOLOGY TO CHARACTERIZE THE VALUE "V" (Cost Effectiveness) OF A SYSTEM

The classical form of "V" is due to W.S.E.I.A.C. (1), it states :

$$V = \frac{\vec{A}.(R).(P).\vec{W}}{\vec{S}.\vec{I}} \qquad (1)$$

Where :
- V is a scalar of dimension : Probability of effectiveness/cost unit.
- $\vec{A}$ line vector. Availability related to the n possible states of the system
- (R) square (n x n) matrix : dependability of the n states (conditional reliability)
- (P) (m x n) matrix : Probability of m performances acquisition from the n possible states
- $\vec{W}$ column vector : weighting of the m performances
- $\vec{S}$ line vector of different costs (acquisition, spares, maintenance gear, operational cost, etc ..)
- $\vec{I}$ column vector : weighting of S terms.

Formulation of equation (1) requires some simplifications, if we wish to use it practically for comparative approximation purposes. It needs, above all, submodels of prediction to weigh in time the consequences of an eventual choice.

The following remarks are for the sake of simplification.

2.1. Pertinency at equipment level : expression of "V"

First, (R) - matrix of dependability - can reduce to scalar R (Reliability) if the three following conditions are met :

a) the equipment has exclusively one of two states (good/no good)
b) it cannot be repaired during mission time
c) it is good at the start of the mission

A great deal of avionics - taken separately - belong to this type of device (no possible reconfiguration)or can be broken down to equivalent "boxes" of this class. They are also fitted with a self contained built-in-test facility which allows comprehensive checking before take-off. In this case of "serial" type of structure the availability steady state value yields further simplification to :

$A = \frac{MTBF}{MTBF+MTTR}$ and when it is agreed to consider a probability of performance (i.e acquisition of a signal S, in given signal/signal + noise conditions)or a synthesis of performances expressed on a mix, then (P) also comes to a more tractable scalar form : P.

NB : If performance acquisition is not written in term of probability, one can always express the capability level by a dimensionless ratio (percentile) relatively to an ideal reference that shall be attained in foreseeable future (when t ↗ or if the number of trials ↗). Such a normalized P is homogeneous to a probability.

Last, the denominator of "V" shall figure the cost of ownership over the equipment's life cycle, but cost conversion through I coefficients of weight, volume, power consumption parameters is not necessary here. At equipment level these variables ought to be treated as performances like stated above. So it resumes to acquisition cost C_A and planned life expenditures $C_E(t)$, both normalized to the reference version.

We shall reckon up : the probability of mission effectiveness E (t).

$$E(t) = R(t_1) \cdot A \cdot P \qquad (2)$$ where t_1 is mission time

the normalized cost Cg(t) over the life cycle t_2

$$Cg(t) = C_A + C_E(t_2) \qquad (3)$$

the pertinency V (factor of merit) : mission effectiveness per life cycle cost units.

$$V = \frac{E(t)}{Cg(t)} \qquad (4)$$

"V" is the criterion on which should be assessed the decision ; when "V" is maximum the technology chosen is the best for the required need. One notices that normalizing the denominator to the reference version infers that "V" is not a probability anymore.

2.2. Problems related to acquisition of the transfer coefficients

2.2.1. Reliability

Reliability can be approximated in some "global" manner, once a standard set of circuits has been selected in which integral technological transfer is feasible for λ ratios computation from MIL-STD-HDBK 217B models. Those will be applied to the particular equipment baseline.

Unfortunately there are no such models or sufficient data available yet for storage failure rate prediction of hybrids or LSI, so no λ ratio is given in this case.

Transfer coefficients have been worked out from a Mixed Set of Schematics called MSS which single finality is to provide entirely convertible circuitry - logical and analog - from conventional to hybrids or LSI monolithic types on hybrids.

Therefore :

- MSS should present similar performances whatever the version.
- In fact it has none definite, as an equipment.
- No general relationships other than λ ratios can be derived out of it because maintenance of such "equipment" is irrelevant and unit's cost ratios would restrict to some arbitrary production rate.

The reliability transfer coefficients are simply :

$$K = \frac{\lambda \text{ MSS version A}}{\lambda \text{ MSS version B/C/D}} = \frac{\text{MTBF MSS version B/C/D}}{\text{MTBF MSS version A}} \qquad (5)$$ λ is the failure rate

Where : A is the conventional technology (discrete components or SSI/MSI on printed boards)
B is encapsulated thick film hybrids (capacitors chips, transistors and IC dies)
C/D are custom designed LSI monolithic chips (bipolar and CMOS respectively)

K has been computed for temperatures ranging from 10° to 110°C (with 20° increments) in 4 typical environments of MIL-STD-HDBK 217B. Hybrid model used is from March 78 revision.

5 different values of K are given each time, they allow computations of Reliability transfer from: logical A to B, C or D versions and linear A to B or C versions.

Table 1 - sums up the results - and the graphs 1 to 5 - provide for interpolation if necessary.

2.2.2. Maintainability

The MTTR of an equipment is a more difficult variable to forecast than MTBF, because it is not only a function of the equipment structure but also of human intervention ; often it applies to specific cases where data is lacking : Maintainability being tightly related to the logistic itself, to the modularity achieved, to the type of servicing and the applicable degree of MIL-STD-470, etc ...

Therefore :

- Availability in its asymptotic form - MTBF/(MTBF + MTTR) - is partially derived from the predicted reliability but transfer coefficients over MTTR cannot be found so easily : they depend on what is exactly the maintainability concept we are interested in. Is it just the time for technical repair ? Or does it include the consequences of the maintenance strategy ? (stocks size, servicing facilities, manpower needed and so on).

In our case - military equipments - due to a particular maintenance structure, the technical part of MTTR is of second order compared to other time contributions (administrative, transport, etc ...) when repair has to be done by the manufacturer.

Anyhow the maintainability M transfer coefficient is with the usual notation,

$$M = \frac{\mu \text{ version A}}{\mu \text{ version B/C}} = \frac{\text{MTTR B/C}}{\text{MTTR A}} \qquad (6)$$

μ is the repair rate

2.2.3. Costs

Costs are function of the present time, the manufacturer's know-how etc ... So we emphasize that all cost coefficients shown on slides are merely for illustration purpose and may in no circumstances been applied elsewhere, although they can be calculated in a similar manner by other manufacturers with their own references.

In addition there is no attempt to modelize $C_E(t)$ from a reference version directly, not knowing what the maintenance policy is going to be. So models deal mainly with acquisition cost C_A covering : the design phase, the development phase and the production phase.

Only the last one brings a cost that can be partially forecast in a similar manner as K, once sufficient statistics have been accumulated over the different technologies. Transfer of development costs can only be expressed in terms of differences Δ ; not by coefficients.

Transfer of design costs is a matter of hazard. It is essentially dependent on the technical quality of the teams involved, so we shall take it as a constant in the model.

We now come to the submodels.

3. EFFECTIVENESS SUBMODELS DERIVED FROM A TECHNOLOGICAL BASELINE VERSION

3.1. Reliability

We suppose an identical Reliability "tree" between versions and assume the previous failure rate knowledge of the conventional one ; so we know λ_A and we want to approximate what would be λ_B for instance.

In the general case the equipment considered is not entirely convertible from conventional to another technology, so :

. the equipment's designer will define in the items list :

- the envelope of the non hybridable remaining components : α
- the envelope of those that disappear (if any) : β
- the envelope of new components which will appear (if any) : γ

. the reliability engineer can then derive λ_α and λ_γ and then calculate with the K table :

$$\lambda_B = \frac{\lambda_A - (\lambda_\alpha + \lambda_\beta)}{K!} + \lambda_\alpha + \lambda_\gamma \qquad (7)$$

NB :

. the term $\frac{\lambda_A - (\lambda_\alpha + \lambda_\beta)}{K1}$ figures the contribution of hybridable part to the total λ_B

. equation (7) is the simpliest form applicable : that is for example if the entire design is from logical type. If not, one should just add to (7) a similar expression with K4 at the denominator, to take care of linear parts.

. the K factor provides for a 10°C internal temperature increase in hybrids logic and a 20°C internal increase for analog type.

. the K factor given assumes : a mean density of 20 dies (transistors and ICs) and 5 attached components for a 1' x 1' alumine substrate with 42 I/O leads max in a logical type or,

a mean desity of 80 components in analog case (60 % thick film resistors, 20 % chips capacitors and 20 % IC, transistors and diodes) on same size substrate.

All hybrids have $\pi Q = 2$ (just under MIL-STD-217 B level B) ; elsewere level C.

In version C or D (LSI) the electrical schematic will be very different of course, but nevertheless the approach remains the same using K_2 or K_3 at the denominator.

In this case (logic diagram) the K factor assumes : 4000 transistors CMOS integrated on a 20 mm2 chip with 60 I/O available or 1000 bipolar on a 20 mm2 chip with 40 I/O.

For linear type, integration will be of the order of 120 bipolar transistors on a 10 mm2 monolithic chip with 32 I/O. Then K_5 shall be used.

In the Large Scale Integration process of logical circuits, we consider as a corresponding basis relatively to conventional ICs, the following ratios :

Logical Function	Conventional IC (T2L)	LSI (CMOS)	LSI (bipolar)
N inputs Nand Gate	4 transistors	2N transistors	3 transistors
N inputs Nor gate	2(N + 1) transistors	2N transistors	4 transistors
D Flip Flop	24 transistors	34 transistors	18 transistors
J-K Flip Flop	32 transistors	36 transistors	27 transistors

3.2. Availability

At paragraph 2.2.2, we take the asymptotic form of A, using a similar approach we find out what A becomes in a transfer process, once the conventional $MTTR_A$ is known and the predicted $MTBF_{B/C}$ given by (7).

We state that the new $MTTR_{B/C}$ results from 2 parts :

- One has for origin the repair time due to the non hybridable (or non integrable) remaining components of failure rate λ_α . Its corresponding MTTR is approximated by the original $MTTR_A$ to the proportion $\lambda_\alpha / \lambda_{B/C}$

- Similary, the second part of $MTTR_{B/C}$ originates from the design evolutive portion. We also assume a proportionality factor to the failure rate.

This gives :

$$\frac{K\lambda_\gamma + \lambda_A - (\lambda_\alpha + \lambda_\beta)}{K\lambda_{B/C}} = y$$

y is the weighing coefficient for the evolutive $MTTR_{B/C}$

Then if M is known somehow :

$$A_{B/C} = \frac{MTBF_{B/C}}{MTBF_{B/C} + MTTR_A \left(\frac{\lambda_\alpha}{\lambda_{B/C}} + y \cdot M\right)} \qquad (8)$$

applicable if MTTR variations are due to technical reasons only (modularity, automatic testing, etc ...) and not if stocks size are involved or other parameters.

We also point out that if technical repair times are of second order relatively to transport/ administrative delays, we can assume then M = 1 and use (8) for a reasonable approximation.

NB : For comparative evaluation of "V" factor it is sufficient to state an arbitrary value of A per mission (say 90 or 99 %) and infer the number of spares associated with each version for cost computation - see paragraph 4.2.2. -

3.3. Performance

The problem restricts to selection and normalization of the proper mix for transfer computation (see paragraph 4.1.)

3.4. Cost submodels derivation

The acquisition cost is an investment, the exploitation cost is the return on investment. Figures 7 and 8 schematize the cost problem. We notice that acquisition cost C_A defines the ordinate at time's origin of the life cycle cost Cg(t).

For the supplier, whatever the equipment version , total expenditure over the manufacturing process should write :

$$C_e = C_{\emptyset 1} + (C_{\emptyset 2} + C_{\emptyset 3})\,\tau \qquad (9)$$

with $C_{\emptyset 1}$ = Cost of design phase
$C_{\emptyset 2}$ = Cost of development phase
$C_{\emptyset 3}$ = Cost of production phase
τ = Time varying index

NB : CØ3 = N.Cu if N units are to be ordered.

From internal cost statistics collected at EMD on likewise applications predictions submodels were derived for estimation of development cost differences Δ and production cost ratios ρ suitable to versions B and C when version A was defined.

giving :
$$C'_e = C_{\emptyset 1} + (C_{\emptyset 2} + \Delta C_{\emptyset 2}) + \rho C_{\emptyset 3} \qquad (10)$$

Obviously the method aims mainly to approximate tendancy for rapid appraisal of cost brackets rather than precise statements (Δ and ρ data relates to a particular time : 1977). So models validity is subject to constant evolution. Nevertheless they brought satisfactory results when applied to electronics of similar type that excludes at present high power circuitry or microwave filters and amplifiers.

3.4.1. Development cost differences

They are given in terms of proportions to man-hours and outlays expressed in financial units on such contributing parameters as :

. electrical parts (hybrids or LSI chips)	: ΔC1	
. multilayer printed boards	: ΔC2	
. design and implementation of automatic testing process	: ΔC3	see slides
. inner interconnection	: ΔC4	
. engineering of mechanical envelope	: ΔC5	

3.4.2. Production cost ratios

The reference technological version is based on discrete components and Dual In Line ICs. The production cost of a variant shall be the sum of the almost constant part of the cost (ie = mechanical components acquisition, wiring and assembly of cards, quality control inspection, intermediate and final testing, packaging and shipping, etc ...) and of the evolutive part contribution namely given by cost ratios of :

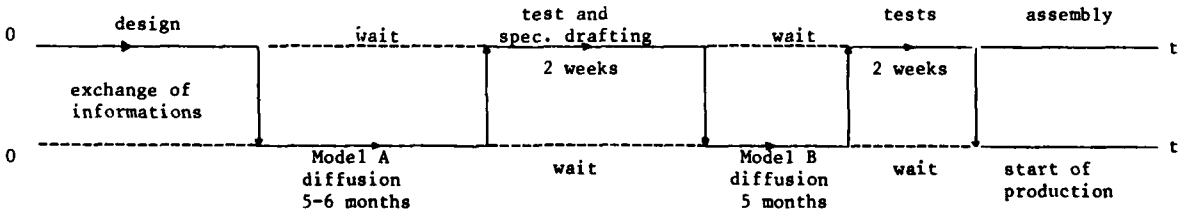
. electrical parts (hybrids or LSI)	: ρC1	
. multilayer printed boards	: ρC2	see slides
. inner interconnection	: ρC3	

As an example of unit cost ratio brackets prediction see fig. 6. Where 3 versions have been plotted versus serial length. The B and C cost ratios were all along normalized to the production cost of the DIL reference version, which remains then equal to one.

3.4.3. The time problem, τ estimation

When contemplating C or D variants for development and production, it is essential to foresee what kind of time delays are involved. They might be inconsistent with the required procurement deadline.

As an example we consider as optimum the following time schedule where the upper t axis is the designer's time table and the lower one the LSI chip manufacturer's time table.



The experience proves that it is very likely that some modifications occur between diffusions of model A and B, this implies a shift of about three months in the time schedule. So

. if the schematics are good : 11 months $<$ τ $<$ 12 months
. if one more diffusion is needed : $\simeq$ 15 months
. if other "snags" happen : $\simeq$ 18 months

fig. 6 - shows what are the cost brackets induced by such rework when needed in versions B (hybrids) or C (LSI). Shaded area.

3.4.4. Exploitation Cost $C_E(t)$

It is a time variable function. It covers fixed and recurring costs associated with maintenance and utilisation of the equipment over its life cycle. No model is given for transfer comparison because it is essentially dependent on the maintenance policy chosen. The number of spares are computed through Poisson's law if the equipment is not subject to wear-out (then it has a constant λ like most electronics) generally speaking variants with high percentage of LSI chips exhibit lower operating costs than others if modules size has been optimized to allow a discard policy. Encapsulated hybrids can be repaired eventually, but a very limited number of times - if it is worth it -

4. APPLICATION OF THE METHOD

Among three avionics presently designed or produced at EMD, three different technological versions were imagined in each case.

The aim being twofold :

- appreciation of the technological impacts related to the choice through "V" factor evaluation by the E.P.A.C. system (2)
- comparison of acquired results, via this comprehensive computer assisted method, to those given by the predicting models described earlier. That is through a selective "global" application of the transfer coefficients onto the known reference baseline.

Of course none of the variants considered here is an "off the shelf" existing type ever produced at EMD. They are solely designed for comparative purpose along the exercise.

4.1. Equipment characteristics for V evaluation

Their technological composition is briefly summarized on tables 2, 3 and 4.

4.1.1. General purpose digital computers

- Word format is 16 or 32 bits. All versions features the same set of instructions (120) although execution time varies from one to another. Core memory size has been reduced here to 8 K words whatever variant.

- The reliability organization presents no redundant sub units. Life time t_2 profile assumed is :
  25°C Storage ; ground fixed 89.6 % t2
  70°C Operation ; ground fixed 6.2 % t2 t2 = 10 years
  70°C Operation ; aircraft inhabited zone 4.2 % t2

- Flight time per mission t1 = 1 hour

- For V factor evaluation availability is set at 99 %, but also computed separately via EPAC and submodels for comparisons over 20 equipments.

- Functional structure is the following :

  . A.L.U. (Arithmetic and Logical Unit) : 2 accumulators and 16 adressable registers, data exchange is through buses R, S, T (16 bits)

  . Control unit (microprogrammable) includes a 1024 words memory command, the mappers and rythming device.

  . Core memory capacity is 8 K (16 bits words + 2 parity bits) address system allows for 64 K extension.

  . I/O unit exchanges parallel 16 bits data with ALU and through serial redundant buses with peripherals.

  . Power supply DC-DC regulated converter isolating the computer supply from the 28V aircraft mains which are under permanent survey for interrupts generation in case of power failure/ restore.

- Performances : the efficiency of the computer is currently defined on a standard "mix" which assumes a fixed percentage of arithmetic kilo-operations/sec. (for a given memory size) this performance is P1.

As in that case, memory is constant and implemented in the same technology (cores), the memory volume has been replaced by the computer's physical volume defining performance P2.

Also, as we are unable to account for the electrical power consumed in operational cost evaluation, this parameter has been assimilated to a performance P3.

Therefore the cost effectiveness (V) has been computed through successive normalization of the performances and cost parameters relative to version C using P1, P1 x P2 and P1 x P2 x P3 as performance factors, if so desired. See table 5.

4.1.2. Radars

Transmit and receive Continuous Wave (C.W) in Ke band (13.325 MHz)

- Flight time / mission t1 = 1 hour
- Same life time profile t2 as computers but flight environment is uninhabited zone 60°C
- Same reliability structure = serial
- Availability fixed at 99 % for V evaluation, but computed separately for comparison by EPAC and submodel 8 (see paragraph 4.2.2.) over a 20 equipments set.
- Performance is expressed by successive account of parameters normalized to version C such as P1: probability of aberrant speeds detection when flying over quiet sea.

P1 x P2	:	P1	x	weight
P1 x P2 x P3	:	P1	x	P2 x power consumption

- Functional structure :

Basically designs of version 1 and 2 are similar but version 3 features a completely different approach.

. Aerials : 2 separate side by side fixed antennas of Y configuration for transmission and reception of 3 simultaneous beams (versions 1 and 2) or sequentially along those axes (in version 3)

. Microwave sources : version 1 has 2 crystals plus separate frequency multipliers channels (varactors) for front beams (13.325 MHz) and rear beam (13.314 MHz).
version 2 uses 2 Gunn diodes as references on same frequencies,
version 3 features only one 13.225 MHz channel, the beams being sequentially transmitted and received.

. Signal processing : needs 3 separate channels in version 1 and 2 but a single sequential processing unit in version 3 via a Pin diodes commutator ; the number of components attached is drastically reduced. Moreover data is then processed in two phases : linearly first then numerically through a fast microprocessor (type 10800). This reduces again the number of items and consequently acquisition and operational costs (through the better reliability figure). It also draws some significant differences on weight and power consumption performance parameters.

> Therefore on this equipment."V"factor should be quite different according to variants considered
> . Application of predicting models based on a reference version is much more dubious.

4.1.3. Beacons

Transmit and receive phone-modulation over 2 separate channels (243 and 282.8 MHz) but main purpose is transmission of beacon type modulated signals on 243 MHz. Those beacons being normally used in critical situation, life time profile is quite different from other equipments. We considered :

25°C	ground fixed ; storage	95.9% t2
25°C	aircraft inhabited ; storage	4 % t2
25°C	test time on ground fixed	0.1% t2

t2 = 10 years

- Flight time per mission t1 = 1 hour (storage)
- Reliability structure = serial
- Availability fixed at 99 % for V evaluation
- Performances evaluation has been restricted to comprehensiveness of the built-in self test used before take-off (for security reasons). The characteristics of the test parameters are the following (in relative importance) with version C as reference for normalization :

Beacon transmission parameters :

. Power transmitted	: 33 %	P1
. F. stability	: 33 %	
. Modulation depth	: 33 %	

Phone reception parameter :

. Sensitivity	: 1 %	

This time, design of version 1 differs from 2 and 3 (which are identical except for technological considerations).

In version 1 frequencies are derived through 2 stages of crystal source multipliers ; in the others, source signals are divided at a phase locked VCO output through a referenced comparator.

> This discrepancy has drawn problems in using the predicting models ; but V biasing is insignificant this time, regarding versions B and C, relatively to each other.

Results of V factor evaluation are presented on table 5.

4.2. Comparisons of parameters evaluation via EPAC and the models

They are restricted here to specific cases where the use of transfer coefficient from the reference version makes sense. That is if the design is unaltered from reference to variants (apart from technological "fall out").

We shall therefore mainly use the computer example.

4.2.1. Reliability

In the case of computer variant B (hybrid) using EPAC results of reference A (conventional) :

. Reliability of power supply unit has been predicted within $\pm$ 3 % of EPAC's value B in all environments, using K4 coefficient and model 7.

. Reliability of arithmetic and logical unit has been predicted within $\pm$ 5 % in all environments, using K1 coefficient and model 7.

. Reliability of Radar version B could have been predicted by the model but this was not worth it, because percentage of hybrids was too low.

. Reliability of hybrid or LSI beacon version had no homogeneous reference to use. It was also concerning storage conditions with no possible transfer.

4.2.2. Availability

. Availability of a 20 computers set was computed through EPAC for a given number of spares :

	Version A	Version B	Version C
Spares number	12	11	9
Availability	0,990	0,995	0,994

. Using $MTTR_A$ result given by EPAC and setting M coefficient equal to 1 one finds with model 8.

Availability	0,990	0,992	0,993

. For 20 radars we found by EPAC.

	Version A	Version B	Version C
Spares number	10	5	4
Availability	0,998	0,998	0,999

. Using $MTTR_A$ result given by EPAC and M coefficient equal to 1 in model 8.
Availability of versions B and C comes to 0.9989 and 0.999 respectively.

4.2.3. Production cost

Using the various ρ cost ratios mentioned at paragraph 3.4.2. (on electrical parts, multilayer printed boards and inner interconnection) we tried to find out the production cost of the hybrid version B of the computer's I/O section, control section and arithmetic and logical unit (ALU), knowing the cost of the reference version A.

NB : This represents over 50 % of the total (core memory was no point, being invariant).

The parameters of the reference were :

. Electrical parts (539 DIL + 226 transistors + 18 PROM) : 215 financial units
. 5 (285 x 140 mm) 8 layers printed boards : 200 financial units
. 5 interconnections : 6 financial units

We determined by the submodels : the number of hybrids modules assuming a mean density of 20 dies per module.

$$\frac{(539 - 18) + 22}{20} = 37 \text{ (1' x 1") hybrids}$$

. the number of printed boards (same size) and the number of layers per board assuming 4 hybrid modules / usable dm2 for an 8 layers board.
In this case usable board surface is 3 dm2 giving 12 hybrids per 8 layers board. So 3 cards at least are needed : 2 eight layers and one 12 layers to take account of the remaining non integrable components (18 PROM + some capacitors).

. Interconnection is invariant for 8 layers boards (1 connector/card) but we shall use 2 connectors on the 12 layers board.

Then using the adequate ρ transfer coefficient we found.

C1 electrical parts : 215 x 0,73 = 157 financial units

C2 8 layers boards : $\frac{200}{5}$ x 2 = 80 financial units (40 F.U./board)

12 layers boards : 1 x 4 x 2.25 = 90 financial units

C3 interconnection : 6 x $\frac{4}{5}$ = 4.8 financial units that is Σ = 332 financial units

Which is 5.5 % lower than the exact production price standing at 351 financial units

NB : The slight error is mainly due to application of ρ C1 coefficient (.73) to the total of electrical parts where in fact the 18 PROMS are not hybridable components.

5. CONCLUSIONS

At first sight V results seem to be merely dependent upon performance and cost parameters ; this is partially due to the relatively short time / mission (t1) considered ; also because availability was uniformly set at 99 %. But this is only apparent since MTBF figures determine in each case the number of spares needed over t2 therefore the exploitation cost CE (t2). If we had to estimate V for a synchronous satellite (for which t2 = t1) then the reliability parameter should be much more contributing at the numerator than at denominator, where repairs are irrelevant in this case. So V is quite a flexible estimator for the decision maker.

As regards to the predicting models, it appears that satisfactory results can be encountered if the basic principle of the design remains roughly unaltered by the transfer ; if not, distortions will obviously occur.

But they provide a helpful time saving if the proportion of technological evolution is high and when no computer assistance is available for evaluation.

Last we do admit that lack of MTTR transfer coefficient might restrict application of availability model to specific cases, also that statistical asset underlying K coefficients would improve with further work on storage failure rate prediction.

6. ACKNOWLEDGEMENT

This paper accounts for recent developments of studies initiated by S.T.T.A. Service Technique des Télécommunications de l'Air - 129, rue de la Convention - PARIS 15°

"Influence des choix technologiques sur la Fiabilité, la Disponibilité et le Coût des Equipements" contrat n° 76-86-005.

7. REFERENCES

(1) W.S.E.I.A.C.

Weapon System Effectiveness Industry Advisory Committee

(2) E.P.A.C.

VEDEL J.P. 1973 - AGARD C.P n° 130/6-1

"La conception de projet assistée par ordinateur".

This package has been designed at E.M.D. under S.E.F.T. (Section d'Etude et de Fabrication de 1'Armée de Terre) contract n°500/223/50 it is intended for : previsional equipment's reliability evaluation through programs FIAB B (serial configuration) or FIAB C (redundant structures) ; operational availability prediction through SIMEX simulator ; spares support determination (RECHANGES) and Life Cycle Cost computation (COAC and COPO modules).

TABLE 1

RELIABILITY TRANSFER COEFFICIENT Ki = f (temperature in °C)

Type of Environment	Ratio	10°	30°	50°	70°	90°	110°
Ground fixed	K 1	2.6	2.12	1.63	1.17	0.88	0.74
	K 2	3.86	3.31	2.79	2.22	1.83	1.59
	K 3	5	3.81	2.4	1.27	0.75	0.41
	K 4	2.59	2.33	2.09	1.74	1.67	1.76
	K 5	1.96	2.04	2.08	2.08	2.06	2.09
Aircraft inhabited zone	K 1	2.88	2.6	2.02	1.42	0.96	0.67
	K 2	5.17	4.77	3.95	2.95	2.12	1.56
	K 3	7.31	6.2	4.16	2.26	1.13	0.59
	K 4	2.63	2.36	2.01	1.64	1.43	1.43
	K 5	2.48	2.58	2.6	2.52	2.32	2.17
Aircraft Uninhabited zone	K 1	2.88	2.63	2.08	1.46	0.97	0.67
	K 2	4.53	4.23	3.61	2.83	2.07	1.54
	K 3	6.02	5.33	3.95	2.43	1.32	0.72
	K 4	2.46	2.24	2.06	1.83	1.71	1.71
	K 5	2	2.11	2.18	2.22	2.2	2.16
Missile or satellite launch	K 1	5.21	4.61	3.56	2.38	1.46	0.9
	K 2	8.4	7.67	6.41	4.78	3.25	2.16
	K 3	11.33	9.84	7.26	4.44	2.37	1.23
	K 4	4.25	3.96	3.37	2.65	2.06	1.74
	K 5	3.61	3.61	3.53	3.32	2.98	2.62

Digital
K1 = λMSS LOGICAL A /λMSS LOGICAL B = conventional → conventional on hybrid
K2 = λMSS LOGICAL A /λMSS LOGICAL C = conventional → bipolar LSI on hybrid
K3 = λMSS LOGICAL A /λMSS LOGICAL D = conventional → CMOS LSI on hybrid

Linear
K4 = λMSS ANALOG A /λMSS ANALOG B = conventional → conventional on hybrid
K5 = λMSS ANALOG A /λMSS ANALOG C = conventional → bipolar LSI on hybrid

TABLE 2

COMPUTERS

ITEM LIST	Version A Conventional		Version B hybrids		Version C Hybrids and µP	
. Discrete components						
Transistors, diodes, capacitors, Résistances, etc ...	1790		580		460	
MSI/SSI DIL (log/analog)	618		68		87	
PROM	18		34		32	
Multilayer boards (main)	7		5		5	
Connectors	12		12		9	
-Hybrids (1" x 2")	0		43		36	
-micro processors	0		0		8	
Cores (8K memory)	131.072		131.072		131.072	
Mechanical and electrical characteristics (performances)						
Volume (in liters)	11		9,4		7,7	
Power consumption (in Watts)	140		120		110	
Computation speed (µs)	16 bits	32 bits	16 bits	32 bits	16 bits	32 bits
Addition (simple and double length)	4.8	7.2	2.4	7.5	2	6.25
Multiplication } Floating point	15	46.8	7.2	17.1	6	14.25
Division } Floating point	18	48	8.4	18	7	15

TABLE 3

RADARS

ITEM LIST	Version A Conventional	Version B Hybrids	Version C Hybrids and μP
Discrete components	2200	1600	600
Hybrids (2" x 1")	0	4	4
Micro processors	0	0	1
Mechanical and electrical characteristics (Performances)			
Volume (in liters)	27.2	27.2	16
Weight (in kg)	16	14	10
Power consumption (V.A.)	180	120	80
Detection of aberration (estimated)	95 %	100 %	100 %

TABLE 4

UHF BEACONS

ITEM LIST	Version A Conventional	Version B hybrids	Version C LSI on hybrids
Discrete components	320	21	21
Hybrids (1" x 2")	0	6	4
Custom designed LSI (linear)	0	0	3
Mechanical and electrical characteristics - Not considered as performances			
Volume (l)	.65	.47	.42
Weight (g)	850	520	470
Transmitted power (peak)	300 mW	300 mW	300 mW
Performance is defined here as self test "depth" relative to version B/C			
	0.67	1	1

TABLE 5

Cost effectiveness V

$$V = \frac{\text{Reliability/mission x required availability x performance index relative to version C}}{\text{Cost of ownership over 10 years relative to version A}}$$

COMPUTERS

		Mission's reliability	Availab. (required)	Relative cost of ownership	Relative index of performance P1	P2	P3	Absolute cost effectiveness	Relative cost effectiveness
P1	Version A	0.9968	0.99	1	0.8			0.78	1
	Version B	0.9975	0.99	0.96	0.8			0.81	1.03
	Version C	0.9976	0.99	0.87	1			1.13	1.43
P1xP2	Version A	0.9968	0.99	1	0.8	0.7		0.55	1
	Version B	0.9975	0.99	0.96	0.8	0.82		0.67	1.21
	Version C	0.9976	0.99	0.87	1	1		1.13	2.04
P1xP2xP3	Version A	0.9968	0.99	1	0.8	0.7	0.78	0.43	1
	Version B	0.9975	0.99	0.96	0.8	0.82	1	0.67	1.56
	Version C	0.9976	0.99	0.87	1	1	1	1.13	2.63

	Kop/s ratio	Volume ratio	Power consumption ratio

RADARS

					detection ratio	Weight ratio	Power consumption ratio		
P1	Version A	0.997	0.99	1	0.95			0.94	1
	Version B	0.998	0.99	0.68	1			1.45	1.54
	Version C	0.999	0.99	0.41	1			2.41	2.56
P1xP2	Version A	0.997	0.99	1	0.95	0.63		0.53	1
	Version B	0.998	0.99	0.68	1	0.71		1.03	1.74
	Version C	0.999	0.99	0.41	1	1		2.41	4.08
P1xP2xP3	Version A	0.997	0.99	1	0.95	0.63	0.44	0.25	1
	Version B	0.998	0.99	0.68	1	0.71	0.67	0.69	2.76
	Version C	0.999	0.99	0.41	1	1	1	2.41	9.64

BEACONS

Version A	0.9976	0.99	1	0.67*	Self test	0.66	1
Version B	0.9999	0.99	1.25	1	comprehensiveness	0.79	1.2
Version C	0.9998	0.99	0.73	1		1.36	2.06

* Version A has no depth modulation test.

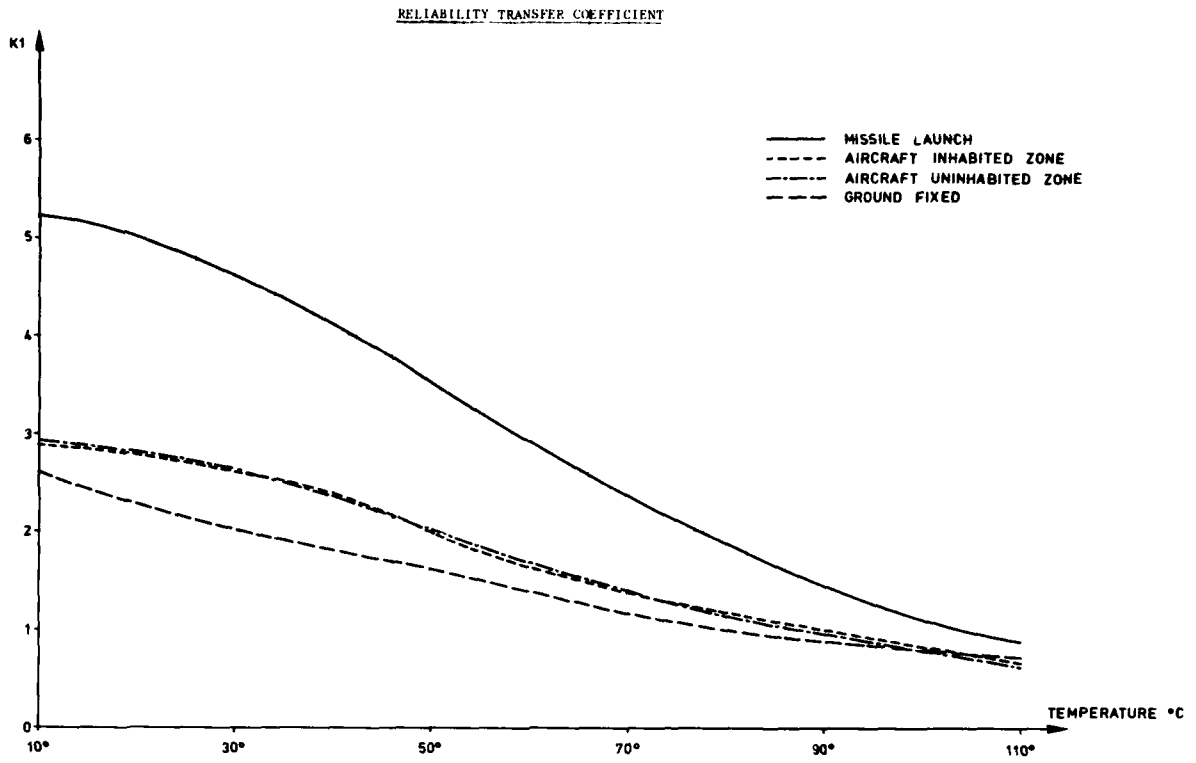


Fig.1 K1 = λ conventional/λ conventional on hybrid (logic)

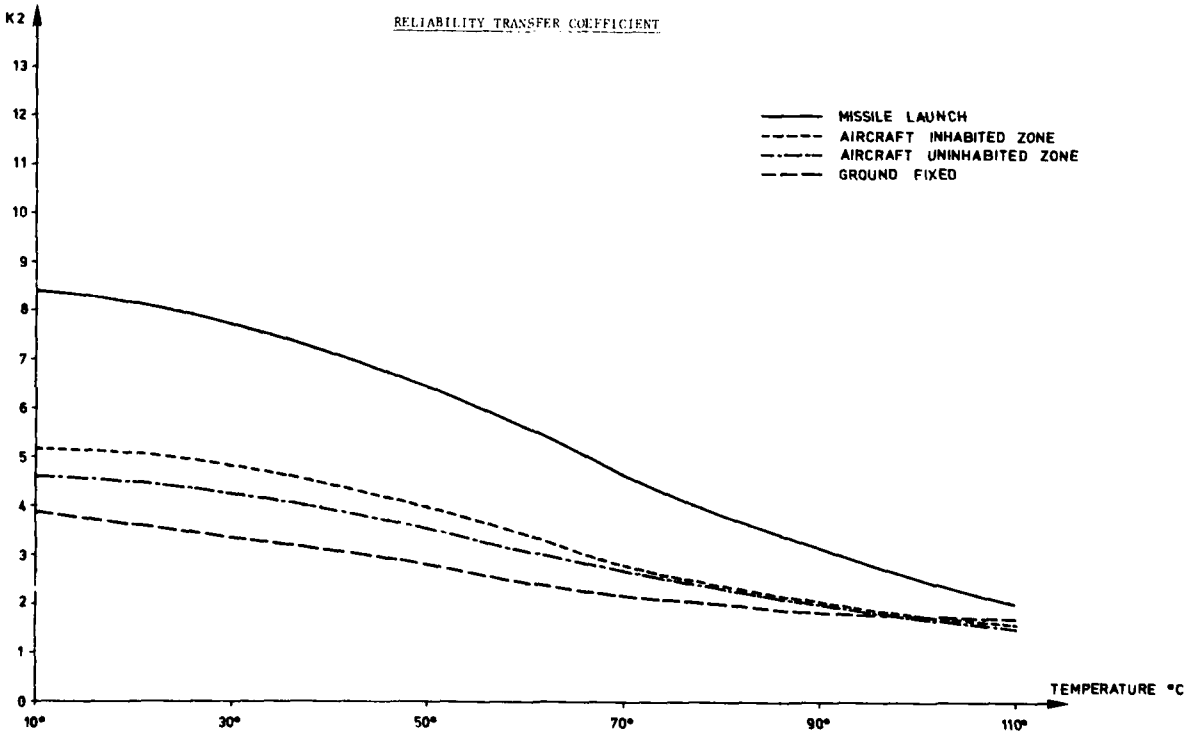


Fig.2 K2 = λ conventional/λ bipolar LSI on hybrid (logic)

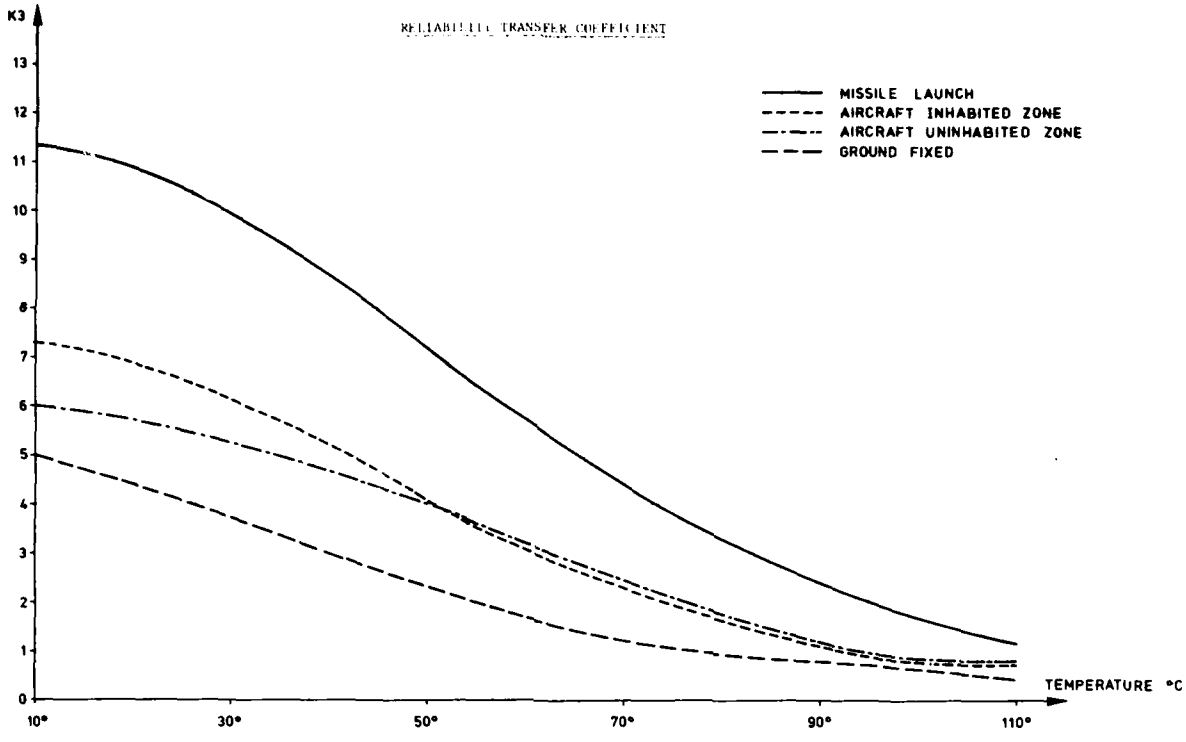


Fig.3 K3 = λ conventional/λ CMOS LSI on hybrid (logic)

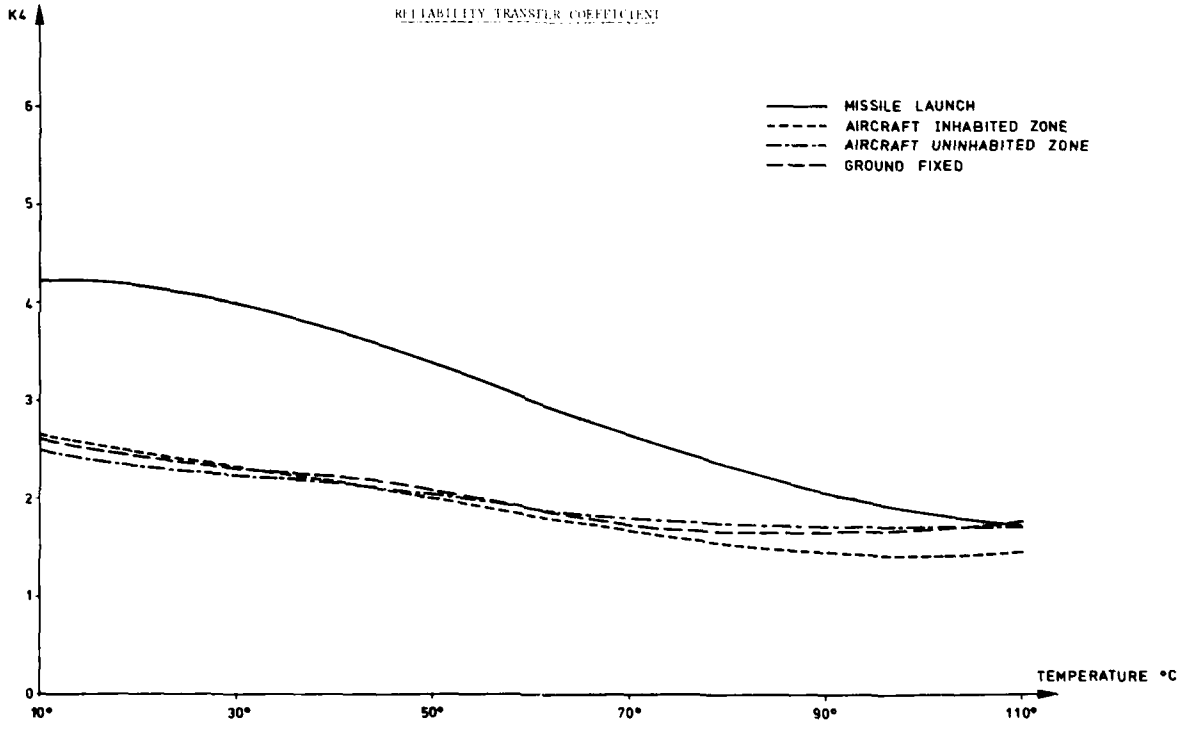


Fig.4 K4 = λ conventional/λ conventional on hybrid (linear)

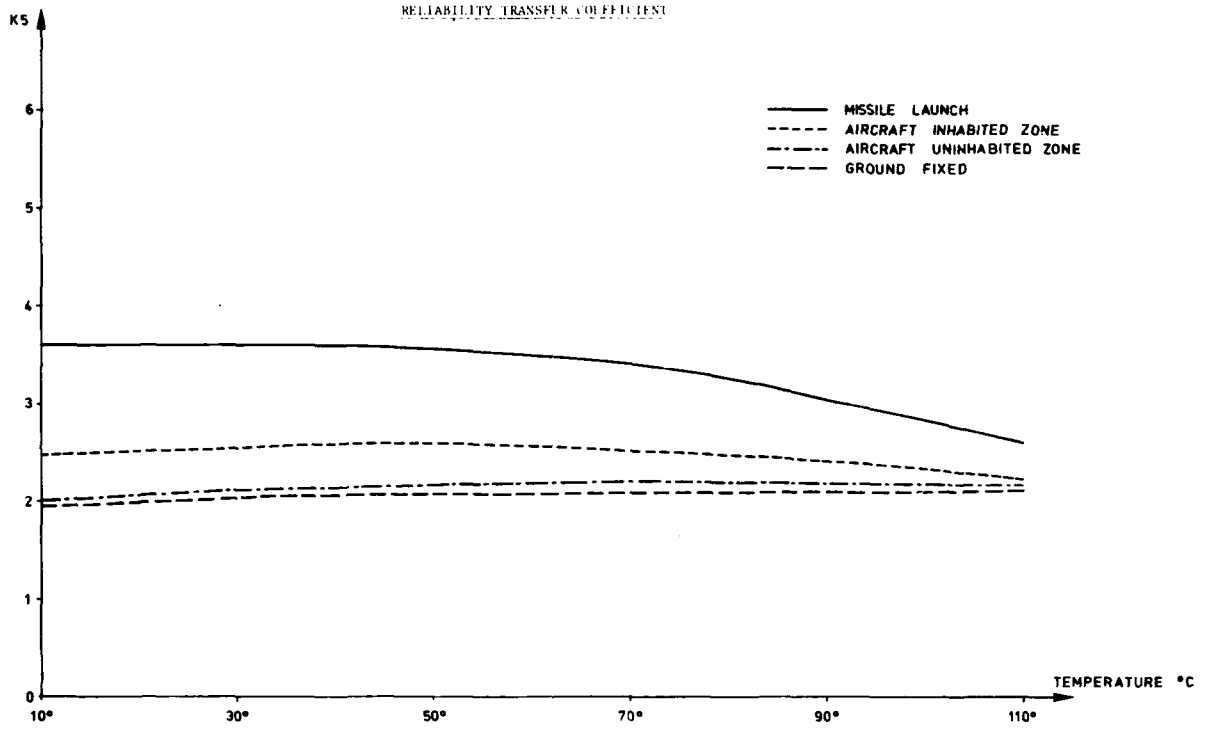


Fig.5 K5 = λ conventional/λ bipolar LSI on hybrid (linear)

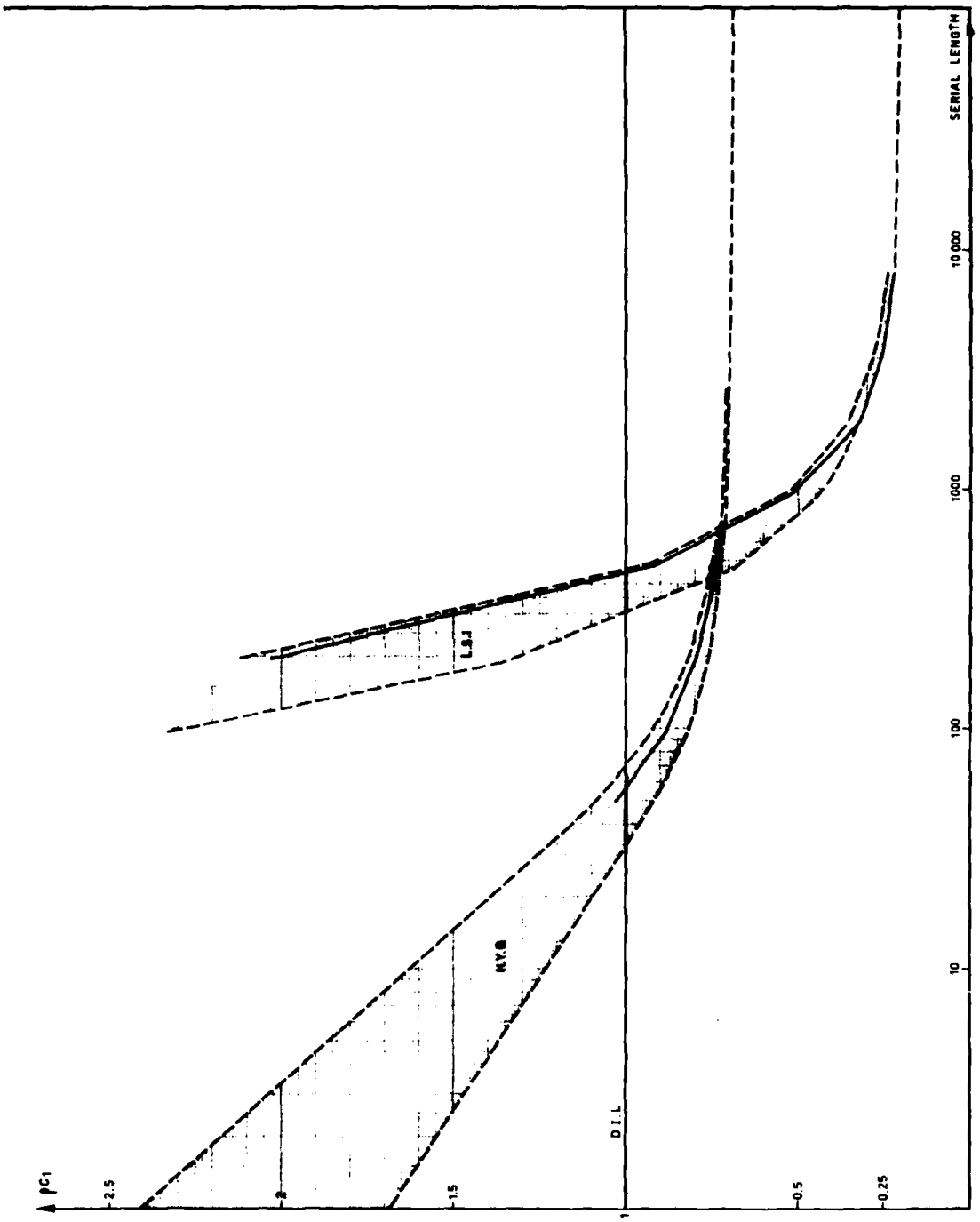


Fig.6 Electrical parts cost ratio normalized to conventional (DIL)

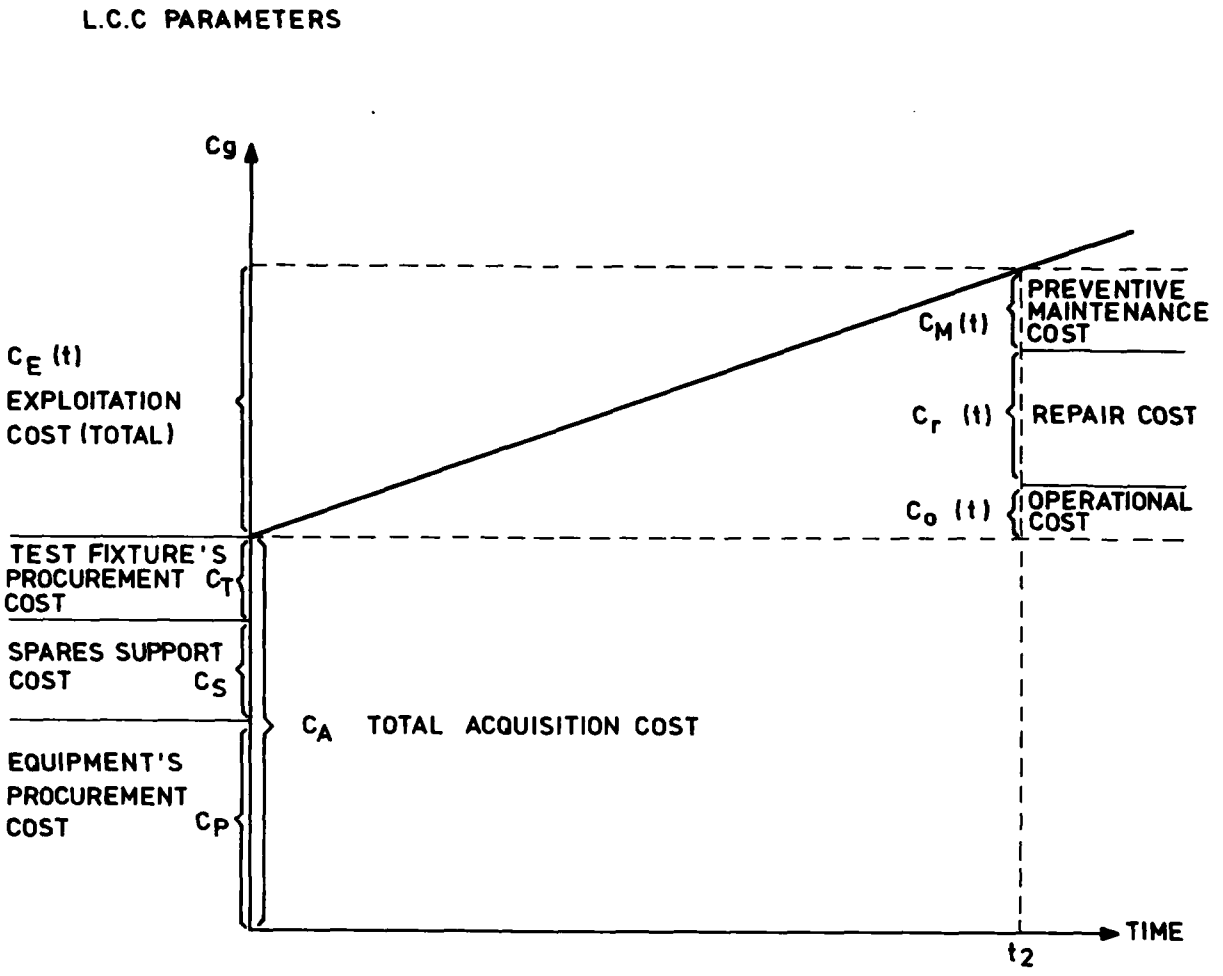


Fig.7 Time evolution of Cg

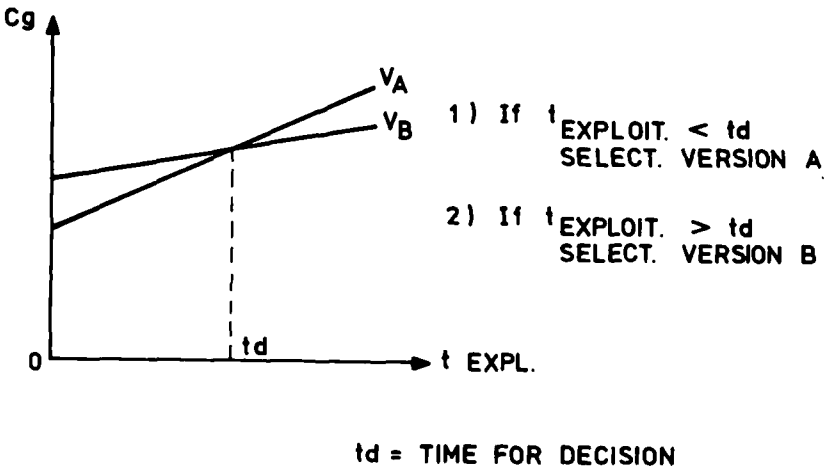


Fig.8 Cost criterion for decision

A NEW APPROACH TO MAINTAINABILITY PREDICTION

Joseph J. Naresky
Rome Air Development Center (RADC)
Griffiss AFB NY 13441, USA

ABSTRACT

Existing maintainability techniques, as described in MIL-HDBK-472, "Maintainability Prediction," are not accurate estimators of current electronic equipment/system maintainability. They were developed more than 20 years ago and, hence, reflect the technology of that era. Since then, there have been rapid advances in electronic technology which greatly enhance maintainability. They are: widespread application of semiconductor microcircuits and large scale integration; designs for equipment modularity; and improved methods of fault detection/isolation. Existing maintainability techniques have another shortcoming in that they have little, if any, effect during the design stage: they can only be applied after the equipment/system is fabricated.

To overcome the previously mentioned shortcomings, a new maintainability prediction method was developed by the Rome Air Development Center (RADC) during the past several years. The new methodology, a time synthesis procedure, directly relates diagnostic/isolation/test subsystem characteristics, and other design characteristics, to equipment/system maintainability parameters. It also provides a comprehensive set of time standards applicable to physical maintenance actions associated with current equipment construction and packaging procedures. Predicted parameters include mean-time-to-repair (MTTR), a maximum (percentile)-time-to-repair, mean maintenance man-hours per repair, and fault isolation resolution. The developed methodology includes two techniques: one for use when final design data is available; the other, an early prediction procedure, for use when only preliminary data is available. The latter is another feature not provided by previous techniques.

1. INTRODUCTION

Many features affect the mean-time-to-repair (MTTR), the most commonly accepted measure of the maintainability of an equipment/system. Some of these factors are: the packaging of the equipment, degree and design of the diagnostics and diagnostic routines or procedures, component failure rates, the operating environment, and the competence and training of the maintenance personnel. Each of these factors affects maintainability differently, and must be taken into account when designing the equipment to maximize maintainability.

Usually, the defects inherent in the maintenance design of an equipment do not surface until field operation begins, at which time most design changes that must be made become exorbitantly expensive. It is usually necessary to change the manufacturing process to incorporate the design changes, and, possibly, retrofit those units already fielded. This not only costs money, but also means that considerable time is lost retrofitting the fielded units (especially if they must be shipped back to the manufacturer).

Maintainability prediction techniques, presently in use have the basic drawback that equipment design parameters cannot be directly related to their impact on the maintainability of various portions of the equipment in time to take proper design, or redesign, actions.

Therefore, several years ago, a program was initiated at RADC to develop improved, more accurate, maintainability prediction and analysis procedures for electronic equipments and systems. A basic premise was that the new procedures would be capable of directly relating diagnostics/isolation/test subsystem characteristics and other engineering characteristics to equipment system maintainability.

Also, to be investigated was the development of a set of time standards applicable to physical maintenance actions associated with current equipment construction and packaging techniques. Finally, the developed techniques had to be applicable to avionics, ground, and shipboard electronics at the organizational, intermediate, and depot levels of maintenance. This paper describes the procedure and the achieved results.

1.1. APPROACH

The approach used to accomplish the above objectives included three tasks. The first was to perform a literature survey to define and evaluate the existing maintainability prediction techniques and maintenance action time standards, and their applicability to modern electronic equipments and systems. The second task consisted of reviewing present day electronic equipment system characteristics and maintainability prediction needs to identify the parameters to be predicted and the general approach to the prediction methodology. The last of the tasks involved reviewing the maintenance policies in current use, and developing prediction techniques consistent with the manner in which maintenance is accomplished.

2.1. SURVEY OF EXISTING TECHNIQUES

The maintainability prediction techniques presently in use were surveyed and reviewed to determine the basic prediction hypothesis, data base, detailed procedure, and shortcomings of each. All had substantial drawbacks relative to their ability to adequately evaluate complex modern systems. The most serious shortcoming was a lack of significant correlation between the quantitative maintainability parameters (i.e., MTTR) and the system fault detection and isolation (FD&I) features.

The prediction techniques surveyed can be divided into time synthesis models and correlation models. Time synthesis models are those in which the maintenance activity is broken down into elemental maintenance steps, each step is assigned a fixed time or time function, and the steps combined, or synthesized, to yield the designed maintainability parameter. The correlation models are those that utilize a checklist, or other mechanism, to score maintenance-related attributes of a system; the score is inserted into a regression equation to yield the desired maintainability parameter. With rare exception, the regres-

sion techniques currently in use were developed anywhere from 10 to 20 years ago and, therefore, are not sensitive to the maintenance features and characteristics of modern electronic equipments and systems. The time synthesis models are based on the premise that, given that a certain part has failed, what time is required to repair the part by replacing it? The drawback to this is that most parts exhibit more than one failure mode and/or effect. Different failure modes result in significantly different corrective maintenance times due to the methodology used for fault isolation/resolution/ambiguity of the fault isolation procedure corresponding to the various failure symptoms. Yet, maintenance is symptom oriented not failure oriented, as the time synthesis models surveyed indicated.

From the results of the survey, it was determined that the prediction methodology should reflect the manner in which maintenance is actually performed. This ground rule implies:

(a) Fault isolation time estimates must be based on how the failure presents itself in terms of external failure effects, and the results of the fault isolation procedures that are available to maintenance personnel.

(b) Variability caused by different failure modes and effects of each replaceable item (RI) must be considered, particularly variations in fault isolation time and fault correction time. A replaceable item is any of the physical entities [(line replaceable unit (LRU), weapon replaceable assembly (WRA), component part, etc., etc.)] normally removed and replaced to accomplish repair at the maintenance level for which the prediction is being made.

(c) Ambiguities (isolation to more than one RI), including consideration of secondary maintenance which is needed when the primary fault correction procedure does not correct the problem, must be taken into account.

(d) The prediction methodology should not be susceptible to technician variance (except possibly maintenance personnel skill level); it should be based on established procedures for each corrective maintenance action.

3.1. GENERALIZED PREDICTION MODEL DEVELOPMENT

In addition to the above general ground rules, some specific ground rules were also followed in developing the generalized prediction methodology:

(a) Failures occur at the predicted failure rates.
(b) Only hard failures were considered.
(c) Only single failures were considered.
(d) Only randomly occurring failures were considered.
(e) Maintenance is performed in accordance with established maintenance procedures.
(f) Maintenance is performed by technicians with appropriate skills and training.
(g) Only active maintenance time is considered; thus, excluding administrative and logistic delay time, fault detection time, clean up time, etc.

The primary maintainability parameter considered in developing the prediction technique was MTTR (mean-time-to-repair). MTTR is the mean value of the probability distribution of times to complete active corrective maintenance over all predictable unscheduled maintenance actions, weighted by the relative frequencies of occurrence of these actions. With minor modifications to the MTTR prediction technique, a number of other parameters can be accurately estimated. These parameters are:

$M_{max}(\emptyset)$ - maximum corrective maintenance time at the $\emptyset$ percentile

I_1 - fault isolation resolution to a single RI

I_N - fault isolation resolution to $<$ N RIs

$\overline{MMH}$/Repair - mean maintenance man-hours per repair

$\overline{MMH}$/MA - mean maintenance man-hours per maintenance action (including false alarms)

$\overline{MMH}$/OH - mean maintenance man-hours per operating hour

4.1. MTTR Elements

As mentioned previously, the methodology used in a typical time synthesis technique. The times associated with each portion of a maintenance action are summed to yield the total maintenance time for that action. For each individual maintenance action, the predicted/estimated maintenance time is the expected average time to complete the maintenance action. Admittedly, there is some variability to the time to complete each maintenance task; this is only addressed in predicting maximum corrective maintenance time.

Table 1 indicates the breakdown of those task elements used in MTTR prediction for various classes of maintenance elements is provided in Table 2. The methods applicable to estimating each of the maintenance element times are presented in Table 3.

4.2. GENERALIZED MATHEMATICAL MODELS

The generalized equation for computing MTTR is:

$$MTTR = \frac{\sum_{n=1}^{N} \lambda_n R_n}{\sum_{n=1}^{N}} \qquad (1)$$

where:

N = number of replaceable items (RI)

λ_n = failure rate of the nth RI (failures/10^6 hours)

R_n = mean repair time (minutes) of the nth RI as computed below

$$\bar{R}_n = \frac{\sum_{j=1}^{J} \lambda_{nj} R_{nj}}{\sum_{j=1}^{J} \lambda_{nj}} \qquad (2)$$

where: J = number of unique fault isolation results

[fault detection and isolation (FD&I) output]

- see Figure 1, page 20-10

λ_{nj} = failure rate of those parts of the nth RI which would cause the nth RI to be called out in the jth fault isolation result

R_{nj} = average repair time of the nth RI when called out in the jth isolation result as computed below:

$$R_{nj} = \sum_{m=1}^{M_{nj}} Tm_{nj} \qquad (3)$$

where:

M_{nj} = number of steps to perform corrective maintenance when a failure occurs in the nth RI and results in the jth fault isolation result--includes all maintenance elements (preparation, isolation, spare retrieval, et al).

Tm_{nj} = average time to perform the nth corrective maintenance step for the nth RI, given the jth fault isolation result

The generalized models were used to develop specialized models for the various situations shown in Table 1. For example, the model for the spcial case of isolation to a single RI and replacement of that RI is:

$$R_{nj} = T_{P_{nj}} + T_{FI_{nj}} + T_{SR_{nj}} + T_{D_{nj}} + T_{I_{nj}} + T_{R_{nj}} + T_{A_{nj}} + T_{C_{nj}} + T_{ST_{nj}} \qquad (4)$$

(see Table 2 for definitions of the "T" terms)

From the generalized models, two procedures were developed for predicting the MTTR (and other maintainability parameters of an equipment/system).

(a) A detailed procedure for use when detailed design and support data is available, and;
(b) An early procedure for use when only preliminary design data is available.

4.3. DETAILED PREDICTION METHOD

The procedure for the detailed prediction technique is as follows:

(a) First define the maintainability parameter(s) to be evaluated, the prediction ground rules, and the maintenance level for which the prediction is being made. For simplicity, in this paper, I shall deal only with MTTR.

(b) Next, define the maintenance concept. For a maintainability prediction this means how repairs are made, and what the replaceable items are.

(c) Construct a list of all the possible failure symptoms, or results of fault detection/isolation procedures, which includes all of the possible indications that an operator or technician might experience in identifying the fault correction procedures to be performed.

(d) Correlate the RIs of the system with the identified failure symptoms developed in (c) which is usually done using a failure mode and effects analysis (or any similar analysis).

(e) Prepare a maintenance correlation matrix similar to the one shown in Figure 1. This matrix provides: (1) the failure rate (λ_{nj}) of each RI that is associated with each failure symptom; (2) the repair time (R_{nj}) for an RI given that a specific failure symptom (failure detection and isolation output) occurs; and, (3) the replacement order (K_{nj}) of RIs given that a specific failure detection and isolation output (FD&I) occurs and the associated maintenance concept is iterative replacement.

(f) Prepare a maintenance flow diagram (MFD) to establish the R_{nj}) values for insertion in the Maintenance Correlation Matrix (Figure 1). The MFD is prepared to illustrate the sequencing of maintenance as performed by the designated maintenance technician.

A sample MFD (shown in Figure 2) starts on the left hand side of the figure as a "Fault Occurs and Detected" event. The oval outputs on the left, following "start," designate the FD&I output which defines the subsequent maintenance activity to be performed. The "j" associated with the output is entered in the circle next to the oval. Following the FD&I outputs, are shown the activities required for fault correction and verification.

The R_{nj} values inserted in the Maintenance Correlation Matrix are computed by adding the times associated with each activity block from the "fault occurs and is detected" event to the "end" event for the subject (n,j) pair. The time entered in the individual activity blocks is computed from a time line analysis. A time line analysis consists of computing the total elapsed time of a maintenance action by summing the times required to perform each step. The individual times can be obtained from: actual times experienced on subject equipment; published maintenance time standards; actual times experienced on similar equipment; or engineering judgment. For example Figure 2 contains the R_{11} through R_{99} values of maintenance times of an airborne radar example described in the report.[1] These values are then inserted into the Maintenance Correlation Matrix as shown in Figure 3, along with the λ_{nj} failure rate values. The numbers across the top, e.g. 001, 011, etc. merely refer to the nine RIs of the previously mentioned airborne radar example described in the report.

(g) Compute the maintainability parameter of interest e.g. MTRR. Once the MFD and the Maintenance Correlation Matrix have been completed, it is an easy matter to compute the desired maintainability.

For example:

(1) MEAN REPAIR TIME OF nth RI

$$R_n = \frac{\sum_{j=1}^{J} \lambda_{nj} R_{nj}}{\sum_{j=1}^{J} \lambda_{nj}} \quad \text{-- obtained from MFD} \qquad (5)$$

(2) EQUIPMENT MTTR

$$MTTR = \frac{\sum_{n=1}^{N} \lambda_n R_n}{\sum_{n=1}^{N} \lambda_n} \qquad (6)$$

The R_{ij} values are obtained from the MFD (Figure 2) and inserted into the Maintenance Correlation Matrix (Figure 3). The R_n and λ_n values can be calculated from the Maintenance Correlation Matrix. Given these, one can then compute the MTTR, as shown for the airborne radar example (see Figure 4).

4.4. EARLY PREDICTION PROCEDURE

This technique was developed for use in the design phase of a program when only preliminary design data is available.

For an early prediction, it is assumed that the following data is available:

(a) A configuration index from which a definition of the primary replaceable items can be derived.
(b) The failure rate of each of the primary replaceable items.
(c) The overall fault isolation concept (i.e. fault isolation to a single RI or group of RIs).
(d) The replacement concept when fault isolation is to a group of RIs (i.e. group or iterative replacement).
(e) The basic packaging philosophy including preliminary access and interchange characteristics of each RI.
(f) The primary fault isolation technique to be implemented for each primary RI.
(g) The fault isolation resolution which is defined in one of two ways:

(1) average RI group side
(2) x_1 % isolation to a single RI
x_2 % isolation to >1 RI but $\leq N_1$ RIs
x_3 % isolation to $>N_1$ RIs but $\leq N_2$ RIs

where

$x_1 + x_2 + x_3 = 100\%$

The prediction model is based on the generalized version of the detailed model. That is

$$MTTR = \bar{T}_P + \bar{T}_{FI} + \bar{T}_{SR} + \bar{T}_D + \bar{T}_I + \bar{T}_R + \bar{T}_A + \bar{T}_C + \bar{T}_{ST} \quad (7)$$

or

$$MTTR = \sum_{m\,1}^{9} \bar{T}_m \quad (8)$$

where

$\bar{T}_m$ = average time of the mth element

m = the elemental maintenance tasks

(P, FI, SR, D, I, R, A, C, ST) as defined in Table 2.

Thus the procedure used in the model is to: define the major ways in which elemental maintenance tasks are performed, assign failure rates and times to each of the different elemental task types, determine a failure rate weighted average for each maintenance element, and find the MTTR by adding the average times of each element.

Two methods are available for determing the time associated with each maintenance element.

The first method is summarized by the following model:

$$\bar{T}_m = \frac{\sum_{n=1}^{N} \lambda_n T_{m_n}}{\sum_{n=1}^{N} \lambda_n} \qquad (9)$$

where

N = the number of primary RIs

λ_n = failure rate of the nth RI

Tm_n = the synthesized time for the mth elemental maintenance task of the nth RI.

This model assumes that Tm_n is available for each maintenance element of each RI. For those maintenance elements where this is not true, the second method determines an average value for the elemental maintenance times by using the following model:

$$\bar{T}_m = \frac{\sum_{V=1}^{Vm} \lambda_{m_V} T_{m_V}}{\sum_{V=1}^{Vm} \lambda_{m_V}} \qquad (10)$$

where

Vm = the number of major unique methods of performing the mth elemental task

λ_{m_V} = the failure rate associated with the set of faults involving the vth method of performing the elemental task

T_{m_V} = the time required to perform the mth elemental task using the vth method

For example, the number of ways of performing fault isolation on a display console might be: test pattern interpreation for the majority of display circuitry; maintenance panel readings for power supplies; computer controlled loop testing for I/O circuits; and manual isolation of miscellaneous circuit electronics. A time would be assigned to each of these methods of fault isolation, and an average fault isolation time would be computed based on the estimated failure rate of the circuitry associated with each method. A similar procedure would be followed for each maintenance element, and the MTTR computerd by adding all of the element times.

The simplied models shown above apply to the most general case where fault isolation is to a single RI. Models were also developed for various types of maintenance concepts and repair policies.

5.0 CONCLUSIONS

The prediction methodology described in this paper achieves the objective for which it was intended. It represents a new approach to maintainability prediction, based upon recent advances in electronic technology such as widespread usage of microcircuits and improved methods of fault detection/isolation and test. The methodology can be applied at any maintenance level, for any maintenance concept, and to all classes of electronic equipment/systems--avionics, ground, and shipboard.

In comparison to previously available techniques, the one described in this paper permits maintainability predictions to be made during the design phase of an equipment development. Thus, the user and/or designer can predict whether the specified maintainability requirement will be met before final design or fabrication is completed. The ability to introduce early design changes to improve maintainability should negate the need for costly, after the fact, retrofitting.

A by-product of the prediction techniques development has been the updating of maintenance time standards which more accurately reflect modern packaging and construction methods.

Additional details on the development and implementation of this new maintainability prediction technique can be found in RADC-TR-78-169, "Maintainability Prediction and Analysis Study," which is available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161, telephone (703)-557-4600. Document Number is AD-A059753.

6.0 REFERENCES

Pliska, T.F., Jew, F. L., and Angus, J.E., 1978, Maintainability Prediction and Analysis Study, Hughes Aircraft Co., RADC-TR-78-169.

Lipa, J. F., March, 1976, Maintainability Prediction Survey, RADC-TM-76-3.

TABLE I

MTTR Elements for New Prediction Methodology

Proposed Breakdown (Basic)	Isolation to Single RI	Isolation to Group with Group Replacement	Isolation to Group with Iterative Replacement	Isolation with Ambiguity	
Preparation	Preparation	Preparation	Preparation	Preparation	Secondary Preparation
Isolation	Isolation	Isolation	Isolation	Isolation	Secondary Isolation
Fault Correction	Spare Retrieval	Spare Retrieval	Fault Correction	Fault Correction	Secondary Fault Correction
	Disassembly	Disassembly			
	Interchange	Interchange			
	Reassembly	Reassembly			
	Alignment	Alignment			
	Checkout	Checkout			
Start Up	Start Up	Start Up	Start Up	Continue	Start-Up

TABLE 2

Definition of Maintenance Task Times

Maintenance Element Time	Abbreviation*	Definition
Preparation	$T_{P_{nj}}$	Time associated with those tasks required to be performed before fault isolation can be executed. Examples: Obtain, set-up and warm up test equipment; Apply power and cooling to system, warm up and stabilize; Input system initialization parameters.
Fault Isolation	$T_{FI_{nj}}$	Time associated with those tasks required to isolate the fault to the level at which fault correction begins. Examples: Load, run, and interpret results of a diagnostic program; Examine fault isolation symptoms, locate symptoms in maintenance manual, follow manual procedures to point where replaceable item or group of replaceable items is identified.
Fault Correction		
• Spare Retrieval	$T_{SR_{nj}}$	Time associated with obtaining a spare replaceable item or group of replaceable items from the designated spares area.
• Disassembly	$T_{D_{nj}}$	Time associated with gaining access to the replaceable item(s) dentified during the fault isolation process. Examples Opening cabinet doors, pulling out equipment drawers, removing CCA retaining bars; Technician transit time to a remote equipment.
• Interchange	$T_{I_{nj}}$	Time associated with the removal and replacement of a faulty replaceable item or suspected faulty items. Examples: Removing screws, connectors, solder joints; Extracting and inserting the replaceable item; Application of conformal coating, heat transfer paste.
• Reassembly	$T_{R_{nj}}$	Time associated with closing up the equipment after interchange is performed, i.e., the opposite process of disassembly.

*Abbreviations used in the prediction math models; Time to perform the m^{th} elemental task (P, FI, SR, D, I, R, A, C, ST) for the n^{th} RI given the j^{th} fault isolation result.

TABLE 2 (concluded)

Definition of Maintenance Task Times

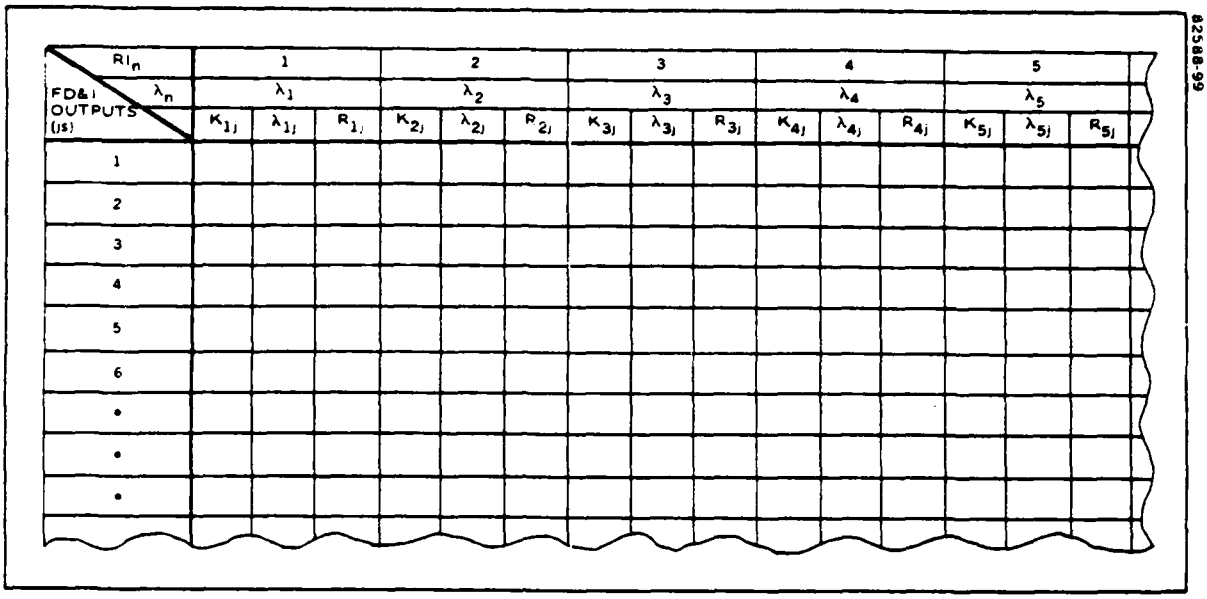
Maintenance Time	Abbreviation*	Definition
• Alignment	$T_{A_{nj}}$	Time associated with aligning or calibrating the system or RI after a fault has been corrected.
• Checkout	$T_{C_{nj}}$	Time associated with the verification that a fault has been corrected and the system is operational.
Start-up	$T_{ST_{nj}}$	Time associated bringing a system up to the operational state it was in prior to failure, once a fault has been corrected and verified.

*Abbreviation used in the prediction math models.

TABLE 3

Corrective Maintenance Time Elements and Methods of Estimation

	Time Standards	Fixed Time	Field History	Engineering Judgement
PREPARATION - $T_{P_{nj}}$		X	X	X
FAULT ISOLATION - $T_{FI_{nj}}$		X	X	X
SPARE RETRIEVAL - $T_{SR_{nj}}$			X	X
DISASSEMBLY - $T_{D_{nj}}$	X			X
INTERCHANGE - $T_{I_{nj}}$	X			X
REASSEMBLY - $T_{R_{nj}}$	X			X
ALIGNMENT - $T_{A_{nj}}$		X	X	X
CHECKOUT - $T_{C_{nj}}$		X	X	X
START UP - $T_{ST_{nj}}$		X	X	X



Under each RI column, enter the failure rate (λ_{nj}) of the RI that could result in the jth output. Now, for each unique output which has only one RI associated with it, enter a 1 in the K_{nj} column for that combination. For those outputs which are associated with 2 or more RIs, the K_{nj} value depends on the maintenance concept. If the maintenance concept is group RI replacement, enter under K_{nj} the number of RIs associated with each output. For example, if three RIs could contribute to the same FD &I output, then a 3 is entered in the K_{nj} for each of those RIs. If the maintenance concept is iterative replacement, then K_{nj} is assigned based on the order of replacement. That is, the first RI to be replaced upon recognition of the subject FD&I output is designated as $K_{nj} = 1$, the second $K_{nj} = 2$ and so forth. The typical assignment of values for each K_{nj} is based on the relative failure rates of the RIs, with the highest failure rate RI assigned as the first replacement item.

Fig.1 Maintenance correlation matrix format

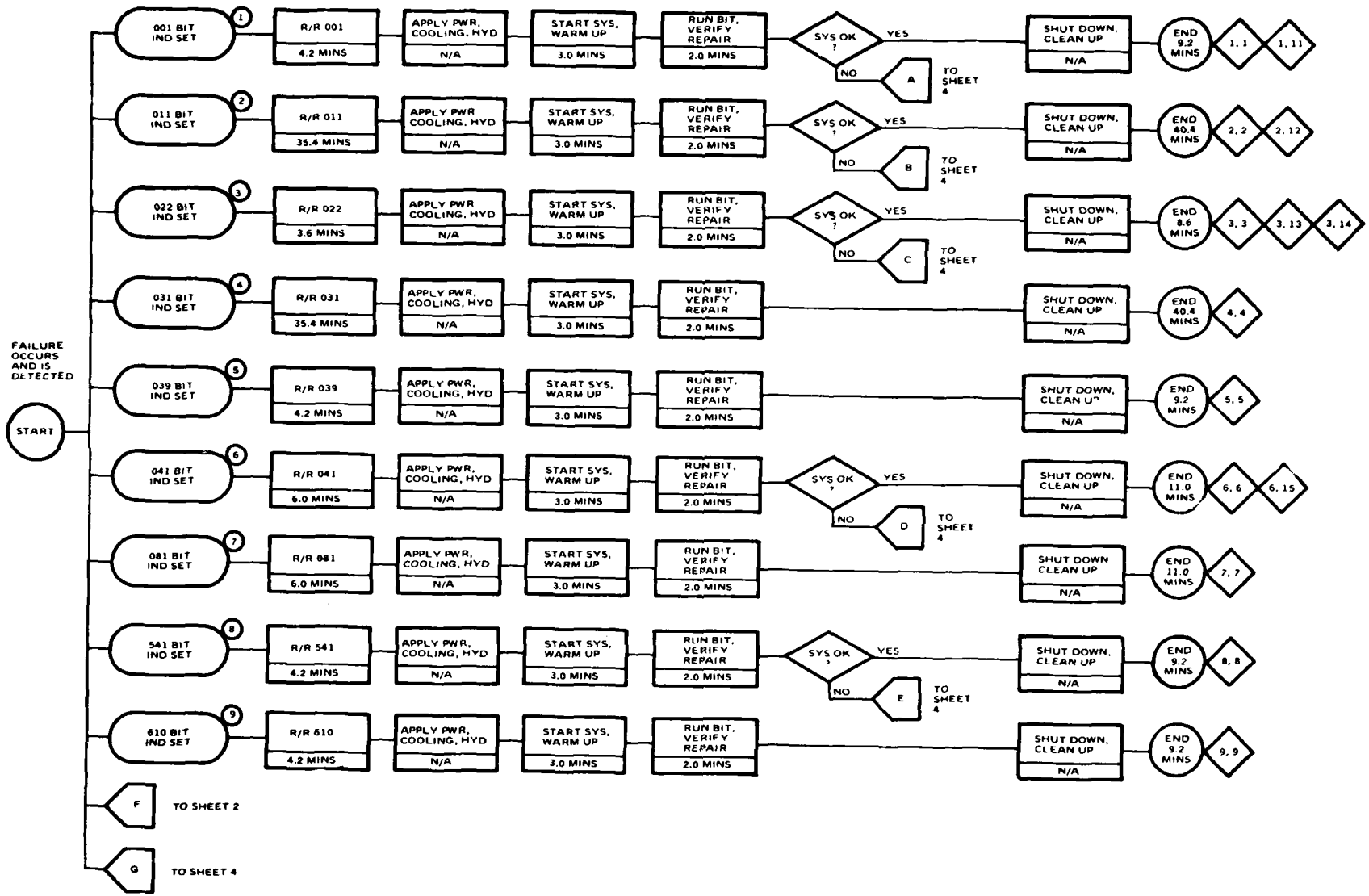


Fig.2 Sample maintenance flow diagram

EQUIPMENT	001			011			022			031			039			041			081			541			610		
λ_{EQUIP}	79.720			226.957			40.779			233.571			126.923			663.127			181.636			9.961			27.476		
J	K_{1j}	λ_{1j}	R_{1j}	K_{2j}	λ_{2j}	R_{2j}	K_{3j}	λ_{3j}	R_{3j}	K_{4j}	λ_{4j}	R_{4j}	K_{5j}	λ_{5j}	R_{5j}	K_{6j}	λ_{6j}	R_{6j}	K_{7j}	λ_{7j}	R_{7j}	K_{8j}	λ_{8j}	R_{8j}	K_{9j}	λ_{9j}	R_{9j}
1	1	22.103	9.2																							
2				1	10.920	39.4																				
3							1	7.778	8.6																	
4											192.388	39.4														
5													1	71.061	9.2											
6																1	599.541	11.0								
7																			1	168.879	11.0					
8																			2	0.088	20.2	1	8.276	9.2		
9																									1	25.660	9.2
10				1	3.456	40.4				2	1.944	80.8														
11	1	49.362	9.2							2	4.310	49.6														
12				1	198.645	39.4				2	4.310	78.8														
13							1	2.132	8.6	2	0.119	50.5														
14				3	1.171	59.7	1	16.997	8.6				2	3.079	20.3												
15							3	0.327	28.8							1	43.761	11.0				2	1.429	20.2		
16																										
17	1	0.059	9.2																							
18				1	0.059	47.9																				
19							1	0.059	16.1																	
20													1	0.059	16.7											

FAILURE RATES (λ) IN FAILURES/10^6 HOURS
REPAIR TIMES (R) IN MINUTES

82595-1P

Fig.3 Sample maintenance correlation matrix

n	λ_n	n	$\lambda_n R_n$
1	79.720	12.88	1026.95
2	226.957	41.10	9327.53
3	40.779	18.45	752.21
4	233.571	43.78	10226.87
5	126.982	13.61	1727.71
6	663.186	11.28	7479.36
7	181.636	11.60	2106.55
8	9.961	11.36	113.11
9	27.476	10.46	287.51
Σ	1590.268		33047.8

$$MTTR = \frac{\sum_{n=1}^{N} \lambda_n R_n}{\sum_{n=1}^{N} \lambda_n} = \frac{33047.8}{1590.268} = 20.78 \text{ minutes}$$

Fig.4 Predicted RI repair times and system MTTR

DISCUSSION

M.B.Kline, US

Could you say a few words about the data base used for your test times? These test times like the failure rates used in reliability prediction are very critical for making valid and useful predictions.

Author's Reply

It is a rather extensive data base and all the test times were redone based upon modern technology, modular design of today, the use of integrated circuits, circuit boards etc.

The data base was prepared using a wide range of systems built over the last few years and is much more modern than previous data bases. One of the purposes of this prediction technique was to make sure that the test times represented modern technology. All new test times are contained in the report.

B.G.Peyret, Fr

Est-ce que dans le MTTR qui vient d'être exposé, le temps de démontage de l'équipement supposément hors de l'avion est compris ou non? En d'autres termes est-ce que le MTTR qui vient d'être exposé, s'occupe simplement des réparations en usine ou bien quel qu'il compte des temps qu'il faut pour démonter l'équipement de l'avion et de le remonter en suite sur l'avion?

Author's Reply

Yes it does. This includes the dissembly, the interchange where necessary and the time involved. It does not include the time involved in waiting for a part for several weeks, administrative or logistic delay time.

RELIABILITY GROWTH THROUGH ENVIRONMENTAL STIMULATION

by

Lawrence J.Phaller
Westinghouse Electric Corporation
Defense & Electronic Systems Center
Box 746, Baltimore, Maryland 21203
USA

SUMMARY

History has shown that a large disparity exist between field or operational MTBF's and predicted MTBF's for electronic equipment. Differences up to and exceeding 10:1 have been recorded. The cause of these differences have been atrributed to many factors. However, most agree that field environments such as temperature, vibration and humidity, are sufficiently different from the contractor tested and qualified environments and these differences are a major part of this disparity.

To minimize this disparity, customers have taken several steps to control and even strengthen the design through the early use of extended environment/mission profile testing. Westinghouse has successfully employed this technique on several programs.

However, the AN/AWG-10 Reliability Improvement Program (RIP) was the first program at Westinghouse in which a Test, Analyze, and Fix (TAAF) or Reliability Growth approach was employed on an already existing field deployed system. In this program, field data were used to identify "unreliable" LRU's and design modification were implemented based on this data. Several conclusion were made as a result of this program:

(1) Field reliability can be enhanced with a relatively moderate program "front-end" investment.

(2) The rate of reliability growth depends no only on the degree of management commenttment to the program, but also on the unit complexity and state of the art of the unit design.

(3) Reliability growth can be experienced in two basic forms: design and quality control.

INTRODUCTION

The need for developing reliability growth programs has been a continuing concern to both DOD and industry. This concern is evidenced by the many coordinated efforts that have been conducted in this field. One of these efforts was the AN/AWG-10 Reliability Improvement Program in which a field deployed system was redesigned and matured through environmental testing to attain the required high level of operational reliability.

This program began with the AWG-10 data feedback system which has been in use since 1967. Over 500 AWG-10 radar systems, each having over 30,000 electrical parts packaged in 29 LRU's (Line replaceable Units), have been deployed throughout the world with over 900,000 field operating hours.

All maintenance actions done on these equipments are recorded by the Navy Maintenance Material Management (3M) Data Collection Program. This data, fed back on a quarterly basis for analysis and use, was instrumental in the development of the AWG-10 Reliability Improvement Program (RIP). This data and knowledge of frequency of field maintenance actions on respective system LRU's, allowed relative unreliable LRU's to be identified. New reliability goals, as shown in figure 1, were then developed for the system LRU's exhibiting this field "unreliability." There goals were used as the basis for the AN/AWG-10 RIP.

This program basically consisted of two elements:

- Redesign of identified LRU's to improve inherent reliability
- "Test and fix" environmental reliability growth test program to develop maturity.

During the design portion of this program. eight of the system's 29 LRU's were extensively modified. In general. all parts were derated to at least 50 percent of their maximum rating (i.e., power, voltage, or junction temperature). Established Reliability (ER) screened parts, level P, were used for resistors and capacitors. JAN TX (Tested Extra) semiconductors and MIL-STD-883, Class B equivalent integrated circuits were used throughout the design. Upon fabrication, each printed circuit board was subjected to "in-line" environmental thermal cycling in which individual units were exposed to five thermal cycles ranging from (-54 to 71^{o}C). Thermal stabilization occurred at each temperature extreme for 30 minutes per cycle. Data was gathered and improvements generated where necessary. Once the printed circuit boards completed this test program, they were installed in LRU's which were then functionally tested and configured into systems. Four of these systems were used in the formal Reliability Development Test (RDT) program.

As a result of the RIP, LRU reliability grew from an initial instantaneous MTBF (as defined by Duane) of 24 hours to a final instantaneous MTBF of 128 hours.

TEST OBJECTIVES

The objective of the Environmental Reliability Growth Test (RGT) was to achieve an increase in system field reliability by accelerating environmental induced failures, isolating the cause of failure, and determining suitable changes to preclude similar failures in production equipment. To achieve this objective, four sets of redesigned system LRU's were simultaneously subjected to thermal cycling and vibration for an aggregate of 4,237 system operating hours. MIL-STD-781B, Test Level F, was used as a basis for the temperature and vibration environment: however, in the initial phases of the program, less severe temperature extremes were used to aid in isolation.

The basic test consisted of one complete 24-hour temperature cycle period, with numerous power "on-off" cycles. Again, however, early in the program when failures were numerous, more operation at steady-state temperature extremes was necessary for failure investigation. Each set of redesigned LRU's tested was installed in a temperature chamber equipped with a sinusoidal vibration exciter. Other equipment necessary for electrical operation was placed external to the chamber and interconnected by extension cables to the LRU's under test. Figure 2 shows the basic test installation configuration used in the RGT. Periodic checks of LRU performance were made during the low, low-to-high and high temperature portions of the thermal cycle.

A total of 6,933 hours of thermal cycling was accumulated on the four sets of system LRU's. Of this 4,237 hours were accrued with the radar in its operating modes.

PROPER TEST PLANNING IS IMPORTANT

As previously stated, the purpose of the RGT was to achieve maturity through a thorough test/failure analysis/fix/retest program. The length of this test program was based on similar industry programs for which results have been published. Among these, the "Golovin Report"[1] and the widely recognized report published by J.T. Duane[2] present approaches to developing system reliability through such a growth program. The Duane Postulate, which states "as long as reliability improvement effort continues, the Duane mathematical model will be valid", was the basis of the AWG-10 RGT.

The mathematical expression used for modeling the Duane Postulate is:

$$\lambda_{\Sigma} = \frac{F}{H} = KH^{-\alpha}$$

Where:

λ_{Σ} = cumulative failure rate
H = total test hours
F = failures during H
K = constant determined by circumstances
α = growth rate

In practice, the available date (test hours and test failures) is plotted on log paper with total test hours as the ordinate and cumulative MTBF as the abscissa. The constant K and the growth rate exponent (α) are then evaluated from the data.

Generally, an α of 0.6 is considered the theoretical limit of maximum growth and an α of 0.1, the "business as usual" level. General industry tendencies indicate an α of 0.4 to 0.5 is a practical level. For the AWG-10 RGT, an α of 0.46 was chosen for planning purposes. Then a system Duane entry MTBF was chosen. This is the system's operational (not predicted) MTBF prior to RGT and is generally significantly less than the predicted value.

Figure 3 illustrates the planned AWG-10 growth curve. As shown, the final achieved MTBF is approximately 80 percent of the "handbook predicted" MTBF and the planned entry MTBF is 10 percent of the final predicted growth MTBF. This entry MTBF can vary depending on the amount of pre-RGT testing and failure investigation; however, for the AWG-10 RGT, the conventional 10 percent entry point was chosen. The industry-accepted test time of 10,000 hours was used for the AWG-10 RGT.

As illustrated in table 1, the predicted system MTBF was 170 hours. Using the criteria above, the final system growth MTBF over 10,000 hours of operation was to be 140 hours with a Duane entry MTBF of 17 hours. As shown in figure 3, the planned initial Duane entry MTBF (17 hours) occurred at 70 hours system cumulative test time. Generally, with some previous subassembly testing, this is enough system test time to allow for "burn-in" and manufacturing problems to become apparent and be corrected, after which real system reliability growth should occur through design modifications and process changes.

HOW EFFECTIVE IS THERMAL CYCLING?

The thermal cycle used for this test is illustrated in figure 4. In addition to the thermal and mode variations shown, vibration and equipment "on-off" cycling was applied. Sinusoidal vibration at the nonresonant frequency of 22 cps (0.01 inches double amplitude) and applied for 10 minutes every system operating hour. Equipment "on-off" cycling was performed every two hours, during which the equipment was cycled "off" for ten minutes and then turned back to operate, thus generating nine thermal changes in the system. These changes were generated in this manner to preclude the

inefficiencies of system level thermal cycling (4 hours cool down) and still get the benefit of local internal thermal changes at the board and LRU level.

The original thermal cycle was modified four times during the RGT. These modifications were necessary since early test failures severely limited equipment operating time under environment. When the extremes were relaxed, more operating time was accumulated and individual failures systematically investigated. As system reliability improved, the modified thermal cycle modifications and figure 5 correlates these cycle modifications with MTBF improvement.

PLANNED RELIABILITY GROWTH CAN BE ACHIEVED

Table 3 summarizes the data derived from the AWG-10 RIP test. Since the Duane concept for reliability growth on which this program was designed is based on cumulative data, (that is, total failure and total hours at any point in the test program) it was necessary to express the effective MTBF of the units under test at any given time in the growth program. The system instantaneous or current MTBF, defined as the MTBF that the equipment currently on test would exhibit if reliability growth (test, redesign, and fix) were stopped and testing were continued, is expressed as follows:

$$\lambda_{\Sigma} = \frac{F}{H} = KH^{-\alpha} \tag{1}$$

$$\lambda(t) = \lim_{\Delta H \to 0} \left(\frac{\Delta F}{\Delta H}\right) = \frac{\partial F}{\partial H} = K(-\alpha + 1)H^{-\alpha} \tag{2}$$

$$\lambda(t) = \lambda_{\Sigma}(1 - \alpha) \tag{3}$$

$$\theta(t) = \frac{\theta_{\Sigma}}{1 - \alpha} \tag{4}$$

Where:

$\lambda(t)$ = instantaneous failure rate
$\theta(t)$ = instantaneous MTBF
θ_{Σ} = cumulative MTBF

From this model, growth rates were calculated for LRU's 2A1, 2A8, and 5A3. Growth rates for LRU's 2A3, 5, 3 and 8 were not calculated since there were no RIP primary failures in these LRU's, and in theory, the growth rates of these units in RDT would be zero. Also, no growth rates were calculated for LRU 2A2 since no fixes resulting from RGT failures were installed in the unit. Table 3 shows that the overall growth rate, considering primary failures only for the RIP LRU's in RGT was 0.5 with a cumulative system MTBF of 64.2 hours. Using the Duane model for caluclating current MTBF, the instantaneous MTBF for the 8 LRU's is 118.4 hours (64.2/.5). Cumulative test times and failures were plotted at the end of each week. Figure 5 illustrates the growth data resulting from RGT.

Figure 5 shows that the instantaneous or current status curves are lines displaced from the cumulative plots by a factor $(1 - \alpha)$, which is fixed distance on a logarithmic plot. Thus, data can be used for a forecast line of appropriate slope (α) which can be extended from the last data point for predicting and planning additional program elements. For example, if the RGT extended to the planned 10,000 hour limit, with the existing α, the projected instantaneous system MTBF would be 215 hours.

Also shown in figure 5 are:

- o the times when the temperature profile were altered during the test
- o the cumulative and instantaneous MTBF plots based on the number of operational hours per week in thermal cycling
- o the composite calculated growth rate.

The first 108 thermal cycling hours were not included in the calculation of the overall growth rate since the data appears to be "noisy." This 108-hour value compares favorably with the 70-hour entry point for the Duane method. Also, the system growth rate of 0.5 compares favorably with the planned growth rate of 0.46.

CONCLUSIONS

Generally, with any type of management, a degree of reliability can be achieved. As illustrated in figure 3, a growth rate (α) of 0.46 in the specified environment was planned, and an actual growth rate of 0.5 was achieved. However, table 3 shows that individual subassembly growth rates vary as unit complexities and "state of art" design approaches vary. LRU 2A1 had the highest cumulative growth rate at 0.63. In this LRU, the transmitter's pulser had a large amount of failures which were easily detectable and correctable since the majority of its design consisted of standard transistor/diode/resistor circuitry. LRU 2A8 realized the lowest growth at 0.42. This unit, the parametric amplifier, incurred the most failures (36) and its predominant failure mechanism centered in the frequency tunable portion of the unit. This was attributed to such factors as inherent ultrahigh frequency sensitive geometry problems associated with microelectric parametric amplifier design, and incompatibility of materials relative to temperature cycling.

For these reasons, "design fixes" could not be determined as quickly as they could be in more conventional design, thus the growth rate was lower. A growth rate was not calculated for LRU 2A2, the system's power amplifier. Even though the unit experienced five failures, no design changes were implemented as part of the test program since the cause of failure was not determined until well into the program. By the time the proper corrective action could be determined, the test was completed. In spite of this, some LRU growth was experienced. This resulted from the "awareness" of a problem with the unit subsequently resulting in a higher degree of quality control while fixing each failure. This effectively resulted in apparent MTBF growth in the test environment; however, in a "business as usual" atmosphere, this type of growth may not occur unless specifically addressed. Thus, growth in a test program can be experienced in design as well as quality control growth. However, only design growth can be construed as permanent growth.

Finally, one asks the question "How much field reliability improvement can be expected as a result of this type of environmental stimulation?" At present, modification kits have been sent throughout the world to incorporate these new changes in the present deployed systems. Data taken from the 3M report on these deployed systems have shown a 4.5:1 improvement in field reliability for the redesigned RIP LRU's and an overall system reliability improvement of 2:1. This in itself indicates that the RGT approach can be an effective way to improve field reliability.

Thus, three basic conclusions can be made as a result of this program:

(1) The rate of reliability growth depends not only on the degree of management commitment to the program, but also to the unit complexity and state of the art of the unit's design.

(2) Reliability growth can be experienced in two basic forms:
   a. growth in the design (permanent growth)
   b. growth in the quality control procedures (short-term growth).

(3) Reliability growth through environmental simulation is a viable means to achieve improved field reliability.

REFERENCES

(1) Golovin, N. 1962. Final Report of Larte Launch Vehicle Planning Group. General Electric Tempo Report SP 312 Jan. 1965.

(2) Duane, J.T. 1964. Learning Curve Approach to Reliability Monitoring *IEEE Transactions in Aerospace.* 2:563-566.

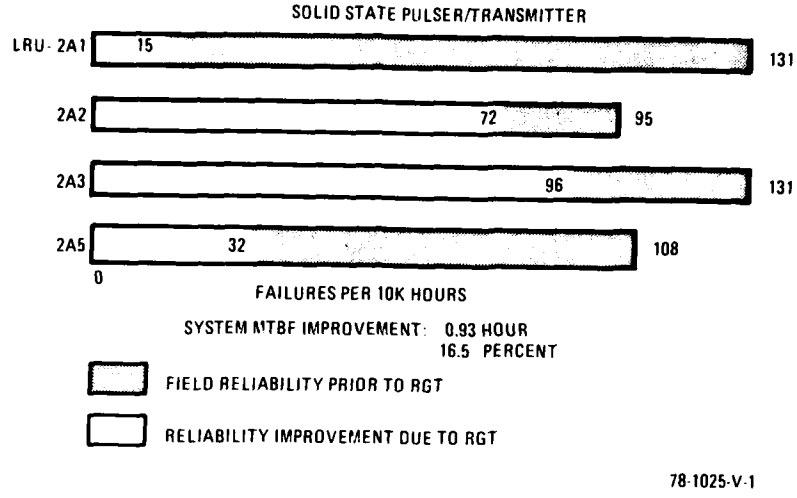


Figure 1. Reliability Improvement RGT Goals Initial Field Reliability

Figure 2. Reliability Growth Test Configuration

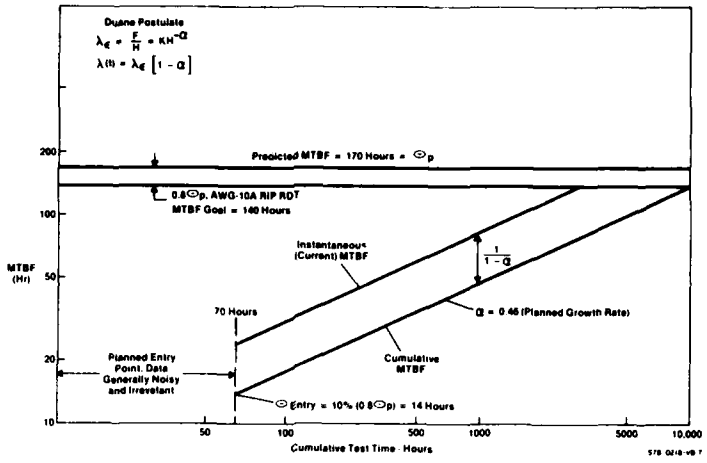


Figure 3. Planned AWG-10A RIP RGT Growth Curve

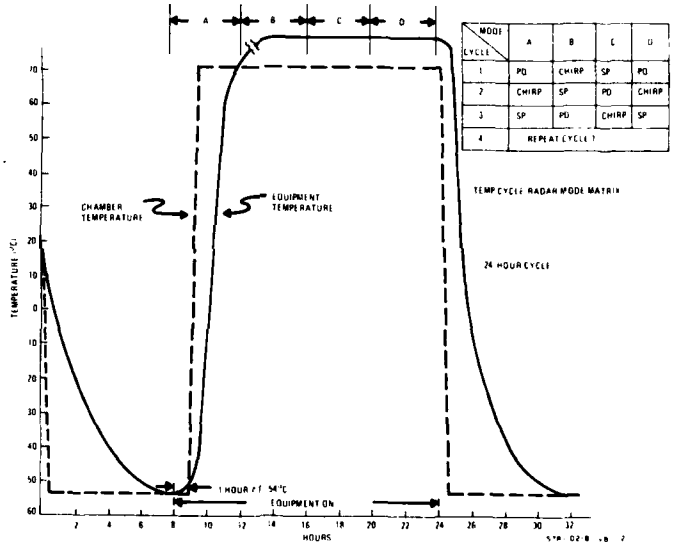


Figure 4. AN/AWG-10A RGT Temperature Cycle

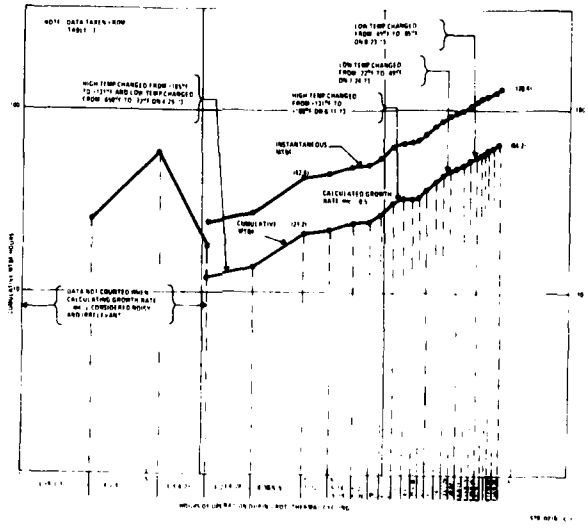


Figure 5. AN/AWG-10A RGT Growth Achievement

TABLE 1
AWG-10 RGT HANDBOOK PREDICTION

LRU	MTBF (Hours)	Failure Rate (Fail/10^6 Hrs)
2A1	6000	167
2A2	315	3174
2A3	15400	65
2A5	11300	88
2A8	1450	690
3	694	1441
5A3	5040	198
8	14000	71
SYSTEM	170	5882

78-1025-T-3

TABLE 2
RGT CYCLE MODIFICATIONS

	Start Test	Modification No.1	Modification No.2	Modification No.3	Finish Test
Thermal Cycle	–30 C to +71C	–30 C to +55 C	–30 C to +71 C	–45 C to +71 C	–54 C to +71 C
System Operate Hrs	119	1007	1022	940	1149
No. of Failures	6	30	15	8	7
Cumulative MTBF per Test Probe	20	34	68	117	164

78-1025-T-4

TABLE 3
AWG-10A RGT MTBF CALCULATED GROWTH RATES AND MTBF'S

LRU	RDT OPERATED HOURS	PRIMARY FAILURE (R/MIP ONLY)	CUMULATIVE MTBF (Hours)	OVERALL GROWTH RATE	GROWTH RATE FOR LAST 75% OF RDT OPERATE HOURS	CALCULATED INSTANTANEOUS MTBF (Hours)
2A1	4237	19	223	0.83	0.78	1050
2A2	4237	5	848	Not Calculated		
2A3	4237	0	No Failures	No Failures		
2A5	4237	0	No Failures	No Failures		
2A8	4237	35	118	0.42	0.53	252
3	4237	0	No Failures	No Failures		
5A3	4237	6	706	0.54	0.54	1540
8	4237	0	No Failures	No Failures		
Overall R/MIP LRU s	4237	66	64.2	0.5	0.50	128.4

78-1025-T-5

DISCUSSION

T.L.Regulinsky, US

Could you give me some idea of the transmitted power you are handling in this equipment?

Author's Reply

It was in the kilowatts but I cannot specify. It was generated by a travelling wave tube (KTS).

J.C.Charlot, Fr

Vous avez parlé d'un essai de 10,000 heures. Pouvez-vous nous indiquer sur combien d'équipements s'il vous plaît?

Author's Reply

We used 4 units for tests on modified LRUs.

P.D.T.O'Connor, UK

Did you find that the effectiveness of the reliability improvement was less than expected, due to the difficulty of incorporating the modifications to equipment already in service?

Author's Reply

Typically this would be the case; however, we recognised this problem before modifications and as a result cycled the units to be modified back through our factory for change.

THE A-7 HEAD-UP DISPLAY RELIABILITY PROGRAMME

Mr K W Boardman

Marconi Avionics Limited
Airport Works
Rochester
Kent
England

SUMMARY

The evolution of the head-up display from earlier forms of weapon aiming techniques is described. The A-7 Head-Up Display is then introduced in terms of the reliability requirement and the contracting environment.

The influence that the reliability requirement had in the introduction of technical innovations is alluded to, particularly in the areas of thermal design, ruggedised long life cathode ray tube technology and durability of low voltage printed circuit connectors.

A complementary nature of the various reliability techniques is suggested together with a generalised cost-benefit statement which relates the reliability programme costs to significant improvements in reliability.

Finally some lessons learnt from this programme will be advanced for consideration in future avionic reliability programmes.

Head-Up Displays can be considered in two ways; either as a new piece of electronic equipment that first saw operational use in the H.S. Buccaneer, or alternatively as a part of an evolutionary sequence spanning half a century.

Very broadly the steps in this sequence were:-

1. The ring and bead sight which enabled the pilot to 'aim-off' during deflection shooting to allow for target speed.

2. The telescopic sight magnified the target but reduced field of view to such an extent that both ring and bead and telescopic sights were often used together.

3. The collimated gunsight simplified target alignment by projecting ring and bead graticules on a semi-reflective glass in front of the pilot thus improving his field of view; this was sufficient until target speeds increased dramatically in the immediate post WWII era.

4. The problem of target speed was overcome to some extent by the electromechanical gunsight, the servoed reticles of which improved aiming accuracy, but their mechanical configuration has always resulted in limited arrangements of non-variable symbology.

5. In the Analogue Head-Up Display the introduction of a cathode ray tube offered a significant increase in display versatility. Symbology could be continuously updated in accordance with aircraft performance, target type and movement. In addition, primary aircraft information could be displayed. Speed, vertical speed, height, distance to navigational waypoint or target, compass heading, the projected path of aircraft or weapon could all be displayed in glowing symbols on the combiner glass in front of the pilot's eye.

6. Finally the replacement of the analogue computer with a digital one yielded yet another step forward in display versatility.

Having briefly considered the evolution of HUD's let us look at the A-7 aircraft.

The A-7 started life as a subsonic strike aircraft derived from the Vought F-8 Crusader. Avionics for the A and B versions were largely 'off the shelf' and the engine was the already proven Pratt and Whitney TF 30-P-6. Even so, the US Navy established a contracting precedent when it ordered the A-7 by requiring LTV to achieve its guaranteed goals or pay stiff penalties.

Many penalties were involved in this fixed price contract. For example, failure to meet sea-level speed and take-off and landing distance guarantees could have cost LTV up to $400,000 (at middle 60's prices), while failure to meet weight empty guarantees was based on a sliding scale per pound excess.

The weight empty guarantee was in fact the only one not met by LTV, the contractor having to pay a penalty for some 600 pounds excess weight, this failure being traded-off against the requirement for a 4,000 hour airframe life.

Because the A-7 provided a stable weapons platform capable of carrying a heavy load over long distances it was chosen for the development of a new navigation and attack system with an accuracy previously unattainable.

Partly as a result of the system study and partly through trade-off studies in the course of competitive tendering the impact of this advanced avionic system on the airframe became known in terms of the requirements for air conditioning, display area, extra space in the equipment bay and increase in aircraft weight.

The contractual reliability requirement for the two-unit HUD system was a staggering (by 1967 standards) 700 hour MTBF with financial holdback until demonstration was complete. Furthermore costs of modifications, both in the factory and in the field, proven necessary by the various qualification programmes were placed firmly in the contractor's court, no post design services as such being provisioned.

LTV's sincerity in the reliability requirement was apparent throughout the contract negotiation, in particular:-

(i) Allocation of cooling air was nearly doubled when preliminary reliability analysis indicated that the MTBF would be otherwise degraded.

(ii) The reliability programme was adequately funded within the non-recurring engineering activity.

(iii) The effects of the reliability programme on extra recurring costs such as component and unit burn-in were accepted.

We were awarded the contract in the Autumn of 1967 and the contractual emphasis on reliability and maintainability was soon apparent as the design got under way.

During contractual negotiations it had become apparent that the thermal design was critical. Although the customer had virtually doubled his cooling air supply to us, we in turn had to utilise it properly. We did this in several ways:-

(i) A cold wall, honeycombed for good thermal transfer, was installed in the mid-section of the computer.

(ii) Heat from all the integrated circuits was extracted through a highly conductive thermal layer on the multilayer printed circuit board to the cold wall.

(iii) Power Supplies were bolted directly onto the cold wall.

Reliability analysis indicated that measures would have to be taken to reduce the potential failure contribution of electrical connections and the cathode ray tube.

This resulted in three main innovations:-

(i) Specific reflow soldering techniques with tight process controls were introduced with particular emphasis on the high integrity soldering of micrologic elements onto multilayer boards.

(ii) An elegant series of controlled experiments was carried out on edge connectors to determine how best to improve their reliability. Only one conclusion could be drawn from these experiments; a minimum of 5 microns of gold was required on the contacts to ensure durability and low contact resistance over a ten year service life.

(iii) As no scheduled maintenance was allowed, considerable time and effort was spent on specialised glass technology to ensure the rugged long-life characteristic of the cathode ray tube.

Fifteen months or so after contract award, the pre-production prototypes were being burnt-in. The burn-in requirement on the units was based very much on the Test Cycles of MIL-STD-781 and we should have continued burn-in testing on each unit until twenty consecutive failure-free cycles were accumulated before the unit was shipped.

It was at this point that the Reliability Programme reached its lowest ebb. The first fifty or so equipments were delivered without meeting their full burn-in requirements. Subsequently the initial in-service data indicated a reliability which was an order of magnitude lower than it should have been. The customer was obviously dissatisfied, R & D funding was being withheld and every sub-standard equipment delivered added to our future liability in terms of retro-fit costs. Furthermore, an initial attempt at reliability demonstration had to be aborted. Yet performance-wise the equipment was just about meeting all the other requirements. Gradually some sense began to emerge from all the noise and disturbance. It soon became apparent that there was a strong degree of correlation between failures of the equipment:-

(i) On burn-in.

(ii) During the abortive reliability demonstration.

(iii) From early in-service usage.

Appropriate statistical data capture showed that the bulk of the failures were occurring in a relatively small number of circuit locations.

A brief but intensive 'get-well' programme soon produced a dramatic improvement in reliability as technical fixes significantly reduced the number of pattern failures.

A second attempt at reliability demonstration was successful and with pattern failures out of the way for all practical purposes we were able to look into process controls more critically and improve the reliability yet further.

Another aspect was brought home to us at this point in time; reliable equipment is cheaper to produce. A fairly obvious point really, but equipment which fails regularly in the field is probably going to fail more often in the factory, and the scrap and re-work cycle is more expensive than many people imagine.

This is of course the parochial viewpoint of an equipment manufacturer. To the user of the equipment the total cost can be conveniently split into two elements. First of all there is the first cost of obtaining the equipment including the R & D and Production. This can be called the Procurement Cost. Secondly, the cost of maintaining and supporting that equipment throughout its life which I shall call the Logistic Cost. The sum of the two is usually called the Total Cost, or Life Cycle Cost.

The effect of equipment reliability on these costs is illustrated in this slide. Although the exact shape of the two curves is difficult to determine with any degree of accuracy, particularly the Logistic Cost, there is little doubt about the general shape.

The Logistic Cost obviously rises very steeply as the reliability falls and the Procurement Cost probably rises steeply as the very high levels of reliability are demanded.

Of course such cost optimisation curves appear in most text-books on reliability, but how does it relate with a real world situation such as this one? Fairly well I think, although I cannot claim more than a first order approximation because inflation and variable exchange rates makes anything more accurate an extremely tedious exercise.

In the middle seventies with a thousand or so head-up displays in-service we calculated that approximately $4M had been spent on the reliability programme to date, but in all probability this was to save over $100M in the first ten years of service life on two items alone:-

(i) Cost to repair.

(ii) Spares.

These figures are somewhat artificial in the sense that the costs of reliability may be real whilst the savings may be appropriated to the procurement of more equipment in an environment where the scale of purchase is often fund limited. I also suspect that in limiting the cost equation to reduced repair and spares costs, these may not represent the totality of benefits that accrue from such a programme, but surely a first order assessment is better than to turn one's back on the problem.

Although I have selected the A-7 Head-Up Display for some special attention, it is by no means our only successful experience on Reliability Programmes properly integrated into the project throughout its life cycle.

In casting back over the last decade there appear to be some factors which are common to programmes where significantly improved equipment reliability has been apparent. I would like to list these for you now without making any claim as to their relative importance.

(a) The Closed Loop Approach

The closed loop approach means measuring deviation from the Reliability Performance Objectives and feeding back corrections.

In this somewhat over-simplified diagram the main stream of activities in the product cycle is represented by progression of the large arrows in the centre from left to right. If one discounts the mathematical folklore of a Reliability Programme, it will be seen that the costs of a Reliability Programme are largely accounted for by a series of servo loops which through 'Failure Analysis and Corrective Action' makes failure rate a decreasing function of time and equipment serial number.

At this point in time, we believe that all these activities are complementary, although with further experience it could be established that some have a greater impact on reliability improvement than others. For the present I suggest that we attempt to minimise the Cost of Ownership to something nearer the optimum before any attempts at fine tuning on a relatively imprecise discipline.

(b) Dedication

Dedication by the customer and supplier means that those in charge of the project in both organisations must believe that reliability is of concern and not simply treat it as a requirement from some specialist which can be dropped when things get tough. This means promotion of the reliability objective from the passive numbers game played by many specialists to the line Manager's acceptance that what he is doing really matters. Implicit in this is a cultural change that envisages months of failure free operation rather than hours.

(c) Contracts, Objectives and Incentives

Quantified objectives are necessary. Not only must the numbers be spelt out unambiguously in the contract, with suitable incentives and penalties where appropriate, but the methods of proving the achievement must also be specified. True there are problems in the absolute measurement of reliability, not only in the validity of clinical laboratory conditions of Reliability Demonstration in relation to those of the field, but because Reliability is in itself such a dynamic parameter. However, as long as Reliability is so significantly lower than its optimum I would suggest that we should not get too excited about precise measurement but concentrate our resources towards really significant improvements.

Perhaps the most important lesson is that all these factors have to be invoked from the inception of the programme. Trying to build on the reliability after the main design features have been established can be very unrewarding in that reliability improvements occur sluggishly, if at all, and far more expensively than if tackled earlier.

In the Aerospace Industry there seems to be a growing consensus of opinion that much of the unreliability of equipment of the past was neither necessary nor desirable. This was brought home to us in many ways, high Cost of Ownership, high Production Costs, a degree of uncompetitiveness internationally and the poor reputation that went with it.

Against this we see the benefits of more reliable equipment. Apart from the obvious desirability of giving the existing performance with more dependability there is the implicit promise that the customer may be able to purchase more equipment because his requirements are often scaled down due to lack of funds, and this can only be exacerbated by lower orders of reliability that swallow up financial appropriation in maintenance and spares.

Finally I will suggest to you several propositions which may go some way towards improving the situation.

1. Unreliability does not occur in a vacuum, it always occurs in the context of systems: Management, Technical, Manufacturing, Procurement and the like.

2. If we accept that unreliable equipment is to a greater or lesser extent a failure of the system that purchases or produces it, then we have to accept that unreliability may not be just 'random' or 'unfortunate' or 'hazardous', but may be an actual output of the system.

3. In order to arrive at a useful understanding of the causes of unreliability in order to reduce it, we will have in turn to consider these systems very closely and be prepared to modify them if improved reliability is a genuine requirement.

4. This improvement must recognise some of the long list of causes of unreliability. The most significant appear to be:-

   Firstly immature engineering design which is considered to be closely related to what, in many instances, appears to be an inadequate amount of development testing.

   This is compounded by the all too prevalent incorporation of unsuitable and defective components and materials into equipment.

   There is a scope also for better manufacturing planning and control, vis-a-vis reliability, to ensure that the reliability levels achieved in design are maintained throughout production life.

   Lastly it is recognised that failures induced by operator mistakes during manufacture and improper use in the field are in fact parts of the broader field of human induced failure.

   Such ergonomic considerations of reliability are, as yet, in their infancy, but evidence is rapidly accumulating to indicate that design discipline in this area can be very rewarding indeed.

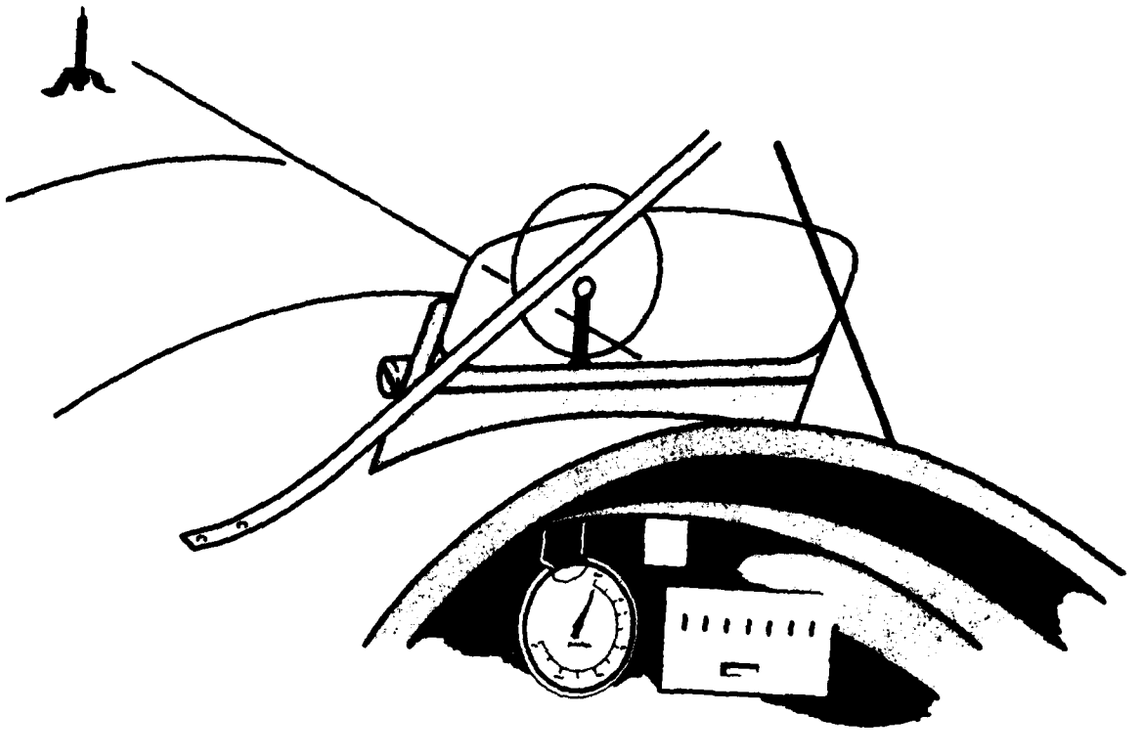


Fig.1 Ring and bead sight

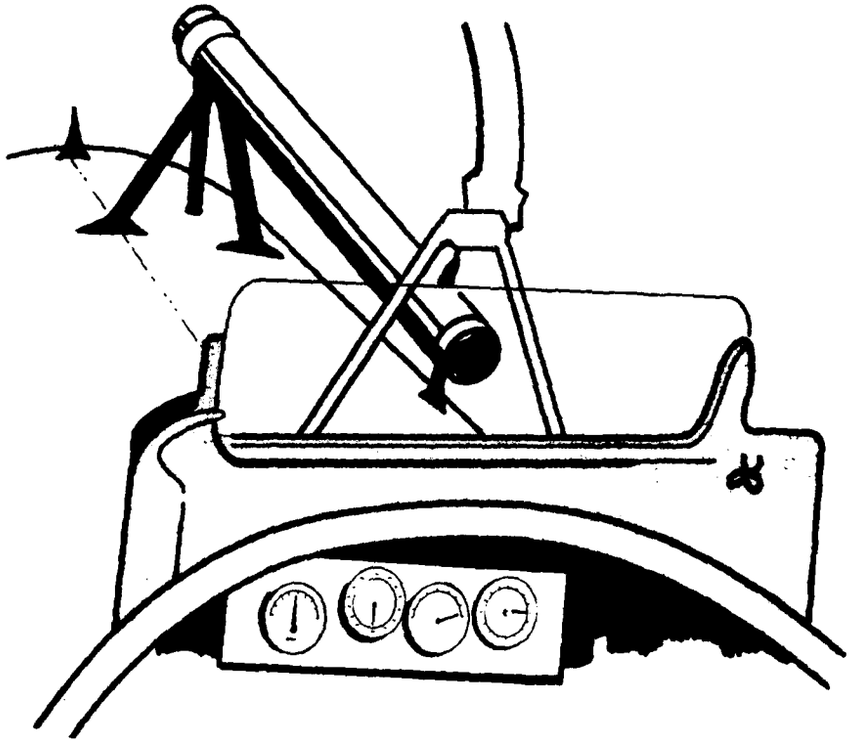


Fig.2 Telescopic sight

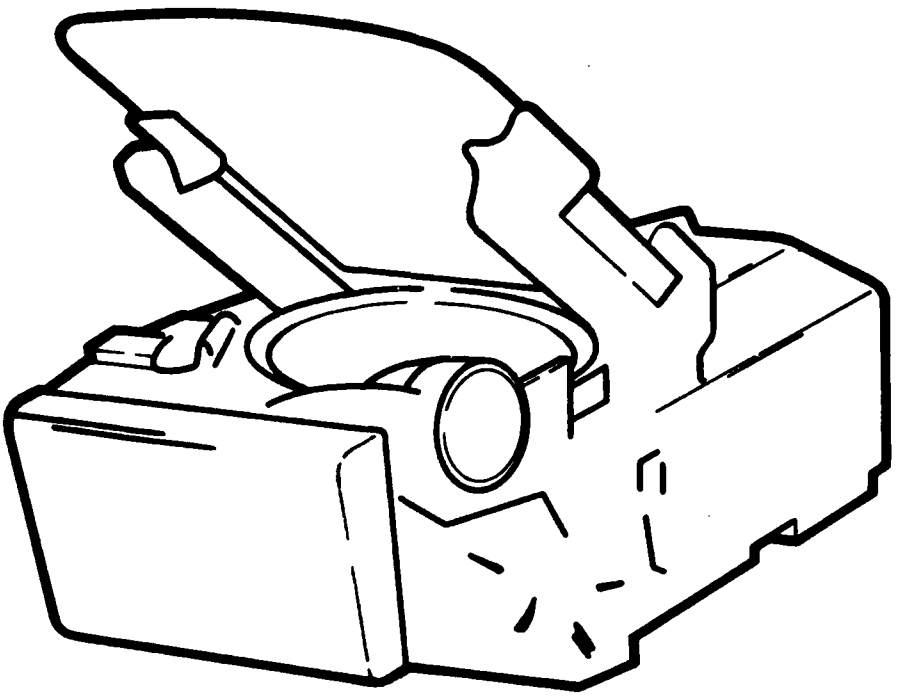


Fig.3 Collimated reflector sight

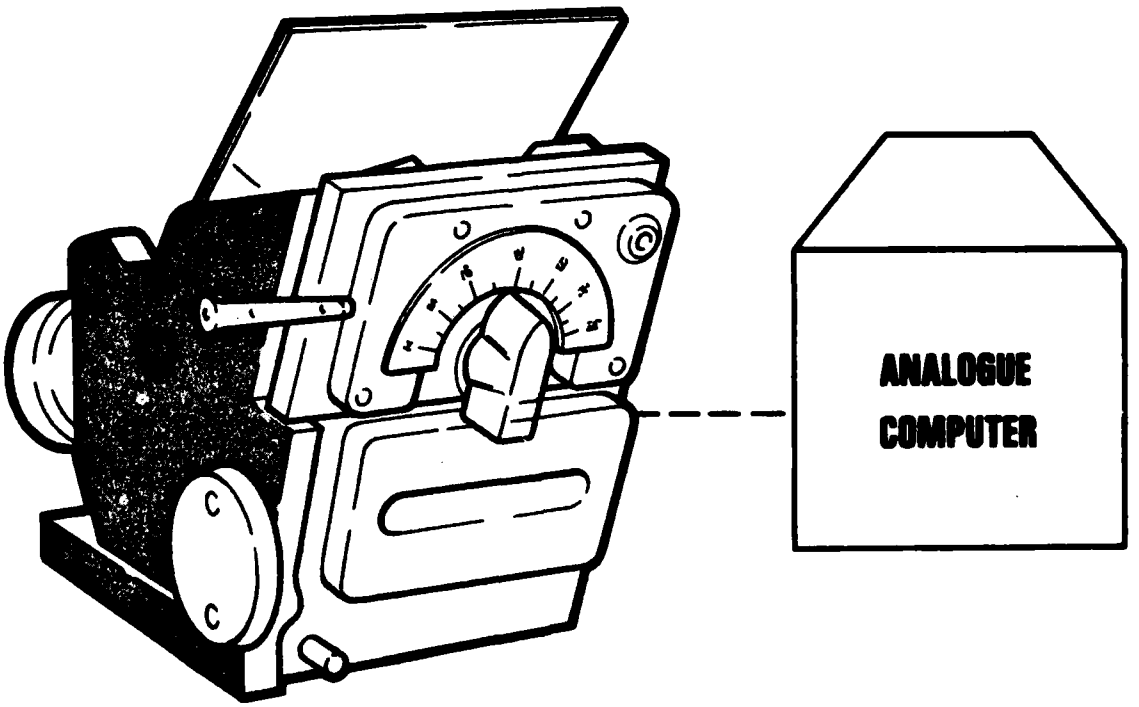


Fig.4 Electro mechanical head-up display

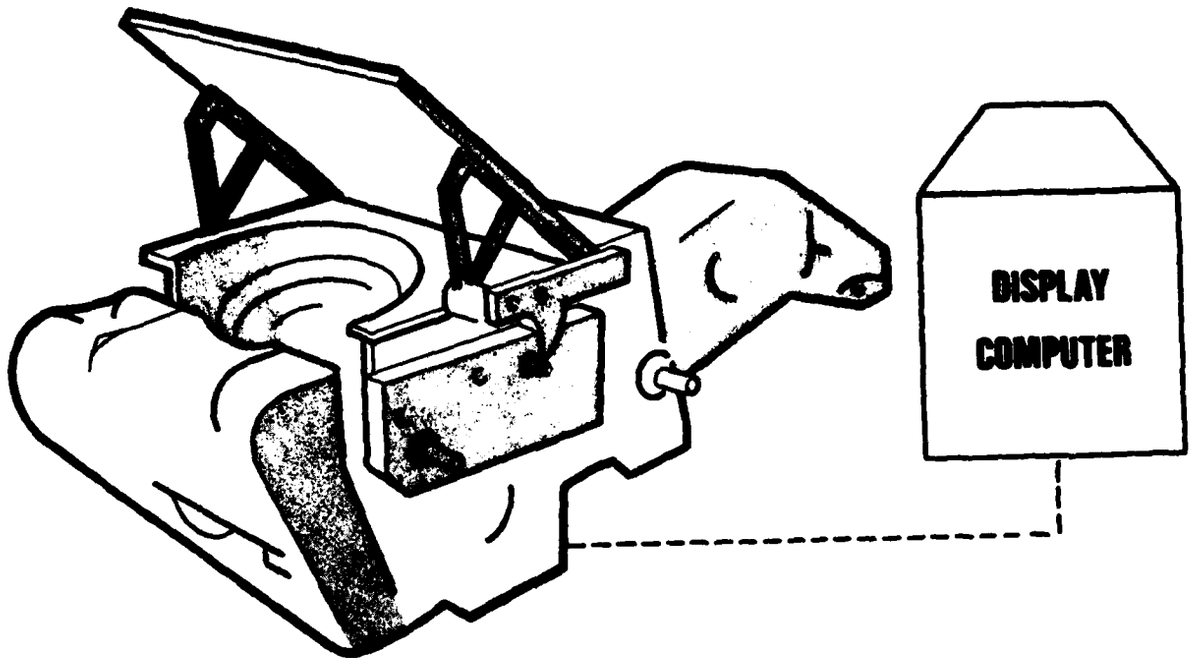


Fig.5 Electronic head-up display

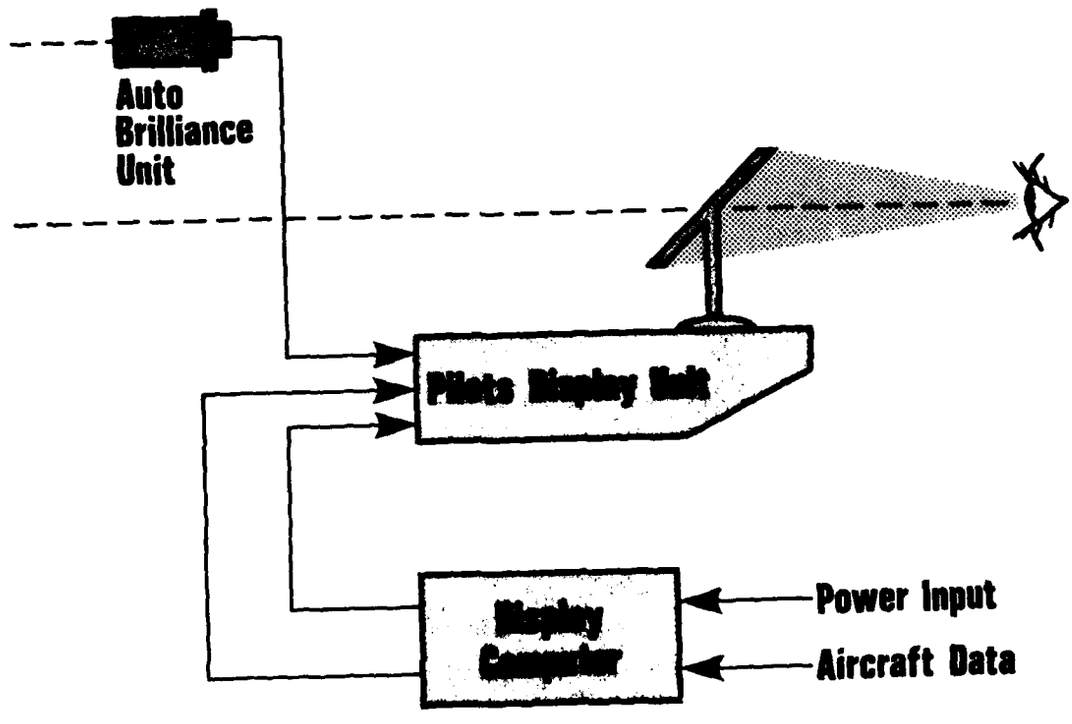


Fig.6 Block schematic of HUD

Air Conditioning	**200% increase in capacity.**
Cockpit	**64 in^2 of display added.**
Equipment bay	**25 ft^3 extra space required.**
Aircraft weight	**500 lb increase.**

Figure 7

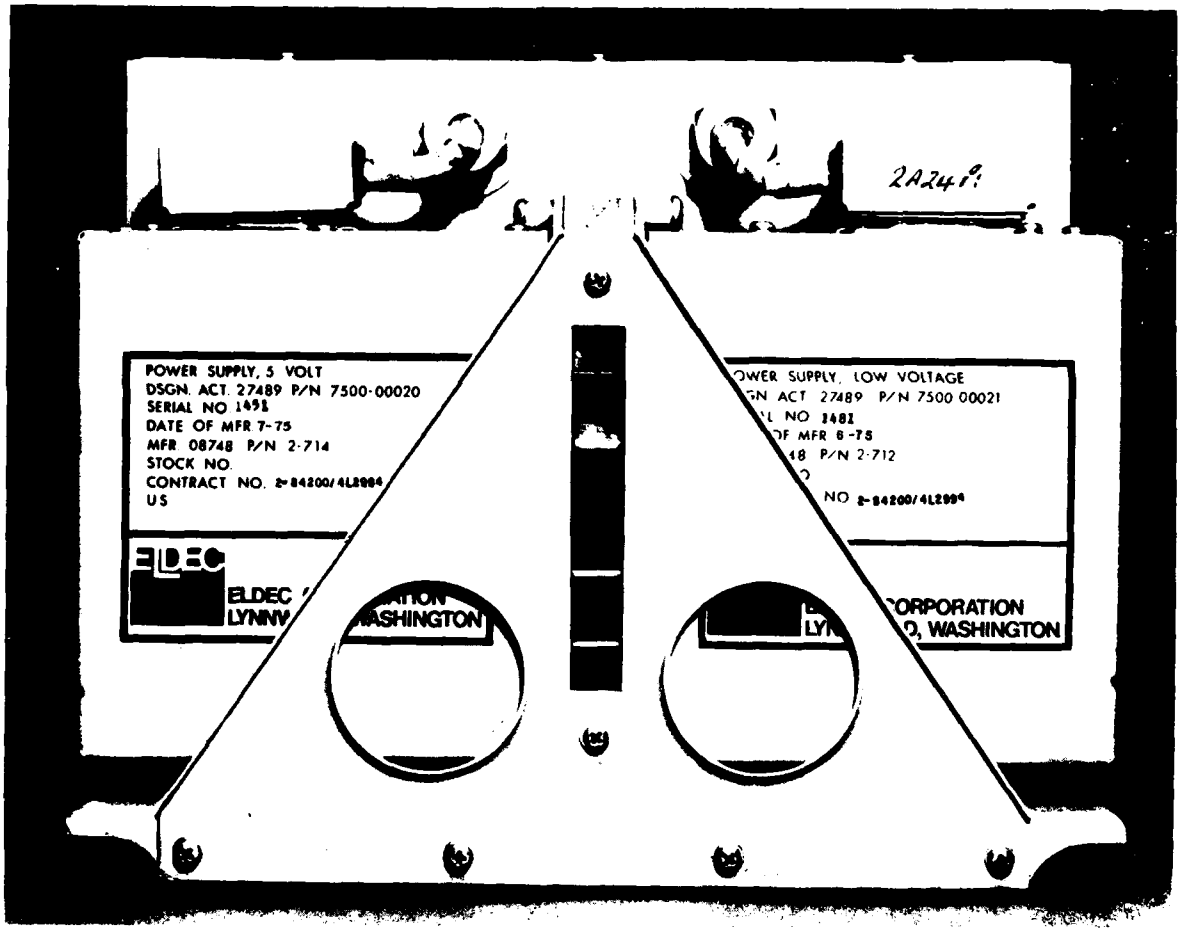


Figure 8

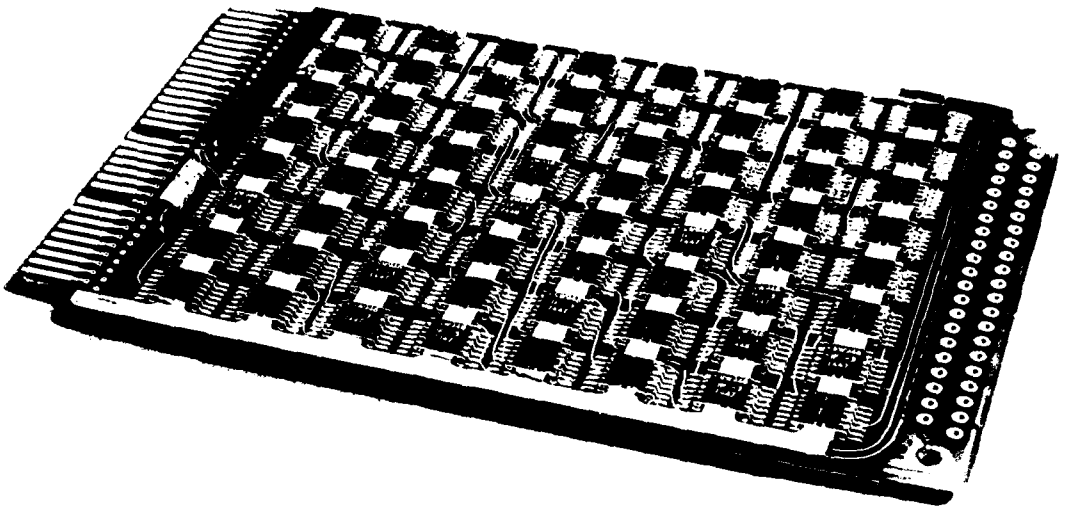


Figure 9

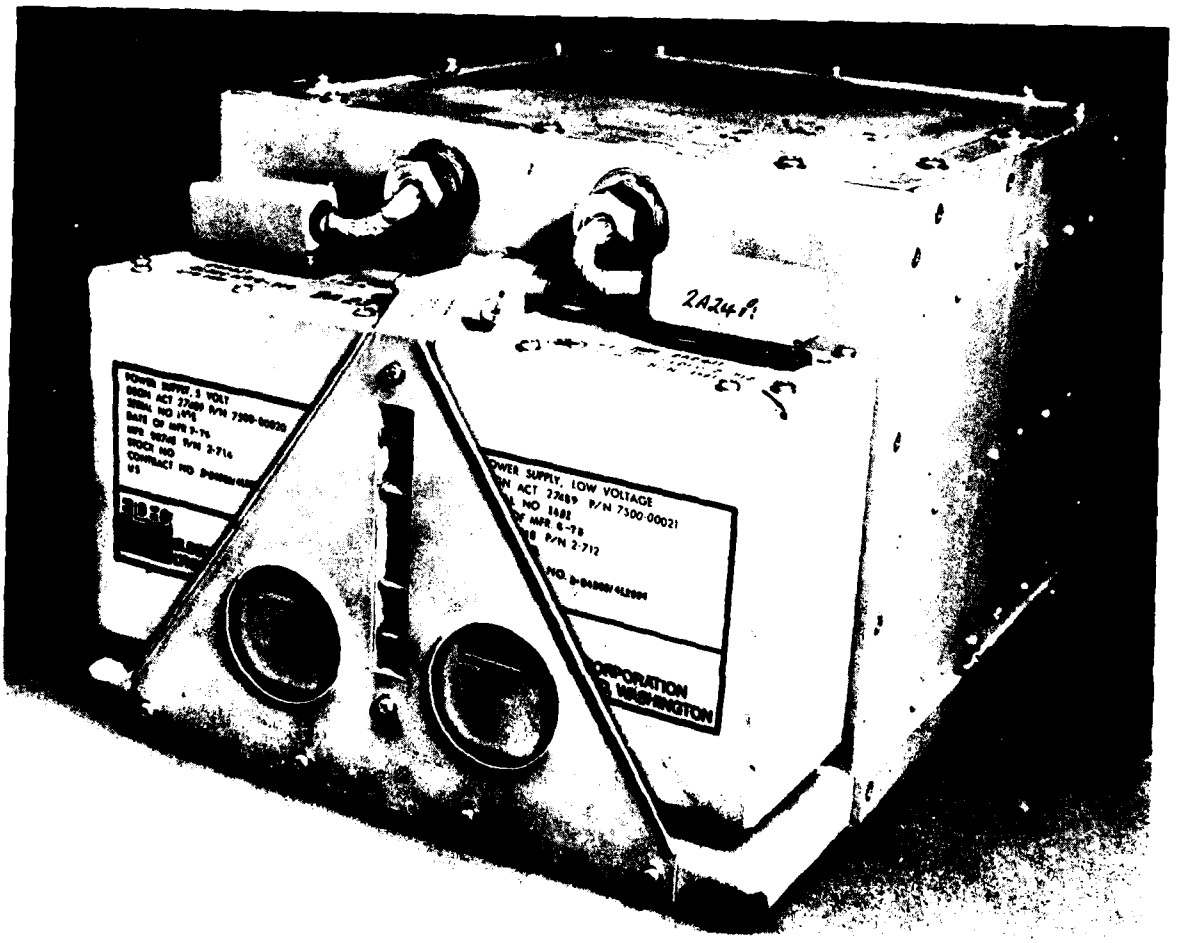


Figure 10

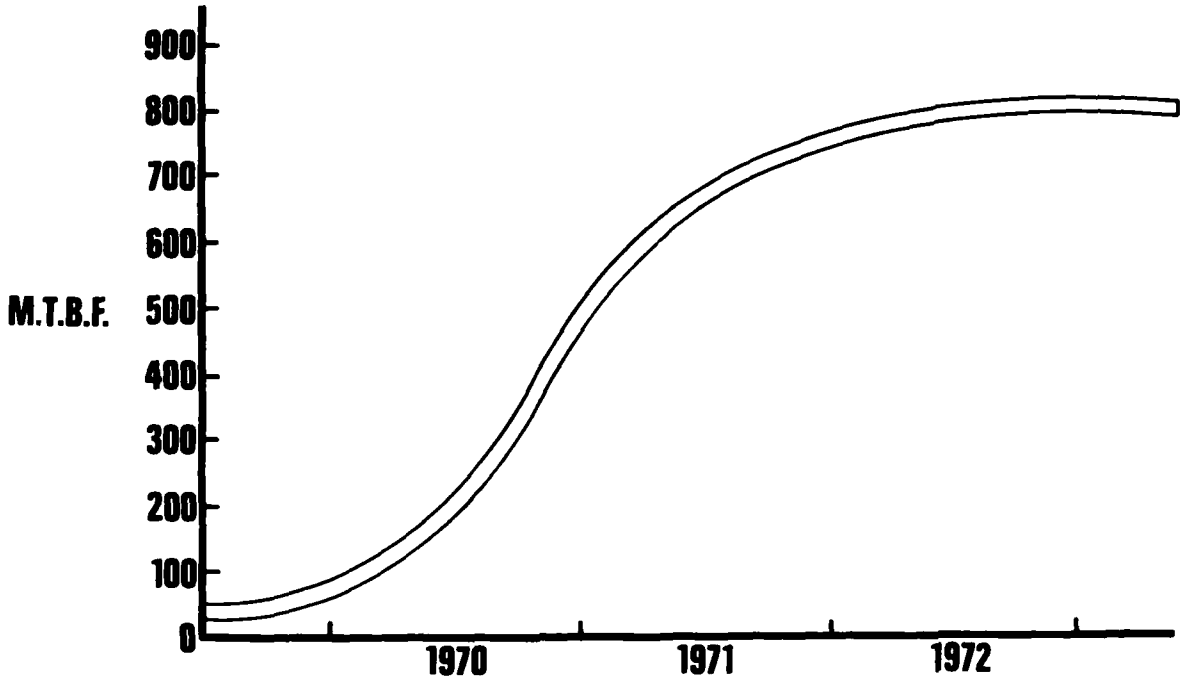


Fig.11 Reliability growth A-7 head-up display

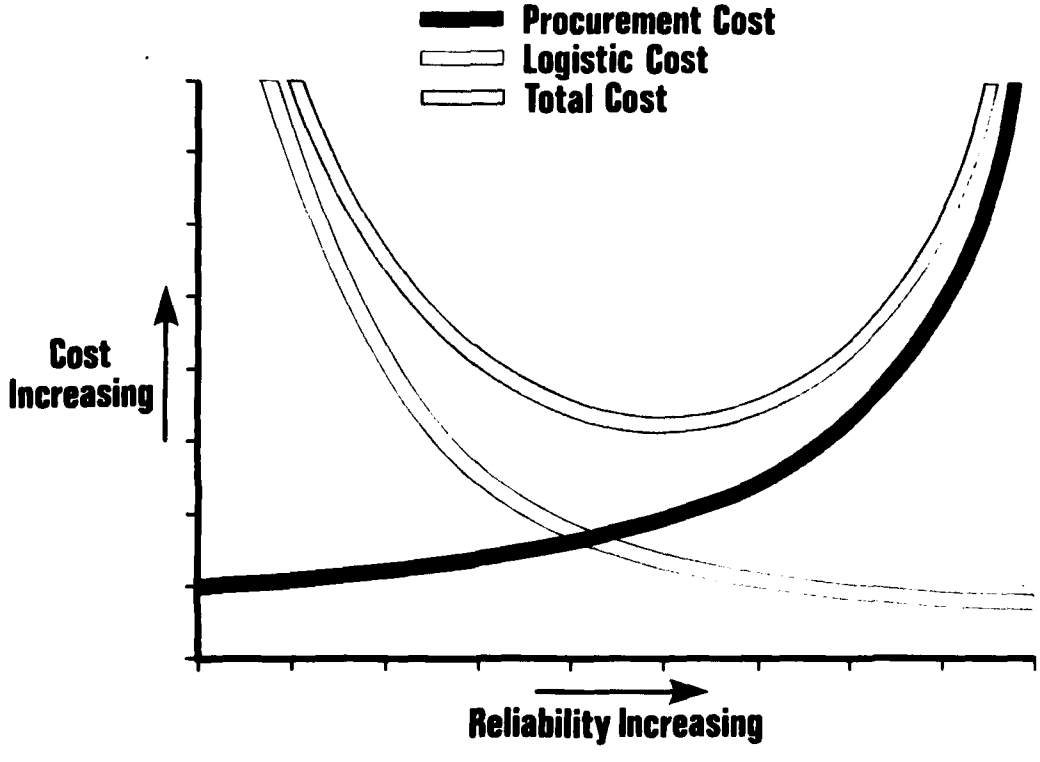


Fig.12 Optimized cost curves

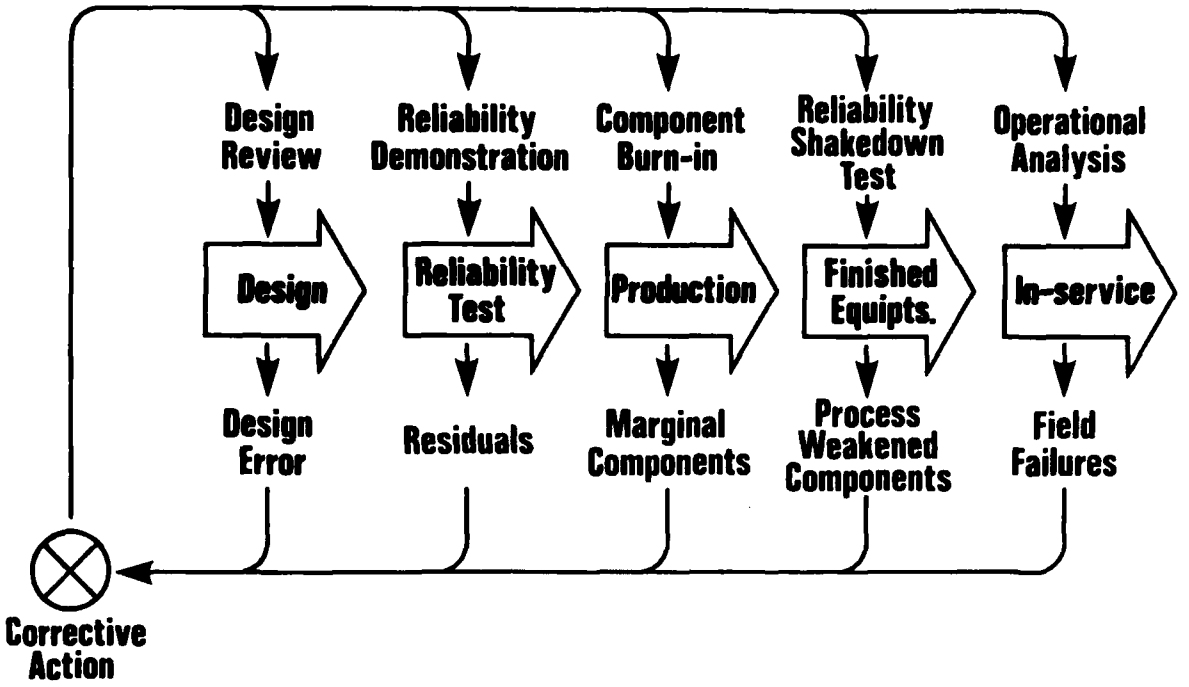


Fig.13 Corrective action closed loop

Dedication

Figure 14

CONTRACTS

OBJECTIVES

INCENTIVES

Figure 15

DISCUSSION

T.L.Regulinsky, US

Could you please tell us how you quantify the life-cycle cost curve?

Is it then fair to say that the life-cycle cost curve was more an exercise in memorisation, i.e. you did not have an analytical expression for that?

Author's Reply

With difficulty. We have certain costs to repair. We had in fact to support this equipment in its first two or three years in operation. We knew the cost to repair and cost of spares and we calculated as if we had been supporting the US Navy (seen as an airline, knowing such things as the number of defects, number of aircraft, flying hours etc. in relation to a simple logistic model; we worked with what we knew to be the costs to repair and the spares necessary for a fleet of that size).

No

F.S.Stringer, UK

A lot of capital facility was required on this project in terms of units which enable you to make the measurements.

Do you see a bottleneck i.e. we don't have enough facilities to enable us to measure the performance of equipment and to get proper burn-in times as well as meeting the specifications?

Author's Reply

Are we talking about the environmental facilities or the complex? The complex, of course, is the environmental facility and has to be backed up by dedicated test equipment ideally capable of a certain amount of diagnostics, and trained operators. So the system or testing system has to comprise these. There are difficulties in obtaining capital, space and trained people. It is a bottleneck as this type of discipline is being adopted very rapidly by the avionics industry.

A.Andrews, UK

In diagnosing the initial cause of unreliability on the A7 HUD, did you require feed-back from the US Navy defect recording systems, or rely on your own repair line data?

Author's Reply

The feed-back came mainly from our own repair facility in Atlanta. The data from the customer was mainly corroborative.

P.D.T.O'Connor, UK

In the period of the development program there were no UK standards for high reliability ICs. Could you say what measures were taken to ensure the reliability of ICs and what failure rate was achieved?

Author's Reply

You are correct in stating that there were no UK standards for high reliability devices, there were no US standards either. We did in fact work to project standards which anticipated later military standards.

These were quite successful in that an operational failure rate of just over 0.2 parts per million was achieved. This was incidentally several orders of magnitude lower than that found on burn-in.

F.Wishart, UK

What impact does the author believe Mil-Std-781C will have on the presented situation from the point of view of facilities and effect on reliability?

Author's Reply

No experience of programs involving 781C, but impact on cost and availability of facilities will be very significant. I would expect some improvement in reliability due to more severe test environment but feel we should have learned to fully exploit 781B before starting along yet another road.

M.W.Watts, UK

You stated the need to safeguard reliability standards during production. This morning we were given a presentation on Production Reliability Assurance Testing. Could you give us a contractor's view of PRAT?

Author's Reply

The evidence for or against Production Reliability Testing is not too abundant at the moment. It will probably require extensive additional resources in terms of Environmental Facilities, Space, Staff and Electrical Energy. Whether this will produce significant easement in the contractor's ability to sustain reliability levels is conjectural at the moment. I suspect it may be over-doing things to apply it universally but I can visualise situations such as the early production of a new design where it could be beneficial.

MILITARY ADAPTION OF A COMMERCIAL VOR/ILS AIRBORNE RADIO WITH A RELIABILITY IMPROVEMENT WARRANTY

Earl I. Feder
Project Leader
USA Avionics R&D Acty
Ft. Monmouth, N.J.

Douglas L. Niemoller
Systems Engineer
Bendix Avionics Division
Ft. Lauderdale, Fla.

SUMMARY

In the realm of military avionics there is a continual striving toward lowering material acquisition costs, increasing reliability and reducing maintenance costs over the life cycle of the equipment. To attain these goals the military adaption of commercial items is becoming increasingly popular. In addition, the military is instituting new logistics techniques with the goal of improving the field Reliability, Availability and Maintainability (RAM) of avionics items.

To meet U.S. Army contractual requirements for a commercial type VOR/ILS receiver, Bendix Avionics selected their FAA TSO certified RN-242A VOR/LOC and GM-247A GS/MB receivers at its basic building blocks. These receivers were reconfigured to meet form-factor and interface specifications. To comply with all operational requirements Bendix analyzed both in-house and field experiences for the basic sets. This review led to incorporation of improved components and circuitry derived from existing ARINC and newer state-of-the-art VOR/ILS receivers within the commercial field.

The U.S. Army contract with Bendix includes a Reliability Improvement Warranty Clause (RIW). The terms and conditions of the RIW Clause require Bendix to assume responsibility for the field reliability and repair of each receiver for a minimum period of four years. Concurrent with the implementation of the RIW, the U.S. Army is studying the overall effectiveness of RIW vs alternative forms of maintenance.

LIST OF SYMBOLS AND ABBREVIATIONS

FAA - Federal Aviation Agency
ICAO - International Civil Aviation Organization
VOR - VHF Omnidirectional Ranging
LOC - Localizer
GS - Glideslope
MB - Marker Beacon
RIW - Reliability Improvement Warranty
O&S - Operational and Support
O&R - Overhaul and Repair
TSO - Technical Standard Order
ILS - Instrument Landing System
RTCA - Radio Technical Commission for Aeronautics
MTBF - Mean Time Between Failures
CDI - Course Deviation Indicator
ARINC - Aeronautical Radio, Inc.
ECP - Engineering Change Proposal
ATR - Air Transport Rack
LCC - Life Cycle Cost
EMI - Electromagnetic Interference
EMC - Electromagnetic Compatibility
RMI - Radio Magnetic Indicator

1. INTRODUCTION

In May 1975 a contract was awarded competitively to Bendix Avionics, Ft. Lauderdale, Florida, for a quantity of 2000 AN/ARN-123(V) VOR/ILS Receiving Sets*. These sets provide U.S. Army fixed and rotary wing aircraft with a relatively low cost, small, lightweight airborne navigation receiver capable of meeting current and proposed FAA and ICAO regulations in this category. The radios are required to enable U.S. Army aircraft to fly civil airways in the United States, Western Europe and other areas where this type of navigation is available. A photograph of the AN/ARN-123(V) is shown in Figure 1 and the R-1963/ARN in Figure 2. A summary of key operational/performance characteristics of the respective receivers is presented in Table 1.

*Under the same contract a quantity of 1000 R-1963/ARN GS/MB Receivers was procured to augment existing AN/ARN-82 VOR/LOC Receivers.

TABLE 1

OPERATIONAL/PERFORMANCE CHARACTERISTICS

DESCRIPTION	AN/ARN-123(V)	R-1963/ARN
Operating Frequencies	VOR/LOC: 108.00 - 117.95 MHz GS: 329.15 - 335.00 MHz MB: 75 MHz	- Same Same
No. of Channels	VOR: 160 (50 KHz Spacing) LOC: 40 (50 KHz Spacing) GS: 40 (150 KHz Spacing)	- - Same
Manual VOR Bearing Accuracy	± 0.75°	-
Rotor Modulation Protection	Yes	-
Self Test	Yes (VOR, MB)	-
Size	Less than short 1/2 ATR IAW ARINC 404	Less than short 3/8 ATR IAW ARINC 404
Weight	12.0 lbs	4.9 lbs
All Solid State - No Moving Parts	Yes	Yes
Power Requirements	27.5 VDC @ 39W 26V @ 400 Hz @ 21VA	27.5 VDC @ 7.5W
Temperature	-46°C to +55°C	Same
Altitude	30,000 Feet	Same
Vibration	5G (Max)	Same

2. TECHNICAL APPROACH

The Bendix Contract required that the AN/ARN-123 be a 50/150 KHz "split" channel receiver capable of meeting current (and proposed future) RTCA requirements, have reduced susceptibility to rotor modulation, have improved EMI characteristics in excess of DO-138 and demonstrate a minimum MTBF of 700 hours at maturity. The associated RIW Clause required that an MTBF and maintenance reporting/assessment system be established to provide the U.S. Army and Bendix with a method of monitoring the status of the RIW program.

3. DESIGN HIGHLIGHTS

The RN-242A and GM-247A receivers were designed to meet the requirements of RTCA documents DO-114 (VOR), DO-131 (LOC), DO-132 (GS), DO-143 (MB) and the DO-138 Environmental Requirements. For the AN/ARN-123, Bendix used existing receiver sub-assemblies, with relatively minor modifications, to meet these RTCA specifications and DO-153 which has superseded DO-114.

The VOR/LOC converter design was updated to provide the desired rotor modulation immunity and bearing accuracy. An electrically synthesized, solid state, automatic bearing output was incorporated. This new circuitry was adapted in part from Bendix RVA-33A VOR and RIA-32A ILS receivers (designed per the latest ARINC requirements) and from the new generation Bendix BX 2000 series models.

The VOR/LOC converter utilizes high-performance operational amplifiers in active-filter and phase-shifting networks and high-stability resistors and capacitors to assure bearing accuracy and stability with time and temperature. The deviation and alarm output circuitry also utilize high-performance operational amplifiers in voltage-source outputs which allow operation with differing numbers of Course Deviation Indicators (CDI's) without requiring external load resistors.

The marker beacon printed circuit board was modified to provide improved preselector and lamp filter circuits to meet the specified EMI requirements of MIL-STD-461. The lamp filters were designed using the same component and circuit types as used in the VOR/LOC converter.

The sub-assemblies, with the exception of the power supply/RMI driver, were mounted on two hinged-interface boards shown in figure 3 which fold out of each side of the chassis. These interface boards provide cost effective, reliable interconnections for the subassemblies, termination for the front panel connectors and a mounting base for shields (where required).

The Power Supply/RMI Driver was mounted on a heat sink on the rear of the unit to allow dissipation of heat generated by the voltage regulators and RMI-output transistors. This subassembly connects to the interface boards by a small wiring harness assembly, the only wiring harness in the unit.

3.1 Rotor Modulation

Susceptibility to Rotor Modulation, (also referred to as Clarksburg Effect) is amplitude modulation of the VOR signal in space by helicopter rotor blades. This susceptibility was a prime concern of the U.S. Army. Many helicopters have rotors that operate in a frequency range around 30 Hz; the bearing signal frequency of VOR. Operation in this range can result in either a bearing error or the presentation of an unstable bearing display to the pilot. This effect cannot be entirely eliminated. However, by judicious selection of receiver AGC-response time, VOR-converter processing circuitry and optimized output-damping circuits the effect of rotor modulation can be greatly reduced. This reduction allows operation of VOR receivers on helicopters as long as the rotor is not operated in the critical range of 29.5 Hz to 30.5 Hz.

Based upon the minimum performance requirements of RTCA DO-153 and known operational requirements the selection of the AGC-response time and output-damping circuits are readily accomplished. However, the receiver overall performance in many applications can still remain unacceptable even after these characteristics have been ostensibly optimized. Additional performance improvement was obtained by modifying the VOR-converter circuitry. The Bendix approach was in the utilization of 400 Hz VOR output signal processing. This technique includes a 30-Hz synchronous detector, that first detects the 30-Hz VOR bearing signal and then applies it to a 0.1 Hz low-pass filter. The D.C. output is then chopped by 400 Hz to provide the output signal for the CDI and RMI. The low-pass filter provides high attenuation to signals as near as 0.3 Hz to the 30 Hz signal allowing the time constant seen by the pilot, when adjusting the bearing selector for the CDI, to be minimized while providing the damping necessary to obtain a stable deviation output signal.

3.2 Electromagnetic Interference (EMI)

Electromagnetic Compatibility (EMC) is an important and critical parameter on a military aircraft. To assure that the AN/ARN-123 did not interfere with and was not susceptible to interference from other sophisticated electronics systems on U.S. Army aircraft the receiver was required to have 20 to 40 dB less emissions or susceptibility than required by RTCA DO-138. Bendix provided this improved EMI performance using commercial techniques, without the use of expensive EMI filters and EMI gasketed dust covers. Chip or disc ceramic capacitors and ferrite beads were used rather than fragile EMI filters. In situations where circuits required shielding printed circuit board-mounted shields were employed utilizing the ground plane of the board as one side. This approach to the EMC solution has proven to be both cost effective and reliable.

3.3 Reliability

The use of existing commercial receiver techniques in the design of the AN/ARN-123 provided a cost effective approach to meet the required MTBF of 700 hours. By using existing commercial receiver circuitry, Bendix was able to meet the MTBF requirement utilizing commercial components which have proven reliable in these receiver applications with no special screening or testing. Only the normal receiving inspection and contractually required equipment burn-in tests were performed. The AN/ARN-123 was subjected to a 48 hour burn-in, the last 24 hours of which are required to be failure free. The set had to meet designated MIL-STD-781B Reliability Tests. These reliability tests have demonstrated a cumulative MTBF in excess of 1000 hours.

At this time the field returns and data are inadequate to determine operational MTBF. However, on a Life Cycle Cost (LCC) demonstration currently conducted by the U.S. Air Force with A-37 aircraft on a VOR/ILS receiver of similar design (AN/ARN-127), the MTBF has reached 800 hours over a eight month period and is continuing to mature. Based on the reliability demonstrations and other field experience, the AN/ARN-123 is expected to mature to an operational MTBF in excess of 1000 hours.

3.4 Data Collection

Inherent within any reliability requirement is a reporting system which facilitates the identification of problems and monitors the reliability of the equipment. To assure that visibility of problems was provided to both Bendix and the U.S. Army, the contract required submittal of several production and field performance reports. The U.S. Army was given Failure Analysis Reports on all production burn-in and sampling test failures. A computer program was developed to facilitate analysis of failure trends within Bendix and the results of all failure analysis reports were entered into the computer. This program aided in the identification of two minor problems which resulted in preparation of the only Bendix initiated ECP's during production of the AN/ARN-123. Bendix generates continuing monthly reports on field performance with annual MTBF assessments which analyze the year's performance in detail. The reports identify returned units by serial number, ECP status, failures experienced, parts replaced, labor required, cost of repair, and hours of operation. The report data will allow the determination of the operational MTBF being experienced and aid in the analysis of the trade-offs of RIW vs organic maintenance by the U.S. Army.

4. THE RELIABILITY IMPROVEMENT WARRANTY

For many years, commercial airlines, for their avionics equipments, have successfully employed the manufacturer warranty repair concept. As a result of these favorable experiences, HQ Dept of the Army directed that the AN/ARN-123(V) be utilized as the pilot program to introduce and test the long term RIW concept for U.S. Army military avionics. (It should be noted that subsequent to the AN/ARN-123 program, a number of other Army avionics equipments were procured with an RIW Clause).

4.1 What is RIW?

Simply speaking, RIW is a concept that commits the contractor to provide total repair support of his equipment for a relatively long period of time (typically four or more years) at a contractual line item bid price. The contractor then shares some of the risk of fielding the equipment.

4.2 Advantages

For the U.S. Army, the RIW offers monetary incentives to the contractor to ensure that his equipment improves in reliability and maintainability after deployment to the field. Improvements in these areas can lead to considerable operational and support (O&S) cost savings. It is estimated that the O&S portion of life cycle costs can vary up to ten times that of the initial material acquisition costs.

For the contractor, the RIW concept provides an excellent opportunity for additional profits if the equipment's reliability and maintainability goals are exceeded; but, as might be expected, losses can be incurred if they are missed. Therefore, the contractor has the continuing incentive to improve field reliability so that equipment failure problems are minimized in order not to have repeated returns to the plant for repair.

4.3 RIW vs Organic Maintenance

The RIW offers a number of potential advantages over in-house (i.e., organic) maintenance. These include the following.

1. The contractor must quote a fixed price for performing total repair services during the initial designated RIW period in a competitive environment. For the AN/ARN-123 program, the RIW cost is about 2.6% per year of the hardware acquisition cost over the initial four year period.

2. Because all failed sets are returned to the contractor for repair, his engineers can analyze failure modes and patterns and take more expeditious corrective action than that possible with depot repair.

3. Initial software costs are substantially reduced as comprehensive Technical Manuals, depot training, provisioning, etc., associated with organic maintenance are not required. It is apparent that if the RIW is implemented for the life of the equipment, then these expenditures will never be necessary.

4.4 AN/ARN-123 RIW

Highlights of the RIW Clause in the Bendix Contract are shown in Table II. Note that the clause contains incentives for both the Army and contractor to improve the reliability, maintainability and the logistics flow of the equipment. If contractual performance criteria are not met, either party will incur penalties.

TABLE II

AN/ARN-123 RELIABILITY IMPROVEMENT WARRANTY CLAUSE

1. Forty-eight months starting upon Gov't acceptance of each receiver.
2. Contractor repairs or replaces equipment failures for price bid on contract.
3. Exclusion clause for Gov't induced damage or failure-Gov't pays via overhaul and repair (O&R) contract.
4. Contractor must repair and re-ship equipment within 20 days or be subject to penalty.
5. Gov't subject to penalty for 30% or greater unverified failures in given period.
6. No cost ECP's to improve reliability and maintainability encouraged.
7. Joint contractor-Gov't inspection of all incoming receivers.

TABLE II

AN/ARN-123 RELIABILITY IMPROVEMENT WARRANTY CLAUSE (Continued)

8. Contractor maintains warranty data accumulation, analysis and reporting system.

9. Initial decision to extend RIW or enter alternative maintenance made two years after initial production start.

4.5 Contractual Adjustments

In addition to the above, the typical RIW Clause may contain one or more of the following adjustments.

(a) Average Operating Hours/Month (AOT) -- If the equipments field AOT utilization is determined to be above a designated contractual figure, the contractor will be payed a certain pro-rated monetary sum; whereas, if it is below, he will have to rebate the Army.

(b) MTBF Guarantee -- If the field MTBF falls below a guaranteed number, the contractor is penalized. The penalty may take the form of providing more spares, additional quality assurance testing not previously programmed, or an extension of the warranty period.

(c) Lost or destroyed sets -- If a set is lost or destroyed, credit for the unused portion of the RIW is given to the Army.

4.6 RIW Logistics

Three types of material logistics and accountability procedures in the repair cycle are currently planned for RIW utilization. Each will be studied for its overall effectiveness. They are the following:

(a) The receiver, which is returned by the user unit to the plant, is repaired and sent back to the same unit. All the float stock is deployed to the field. This approach is being utilized for the AN/ARN-123(V) program.

(b) A bonded warehouse, stocked with a relatively small number of float sets, is set-up at the contractor's plant. When a returned set is received at the plant, another is immediately removed from the warehouse and forwarded to the user unit. The returned receiver is then repaired and placed in the warehouse. With this procedure the turnaround time is thereby reduced by the time necessary to repair that particular set.

(c) A bonded warehouse is also set up that typically contains the bulk of the float stock. After failure of a set in the field, the user unit sends an electronic message (i.e., Telex, TWX, etc.) to the contractor's plant. Upon receipt of the message, the contractor immediately removes a receiver from bonded storage and ships it to the user unit. Meanwhile, the unit expeditiously initiates the return of the failed set to the contractor. With this approach the turnaround time is further reduced by both the time necessary for shipment into the plant and the time required to repair the set.

4.7 Contractor In-Plant Handling

Most sets returned to the plant will be classified as normal RIW returns -- that is, they will be repaired, retested and repackaged for shipment. However, a number of their sets will be "excluded" from coverage in accordance with the terms and conditions of the RIW Clause. Reasons for exclusion include physical damage, broken seals, willful mistreatment, etc. Sets falling into this category are paid for under a separate contractual overhaul and repair agreement.

One of the areas that may result in significant contractor penalties is the repair turnaround time. Therefore, it is vitally important that returned receiver shipments be immediately recognized as being under RIW, categorized as to failure type (verified, unverified or exclusion) by joint contractor and government inspection and rapidly introduced, when applicable, into the repair cycle.

(a) Field Experiences and Observations

It is [illegible] that a large number of returns are not properly documented. For example, it is [illegible] with no identification of the sending unit, apparent failure symptoms [illegible] Such factors as usage rates, MTBF, exclusions [illegible] information. Also, a small amount of sets have [illegible] that unauthorized maintenance may have been

[illegible] maintenance [illegible] supply support
[illegible] [illegible] without proper
[illegible] [illegible]
[illegible] [illegible]

handling and lengthy delays are not in tune with RIW philosophy. Lack of training results in high percentages of non-verified failures, extended turnaround times, misdirection of shipments and poor warranty documentation. All of these problem areas have an impact upon equipment availability.

4.9 RIW Field Team

Based upon early experiences, it was decided that in order to effectively introduce, teach, and monitor RIW procedures for avionics equipments, a U.S. Army RIW Field Training Team be established under the leadership of a field grade officer. The team's mission is to visit the field carrying suitable documentation necessary to educate and train the troops who are handling RIW avionics in the logistics support loop. In addition, the team serves as a focal point to evolve/promulgate RIW policy and procedures within the Ft. Monmouth, N.J. complex.

4.10 RIW Effectiveness Study

With the assistance of ARINC Research, Inc., Annapolis, MD, an RIW Computer Study Model has been developed that will enable the U.S. Army to determine the overall effectiveness of the AN/ARN-123 (and other avionics equipment) warranty as relevant data is programmed. At future maintenance decision points, the computer output will be employed in determining government positions for use in either negotiating further extensions of the RIW with the contractor or, alternatively, the study may recommend immediate transitioning to U.S. Army organic maintenance. In addition, in the case of the AN/ARN-123, with its similarity to the AN/ARN-127, the possibility of an interservice repair agreement exists. This approach is being investigated as one of the maintenance options.

5. CONCLUSIONS

The initial acquisition cost of the AN/ARN-123 and R-1963/ARN was considerably lower than that projected for a comparable military set. Reports from the field indicate that both receivers are operationally performing in a highly satisfactory manner. Thus, the adaption from commercial to military end use has been successfully accomplished.

It is still too early in the AN/ARN-123 program to assess the overall effectiveness of its associated RIW. Every effort is being made to ensure a fair trial through adequate training of field logistics personnel in RIW policy and procedures.

If successfully implemented the RIW offers the prospect, for all avionics equipments, of increased field Reliability, Availability and Maintainability (RAM) with a consequent reduction in overall equipment life cycle costs.

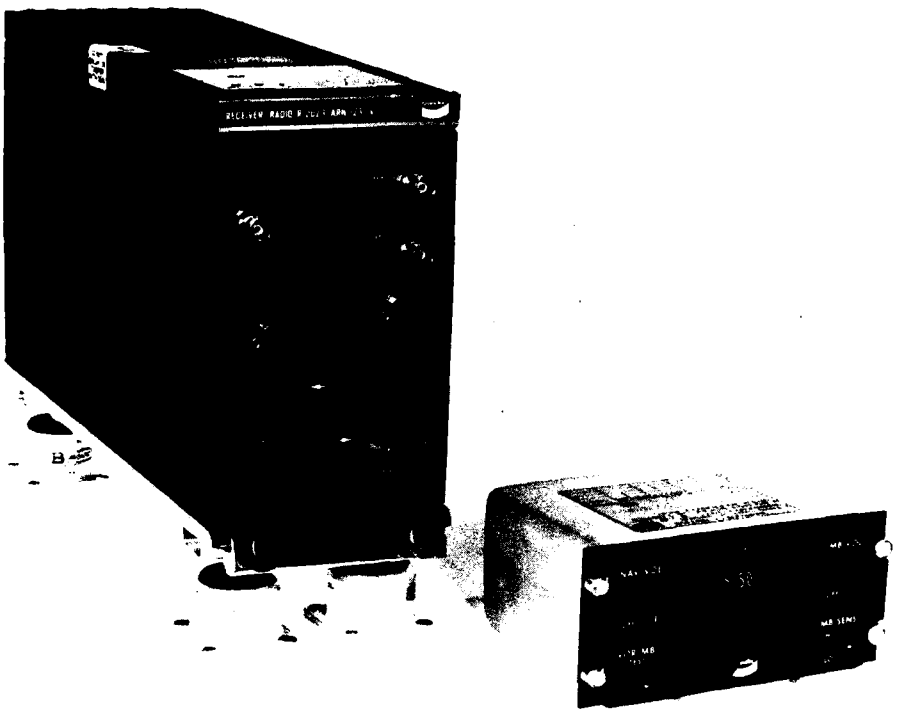


Fig. 1 AN/ARN-123 Radio Receiving Set

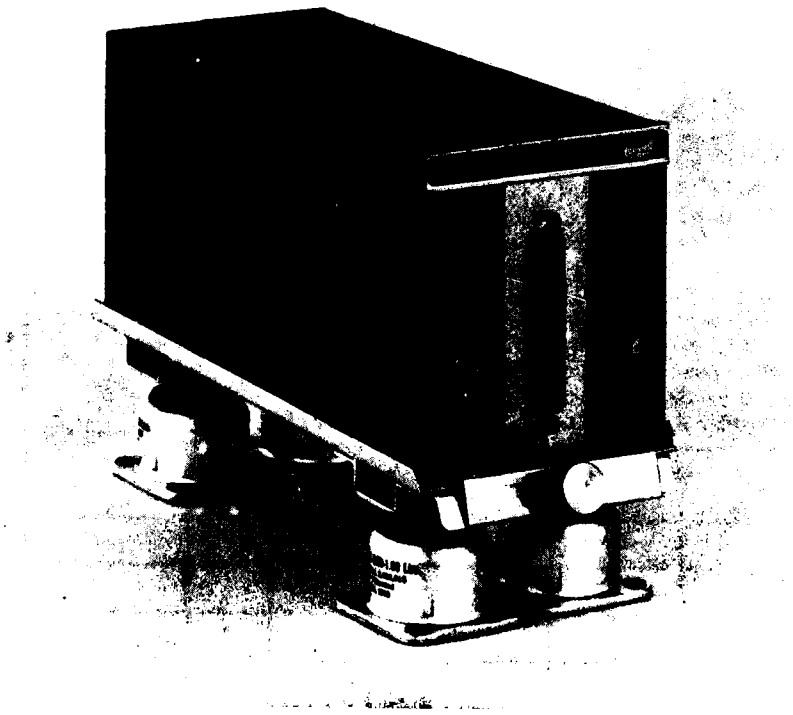


Fig. 2 R-1963/ARN Radio Receiver

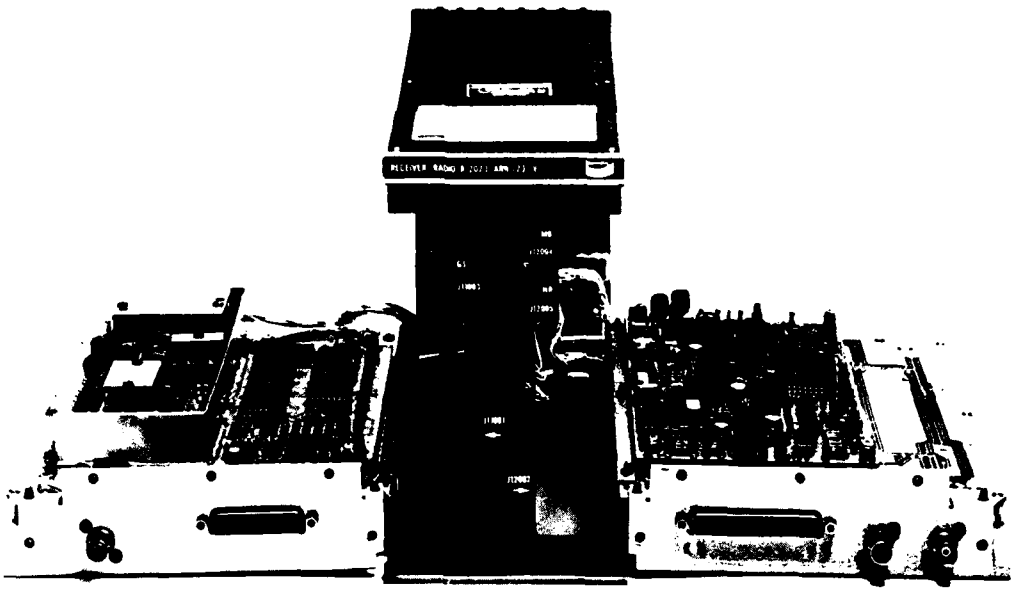


Fig. 3 AN/ARN-123 Radio Receiving Set with Covers Removed

WARRANTY NOTICE Bendix

1. THIS UNIT IS UNDER WARRANTY UNTIL ___/___/___

2. DO NOT BREAK OR TAMPER WITH WARRANTY SEAL.

3. VERIFY FAILURES USING APPROVED PROCEDURES AND TEST EQUIPMENT OF TM 11-5826-258-24
4. RECORD REASON FOR REMOVAL & TEST FINDINGS ON FORM DA-2407
5. PACKAGE IN ACCORDANCE WITH SECTION 11 OF TM 11-5826-258-24 AND RETURN TO BENDIX AVIONICS 2100 N.W. 62nd ST. FT. LAUDERDALE, FLORIDA 33310

C-10048/ARN-123(V) SERIAL NO._____

	INSTALLATION DATA						CONTRACTOR'S USE ONLY CODE NO.
	A/C TYPE	A/C NO.	DATE & TIME TOTALIZING METER				
			IN A/C	TTM	OUT A/C	TTM	
1							
2							
3							
4							

Fig. 4 Typical Warranty Notice

EMULATION APPLIED TO RELIABILITY ANALYSIS OF RECONFIGURABLE, HIGHLY RELIABLE, FAULT-TOLERANT COMPUTING SYSTEMS FOR AVIONICS

Gerard E. Migneault
NASA Langley Research Center
Hampton, Virginia

SUMMARY

This paper proposes that emulation techniques can be a solution to a difficulty arising in the analysis of the reliability of highly reliable computer systems for future commercial aircraft, and thus should warrant investigation and development.

The paper first establishes the difficulty, viz., the lack of credible precision in reliability estimates obtained by analytical modeling techniques. The difficulty is shown to be an unavoidable consequence of: (1) a high reliability requirement so demanding as to make system evaluation by use testing infeasible, (2) a complex system design technique, fault tolerance, (3) system reliability dominated by errors due to flaws in the system definition, and (4) elaborate analytical modeling techniques whose precision outputs are quite sensitive to errors of approximation in their input data.

Next, the technique of emulation is described, indicating how its input is a simple description of the logical structure of a system and its output is the consequent behavior. Use of emulation techniques is discussed for "pseudo-testing" systems to evaluate bounds on the parameter values needed for the analytical techniques.

Finally an illustrative example is presented, albeit for a fanciful small scale application, to demonstrate from actual use the promise of the proposed application of emulation.

INTRODUCTION

Research efforts are underway to develop more efficient civil transport aircraft for the future. One facet of the effort involves active control technology which implies greater reliance upon computer systems in order to obtain maximum benefits. This paper discusses the need and justification of development and investigation of emulation techniques as adjuncts to theoretical reliability analysis models of fault tolerant avionic computer systems.

REQUIREMENT FOR FAULT TOLERANCE

Designs of fault tolerant computer systems have arisen in response to anticipated needs of future civil aircraft (Bjurman, B. E. et al., 1976), (Hopkins, A. L. et al., 1978), (Wensley, J. H. et al., 1978). Requirements for reliability of systems and associated components have been inferred from the expression "extremely improbable" in regulatory documentation pertaining to safety in commercial transport aircraft (FAA, 1970). The following, variously worded, informal statements indicate the range of interpretations:

> "Thus we have a reliability requirement of 10^{-8} per hour of operation for a level 1 or level 2 function with no internal or external backup ..." * (Ratner, R. S. et al., 1973)
>
> "... a number less than or equal to 1×10^{-9} has been imposed ... to represent the probability of an event designated as extremely improbable. ... Loss of the CCV/FBW function, given a fault-free system at dispatch, shall be extremely improbable." ** (Bjurman, B. E. et al., 1976)
>
> "... the computer's failure rate will be designed below 10^{-9} failures per hour in flights of up to ten hours duration, with a preferred goal of 10^{-10} failures per hour." (Smith, T. B. et al., 1978)
>
> "... the extrapolated failure of the design in context with production system application shall not exceed 10^{-9} computer-related system failures in flights up to ten hours." (sic) (NASA, 1978)

As an average of the interpretations, and for discussion purposes, an equally informal statement is adopted here as the requirement, viz.,

> the probability that a system containing no failed components at the start of operation will fail during the first ten hours of operation will be less than approximately 10^{-9}

in which the term "failed components" refers, in a conventional manner, to failures caused by physical defects occurring randomly in time, and in which a system is considered to have failed when it has not correctly performed the function required of it as a subsystem in a larger, encompassing system.

Temporarily disregarding failures due to causes external to systems or to inadequately or incorrectly designed and implemented systems, one can determine that, in order to satisfy the reliability requirement, a computer system constructed of devices (in turn constructed of more basic components) with independent

---

*Levels pertain to criticality of functions.
**CCV/FBW = control configured vehicle / fly by wire.

failure distributions and constant failure rates would require, if it were intolerant of the failure of any of its constituent devices, a mean time to failure (MTTF) of approximately ten billion (10^{10}) hours for the least reliable of the devices. Such a system is unlikely to see the light of day in the near future, to say the least, since realistic, available devices such as processors, memories, etc., from which systems can be constructed, do not have such lengthy MTTF's; values in the range from 10^2 to 10^5 are more reasonable. Consequently, computer systems intended to satisfy the reliability requirement have been designed to tolerate failures.

A CONSEQUENCE OF FAULT TOLERANCE

Several characteristics of fault tolerance give rise to a need to examine explicitly the reliability implications of a failure mode conventionally handled implicitly by testing actual systems.

One rather obvious characteristic of a fault tolerant system is redundancy of components -- at the very least when in an initial condition free of failed components. In the case of systems with requirements for reliability stated in terms of the first few hours or a small fraction of expected equipment lifetimes, the characteristic implies renewal activities which will be often repeated. While some form of verification that systems are still in a (perceived) fault-free condition will be a minimum renewal activity, the MTTF's of realistic, avionic devices insure that a not insignificant amount of repair activity will also be needed -- to return systems to the fault-free, initial condition needed to fulfill the assumptions underlying the reliability estimates. The characteristic further suggests, other things unchanging, that the more "multifunction" the constituent devices are, the more efficient the systems are in terms of total equipment used and maintained. Therefore, there is an economic pressure for designs utilizing multifunction devices such as microprocessors with software. However, a cost is incurred in a different coin, i.e., greater complexity in the synthesis, logic and analysis of systems with parallel and/or intersecting signal and data paths and time-shared use of resources and algorithms.

Another necessary characteristic of a fault tolerant system is its possession of an agent or mechanism capable of detecting failures in devices or components and utilizing available redundancy to nullify failures. This characteristic may be accomplished in a passive manner when some convenient property of nature permits -- a simple example is parallel rather than series wiring of Christmas tree lamps to avoid an open circuit failure caused by one defective lamp -- or, as appears more likely to be necessary in complex systems, in an active manner by the addition to a redundant system of still more devices and/or logic to act as detectors and nullifiers. Of course, a price is paid again in increased complexity.

There is a notion which merits a few words as it occasionally arises at this point. The notion is that the reliability requirement is unnecessarily stringent, as witnessed by the ten billion (10^{10}) hour MTTF previously cited. However, that value was for a fault intolerant system, a "series" system, and is inappropriate as an approximation of the MTTF of a fault tolerant system of equivalent reliability at an extremely early stage of its expected operation, i.e., ten hours, for one reason because the variance of time to failure of fault tolerant systems tends to be much less than that of series systems. For example, Figure 5 compares the failure density distribution of two systems having the same mean (i.e., same MTTF). Density A is a series system. Density B is a representation (specifically a 2 out of 5) of a parallel redundant system. Clearly, at an early stage in their operation, the parallel system has a greater reliability. A better approximation is provided by the MTTF of systems composed of several r-out-of-n subsystems (i.e., n parallel, identical devices of which r must be operating for the subsystem to be operating) in series. A system consisting of a single r-out-of-n subsystem serves as a reasonable upper-bound estimate of the MTTF of a fault tolerant system when the representative constituent device chosen is the fault tolerant system's "worst" (i.e., the device type with the greatest MTTF in the set of constituent devices whose functions cannot be performed by any combination of the other device types of the system; a processor would be in this set). Assuming, as before, that constituent devices have independent failure distributions and constant failure rates, one can show that an r-out-of-n system has a MTTF not very much different from that of its constituent device, and quite likely less because of factors accounted for by "coverage". Figure 1 contains a simplified behavior model of an r-out-of-n system. Each state corresponds to a set of possible configurations having a stated number of operating constituent devices. The transition rate out of a state is the appropriate multiple of the constant failure rate, λ, of one device. Since, given the occurrence of a component failure, a successful transition to another operating state of less redundancy is problematical, so-called "coverage" parameters, C_j, conditional probabilities of successful transition given a failure, are included. Unsuccessful transition is assumed to mean immediate system failure. Usually the coverage parameters are associated with systems having active recovery processes, but they are also applicable to passive mechanisms as long as there are transitions which can go awry among distinguishable, operating states. No distinction is made here. Recognizing this model and assumptions as a Markov process, one can develop the appropriate differential equations for the stochastic process (Feller, W., 1966) and determine in a straightforward manner that the probability of system failure is represented by the expression

$$1 - e^{-n\lambda t} \sum_{j=0}^{n-r} a_j \binom{n}{j} \left(e^{\lambda t} - 1\right)^j$$

where $a_0 = 1$ and $a_j = \prod_{i=1}^{j} C_i$ for $j = 1, 2, \ldots, (n-r)$.

Ratios of system MTTF to constituent device MTTF are tabulated for various combinations of values of r, n, and C_j in Tables 1 and 2. In Table 1, $C_j = 1$ for all j, implying that coverage is perfect. Although the ratios are independent of the constituent device's failure rate (or equivalently, MTTF), not all combinations of r and n are useful, given a specific device failure rate, when the 10^{-9} requirement is considered. For instance, a device with MTTF less than 10^p hours could be used to construct systems of zones p-1 and lower but not zones p or higher. More specifically a device with MTTF of five thousand (5×10^3) hours would not be used to construct systems of zones 4 and 5. In Table 2, $C_1 = 0.9$ and $C_j = 0.$

for all $i \neq 1$, which is excessively poor coverage since systems are all in zones 9 or higher. In all cases in both tables, the ratios do not differ from 1 by an order of magnitude. Hence, to the extent that fault tolerant systems are represented by r-out-of-n systems, a simple and reasonable approximation to the MTTF's of such systems appears to be simply the MTTF of the "worst" device type, a far cry from the ten billion (10^{10}) hour value.

However, having identified a better approximation to MTTF for fault tolerant systems, it is well to note that, in the application of interest, the systems will be effectively renewed every ten hours or so. Hence MTTF, in the conventional sense of an unrenewed system used until system failure as computed above, is not descriptive of system use. In order to consider the relationship of the reliability requirement to safety, it is more meaningful to estimate the probability of system failures, to be considered emergency situations, during the lifetime of a fleet of aircraft with realistic policies for renewal. Therefore, assuming (1) systems meeting the 10^{-9} requirement when all failure modes are considered, (2) system renewal after every ten hours of operation, and (3) a fleet of two thousand (2×10^3) aircraft each with a lifetime of sixty thousand (6×10^4) hours, the probability is approximately 0.01 that one or more emergency situations will occur because of a computer system. It is a matter of judgment, no doubt tempered by economics, whether or not any greater risk to safety is acceptable. Indeed this estimate does not consider latent failures, i.e., conditions where physical defects have occurred but have not yet contributed to a data error because the failed components have not been party to a computation. Such a mechanism could be modeled as an aging effect on the systems -- despite periodic renewals -- indicating that the value 0.01 above is optimistic. And this computation has not included any manner of considering increased complexity as r and n varied.

Ironically the increased complexity, while ostensibly contributing to a reduction in the incidence of system failures resulting from component and device failures, is a source of residual "definitional flaws" in systems. The term "definitional flaw" is adopted here to denote an inadvertent system design which, when the system is in some particular condition with some unexpected data and regardless of the presence or absence of conventional component failures or anomalous environments, produces undesirable results which could have been avoided by another, proper design; the term includes design errors, specification errors or inadequacies, missing requirements, etc. It matters not whether the flaw is in software or hardware or is the result of the correct implementation of an erroneous or incomplete specification; the root cause is human error. One expects the incidence of such flaws to increase with growth in complexity. There is a quite large pool of practical experience with such a failure mode -- everyone's 'bêtes noires', the software bugs found in operational software systems -- which indicates strongly that the failure mode must be included, in some fashion, in the reliability analysis of complex systems. On the other hand, in the avionic application of interest, the level of system reliability required effectively precludes the use of thorough, lifetime/use testing of actual systems to determine with acceptable confidence (in a statistical sense) that the probability of system failure due to residual definitional flaws is compatible with the reliability goals and requirement. As a consequence, more analytical methods -- for example (Costes, A. et al., 1978) -- must be developed and relied upon to address total system (i.e., logic, largely software, and hardware) reliability -- with "acceptable credibility".

TECHNIQUES FOR ADDRESSING DEFINITIONAL FLAWS

Analogously to "hardware redundancy", techniques for designing systems with "logical redundancy" to (attempt to) prevent system failures attributable to residual definitional flaws are becoming a subject of research -- and development. The software fault tolerance studies at the University of Newcastle-upon-Tyne are a leading example of recent innovations (Randell, B., 1975). Largely as a result of the sequential nature of software algorithms, fault tolerant software has been oriented more to a method of sequential test and selection, in accordance with stated acceptance criteria, from among alternate algorithms in a software system, rather than to a method of comparison and voting over the results of a number of alternate algorithms. But parallel alternate hardware logic or concurrent alternate software algorithms in parallel processors are conceivable mechanizations. The "logical redundancy" techniques are therefore seen to parallel hardware.

Fault tolerant software lends itself to an especially simple behavior model, as in Figure 2(a), on the assumption that successful recovery from a software (or logic) failure implies immediate return to the initial (software) state. The rationale for the assumption is that the flaw responsible for the software data error has always been present in the system, having merely not been previously activated, so to speak; the system remains ready to function as before (i.e., correctly) once it has survived the software data error. Indeed, one might expect to not see a second, identical software error, assuming the initial error to have been triggered by unusual data not likely to soon be seen again. (As an aside, experiments using the emulation technique to be discussed suggest themselves to determine whether or not software data errors might not better be modeled as error "bursts".) Figure 2(b) is a simpler representation of the same recovery/failure process. Again, for the sake of simplicity, software is assumed to have a constant failure rate, μ, and fault tolerant software is assumed to have an aggregate recovery parameter, k, analogous to the coverage parameters of the r-out-of-n hardware model. Immediate system failure is assumed to be the result of lack of successful recovery. No further elaboration of a software model is attempted since there has been no credible empirical evidence available for the selection and justification of any particular, more complex, general model of system failure due to software (Thibodeau, R., 1978), let alone the more general case of residual definitional flaws.

ANALYTIC RELIABILITY ANALYSIS: HOW CREDIBLE?

The software model of Figure 2 and the r-out-of-n model of Figure 1 suffice, however, to show the difficulty, when lifetime-use testing of actual systems is not feasible, of establishing with acceptable confidence (in the statistical sense) that systems designed to satisfy the 10^{-9} requirement do achieve the reliability goal. In Figure 3, the two models are combined to represent simply a system subject to and tolerant of both hardware component failures and errors due to residual definitional flaws (here, software). An additional assumption is made -- that the software and hardware are independent -- to keep the

illustration simple again. It is possible to add more complexity in the model, but as stated before, there is no empirical evidence to justify selecting any particular model in preference to another. Also, the conclusion below is not appreciably modified. Again recognizing the model and assumptions as a Markov process, the probability of system failure is computed to be

$$1 - e^{-(n\lambda+\mu(1-k))t} \sum_{j=0}^{n-r} a_j \binom{n}{j} (e^{\lambda t} - 1)^j$$

where a_0 and a_j are as before.

For a typical (and optimistic) value for λ ($\approx 10^{-4}$ failures per hour), typical values for n ($\approx$3 to 5) and the required value for t (=10 hours), bounds on C_1, C_2 and $\mu(1 - k)$ required, in order for the system to satisfy the 10^{-9} requirement, are calculated to be as follows:

$$1 \geq C_1 \geq 0.999999$$

$$1 \geq C_2 \geq 0.9999$$

$$\mu(1 - k) \leq 10^{-10}$$

There appears to be little margin for error in designing systems to satisfy the 10^{-9} requirement. Refinement of the model cannot eliminate the difficulty in estimating precisely the reliability of such systems; it can only transform it into a need for near perfect knowledge of different parameters, for the systems must still achieve the same aggregate behavior as above.

MORE COMPLEX MODELS

In the process of investigating fault tolerant systems (previously, principally studies of hardware) numerous models have been developed for analyzing the reliability of such systems. Of late, investigations have also been undertaken into models to relate the system failure modes to time-variable computational and performance requirements, thus attaching the reliability of a system more tightly to its application (Meyer, J., 1977), (Beaudry, M. D., 1978). Some model evaluation schemes have been "computerized" to serve as more or less general purpose tools for the convenient analysis, in the architectural design stage, of systems composed of complex arrangements of elements, e.g., CAST (Cohn, R. B. et al., 1974), CARE II (Stiffler, J., 1974), CARSRA (Bjurman, B. E. et al., 1976), ARIES (Ng, Y., 1976). Although they consider details of system behavior such as recovery (detection, isolation, reconfiguration) strategies, sparing (active, stand-by, switching) strategies, transient and intermittent fault (duration, periodicity, leakage) modes, functional dependence among devices, nonexponential failure distributions, etc., the models still are constructed essentially from parametric descriptions of aggregate system, subsystem and/or device behavior in order to make use of mathematical techniques applicable to idealized stochastic process models and for reasonably efficient computation. Hence all the models must be provided with parameter values which need to be assumed or known, by some other means, in order to precisely represent any and each particular system design of interest.

EMULATION

Digital Simulation

While the word "simulation" is widely used to denote all manner of techniques for, among other purposes, analyzing the behavior of objects and their environments by means of implementation and manipulation of more conveniently malleable surrogates, here the word is limited to mean the use of computer "systems" as surrogates -- at whatever level of abstraction is meaningful to an application. The concept of system is stressed because usefulness of a simulation scheme depends upon both software and hardware -- a characteristic more effectively utilized by emulation. For example, consider the reliability analysis programs previously mentioned -- CAST, etc. Although they are essentially simulation schemes which are normally discussed without regard to host computer hardware, in any actual application, host computer hardware will be an important constraint upon the amount of detail which it will be feasible to consider with the programs.

Digital simulation, as opposed to emulation, at the level of gate logic has been discussed in the literature on computers and considered as a tool for design and fault (signature) analyses of digital logic circuits at levels of detail ranging from simple (e.g., assuming gates to have only two possible output values) to complex (e.g., allowing undefined values of gate outputs and various timing anomalies) (Szygenda, S. and Thompson, E., 1976). For the analysis of circuits the sizes of microprocessors, memories and larger, in practice simulation techniques at the aggregate, functional behavior level begin to displace gate level simulations (Menon, P. and Chappell, S., 1977) as the gate level simulation costs become prohibitive when compared to perceived benefits.

However, for the purposes of reliability analysis of fault tolerant systems, gate level simulation warrants considerable cost in view of the conclusion to be drawn from the preceding paragraphs that, at the levels of reliability of interest, the probability of failure of such systems is less dependent upon the mode of failure resulting from depletion of redundant resources than it is upon the less well understood and questionably modeled modes considered under the terms "coverage" and "definitional flaws". A similar conclusion to the effect "that the introduction of a redundancy at the hardware level increases the relative influence of software faults" is made elsewhere (Costes, A., 1978). Unfortunately, while the costs

could be suffered, in light of the benefits, gate level simulation is not a feasible technique for application to questions involving chance events and repeated trials because it is time consuming -- orders of magnitude slower than likely target systems.

Emulation vs. Simulation

In ordinary use, the word "emulation" means an endeavor to equal or excel; in the present context, it is reserved for a particular technique of implementing simulation possible when a host computer is microprogrammable. In order to avoid confusion, "simulation" acquires the added meaning here of being distinct from "emulation". Microprogramming is significant because it allows a final definition of a computer's "apparent" instruction set to be postponed until after the definition of hardwired logic is completed, and it does this with an acceptably small risk that the hardware logic will need redesign. This happens because a "real" instruction set is defined by the hardwired logic, is at a quite primitive level, and is tailored especially for executing algorithms which, in turn, become operational definitions of less primitive operations -- the "apparent" instruction set.

Thus it may be said that a computer defined by an "apparent" instruction set does not really exist; it is "emulated" by microprogrammable hardware by means of microcoded algorithms. Admittedly, variations in efficiency of variant microcode operations vis-a-vis various "apparent" instruction sets may exist, but they can be ignored for the present purpose. What is notable is that, given reasonable care not to mismatch host and target computers, microprogrammable computers can perform in the role of an "apparent" computer approximately as efficiently as a hardwired version of the "apparent" computer would. Note that "emulation" is at a level of detail which permits software implemented for another, "apparent", target computer to be executed "directly" by a host computer. That is, no modification of the target software is needed to make it compatible with the host computer, and no special software on the host computer needs to be generated (more specifically, no simulation program in an "apparent" instruction set on the host to interpret the instructions of the target software and mimic the target computer) as would be needed on a nonmicroprogrammable computer.

Use as a Diagnostic Tool

Addition of diagnostic, control functions in the microcode permits a host computer to act not only as a surrogate but also as a device for observing and recording (and possibly analyzing) target software performance in an ostensibly natural environment. Such "diagnostic emulation" use is becoming more common in the development and maintenance of special software systems and is, seemingly, "emulation" in the dictionary sense. As might be expected efficient use of such a diagnostic system requires support capabilities for readily modifying microcoded algorithms defining target computers. Such facilities are beginning to be developed -- for example, EMULAB (Clausen, B. et al., 1977). What has been less well considered is the fact that such capabilities can be extended to permit analysis not only of software but also of systems (i.e., software and hardware) -- and not only as they are intended to be but also as they are not. By generating the defining microcode such that it represents target computers in sufficiently fine detail combinations of failures in individual components, anomalous data, and definitional flaws can be introduced and their effects at the system level observed rather than assumed. Thus emulation provides a conveniently manipulated failure effects analysis tool. In addition the manner in which an emulation technique is implemented, with automated diagnostic and system and environment controls, lends itself to use for "pseudo-testing" as in Figure 4.

In general, emulation can be used to generate repeated trials of "emulated" systems from which failure ratios and histograms can be tabulated for analysis -- hence, aggregate behavior models verified and parameter values estimated with some measure of confidence (in a statistical sense). Clearly, assumptions about the manners and rates of occurrence of failures and flaws must still be made in order to introduce these last into the emulations. However, while the credibility of precise assumptions will still be questionable, it should be possible to develop credibly pessimistic assumptions to attempt to demonstrate that particular fault tolerant system designs exceed the reliability requirement.

While, also in general, the use of emulation to perform such "pseudo-testing" is limited by the efficiency (i.e., computation speed) of the emulation technique and equipment, it appears reasonable to state that it is less restricted than in the case of digital simulation. Given the previously described need and difficulty of establishing the reliability of the fault tolerant avionic computer systems of interest, emulation techniques merit further investigation.

SAMPLE EXPERIMENT

Scope

An effort of limited scale was undertaken in order to determine whether or not an emulation scheme could be devised which would be sufficiently efficient to support analyses of target systems of meaningful sizes and complexities, and to demonstrate that such a scheme could be implemented in a manner convenient for analysis purposes by users not well versed, if at all, in the emulation scheme itself. As a demonstration, a sample analysis bearing upon reliability of fault tolerant systems was chosen.

The effort was experimental; time and effort were expended searching out efficient implementations and superior microprogramming capabilities to support the implementations. Consequently no commitment to any specific microprogrammable hardware was desirable initially. The experiment was performed on a large, general purpose computer whose underlying microcode was sacrosanct. For this reason emulation was really simulated. This last level of complication can be accounted for by introducing a time scale factor; it is otherwise ignored here. While some variant emulation algorithms which have been conceived have not yet been implemented and examined, the effort has provided a basis for selecting microprogrammable hardware

for further studies. Here, however, the experiment is discussed merely to illustrate an actual, rather than speculated, application of emulation to reliability analysis.

Emulation Technique

The scheme selected consists of an algorithm generated independently of any target computer. Descriptions of particular systems to be emulated are provided to the algorithm at the time of operation. The method is referred to as "table-driven" in contrast to a "compilation" method in which a hardware description is input to a hardware description language "compiler" which generates a computer program to emulate one specifically defined computer. The table-driven method was chosen because it was believed to facilitate the infusion of failures and to provide better visibility to a user. That is, the target hardware is visible as a distinct entity at emulation time rather than being dispersed and buried inside the workings of an emulation program, and failures and faults can be added and removed without altering the cyclic nature of the algorithm.

From a user's viewpoint, the emulation is visualized as the repeated transformations of two variables. One variable, S_n, describes the structure of the system at time step n. The variable is essentially a matrix which identifies the interconnections among the logic elements in a system, and also identifies the functional behavior of each element. The most primitive element permitted is a generalized gate to which constant behavior characteristics (neither correct nor faulty to the emulation algorithm) are attached. More complex elements such as flip-flops and tristate devices are also permitted, if desired, as primitive elements to be manipulated as indivisible entities by the emulation algorithm. (For the experiment, the algorithm was limited to elements with scalar output values.) For example, a logic element X might have been identified to act as a four (4) input NAND gate driving six (6) other identified elements and supposed to have an irregular input-to-output signal propagation time. Hence, S_n is effectively a time-varying, annotated logic diagram.

A second variable, V_n, is a vector containing the output values, at time step n, of each of the logic elements defined in S_n. Target software corresponds to a subset of this variable, viz., those values corresponding to logic elements defining some of the emulated system's memory.

A third auxiliary variable, F_n, can be visualized as a source of external perturbations into the emulated system -- affecting S_n, V_n, or both. As currently implemented, this variable is generated separately from the others in order to increase the speed of the emulation computations. It represents the source of random failures, flaws, and anomalies at either preselected or random times and control over the emulation process.

The emulation algorithm, a time invariant transformation, is a collection of techniques (so-called "selective trace", linked lists, data compression, parallel processing -- untested because of the limitations of the general computers previously mentioned --, event scheduling) consistent with a model of the behavior of a "generalized" logic element over an arbitrary time step.

SAMPLE ANALYSIS: LATENT FAILURES

The experimental analysis performed was a study of the efficacy of five (5) particular algorithms, each with a different instruction mix, as detectors of component "stuck-at" faults (i.e., latent failures) in a particular "play" system. The analysis is documented in detail in (Nagel, P., 1978).

The "play" target computer was originally generated (i.e., defined at the gate logic level) as a vehicle for checking out the initial and modified versions of the emulation algorithms, and for demonstrating the ability of support software, a hardware description language translator and meta-assembler for regenerating target software, to respond semiautomatically to hardware design changes. The "play" computer has a memory of 8192, 16 bit wide words, a CPU with a count of approximately 2000 gate equivalents, and a single input-output register/port. The logic is arbitrarily assigned to four (4) hypothetical chips: a "clock" chip, an "adder" chip, an "op-decode" chip, and a miscellaneous odds and ends chip. The instruction set contains about a dozen basic instructions.

The emulated system were simple. The five algorithms, ranging in length from about a dozen instructions to several hundreds, were repeatedly executed, with randomly selected initial data, and randomly selected faults of random components. Distributions of time from fault occurrence to fault detection (i.e., fault latency duration) were generated. Two analyses of the sort that would be of interest in studies of fault tolerant systems were made. For one, the observed distributions were fitted against commonly used mathematical models, e.g., exponentials, as would be done in order to determine models and parameter values for use in reliability analysis programs. The results, of course, are not significant, owing to the fanciful nature of the input data; still, it is interesting that the distributions were best fit by models of balls selected at random from urns. Another result, that the distributions each exhibited different nonzero probabilities of never detecting the faults, was predictable, but only an experiment of this nature could determine the differences in magnitude. A second effort was a search for correlations among the distinguishable characteristics of the algorithms and the distributions. The only significant correlation found was between instruction mix and detection probability. Here too, because of the nature of the target system, the magnitudes of the correlations can only be considered fanciful. But the concept is useful in considering characteristics which should be avoided in algorithms whose function is to reconfigure a system after a failure has been detected.

CONCLUSION

A case has been made for the use of emulation techniques as a needed adjunct to reliability analysis models for highly reliable avionic computer systems. Although no conclusion about the technique's

eventual usefulness is yet warranted, in light of its apparent usefulness as a failure modes effects analysis tool and the promise and potential rewards of its use for probability distribution uses, further development and investigation of the technique appears warranted and is being pursued by the NASA.

REFERENCES

Barlow, Richard E., and Proschan, Frank, 1965, Statistical Theory of Reliability and Testing, Holt, Rinehart and Winston, Inc.

Beaudry, M. D., June 1978, "Performance-Related Reliability Measures for Computing Systems", in IEEE Transactions on Computers, Vol. C-27, No. 6, pp. 540-47.

Bjurman, B. E., Jenkins, G. M., Masreliez, C. J., McClellan, K. L., and Templeman, J. E., August 1976, Airborne Advanced Reconfigurable Computer System, Boeing Commercial Airplane Company, NASA Contractor Report # 145024.

Clausen, B. A., and Wachs, R. W., 1977, EMULAB, Unique Systems Development and Integration Laboratory, AIAA/IEEE/ACM/NASA Computers in Aerospace Conference, Los Angeles.

Cohn, R. B. et al., 1974, Definition and Trade-Off Study of Reconfigurable Airborne Digital Computer System Organizations, Ultrasystems, Inc., NASA Contractor Report # 132552.

Costes, A., Landrault, C., and Laprie, J. C., June 1978, "Reliability and Availability Models for Maintained Systems Featuring Hardware Failures and Design Faults", in IEEE Transactions on Computers, Vol. C-27, No. 6, pp. 548-60.

FAA FAR part 25, paragraph 25.1309(b), dated 5 August 1970.

Feller, William, 1966, An Introduction to Probability Theory and Its Applications, Volume I, John Wiley & Sons, Inc.

Hopkins, A. L., Smith, T. B., and Lala, J. H., October 1978, "FTMP -- A Highly Reliable Fault-Tolerant Multiprocessor for Aircraft", in Proceedings of the IEEE, Vol. 66, No. 10, pp. 1221-1239.

Menon, Premachandran R., and Chappell, Stephen G., August 1978, "Deductive Fault Simulation with Functional Blocks", in IEEE Transactions on Computers, Vol. C-27, No. 8, pp. 689-95.

Meyer, John F., 1977, "Reliable Design of Software", in Rational Fault Analysis, Sacks and Liberty, eds., Marcel Dekker, Inc.

Meyer, J. F., July 1978, Models and Techniques for Evaluating the Effectiveness of Aircraft Computing Systems, University of Michigan, NASA Contractor Report # 158993.

Nagel, Phyllis, September 1978, Modeling of a Latent Fault Detector in a Digital System, Vought Corporation, NASA Contractor Report # 145371.

NASA Contract NAS1-15428, July 1978, Statement of Work for "Development and Evaluation of a Software Implemented Fault Tolerance (SIFT) Computer", NASA Langley Research Center.

Ng, Ying-Wah, September 1976, Reliability Modeling and Analysis for Fault-Tolerant Computers, University of California at Los Angeles, Engineering Document UCLA-ENG-7698.

Randell, B., 1975, "System Structure for Software Fault Tolerance", in Proceedings of the 1975 International Conference on Reliable Software, Los Angeles.

Ratner, R. S., Shapiro, E. B., Zeidler, H. M., Wahlstrom, S. E., Clark, C. B., and Goldberg, J., October 1973, Design of a Fault Tolerant Airborne Digital Computer, Volume II - Computational Requirements and Technology, Stanford Research Institute, NASA Contractor Report # 132253.

Smith, T. B., Hopkins, A. L., Taylor, W., Ausrotas, R. A., Lala, J. H., Hanley, L. D., and Martin, J. H., July 1978, A Fault-Tolerant Multiprocessor Architecture for Aircraft, Volume I, The Charles Stark Draper Laboratory, NASA Contractor Report # 3010.

Stiffler, J., 1974, Reliability Model Derivation of Fault-Tolerant, Dual, Spare Switching, Digital Computer System, NASA Contractor Report # 132441.

Szygenda, Stephen A., and Thompson, Edward W., December 1976, "Modeling and Digital Simulation for Design Verification and Diagnosis", in IEEE Transactions on Computers, Vol. C-25, No. 12, pp. 1242-53.

Thibodeau, R., 1978, The State-of-the-Art in Software Error Data Collection and Analysis, General Research Corporation, AIRMICS Contract # DAAG29-76-c-0100/0598.

Wensley, J. H., Lamport, L., Goldberg, J., Green, M. W., Levitt, K. N., Melliar-Smith, P. M., Shostak, R. E., and Weinstock, C. B., October 1978, "SIFT: The Design and Analysis of a Fault-Tolerant Computer for Aircraft Control", in Proceedings of the IEEE, Vol. 66, No. 10, pp. 1240-54.

TABLE 1

n \ r	2	3	4	5	
3	.83				
4	1.08	.58			
5	1.28	.78	.45		
6	1.45	.95	.62	.37	
7	1.59	1.09	.76	.51	zone 5
8	1.72	1.22	.89	.64	zone 4
9			1.00	.75	zone 3
10				.85	zone 2

Ratio of $\frac{\text{MTTF (r/n system, } C_i = 1)}{\text{MTTF (constituent device)}}$

TABLE 2

n \ r	2	3	4	5
3	.78			
4	.6	.55		
5	.46	.45	.43	
6	.38	.37	.37	.35
7		.31	.31	.31
8		.27	.27	.27
9				.23
10				.21

Ratio of $\frac{\text{MTTF (r/n system, } C_1 = 0.9, C_{i\neq 1} = 0.1)}{\text{MTTF (constituent device)}}$

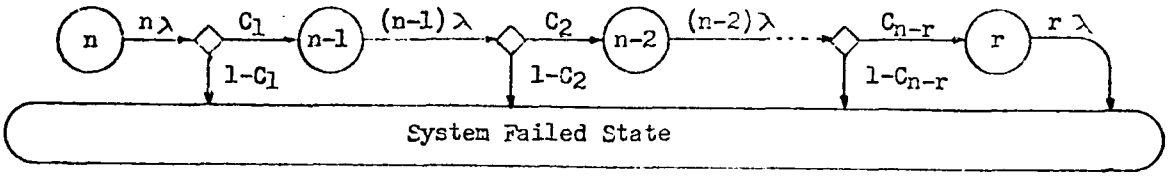


Fig.1 r-out-of-n system with coverage parameters

Fig.2 Fault tolerant software

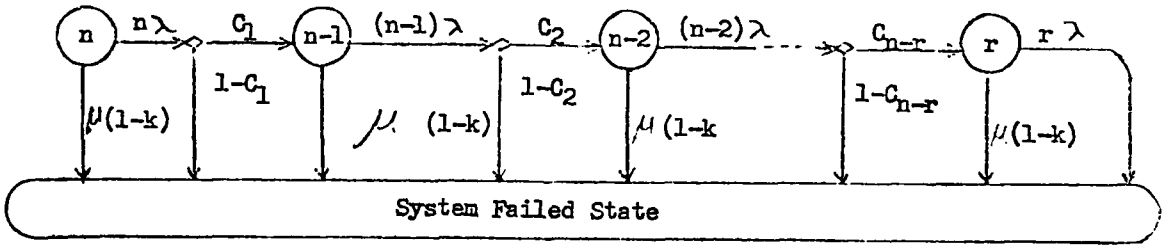


Fig.3 r-out-of-n system with fault tolerant software

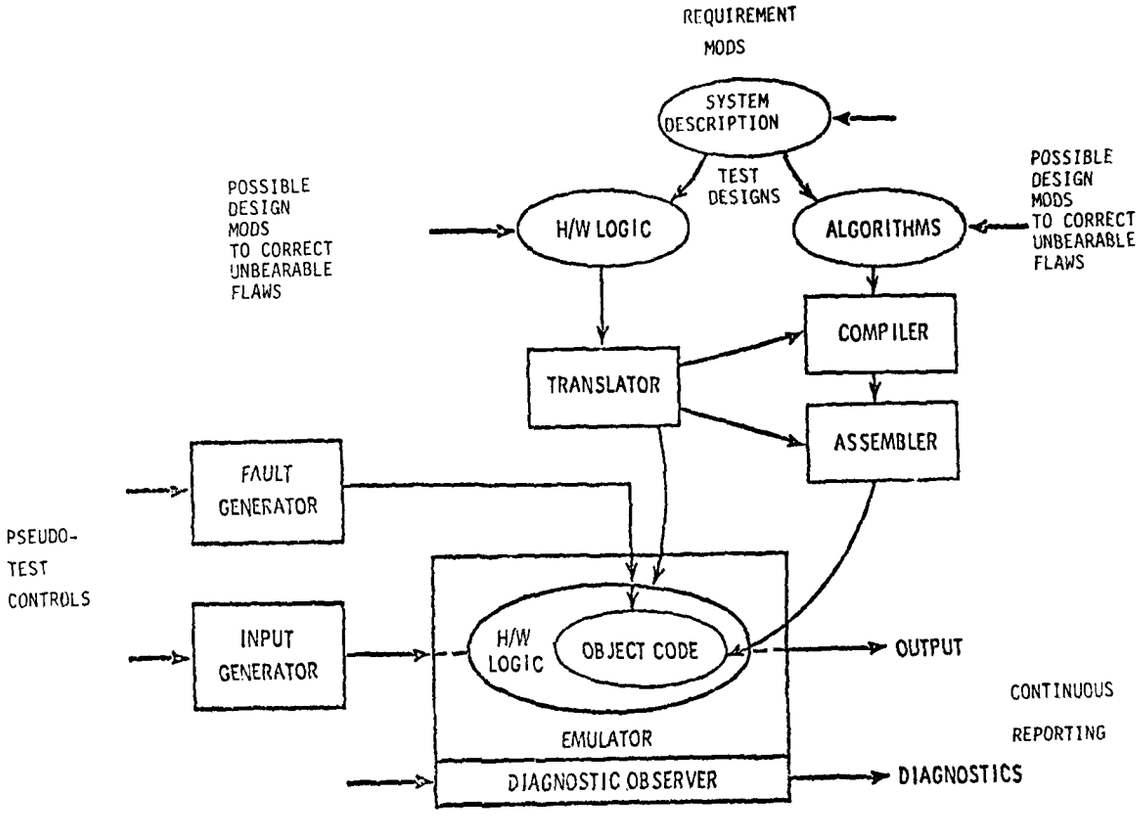


Fig.4 Emulation "psuedo-testing" activity

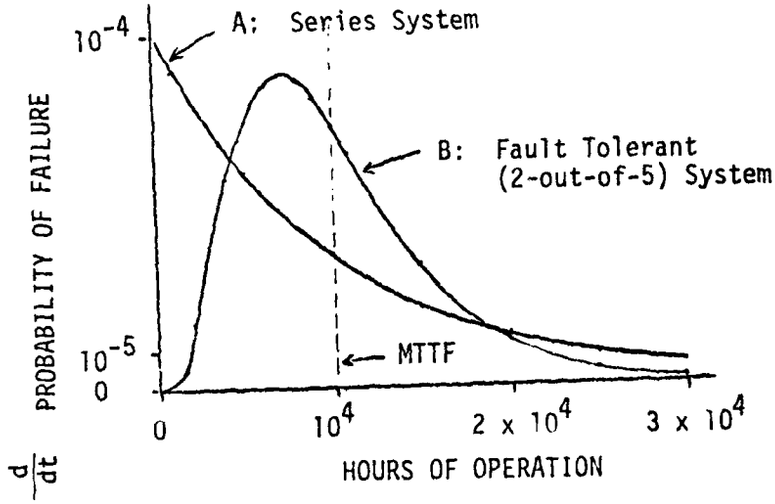
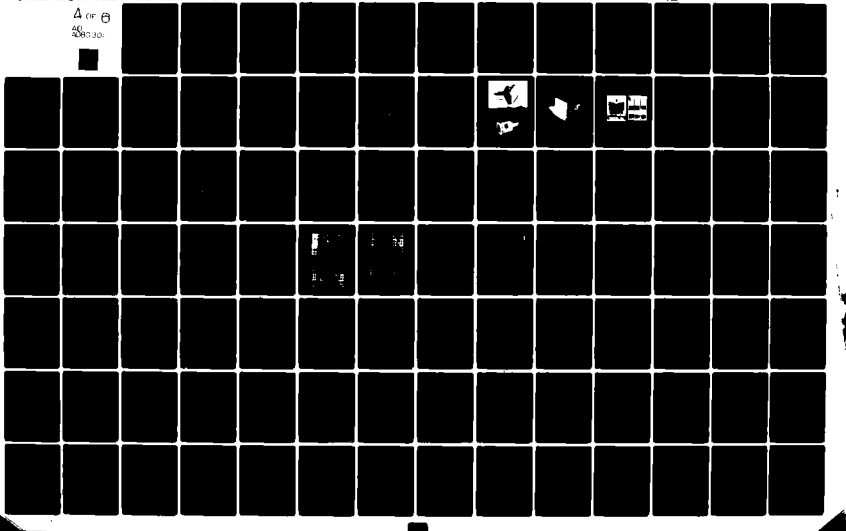


Fig.5 Failure distributions with same MTTF

AD-A080 301 ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT--ETC F/G 9/5
AVIONICS RELIABILITY, ITS TECHNIQUES AND RELATED DISCIPLINES.(U)
OCT 79 M C JACOBSEN

UNCLASSIFIED AGARD-CP-261 NL



DISCUSSION

T.L.Regulinsky, US

Did I understand you correctly that you are somehow computing the time to detection of errors from that Histogram? How did you compute the times?

Author's Reply

We insert a fault into the emulation scheme and plot the time to breakdown.

T.L.Regulinsky, US

(1) Was the simulation Montecarlo?

(2) What distribution did you use?

(3) I'm driving at the density function. You are formulating the model as a Markovian model, you are obviously simulating from a density function. Is this correct?

Author's Reply

(1) You are inserting faults or you have to sample times of faults from some distribution.

(2) The distributions have to come from somebody else, not from the people who deal with the emulator. They come from the hardware people who have some knowledge about the devices and logic to interconnect the devices.

(3) This is different from the previous model. If we want to calculate the difference between two fault computers using analytical techniques then we have to know the coverage parameters out to 6, 7, 8 significant digits. These are not too well known and so reliability cannot be used until some information is known about the parameters. We are trying to provide a technique by which we can justify statements about the difference in parameters for two different computers with different logic.

J.C.Robertson, UK

Will the emulation scheme cater for transient as well as permanent faults in the system being emulated?

Author's Reply

Yes, provided that the pdf applying to transient fault occurrence were known. This might be difficult to determine; one approach would be to assume a set of conditions much worse than those expected in real-life conditions, so that if the system logic survived these, then one would be reasonably sure that it would survive anything it might meet in real-life operation.

RELIABILITY ASSURANCE FOR LARGE SCALE INTEGRATED CIRCUITS

Robert A. McDonald, Captain, USAF

Reliability and Compatibility Division
Rome Air Development Center
Griffiss AFB NY 13441

ABSTRACT

The objective of the Large Scale Integrated Circuit (LSI)/Microprocessor Reliability Program is to ensure the availability of high quality reliable microelectronic devices that will conform to military needs and standards. This is done to avoid device failures to critical systems by selecting, evaluating, characterizing and qualifying individual integrated circuit device types from U.S. vendors. The Reliability Branch of the Rome Air Development Center has been tasked with providing reliability assurance for microprocessors, memories, and related devices.

The program has been divided into three broad areas of responsibility. The Product Evaluation Group selects, analyzes and prepares reports on devices of potential or current interest to the military. The reports are disseminated to government agencies and describe the physical construction, technology, packaging, workmanship, input protection problems, utilization difficulties and electrical design risks. The Electrical Characterization Group is tasked with characterizing specific devices selected for inclusion in the military specification system. The characterization effort generates the required acceptance tests that each part must pass before use and prepares device detail specifications. Overall effects of circuit loading, technology compatibility and test vector generation complicate the problem of developing the detailed specification. The Reliability Assurance Group stress tests the devices, assesses the failure modes and projects a system use failure rate. The complexity and number of functional components on the same chip makes the life cycle testing difficult.

RELIABILITY ASSURANCE FOR LARGE SCALE INTEGRATED CIRCUITS

The United States Air Force requires its defensive and offensive systems to work everywhere, every time, under any environment or circumstance. The reliability of a system is ultimately dependent on the reliability of its individual parts. As systems increase in their complexity, their dependence on large scale integrated circuits is also increasing. The electronic revolution in weapons systems now taking place depends on the use of large microelectronic devices with multiple bit, stored program, digital computing functions. Rome Air Development Center (RADC) has been tasked to assure the reliability of microprocessors, memories and related devices for the USAF as well as DOD. RADC has been developing military standards since the early 1960's and has the laboratories, experience and capability to actively interface between United States commercial large scale integrated circuit (LSI) industry and the needs of USAF.

The objective of the task is simple in concept but complex in practice. RADC is to ensure the availability of high quality, reliable LSI microelectronic devices that will conform to military needs and standards. Additionally, RADC is to develop tools and techniques to evaluate, test and analyze electronic devices; provide reliability support to acquisition divisions; and prepare and coordinate specifications and standards. RADC is to assure that high quality LSI parts are readily available so that highly reliable systems may be acquired and fielded at the lowest possible cost. The objective is complex because the testability and reliability of LSI is not yet completely determined.

The LSI microcircuits that RADC has determined to best fit the needs of the U.S. Air Force are always commercially made, military specified and qualified parts. That is, the devices are originally designed by reputable U.S. vendors for the commercial market but have increased reliability through imposed, stringent qualification standards. Basically, a manufacturer may apply to have a device qualified or an RADC engineer may select for qualification a particular generic type; for example, a 16 thousand bit memory manufactured by several vendors. Standardization of LSI microcircuits used in systems has several important benefits to the user, vendor, and to the U.S. government.

For the vendor, benefits include increased yields from each wafer processed, because the manufacturer can develop increased experience with each device type. For the user, probably the dominant cost of using any programmable device is the software development cost. This cost can be minimized by using standard parts where elements of existing software can be applied to new systems. There are also benefits in that several sources are potentially available thus providing increased capability for obtaining parts during large or long equipment production runs. The improved availability of replacement parts, often many years after a system has been obtained, is important. By using a standard part the user also avoids the problem of obtaining and maintaining the large amount of paperwork associated with any LSI device. For example, when an acquisition division desires to use a nonstandard part, a detailed control drawing must be prepared, the device must be evaluated and qualified and finally the drawings must be stored and any changes annotated. Each time an acquisition division desires to use a nonstandard part, this qualification process is usually repeated even if the part was used in another system. By comparison, once a part has been qualified under military specification, the drawings, evaluation and qualification data are available to all users of the device. The reliability of the parts is assured because the parts are thoroughly tested, analyzed and a detailed procurement specification prepared. When the same parts are used in different systems more reliability data on the failure modes of the devices can be collected and deficiencies corrected. If only a few devices of a particular part type are used their potential reliability problems may not become apparent. The best method of avoiding system failure is to use devices that have widespread application in both the commercial and military fields with increased reliability for the military through the application of a strict qualification and specification program.

The approach to assuring the reliability of LSI/microprocessors that RADC engineers have determined best is a mix of in-house and contractual efforts divided into three basic groups. A product is; one, chosen and evaluated; two, characterized and specified and three, has life testing and quality assurance

studies performed. Additionally, failed parts are examined and deficiencies corrected, and some parts from the warehouse inventory are evaluated (Figure 1).

Figure 1 shows that first a part is chosen for a product evaluation study. Basically, the evaluation is to assess the reliability and performance at a lower cost than a complete characterization. If a part is judged to have potential military usefulness, a Product Evaluation Report recommends that characterization studies be done and specifications written. After specifications are published, devices may have further reliability assurance studies performed if the technology is new or unique device designs are being used. Figure 1 also shows the operating system feedback in the form of failure reports. Device reliability is forecast by Reliability Prediction of Electrical Equipment (MIL-HDBK-217C). Device specification procedures, classes and details are contained in General Specifications for Military Microcircuits (MIL-M-38510D) while Test Methods and Procedures for Microelectronics (MIL-STD-883B) establishes uniform methods and procedures for testing. After a device specification is issued a manufacturer must obtain a Defense Electronic Supply Center (DESC) certification before production (see Figure 1). After certification a specified part is placed on the Qualified Parts List (QPL).

Devices are selected for inclusion in the General Specifications for Microcircuits (MIL-M-38510D), according to a set of criteria developed at RADC. The first criterion is based on usage of the proposed device. If a device is in widespread commercial use, then the USAF can expect to see U.S. defense system designers implement the device in military systems. Presently, RADC is preparing detail specifications on devices that are already in some AF systems. This occurs because the rapid advance in LSI digital technology has caused systems to come into the inventory without military qualified parts being available.

A device may be chosen when it is especially critical in a system. For example, a particular microprocessor was chosen because it forms the heart of a high data speed processor. By studying this control device first, RADC expects to gain information on the potential reliability of the entire system. Finally, under device usage a part may be chosen for its large volume. The part may be a simple controller, but if it has widespread use then its reliability must be assured.

Devices may be selected for study on the basis of device technology. Product evaluations are performed on devices of even remote interest to defense application at the present time, if the new technology promises wide use in the future. Magnetic bubble memories are a typical example of such a device selected for study but not ready for military specification. Some devices are selected in an attempt to discover the physics of their limitations. For example, metal-nitride oxide semiconductor memories (MNOS) have an important non-volatile digital storage capability but presently have a limited operational lifetime. Radiation hardness is also very important to some defense systems and technology areas such as complementary-metal oxide semiconductor (CMOS) have important radiation hardness properties. Devices designed from this technology are thus candidates for inclusion in the military specification system.

Manufacturer interest in having a part qualified is an important factor. If the manufacturer desires to qualify a part to military specification and obtain qualified product listing then RADC first determines if the military has a requirement for the part and then proceeds to develop tests and write specifications. For the entire military parts program, vendor cooperation and interest is crucial. The manufacturer must supply detailed information about architecture, fabrication, and product limitations to the test and specification writer about his product. No military specified part is possible unless the vendor will provide this data.

The first microprocessors chosen for qualification are shown in Figure 2 along with comments about their technology. The 8080A and 6800 were chosen primarily because of product maturity and extensive military use. The 1802 was chosen primarily for the radiation hardness and low power consumption provided by CMOS technology. The 2901A was chosen because of the expansion capability of bit-slice architecture as well as the high speed and the power advantages of low-power Schottky technology. The 9900 was chosen because of a specific military need for a 16 bit architecture.

Along with these processors, appropriate available support devices such as peripheral interface adapters, system clocks, and memories were chosen. The objective of choosing a family of devices is to have an entire family of parts available to a system designer without restricting creativity or system capability.

Any parts qualification program must keep current with technology advances. The rapid pace of microelectronics has made this problem particularly acute. The second and third generations of just one device are shown in Figure 3.

From the second and third generations will come future systems and products. The 8080A microprocessor is an 8-bit parallel processor designed for use in general purpose computing environments. It has 78 basic instructions divided into five groups operating into six registers and an accumulator. As an example of what has happened to the technology consider the 8085 and 8086. The 8085 has 80 instructions and is software compatible with the 8080A but has some of the peripheral circuits built on the chip, notably the clock. This improved version, considered a second generation microprocessor, is under consideration for military qualification. The question to be answered is: Is the increase in vector interrupt capability and increase in clock speed sufficient reason to add another microprocessor to the inventory of qualified parts? Engineering determination about the suitability of this part for military systems is yet to be determined. The third generation parts such as the 8086, 68000, Z8000 are 16 bit processors capable of performing bit, byte, word and block operations. At least one of these 16 bit processors will be chosen for military qualification.

Product Evaluation

The Product Evaluation Group is tasked with tracking current commercial technologies for possible military use. It should be emphasized that the devices subject to product evaluation are not always military qualified parts, but are of possible use to military agencies. Some of the products evaluated will

go on to be military qualified while others will prove to be not acceptable or will lose out to better choices in the process of selecting the minimum set of different devices needed to satisfy military functional needs. A product evaluation report is divided into two basic sections, physical description and electrical features.

The physical description of a device begins with an investigation of the package. Present LSI military integrated circuits are packaged with dual-in-line packages (DIP) with ceramic shells. The die cavity is hermetically sealed with a metal or ceramic lid. This construction is also used by high reliability commercial manufacturers but without the control of military standards. The bonding pad size and type bonding is inspected to insure that the lead wire is firmly affixed to the pad for use in high vibration environment. Furthermore, the lead wire and bonding pad must be of compatible composition to prevent the formation of intermetallic compounds which weaken the wire bond. Die size is ascertained and fabrication details noted by analytical analysis. Scanning electron microscope and x-ray photographs are taken of die to determine if voids are present in the die attachment material which would impair heat dissipation or cause the die to detach from the substrate. Large voids have been observed in some devices of commercial manufacture. The physical description of the construction of a device includes a package gas analysis. This consists of an ambient gas analysis to determine the contents of the internal package atmosphere. Excessive moisture in the package constitutes a known hazard to reliability and its presence in quantities greater than 5000 parts per million (ppm) (MIL-STD-883B, M5005.4, p.8) indicates the manufacturer will have to exercise more control over the packaging environment (Thomas, p.167). Two of the five military selected microprocessors (Figure 2) were found to have water vapor in excess of 1% = 10,000 ppm).

The electrical analysis examines the circuit schematic and determines the surface topology and architecture. This is done to assist a designer in properly using the electronic features of the device and to insure that the manufacturer provided data is correct. An important features of the electrical analysis is examination for input protection. Lack of suitable input protection in often electrically noisy military environments can cause sudden device failure if a transient voltage spike appears at an unprotected input. Additionally, devices without adequate input protection may be damaged by static electricity during handling. In one microprocessor examined, certain input lines have only a substrate diode type of parallel protection with no series resistance to limit current and spoil rise times of electrostatic pulses (Dicken, p.18). Careful examination of manufacturer's literature is done to expose inconsistencies and possible design restrictions. For example, if power requirements increase with temperature, is the amount of increase documented and, if so, can the input lines sustain this current density?

Product evaluations are also performed on devices that are in warehouse stores to determine if military standards are being applied consistently and correctly. This analysis may not be performed to the same depth as initial studies but is useful to control counterfeiting, substitution, unauthorized change, and any lapse of compliance with the military specification (Figure 1). If the construction, fabrication or design of a device is changed, then the military user must know exactly how the device will perform in the system when offered as a replacement for a failed part.

The entire product evaluation program produces a document called a product evaluation report which is then used by designers and other groups to ascertain the desirability of using or qualifying a device for military use. Only proven reliable devices must be used in the military environment and the thorough examination of candidate devices before a specification is issued is essential.

Characterization & Specification

The objective of characterization is to completely describe the electrical performance and parameter limits of a device over the full military temperature range (-55°C to 125°C). The approach used is; first, verify functional design; second, determine the critical parameters; and finally, establish test sequences. The result is a detailed specification published for all users and manufacturers that exactly describes the procurement and testing the device must meet.

One of the basic documents used is General Specifications for Military Microcircuits (MIL-M-38510D) which provides basic guidance for all large scale integrated circuits. It provides three levels of reliability assurance, Classes S, B and C. Class S parts are designed for systems where the reliability and radiation requirements are the most severe, such as space. Class B is the general military part quality level while Class C is similar to Class B except that the production testing is not as stringent (MIL-M-38510D, p.6). Possibly the most important guideline contained in the General Specification is the workmanship requirement which states that microcircuits shall be manufactured in a careful manner in accordance with good engineering practice, with the requirements of the specification followed exactly, and all inspections and tests performed and recorded correctly (MIL-M-38510D, p.17). The General Specification then describes the product assurance provisions for each microcircuit covered by the specification. In general it requires that inspection records be maintained and that certain tests be performed according to the specified product assurance level, Class S, B or C. The tests called out are either screening tests that every circuit must pass (nondestructive) or quality conformance tests that only a selected sample must pass (destructive). All the basic tests are shown in Figure 4.

The Military Standard, Test Methods and Procedures for Microelectronics (MIL-STD-883B) describes in detail how these tests are to be performed. Method 5004.4, Screening Procedures and Method 5005.4, Qualification and Quality Conformance Procedures are two test methods used on all microcircuits. The Screening Procedures method establishes procedures for total lot screening of microelectronics to assist in achieving levels of quality and reliability commensurate with the intended application (Figure 5). Qualification and Quality Conformance Procedures are intended as destructive tests and inspections intended for quality conformance inspection of individual inspection lots as a condition for acceptance for delivery. By using a standard test procedure common to all devices as well as different manufacturers better quality can be maintained. Reliability is enhanced by specifying conditions obtainable in the laboratory equivalent to actual service conditions and by requiring that devices be tested against this environment before being used for military systems.

The development of an electronic test for a microprocessor is a complex and time consuming process. For example, the detailed specification for the 8080A has some 12,000 test patterns. The general procedure is to first generate a detailed functional block diagram by partitioning the processor into basic functional blocks such as registers, multiplexers, arithmetic and logic functions and identifying all data paths. Each of the functional blocks is tested using patterns which are known to fully determine the correct operating condition. Test patterns are then generated to verify the integrity of the data and critical paths. All instructions are tested to verify that they perform the intended operations and, finally, test patterns that check for known processor sensitivities are included (Ostrowski, p.v-1).

Once the detailed specification with test patterns has been developed, it is published as a part of MIL-M-38510D. This document is used by the vendor to manufacture and test his parts before shipment as well as by the user to check parts before installation.

Reliability Assurance

Reliability assurance is a technology-based check of the basic failure modes and mechanisms as a function of time and stress. The approach is to subject military qualified devices to high-stress, short-term testing and then perform analysis on failed devices. The results are used to identify faulty materials, processes, and designs; determine effective and efficient screening; and to determine reliability prediction models.

Reliability prediction models are described in Reliability Prediction of Electronic Equipment (MIL-HDBK-217C). A product evaluation and this prediction is refined as data which is collected and analyzed. The limitations of prediction are the practical ones of data gathering and analysis complexity. Considerable effort is required to generate sufficient data on a part class to report a statistically valid reliability figure for that class.

The two major methods of prediction are parts stress analysis and parts count analysis. The parts stress method determines a failure rate for each part based upon the part quality level and operating environment. The environment is divided into the categories of ground, space, naval, airborne and missile systems. One of the primary factors contributing to increased reliability is the part quality level. The fully qualified Class B military LSI device has less than one-half the expected failure rate of commercial devices procured to non-military standards (MIL-HDBK-217C, p.2.1.5-1). The parts count analysis is simpler but less accurate. A count is made of the number of parts and the expected equipment failure rate determined by comparison to previous experience with equipment of similar complexity.

A recent reliability assurance study confirms the value of using commercially proven parts procured to military standards. The non-military parts were subject to 2.7 times the number of removals as those procured to MIL-M-38510D, Class B (RAMFAS, p.7). The primary failure mechanisms were external package defects and oxide defects in the semiconductors. Reliability assurance seeks to identify such deficiencies and correct them in future systems by changing and improving the military standards.

Summary

Rome Air Development Center has developed a procedure for assuring the reliability of large scale integrated circuits. The procedure uses the parts of commercially proven architecture and circuit design while imposing strict procedures on the manufacture and testing of the devices. The selection of devices for inclusion is based on established criteria and provisions are inherent for technological advances. The rapid pace of microelectronics has not prevented the introduction of new and innovative designs for U.S. defense systems. Data collected from the field demonstrates that it is a valid procedure.

Acknowledgement

Assistance and encouragement from Dr. R. W. Thomas and Mr. J. B. Brauer is gratefully acknowledged.

References

Dicken, Howard, 1978, "MC 6800 Microprocessor and Related Peripheral Devices - Product Evaluations," RADC-TR-78-113, Rome Air Development Center (RBRM), Griffiss AFB, New York.

MIL-HDBK-217C, 1978, "Military Standardization Handbook, Reliability Prediction of Electronic Equipment," Rome Air Development Center (RBRT), Griffiss AFB, New York.

MIL-M-38510D, 1977, "Military Specification, General Specification for Microcircuits," Rome Air Development Center (RBRM), Griffiss AFB, New York.

MIL-STD-883B, 1977, "Military Standard, Test Methods and Procedures for Microelectronics," Department of Defense, Washington, D.C., FSC 5692.

Ostrowski, T. M., 1978, "Characterization of Complex Microprocessors and Support Chips," RADC-TR-78-138, Rome Air Development Center (RBRM), Griffiss AFB, New York.

Thomas, R. W., 1978, "Moisture, Myths, and Microcircuits," IEEE Transactions on Parts, Hybrids, and Packaging, Vol PHP-12, No. 3, September 1976.

"Reliability Analysis of Microcircuit Failures in Avionics Systems (RAMFAS)," RADC-TR-76-3, Rome Air Development Center (RBRP), Griffiss AFB, New York.

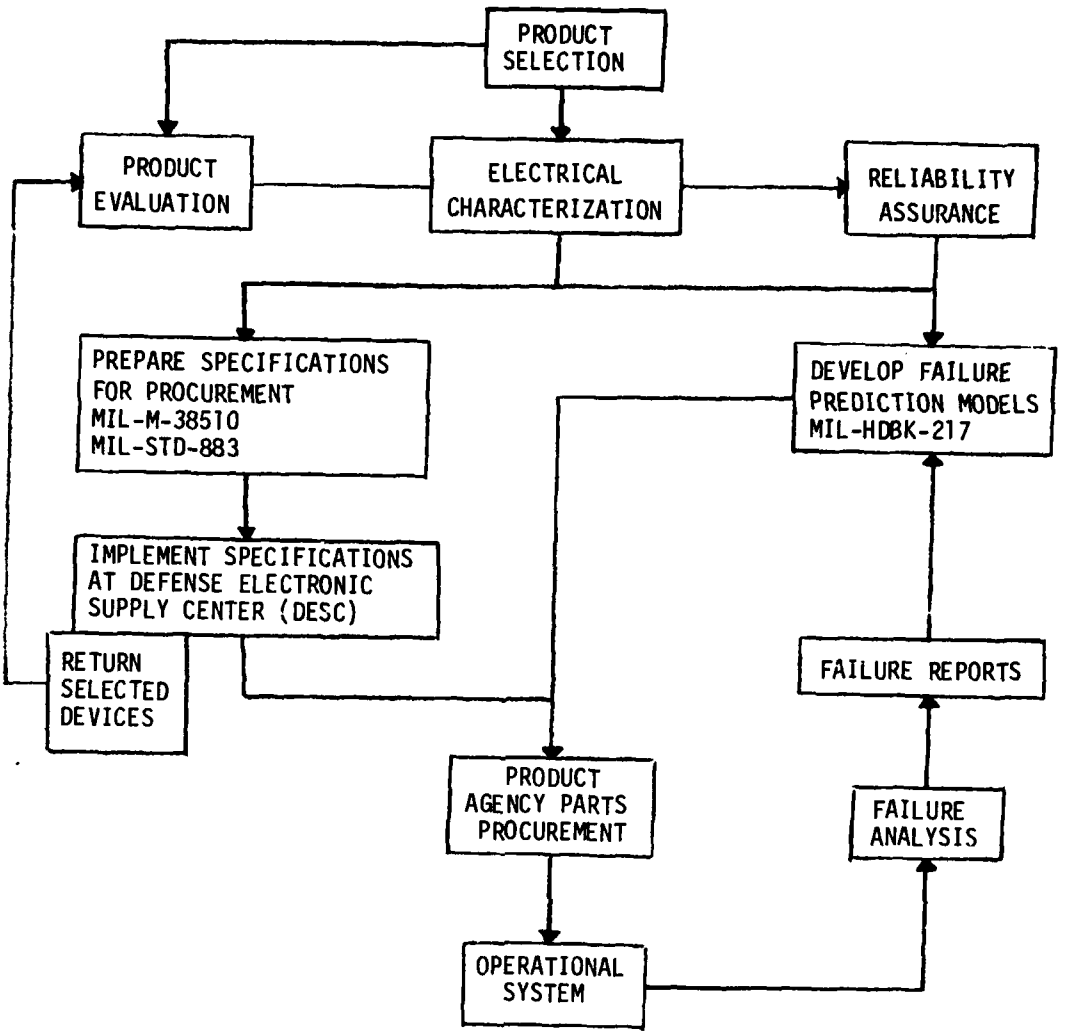


FIGURE 1 - Reliability Assurance for Large Scale Integrated Circuits at Rome Air Development Center

COMMERCIAL MICROPROCESSORS

DEVICE TYPE	TECHNOLOGY	WORD SIZE	STATUS
8080A	N-Channel Metal-Oxide (NMOS)	8 bit	QPL (38510/42001)
6800	N-Channel Metal-Oxide (NMOS)	8 bit	QPL (38510/40001)
1802	Complementary Metal-Oxide (CMOS)	8 bit	Procurement Specification Completed
2901A	Bipolar Low-Power Schottky Transistor	4 bit	Procurement Specification Completed
9900A	Integrated Injection Logic (I^2L)	16 bit	Procurement Specification Completed

FIGURE 2 - First Generation Microprocessors for the U. S. Military

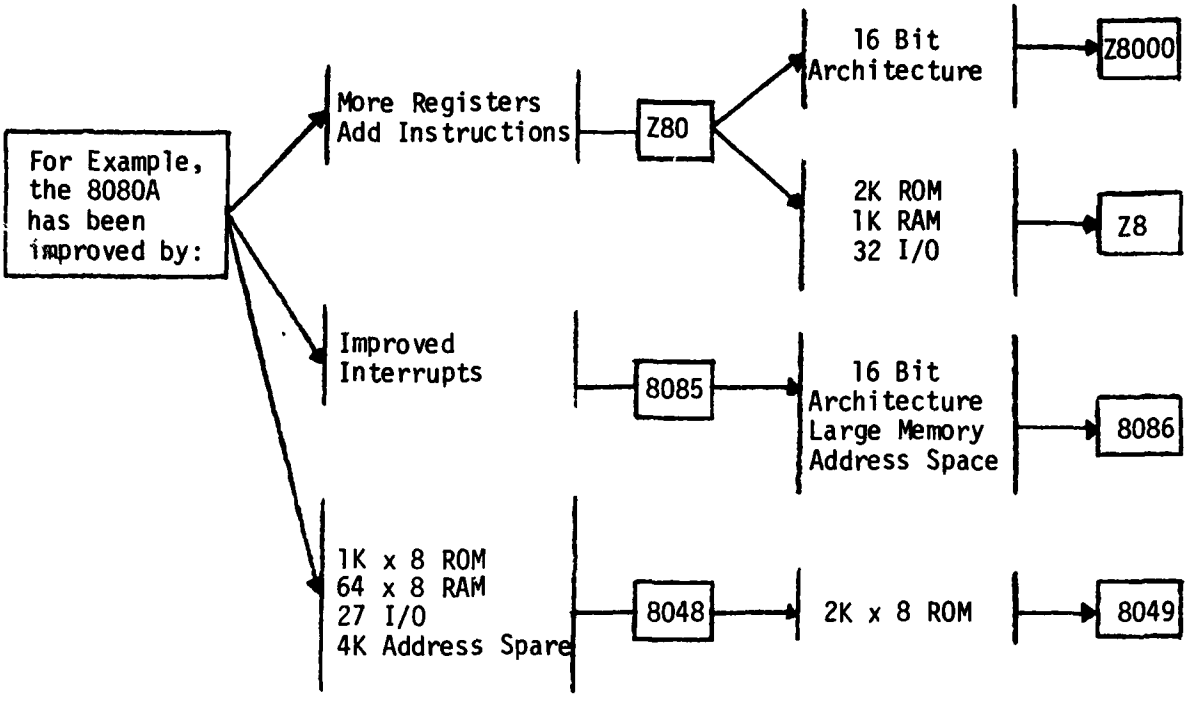


FIGURE 3 - Second and Third Generation Device Proliferation

DESTRUCTIVE TESTS	NON-DESTRUCTIVE TESTS
Internal Visual	Seal
Bond Strength	Internal Visual
Solderability	Radiography
Moisture Resistance	Particle Impact Noise Detection
Lead Integrity	Burn-In Screen
Salt Atmosphere	
Scanning Electron Microscope Inspection	
Steady State Life Test	
Die Shear Strength Test	

FIGURE 4 - Tests for Screening of LSI

SCREEN	CLASS S	CLASS B	CLASS C
Internal Visual	100% at 100X at 200X Magnification	100% at 75X to 150X Magnification	100% at 75X to 150X Magnification
Stabilization Bake	24 hours, 150°C min, all classes		
Temperature Cycling	10 cycles, -65 to +150, all classes		
Constant Acceleration	30,000g, all classes		
Seal: Fine and Gross	All classes		
Particle Impact Noise Detection	Class S only		
Burn-In Test	240 hours @ 125°C min	160 hours @ 125°C min	Not Required
Reverse Bias Burn-In	72 hours @ 150°C min (when specified)	Not Required	Not Required
Final Electrical Test	Per applicable device specification		
External Visual	100%	100%	100%

FIGURE 5 - Selected Screening Procedures for LSI, by Part Class (MIL-STD-883B, Method 5004.4)

DISCUSSION

R.Voles, UK

(1) How long does it take you on average when new devices come on the market before you characterize them in the way you describe?

(2) How many samples do you characterize for each device? Do you do it for every second source supplier too?

(3) The device manufacturers change their technology with time. Do you track them continually?

(4) Are your qualifications accepted by the other Services, Navy and Air Force?

Author's Reply

(1) The latest time to qualify a device completely is approx. 18 - 24 months.

(2) Yes. The samples aren't that many and are not enough to be statistically significant but we do sample for both primary and secondary sources.

(3) Yes. We track them and also get samples of the purchased items out of the warehouse inventory to ensure they haven't changed.

(4) Yes, we are the DOD OPR for microcircuit specifications.

P.D.T.O'Connor, UK

(1) Are you also characterising and qualifying the microprocessor system's complete boards in addition to individual chips?

(2) Would you see any advantage in having standard microprocessor systems rather than standard microprocessor devices?

Author's Reply

(1) Not yet. We qualify all the chips for a given family.

(2) We are doing this except we don't qualify on the board. We qualify the microprocessor, all the peripherals to make up the complete system, so that when the system is put together using the qualified parts we don't qualify the package.

P.Wust, Ge

You said you qualify the system. Do you qualify the hardware and software which comes with the hardware?

Author's Reply

Yes. We qualify the standard software package.

W.Ehrenberger, Ge

Do you recognise failures arising due to short circuits between the crystal and the upper layer, i.e. pinholes?

Author's Reply

The substrate, yes, that's why we use the scanning electron microscope to find pinholes. We magnify in our visual tests up to several thousands to search for these pinholes which will cause possible short circuits.

We are currently doing research to try to conduct external electronic tests to give us the same amount of information that visual tests used to.

RELIABILITY OF HIGH-BRIGHTNESS CRT's FOR AIRBORNE DISPLAYS

J.P. GALVES , J. BRUN
THOMSON-CSF, Electron Tube Division
Paris, France

SUMMARY

High-brightness monochrome or color cathode-ray tubes (CRT's) are becoming increasingly common in modern aircraft for head-up display (HUD) and head-down display (HDD) systems.

The specification, which includes electrooptical performance and environmental conditions, defines the tube quality at zero operating time. Two typical examples of specifications are given. The problems encountered in designing tubes, and the solutions used to obtain the desired level of quality are briefly discussed.

Reliability testing concerns random failures that occur during normal operation of the tube. After a short mathematical treatment of the principles involved, three examples of reliability tests carried out on CRT's manufactured by THOMSON-CSF are given.

The electrooptical characteristics of a CRT change during operation. This is mainly a cathode and screen wearing-out phenomenon. The life expectancy of a CRT depends on this evolution, and is thus a function of tube operating conditions.

1. INTRODUCTION

The use of high-brightness cathode-ray tubes (both monochrome and color) in "head-up display" and "head-down display" systems is becoming increasingly common in modern aircraft.

Because of the difficult environmental conditions (in particular, vibration, shock, ambient temperature, ambient lighting) encountered in aircraft, the cathode-ray tubes must be extremely reliable as they are involved directly in the success of the mission.

These tubes must be very carefully designed so as to avoid damage to their component parts, particular attention being required for the electron gun, the bulb and screen-deposition techniques, and the additional filter.

2. TUBE QUALITY AT ZERO OPERATING TIME

The quality of a tube at zero operating time is defined by its specification. This document defines performance figures, the way in which they are measured, and the Acceptance Tests. It normally consists of two parts : the electrooptical performance, and the allowable environmental conditions.

It should cover the worst possible operating conditions that are likely to be encountered. For example, the tube brightness should be such that, even under the most extreme ambient lighting conditions, the display is always readable.

In environmental testing, the stress levels must represent the worst possible operating conditions. Duration must be such that, at the end of testing, the tube has undergone a total set of stresses that is similar to that which would be encountered during the life of the tube.

Sometimes, only extreme stress levels, corresponding to the worst operating conditions are specified. These short-duration tests do not take the possibility of repeated stresses into account.

Two extracts from typical examples of specifications for avionics CRT's will now be given.

2. 1. TH X1614 E17 (Photo 1)

This is a 5" x 5", color penetration cathode-ray tube for head-down display (HDD) in military aircraft.

It has a directional filter for contrast enhancement and is potted, together with its precision-aligned deflection coils, in a metal shell.

- Electrooptical performance (without filter)

Color	Red	Yellow	Green
Line brightness (cd/m²)	550	1300	1600
TV-raster brightness (cd/m²)	160		1000
Line width (mm)	0. 2 to 0. 4	0. 2 to 0. 4	0. 2 to 0. 4

- Environmental performance

Over-pressure	:	0.3 MPa
Low pressure	:	190 mbar ; t = 2 h
Salt fog	:	T = 35 °C ; t = 48 h
Fungus	:	t = 28 days
Damp heat (cyclic)	:	T = 55 °C ; t = 48 h
Temperature	:	$-40\,°C \leq T \leq +95\,°C$
Shocks	:	30 g ; 11 ms
Sinusoidal vibration	:	10 to 60 Hz ; 1.5 mm peak-to-peak 60 to 500 Hz ; 10 g 500 to 2000 Hz ; 3 g Time per axis : 4 h Total duration : 12 h.

2. 2. TH 8408 E22 (Photo 2)

This is a 3" diameter monochrome tube for head-up display (HUD) systems. It is potted, together with its precision-aligned deflection coils, in a metal shell that provides magnetic shielding and assures precise location on installation in the equipment.

- Electrooptical performance

Line brightness	:	15 000 cd/m²
Line width	:	⩽ 0. 2 mm.

- Environmental performance

Over-pressure	:	0. 3 MPa
Low pressure	:	190 mbar ; t = 2 h
Salt fog	:	T = 35 °C ; t = 48 h
Fungus	:	t = 28 days
Damp heat (cyclic)	:	T = 55 °C ; t = 48 h
Temperature	:	–40 °C ⩽ T ⩽ + 95 °C
Shocks	:	30 g, 11 ms
Sinusoidal vibration	:	10 to 58 Hz ; 3 mm peak-to-peak 58 to 2000 Hz ; 20 g Time per axis : 4 h Total duration : 12 h
Random vibration (1)	:	50 to 100 Hz ; + 6 dB 100 to 1000 Hz ; 0. 3 g² /Hz 1000 to 2000 Hz ; –6 dB Time per axis : 15 min Total duration : 45 min
Random vibration (2)	:	Complex spectrum, reaching 0. 9 g² /Hz, for 6 min 15 s per axis.

2. 3. Problems Encountered In Designing Avionics CRT's

The two specifications given above closely resemble each other. However, the HUD tube, which is fixed to the main frame of the aircraft, is subjected to more severe vibrations.

If these specifications are compared with that of a conventional tube, such as a small CRT for computer terminals, we find that, in the latter case :

- the brightness is too low,
- no environmental guarantee is given.

If such a tube were subjected to the conditions under which avionics tubes must operate, the following problems would be encountered :

- during climatic testing : degradation of the resin bonding of the safety panel ; possible destruction of the tube,
- during mechanical testing : large displacements of the spot, making the image unusable ; broken connections ; broken filaments ; etc.

When designing and developing CRT's for airborne applications, the following points must thus receive very careful consideration :

- the electron optics, which must be capable of giving the very-high beam currents that are required for a high-brightness display,
- the mechanical structure of the electron optics, which must be capable of withstanding the severe vibrations and shocks,
- the tube mounting shell (which permits the tube to be mechanically mounted with precision in the equipment), which must satisfy mechanical and climatic criteria that are usually contradictory.

2. 4. Solutions

2. 4. 1. Electron Optics

Special electron guns, capable of giving very high cathode currents with high efficiency (beam current very close to cathode current), must be developed for avionics CRT's.

Because these tubes must be capable of giving good images in ambient lighting levels varying from night to direct sunlight, the guns must keep a good resolution at high beam currents, and variations in resolution due to changes in beam current must be minimized.

Because the guns use a very high cathode current, special cathodes, capable of giving very high current densities, are required.

The mechanical structure of these guns is also special : the support pins between the electrodes and the support rods are reinforced ; the connections between the gun and the neck and base of the tube are specially strengthened ; the connections between the electrodes and the base pins are either doubled, or of large diameter with multiple connections at the electrode and at the pin.

2. 4. 2 Mounting Shell

The mounting shell, in which the tube is potted, is designed as a function of the space available in the equipment, and of the vibration requirements. The most rigid structure possible is sought.

The selection of suitable potting compounds is of the utmost importance. A flexible resin is required to meet the climatic requirements. However, flexible resins may cause resonance problems under vibration conditions : this will cause the spot to move, and may break the filament or completely destroy the tube.

In practice, the best compromise is sought, this sometimes entailing the use of a combination of several different potting compounds.

3. RELIABILITY

A specification defines the quality of a tube at zero operating time. Because operating conditions are, in general, much less severe than those foreseen in the specification, the gap between the maximum acceptable stress level and the mean stress level has a very favorable effect on the operational reliability of the CRT.

3. 1. Mathematical Aspects of Reliability

The reliability of a product is expressed by the probability that it will not fail (or that it will survive) during a given time t, and under given operating conditions. Exactly what constitutes a failure must also be defined.

Failures are usually classed into two groups :

- catastrophic failures, that are sudden and total,
- degradation, that is progressive and partial.

3. 1. 1. Failure Rate

If R(t) is the reliability of the product (probability of survival during a period of time, t), the failure rate, $\lambda(t)$, is given by :

$$\lambda(t) = \frac{dR/dt}{R}.$$

The failure rate usually varies with product age as shown in Figure 1.

In zone 2, the failure rate is virtually constant, and equal to λ_o.

Zone 1 corresponds to premature failures. These faulty products are eliminated by a preliminary burn-in procedure.

3. 1. 2. Life Expectancy And Corresponding Standard Deviation

Zone 3, with its characteristic constant increase in failure rate, corresponds to the wearing-out of the product. As wearing out generally follows a Gaussian law, a representative sample of the product population can be used to determine a mean life expectancy, M, and a corresponding standard duration, σ, about this mean.

3. 1. 3. Mean Time Between Failures (MTBF)

Zone 2, with its low, constant failure rate (λ_o), corresponds to the useful working life of the product.

The mean time between failures, MTBF, is defined by :

$$\text{MTBF} = \frac{1}{\lambda_o} = m_o.$$

3. 1. 4. Mission Time And Product Age

For a product having an age T at the beginning of a mission, the reliability at a later time t, during the mission, is given by :

$$R(t) = \frac{R_u(T + t)}{R_u(T)} \cdot \exp - \frac{t}{m_o},$$

$$\text{where } R_u(T) = \int_t^{\infty} \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp - \frac{(T-M)}{2\sigma^2} \, dt.$$

It can be shown that the probability of failure due to the product wearing out is negligible, so long as $T < M-4\sigma$:

$$R_u(T + t) \sim R_u(T) \sim 1$$

$$R(t) \sim \exp - \frac{t}{m_o}.$$

R(t) is independent of product age, T, and only depends on the mission time, t.

3. 1. 5. Summary Of Mathematical Aspects

- The life expectancy, M, is characteristic of failures due to the product wearing out. It is defined as being the mean life expectancy of the different components of the product. M follows a Gaussian distribution law as a function of product age.
- The MTBF, m_o, is characteristic of random failures. It is defined as being the mean time between random failures.
- For reliable components, the MTBF, m_o, is appreciably longer than the life expectancy, M. In actual practice, if components are replaced as a preventive measure when their age, T, approaches M (usually when $T < M-4\sigma$), then the bigger the value of MTBF, the lower is the probability of random failure.

3. 2. Application To Reliability Testing Of CRT's

Many types of test for the evaluation of the MTBF of a product have been proposed. Only one type, however, is easily applied to CRT's ; "truncated tests without replacement".

A group of n CRT's are operated for a time T, after which C tubes are found to be defective. Then the MTBF ($\bar{m}$) can be estimated by means of the expression :

$$\bar{m} \sim \frac{1 - C/2n}{C/n} \cdot T$$

For a given confidence level, a corresponding minimum value of MTBF can be determined. For example, for a confidence level of 60 % :

$$m_{60\%} > \frac{n - C}{C + 1} \cdot T.$$

As an example, consider a batch of 10 CRT's that are operated for 500 hours, after which one tube is found to be faulty (C = 1). Then the most probable MTBF, $\bar{m}$, is

$$\bar{m} = 4750 \text{ h}.$$

For a 60 % confidence level, the MTBF is

$$m_{60\%} \geq 2250 \text{ h.}$$

3. 3. Burn-In

The preceding considerations concerning the MTBF ($m_o = \frac{1}{\lambda_o}$) suppose that premature failures have been eliminated. For the user, the term MTBF has a slightly different significance. The operational MTBF, m, is operating time summed over n equipment divided by the number of failures occuring during operation. Premature failures must thus be eliminated as far as possible because, although they do not affect the MTBF as mathematically derived (m_o), they do reduce the operational MTBF, m. Premature failures are eliminated by burn-in procedures.

3. 3. 1. Mathematical Aspects Of Burn-In Procedures

Imagine that, once premature and wearing-out failures have been eliminated, the MTBF ($m_o = \frac{1}{\lambda_o}$) is known. The practical problem is to find the burn-in time, T, that would be necessary to ensure that the MTBF after burn-in,m, was a known fraction K of m_o. That is to say :

$$m \geq \frac{m_o}{K} .$$

To do this, n CRT's are operated for a preliminary time t, after which C failures are noted (which are then replaced or repaired). To obtain $m = \frac{mo}{K}$, the burn-in must be continued for a total time T such that :

$$T = 1.35 \left[\frac{1-\alpha/2}{\alpha(K-0.5)}\right] t,$$

where $\alpha = C/n$.

During this burn-in, faulty tubes are either replaced or repaired. Deciding on a value for t can be very difficult, but it can be taken to be 5 % of the life expectancy.

If the failure rate is very low, a minimum MTBF can no longer be guaranteed because the required burn-in time, T, becomes impractically long. If the failure rate is zero, the burn-in is stopped after a time t : we can then assume that the absence of premature failure has been proved, and that the MTBF, m, is close to m_o.

3. 3. 2. Practical Application

Let us consider the case of a CRT installed in the cockpit of an aircraft. The failure rate is defined by a MIL Spec. as being :

$$\lambda_o = r \times 15 \text{ failures}/10^6 \text{ h,}$$

where r is an environmental factor. This factor, which applies to various types of components, varies from 0.5 for ground-based applications, with zero stress and optimum personnel qualification (operators and maintenance crews), to 80 for missiles. This factor multiplies the failure rate, and results in a reduction in reliability when stress levels increase.

For equipment in aircraft cockpits, r is equal to 6.5. That is to say :

$$\lambda_o = 100 \text{ failures}/10^6 \text{ h,}$$

and :

$$m_o = 10{,}000 \text{ h.}$$

Suppose that the aim is to guarantee an operational failure rate of less than 1 ‰ for missions of 2 h. From paragraph 3.1.4, we have the expression :

$$R(t) \sim \exp -\frac{t}{m} \sim 1 - \frac{t}{m} \geq 0.999,$$

so, with t = 2 h, we have :

$$m \geq 2000 \text{ h.}$$

So, the burn-in must assure that the ratio $\frac{m}{m_o}$ $(= \frac{1}{K})$ is greater than or equal to $\frac{2000}{10{,}000} = \frac{1}{5}$.

For an expected life of 500 hours, the initial burn-in time is set at :

$$t = 500 \times 0.05 = 25 \text{ h.}$$

If, after these 25 hours, no failure is noted, it can be assumed that no premature failures occur, and that the MTBF will be close to m_o = 10,000 h.

If a non-zero failure rate, 10 % for example, is found, then the burn-in should be continued for a time T given by :

$$T = 1.35 \times \frac{1 - 0.05}{0.1 \times (5 - 0.5)} \cdot 25$$

$$= 71 \text{ h.}$$

3. 3. 3. Implementation

Common pratice is to carry out burn-in procedures for 24 h to 50 h. The burn-in is performed with the tube operating under conditions that represent average CRT utilization. This procedure permits most premature electrooptical failures to be eliminated, and allows the tube characteristics (brightness, cathode emission, cut-off voltage) to stabilize. In some cases, burn-in procedure includes vibration and temperature cycles.

Tube burn-in is usually followed by equipment burn-in, which effectively prolongs the former.

Special equipment (tube-driving bays, environmental testing equipment) is required for these tests which are thus expensive. In addition, they increase delivery delays. However, they reduce maintenance costs by reducing the operational failure rate.

3. 4. Reliability Testing

Laboratory testing is commonly used to establish the level of reliability of a new product. The tests must simulate real operating conditions as faithfully as possible. We give, below, three examples of reliability tests carried out by the Electron Tube Division of THOMSON-CSF.

3. 4. 1. TH X613 E17

The TH X613 E17 is a 5" x 4" color penetration CRT for aircraft cockpits. It is the first small-format color CRT to be specially developed for this application.

Reliability tests were carried out using a company-designed, 10-position, automatic test bay that permitted performing complete operating cycles without human intervention.

The operating conditions, chosen as a function of known operational data, were as follows :

- 24 h cycle consisting of 20 h operation and 4 h switched off,
- red mode : 10 h/day,
- green mode : 10 h/day,
- max. brightness : 6 min/day,
- half brightness for the rest of the time,
- image : one stationary trace plus one moving trace,
- sinusoidal vibrations : 0. 3 g at 40 Hz and 80 Hz, each for 1 h/day,

40 tubes were tested in this way for 500 hours, giving a total accumulated test time of 20,000 hours. No failure was noted, so we can say that the MTBF is better than 20,000 hours. MIL-Spec. HDBK 217 requires 10,000 hours for a CRT used in an aircraft cockpit.

In our tests, the stress level was fairly low (tube operated at high brightness for only a short time ; no climatic testing), hence the better MTBF.

3. 4. 2. TH X694 E21 (Photo 3)

The TH X694 E21 is an 8" x 10" color penetration CRT, destined for installation in display consoles for "Space Lab".

A series of 10 independent test bays were built specially for these tubes which were to be operated, non-stop, under the following conditions :

- 13 kV high voltage (yellow mode),
- cathode current, I_k, of 100 μ A,
- full-screen TV raster.

A first set of 10 tubes is at present undergoing the following climatic tests :

- temperature cycle : 20 °C, 50 °C, 0 °C, 50 °C, 20 °C,
- duration : 1 day,
- frequency : at the following times after start-up ; 0 h, 168 h, 500 h, 1000 h, 1600 h, 3000 h.

A second set of 10 tubes is at present undergoing the following vibration tests :

- 5 to 8.5 Hz : 10 mm peak-to-peak,
- 8.5 to 80 Hz : 2 g,
- 80 to 100 Hz : 1.4 g,
- 100 to 2000 Hz : 0.5 g,
- Duration : 1 scan at 4 octaves/min.

As these tests are not finished yet, the results are not known at present.

3. 4. 3 TH 8408 E22

The TH 8408 E22 is a 3" diameter, monochrome CRT for HUD (see paragraph 2.2.). An automatic, 10-position bay has been specially designed for these tests, and the tubes are operated under the following conditions :

- 24 h cycle consisting of 10 simulated missions, each one lasting 1 h 30 min, plus a "standby" and an "off" phase,
- maximum brightness during 6 missions,
- half-brightness during 2 missions and standby,
- low brightness during 2 missions,
- 10-line image : 9 moving and 1 stationary,
- each mission consisting of a climatic cycle :
  - 20/50/20 °C for 8 missions,
  - 20/70/20 °C for 2 missions,
  - 20/−20/20 °C for standby,
  - −30 °C or + 20 °C or + 70 °C for the "off" phase,
- each mission has a random vibration sequence during a stable temperature period :
  - 1 min, 0.1 g^2/Hz at 300 to 1000 Hz for 1 mission,
  - 1 min, 0.001 g^2/Hz at 10 to 2000 Hz for the other nine missions.

10 tubes have undergone these tests for 1000 hours.

- 1 tube developed a repairable defect (stray emission),
- 2 tubes showed a change in cut-off voltage. This drift was continuous and progressive. Provided that provision for such variations has been made in the display system, they would not result in system failure within its design limits.

If the drifts are counted as failures, then the MTBF is given by :

$$m_{60\%} \geq \frac{10 \cdot 3}{4} \cdot 1000 = 1750 \text{ h},$$

$$m \sim \frac{10 \cdot 1.5}{3} \cdot 1000 = 2800 \text{ h}.$$

If the drifts are not counted as failures, then the MTBF is given by :

$$m_{60\%} \geq \frac{10 - 1}{2} \cdot 1000 = 4500 \text{ h},$$

$$m \sim \frac{10 - 0.5}{1} \cdot 1000 = 9500 \text{ h}.$$

If no failure had occured, then we would have been able to show, in the best case, an MTBF of :

$$m_{60\%} \geq \frac{10}{1} \cdot 1000 = 10{,}000 \text{ h}.$$

By comparing these results with those obtained in 3.4.1. we see that, although the HUD tube is designed for a more severe environment than the HDD tube, the inclusion of much more severe environmental conditions in its reliability test procedure results in an MTBF that is lower than that of the HDD tube.

A HUD tube is subjected to higher stress levels than those normally encountered in aircraft cockpits. MIL-Spec. HDBK 217 defines an MTBF of 10,000 hours for normal cockpit conditions. We can thus conclude that the reliability tests carried out on the THOMSON-CSF tubes show that this specification is fulfilled and that, for a HUD CRT, the environmental factor, r, is on the order of 13 instead of the normal-cabin-condition value of 6.5.

4. WEARING-OUT : EVOLUTION OF TUBE CHARACTERISTICS WITH TIME

During operation, the characteristics of a cathode-ray tube change because of normal wearing out of its principal components. This wearing out, or ageing, does not result in tube break-down or "catastrophic failure" but simply in a degradation of its characteristics.

This degradation can be quantified, and it is then up to the user to define the acceptable limits corresponding to his operating conditions.

The two principal CRT components that are subject to wearing out are the screen and the cathode, with consequent modification of the following tube characteristics : brightness, modulation voltage, line width, spatial resolution, etc...

If tube life is to be determined from specific, well-defined criteria, the ways in which these components change must be considered separately.

4. 1. Screen Evolution

All avionics CRT's, whether they be for HUD or HDD applications, must give extremely high-brightness images so that readability is good under the intense lighting levels found at high altitudes.

These high brightnesses are obtained by using extremely high cathode currents, and thus by working with very high screen loading (in $\mu W/cm^2$ or $\mu A/cm^2$). Because of this, screen ageing is much more rapid with avionics tubes than with conventional CRT's.

This ageing effect results in a drop in the screen's luminous efficiency, this leading to a reduction in screen brightness for given, constant operating conditions.

In general, the useful life of a phosphor is considered as being finished (although, in fact, it never completely stops working) when the brightness (and luminance efficiency) has dropped to half of its original value.

It is generally admitted that screen ageing depends mainly on the total electrical charge received by the screen (in coulomb/cm^2) during operation. It is also accepted that ageing, L/L_o, is given by the equation :

$$\frac{L}{L_o} = \frac{1}{1 + \frac{Q}{Q_{0.5}}},$$

where L_o is the initial brightness,

$Q_{0.5}$ is the charge resulting in a 50 % drop in brightness,

λ is the brightness corresponding to a screen charge Q.

For phosphor types P1, P43 and P44, which are commonly used in monochrome avionics CRT's, the coulomb rating ($Q_{0.5}$) is on the order of 100 coulombs (see Figure 2).

4. 1. 1. Color Penetration Screens

We can now consider the evolution of the color penetration screen type E17, used in the THX1614 and other HDD CRT's. When these tubes are used at maximum brightness, the corresponding screen current density is on the order of 15 $\mu A/cm^2$ for each mode (red, yellow and green).

A test procedure has been developed that permits the coulomb rating of such screens to be determined. The screen is scanned in a TV raster that is divided into four zones. The cathode current, I_k, is modulated so that the current density in the zones is 15, 10, 5 and 1 $\mu A/cm^2$ for each of the operating modes (see Photos 4 and 5). The coulomb rating also depends on the current density for penetration phosphors (unlike the case for conventional phosphors) :

$Q_{0.5}$ = 108 C for 5 $\mu A/cm^2$ and 15 kV VHV

$Q_{0.5}$ = 172 C for 10 $\mu A/cm^2$ and 15 kV VHV

$Q_{0.5}$ = 216 C for 15 $\mu A/cm^2$ and 15 kV VHV.

Figures 3 and 4 show an example of values of L/L_o as a function of screen loading and time. The following points should be noted :

- with color penetration screens, the coulomb rating is not independent of screen excitation conditions,
- no matter what operating mode is used, the coulomb rating is at least as large as that of one of the better conventional phosphors ($Q_{0.5}$ = 100 C),
- the coulomb rating always gives corresponding times of at least 4000 hours for L/L_o = 50 % (see Figure 4).

4. 2. Cathode Evolution

The cathode is the second element of a cathode-ray tube whose characteristics change during operation. These changes result in a drop in cathode current if the bias potentials of the other gun electrodes are not changed.

Cathode characteristics can change for several reasons :

- reaction between the emissive elements of the cathode, and degassing products of the metal electrodes, internal coatings, and glass walls of the tube.
- structural change in the emissive elements (oxides),
- ionic bombardment,
- etc.

So far as the tube is concerned, the wearing-out of the cathode results in a change in the I_k-Vg curve (see Figure 5) and, as a consequence :

- the cut-off voltage changes :

  $V_{c0} \rightarrow V_{c1}$

- the bias voltage (and modulation voltage) changes :

  $V_{b0} \rightarrow V_{b1}$

- the maximum cathode current, at Vg = 0, is reduced ($I_{k0} \rightarrow I_{k1}$).

Usually, the spatial resolution also changes, this being mainly due to an increase in the modulation required to obtain the same current.

A test procedure has been established to evaluate and predict the evolution in cathode characteristics. This technique, known as the "Dip Test", permits measuring the remaining emission capability of the cathode ; the electrical characteristics start to change once this remaining emission capability has been used up. The rate at which this reserve is used up can be used to evaluate how long it should be possible to operate the tube before the electrical characteristics change.

The rate at which the reserve is used up obviously depends on the conditions under which the cathode must operate :

- electron gun structure (beam resolution),
- beam current (screen brightness).

In other words, it depends, on the tube characteristics and on the operating conditions.

5. USABLE LIFE

In the preceding sections we have considered the MTBF of CRT's : that is to say, the probability of random, unpredictable breakdowns occuring during operation that make the tube, which is mounted in a piece of equipment, unusable. We have also shown how the electrooptical characteristics evolve during use due to wearing out, or ageing, of the cathode and the screen.

This evolution results in a reduction in tube performance. So, we must define what reduction can be accepted before the tube is considered as being unusable. The time taken to reach this performance level is the usable life of the tube.

We have already indicated that the changes in cathode and screen characteristics that result in reductions in tube performance depend mainly on the operating conditions. So, a usable life cannot be defined unless we know :

- what performance levels correspond to the end of tube life,
- the operating conditions.

These two pieces of information can only be supplied by the user of the tube.

Below, we give an example of operating conditions supplied by a user. The tube in question is a color penetration CRT for use in the cockpit of commercial aircraft.

The tube is operated 16 hours per day with the brightness indicated in Figure 6 and below :

- 3 h at 100 % brightness,
- 10 h at 50 % brightness,
- 3 h at 5 % brightness.

The temporal (cathode) and spatial (screen) duty cycles are always 30 %.

The end of usable tube life is considered to be when the brightness has dropped to 50 % of its initial value.

Special life-testing bays have been constructed to permit defining the usable life that corresponds to these operating conditions. Of course, these bays are not as complex as the equipment for which the tubes are destined, and they do not permit displaying the same images. However, they have been designed so that screen and cathode loading correspond to the required values. These life tests are at present underway in our laboratories.

6. CONCLUSION

First introduced in the early days of electronics, the CRT is a contemporary of the first receiver tubes (vacuum diodes, triodes, etc.). Although receiver tubes have virtually disappeared, having been supplanted by solid-state devices, CRT's are still widely used. This is because CRT technology has evolved greatly, and devices that could replace them have not reached the same levels of performance and reliability.

The reliability of present-day tubes is so good that the CRT is quite suitable for use in airborne display systems Over 10 years experience has now been accumulated with CRT's operating under the very severe requirements of military airborne systems, and the Mirage 2000 will be the first military aircraft in the world to take advantage of this, giving the pilot multicolor displays with all the associated advantages.

We wish to acknowledge the support of the STTA and the STAE for this work.

Note - Tube references including an "X" refer to developmental models.

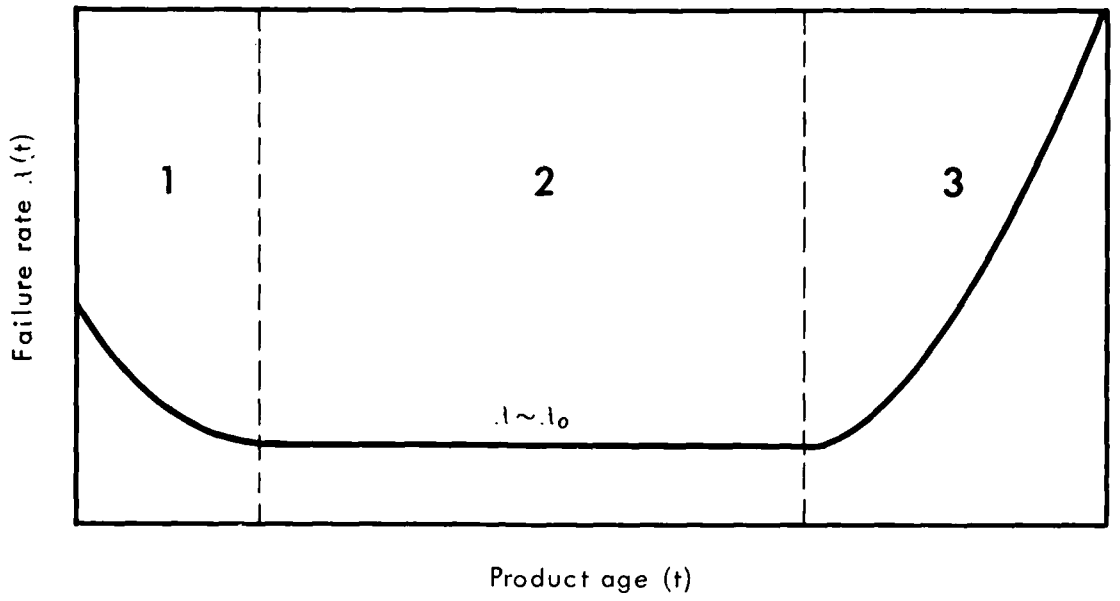


Figure 1 - Variation of failure rate with product age.

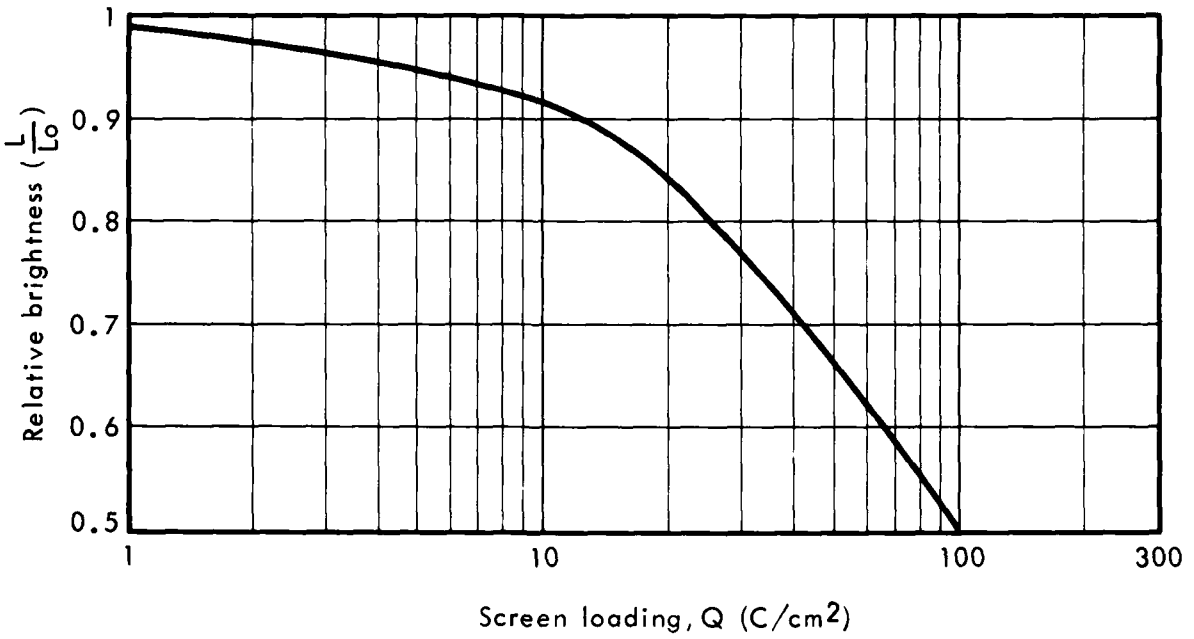


Figure 2 - Change in brightness with screen loading-P1.

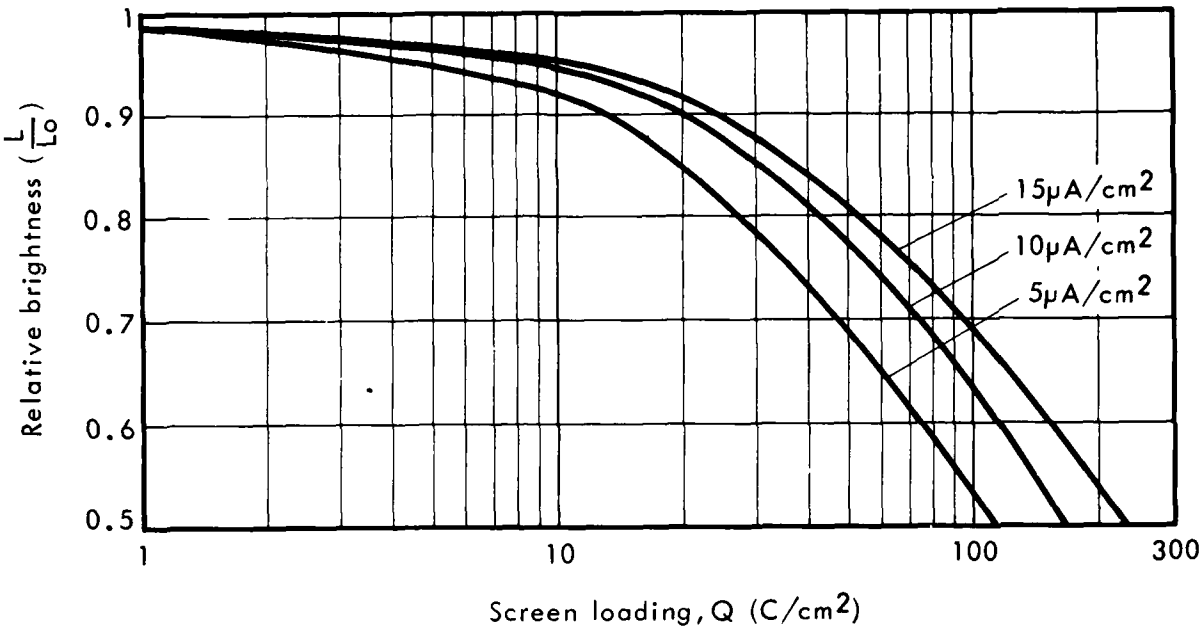


Figure 3 - Change in brightness with screen loading - E17.

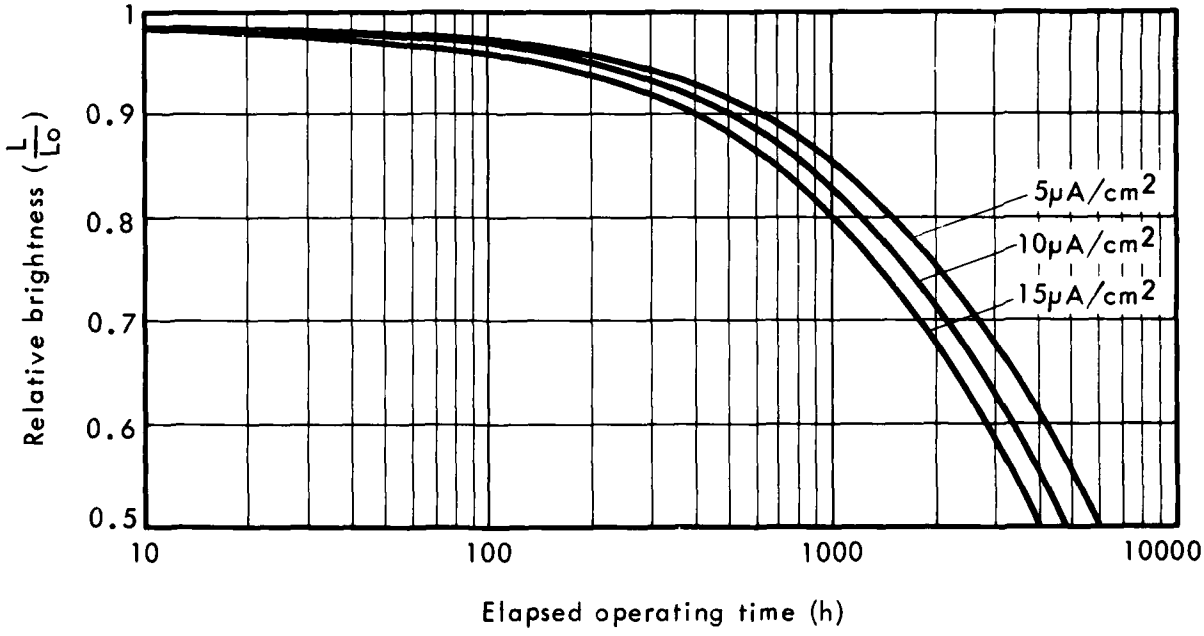


Figure 4 - Change in brightness with elapsed operating time - E17.

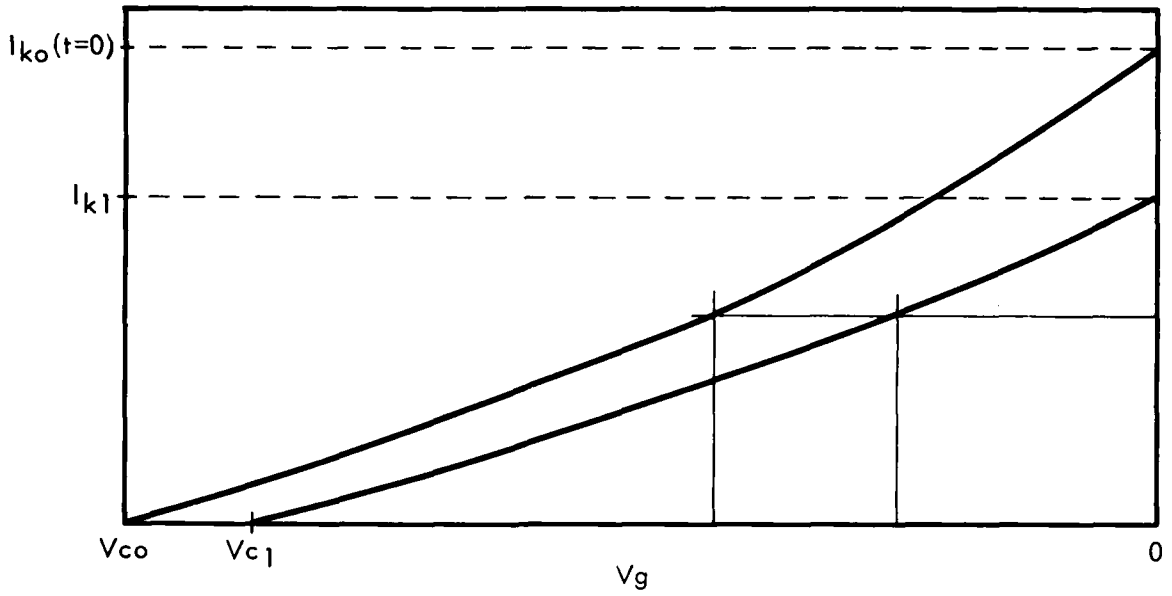


Figure 5 - Change in the I_k vs Vg curve with time.

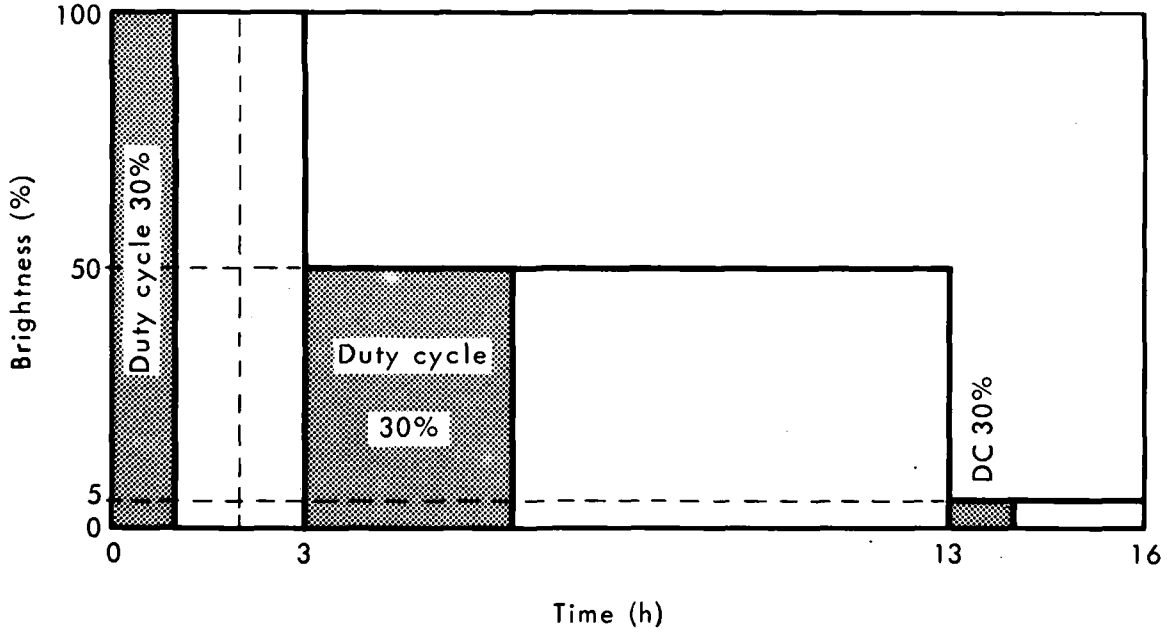


Figure 6 - Typical requested daily operating cycle for a color penetration CRT in a commercial aircraft.

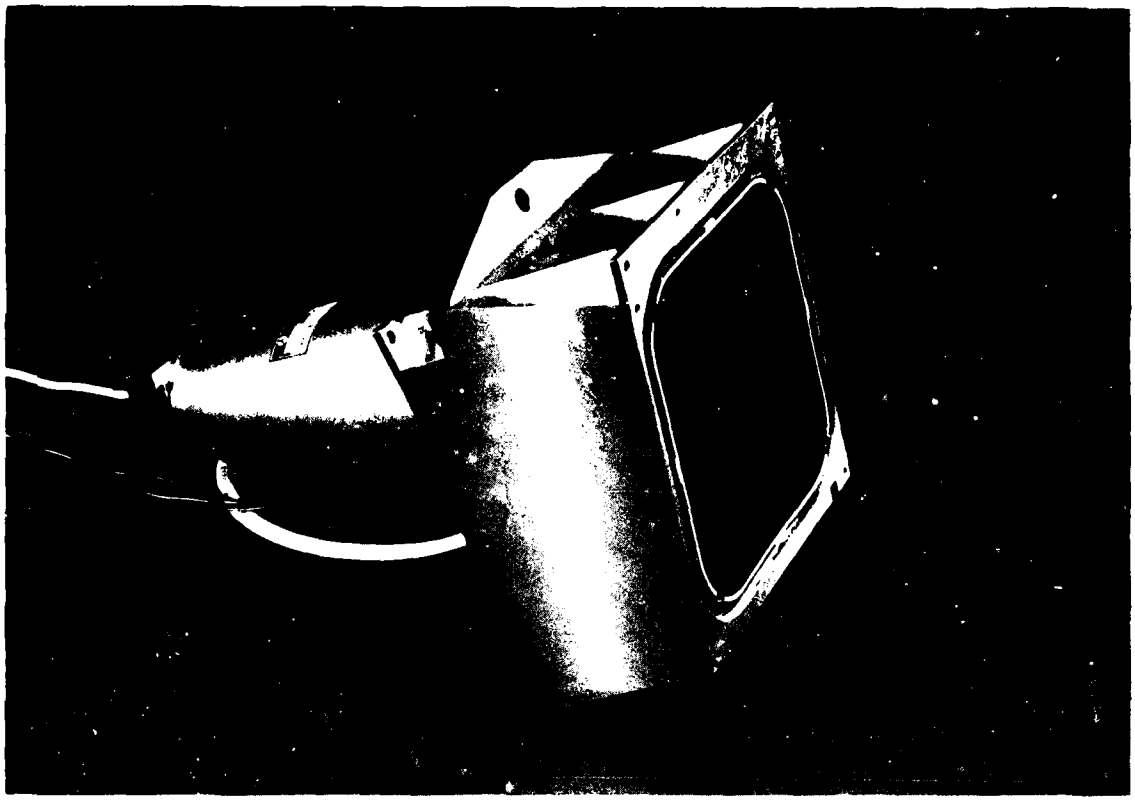


Photo 1 - The TH X1614 E17, a 5" x 5" color penetration CRT for head-down displays.

Photo 2 - The TH 8408 E22, a 3" diameter monochrome tube for head-up displays.

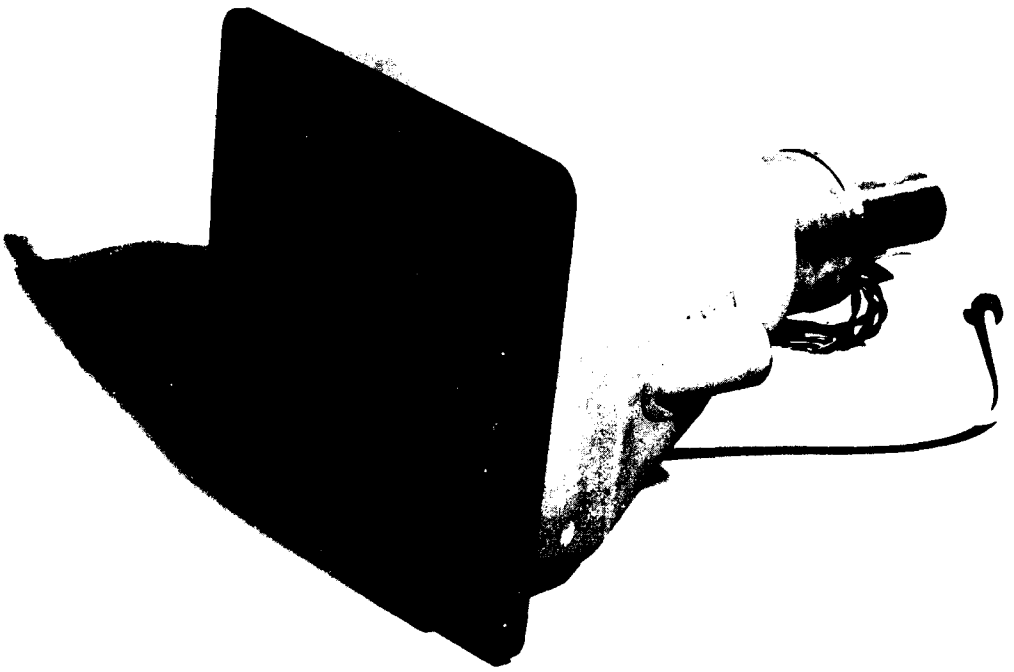
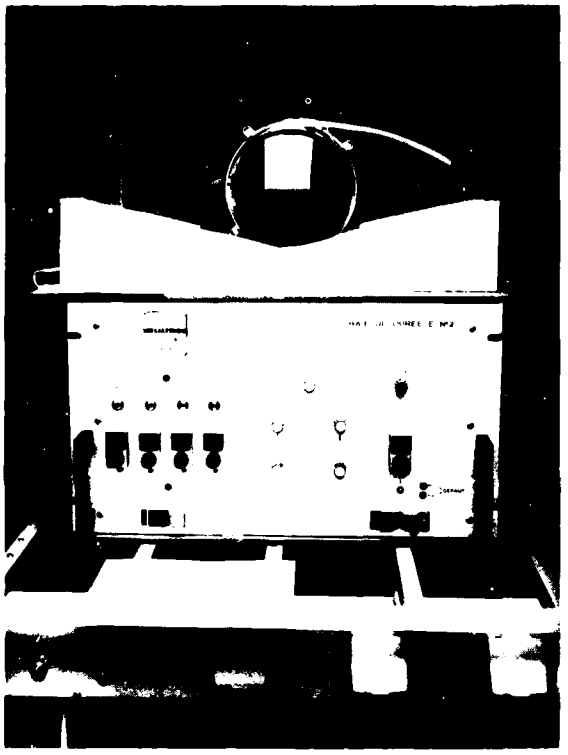
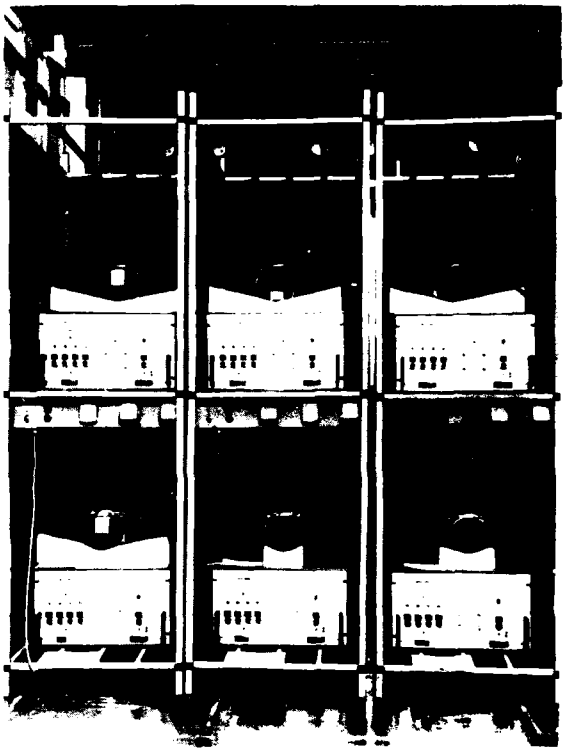


Photo 3 - The TH X694 E21, a 8" x 10" color penetration CRT for Space Lab display consoles.





Photos 4 and 5 - Life testing bays.

RELIABILITY INVESTIGATIONS ON AN AUTOMATIC TEST SYSTEM

Hans-Hermann Molter
Messerschmitt-Bölkow-Blohm GmbH
P.O. Box 801149, D-8000 Munich 80

SUMMARY

Statistical methods are used to determine the reliability of a complex guided missile test system. When operating such a system, failures occur which are documented in failure reports. The failure-free periods between successive failure events are evaluated in the context of a Weibull distribution, yielding statements as to the "Mean Time Between Failures" (MTBF) and the type of failure. The results are used to compare various systems; the influences exerted by differing operating conditions likewise become evident.

The values determined in this way also contribute to localizing weak points, thus enabling purposive design measures to further increase the reliability of the test system.

1. INTRODUCTION

The guided missile KORMORAN is developed by Messerschmitt-Bölkow-Blohm GmbH (contracted by the German Ministry of Defence) and belongs to the air-to-ship weapon system KORMORAN/F104G. It features an active seeker which permits it to navigate itself to the target area.

The missile is approximately 4.4 m long, has a wing span of approx. 1.0 m (cross-winged version) and weighs about 6000 N. The KORMORAN is launched from the F104G or from MRCA-TORNADO.

An automatic test system (ATG) has been developed for the weapon system to subject the guided missile KORMORAN to static and dynamic functional checking.

The ATG features a self-test programme, the tests being controlled fully automatically by a process computer.

The static test basically serves to check all supply voltages, static signals and logic levels of the guided missile.

In the course of the dynamic test, the interaction of all guided missile functions is checked by means of a simulated target approach. This test is effected in a closed control loop, the signals required to simulate targets and jammers being generated and radiated via horn feeds in an anechoic chamber.

The ATG can be shipped by land, sea or air. The entire test equipment, including all accessories, is accommodated in a 20 ft. ISO container. Fig. 1 shows the basic ATG configuration.

Two systems, ATG1 and ATG2, are employed as test systems in the course of the series production of the guided missile KORMORAN.

A third system, ATG3, is currently undergoing integration and will be supplied to the troops in summer. The experience obtained on developing and operating ATG1 was taken into account on developing ATG2 and ATG3. ATG1 thus differs in some points from the other two systems.

When such a complex, highly-integrated system is operated, failures naturally occur which are noted in the log-book and recorded on failure report forms (Fig. 2). The periods between the occurrence of two failures, that is, the failure-free times, are utilized to provide statements concerning the reliability of a unit. A Weibull curve is fitted to this times-to-failure data.

2. MATHEMATICAL PRINCIPLES

2.1 Weibull Distribution

Life observations can frequently be described extremely well by the Weibull distribution which, due to the presence of three parameters t_0, T and b, can be adapted very flexibly to the observations made (Graf/Henning/Wilrich, 1974). Its probability density is $f(t) = \frac{b}{T}\left(\frac{t-t_0}{T}\right)^{b-1} \exp\left[-\left(\frac{t-t_0}{T}\right)^{b}\right]; \quad t > t_0 \geq 0$

whereby t_0 is the point in time at which the forward reckoning begins. T signifies the characteristic life and is a criterion of the location ($T > t_0$), b is a parameter for the shape of the distribution.

The distribution function $F(t) = 1 - \exp\left[-\left(\frac{t-t_0}{T}\right)^{b}\right]$

indicates the probability with which an element fails up to time t.

The failure rate $\lambda(t)$ is defined as $\lambda(t) = \frac{f(t)}{1-F(t)} = \frac{b}{T}\left(\frac{t-t_0}{T}\right)^{b-1}$

and is constant at b = 1 only, and thus does not depend on the life. Inasmuch as $(t-t_0)^0=1$, the Weibull distribution at b = 1 reduces to the exponential distribution.

$\lambda(t)$ decreases monotonically for $b<1$, the failure rate indicating hidden faults or "teething troubles".
$\lambda(t)$ increases monotonically for $b>1$, which indicates aging and wear.

In Fig. 3 and 4, f (t') and $\lambda(t')$ are represented in graph form versus the relative life t' = (t-to)/T .

2.2 The Life Distribution

The distribution function F (t) can be expressed as $\frac{1}{1-F(t)} = \exp\left[\left(\frac{t-t_0}{T}\right)^b\right]$

and, by taking the logarithm twice, as $\ln \ln \frac{1}{1-F(t)} = b \ln (t-t_0) - b \ln T$

which is an equation of straight lines in which b signifies the slope and T the position of the straight lines. This permits drawing a distribution which is applied in the following (Deutsche Gesellschaft für Qualität, 1975).

T is obtained by reading off the abscissa value at the straight line at point F (T) = 63.2% ($F(T)=1-e^{-1}=0.632$). A scale for $a = (\bar{t}-t_0)/T$ is also attached, whereby $\bar{t}$ = MTBF is the arithmetic mean value of the life of the units investigated.

2.3 Test Method for Shape Parameter b

The following table (see Verband der Automobilindustrie, 1976), permits calculating one-sided upper and lower confidence limits β_o and β_u for the shape parameter β of the life distribution with the probability $1-\alpha = 95\%$. The required factors $F_b(n)$ depend on the sampling scope n.

n	5	6	7	8	9	10	11	12	13	14
F_b (n)	1.64	1.58	1.54	1.50	1.47	1.45	1.43	1.41	1.39	1.38

n	15	20	23	25	30	35	40	45	50	55
F_b(n)	1.37	1.32	1.29	1.28	1.26	1.24	1.22	1.21	1.20	1.19

The following apply: $\beta_o = b \cdot F_b(n)$

$\beta_u = b/F_b(n)$

Using β_o or β_u, respectively, permits performing the following statistical test:

Hypothesis H_0 : The true shape parameter β, which is estimated through b, is compatible with a nominal value β_1, $\beta = \beta_1$.

Counter-hypothesis H_1: β is incompatible with β_1, because $\beta < \beta_1$ or $\beta > \beta_1$ applies.

Level: $\alpha = 5\%$
Sample: scope n
result b
Threshold value: $F_b(n)$
Confidence limits: $\beta_o = b \cdot F_b(n)$ or $\beta_u = b/F_b(n)$
Decision: if $\beta_o < \beta_1$ or $\beta_u > \beta_1$, H_o is rejected.

The test can be applied specially to the case $\beta_1 = 1$ in order to distinguish between early failures, failures due to wear, and true random failures.

2.4 Evaluation Procedure

It is immaterial in the case of life investigations whether many units are observed up to failure, or one unit only which is repaired after each failure, that is, regains its original status. In the present case only three units have been built, so that the second possibility alone is feasible.

Each failure of a unit is documented in a failure report evaluated separately for each unit. Design and operating failures are not considered. The following operations are performed:

1. Determination of the times t between two successive failure reports
2. Classification, if appropriate (for n > 30)
3. Formation of the relative cumulative frequencies F (t)
4. Entering F (t) versus t in the Weibull distribution
5. Reading off T
6. Reading off b and a
7. Calculation of MTBF
8. Allocation of failure reports over the four subsystems
   - processor and peripheral equipment
   - missile-oriented RF peripheral equipment
   - missile-oriented LF peripheral equipment
   - container assembly.
9. Repetition of steps 1 to 7 for the four subsystems.

3. EVALUATION OF THE ATG1 FAILURE REPORTS

3.1 Joint Evaluation of all ATG1 Failure Reports

52 Failures which could be evaluated occurred over a period of 21 months in connection with ATG1. Determining the (failure-free) times t between two successive failure reports and subsequent classification yield the following table:

Serial No. i	Upper Category Limit t_i' [h]	No. n_i	$\sum_{j=1}^{i} n_j$	Relative Cumulative Frequency F_i [%]
1	7	15	15	29.4
2	33	12	27	52.9
3	59	7	34	66.7
4	124	6	40	78.4
5	254	7	47	92.2
6	449	3	50	98.0
7	> 449	1	51	100.0
		n = 51		

The representation within the Weibull distribution for $t_0 = o$ shows that points (t_i' , F_i) can be balanced extremely well by a straight line (Fig. 5). Assuming the Weibull distribution for the life t is thus justifiable. Values so derived are:

$$T = 50 \text{ h}$$
$$b = 0.57$$
$$a = \bar{t}/T = 1.6 \quad \bar{t} = a \cdot T = \underline{80 \text{ h} = \text{MTBF}}$$

3.2 Evaluation for the Subsystems

3.2.1 Subsystem Processorand Peripheral Equipment

Determining and ordering the times t between two failures from this subsystem yields the following table when the relative cumulative frequency F_i is determined according to the equation

$$F_i = \frac{i}{n+1}$$

where n is the number of the failure-free times exhibited by this subsystem; in this case n = 23.

Serial No. i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Operating time t_i [h]	3	3	7	7	26	33	39	46	46	46	52	72	91	130	208
F_i [%]	-	8.3	-	16.7	20.8	25.0	29.2	-	-	41.7	45.8	50.0	54.2	58.3	62.5

Serial No. i	16	17	18	19	20	21	22	23
Operating time t_i [h]	234	241	267	312	319	475	494	741
F_i [%]	66.7	70.8	75.0	79.2	83.3	87.5	91.7	95.8

The representation within the Weibull distribution for $t_0 = o$ shows that points (t_i, F_i) can be fitted extremely well by a straight line (Fig.6), yielding:

$$T = 150 \text{ h}$$
$$b = 0.64$$
$$a = \bar{t}/T = 1.4 \quad \bar{t} = \underline{210 \text{ h} = \text{MTBF}}$$

3.2.2 Subsystems RF, LF and Container

The RF and LF subsystems are evaluated in the same way as the subsystem processor. During the period observed, n = 15 (RF) resp. n = 9 (LF) failure-free times accrued to the above. The results are compiled in section 3.3.

The subsystem container cannot be evaluated, since only two failure reports are available for this system.

3.3 Summary and Assessment of ATG1

Evaluation of ATG1 supplied the following values:

	MTBF/h	b	β_o resp. β_u
Processor	210	0.64	0.83
RF	351	0.58	0.79
LF	1164	0.45	0.66
Container	-	-	
ATG1	80	0.57	0.68

The upper confidence limit for the shape parameter β with a probability of 95%, determined according to section 2.3, is yielded for the overall ATG1 as

$$\beta_o = b \cdot F_b = 0.57 \cdot 1.2 = 0.68 < 1.$$

The hypothesis that the shape parameter of the overall ATG1 reaches at least the value 1 is rejected at the level $\alpha = 5\%$.

The shape parameter thus determined indicates that a many hidden defects are still present, which is not surprising in view of the first version of such a complex system. The shape parameters of the three subsystems under consideration lie in the same order of magnitude, so that it can be assumed that the subsystems are not fully mature.

The MTBF value of ATG1 amounts to 80 h. It depends mainly on the MTBF values of the subsystems processor (MTBF = 210 h) and RF (MTBF = 351 h).

4. EVALUATION OF THE ATG2 FAILURE REPORTS

The ATG2 has been employed since January 1978 as a test system in the production of the guided missile KORMORAN. The evaluation of the failure reports is effected here separately for the two periods up to December 1977 (integration) and as of January 1978 (operation). This permits making further comparisons.

4.1 Evaluation for the Integration Period

During the integration of ATG2 over 16 months, n = 40 failure-free times were noted.

Further evaluation is accomplished in the same way as in section 3, so that details can be dispensed with here. Merely one peculiarity should be pointed out which occurred in the case of the subsystem container:

The 5 points (t_i, F_i) for the subsystem container cannot be represented in the life distribution by a straight line, if the failure-free time is assumed as $t_o = 0$, as previously (x in Fig.7).

Serial No. i	1	2	3	4	5
Failure-free time t_i/h	94	132	138	182	435
Converted Failure-free time (t_i-t_o)/h	16	54	60	104	357
F_i/%	16.7	33.3	50.0	66.7	83.3

Since the fitted curve is a convex function towards the top, it can be linearized through suitable selection of $t_o > 0$. There is no exact mathematical rule which applies for the determination of t_o, a suitable value must be found by trial and error or by applying an empirically determined approximation method, as described, for instance, in (Verband der Automobilindustrie, 1976): The balancing curve is divided in the direction of the ordinates into two equally long sections and the lives t_1, t_2, and t_3 read off at the intersection points of the curve, as shown in Fig.7. The failure-free time t_o is calculated according to

$$t_o = t_2 - \frac{(t_3-t_2) \cdot (t_2-t_1)}{(t_3-t_2) - (t_2-t_1)}$$

In this case, t_o was calculated as t_o = 78 h. The 5 points can now be fitted far better by a straight line (Fig. 7). The graphic evaluation permits calculating the MTBF value as $\bar{t}$ = MTBF = a T + t_o. The shape parameter will not be converted.

The following table indicates the evaluation results for ATG2 (integration):

	MTBF/h	n	b	β_o resp. β_u	
Processor	255	15	0.55	0.75	
RF	215	11	1.02		0.71
LF	122	7	1.21		0.79
Container	233	5	0.75	1.23	
ATG2(integr.)	64	41	0.66	0.81	

4.2 Evaluation for the Operational Period

Over the course of the operational period observed (January to November 1978), a total of n = 26 failure-free times were ascertained for ATG2.

The following table provides the evaluation with t_o=0:

	MTBF/h	n	b	β_o resp. β_u
Processor and peripheral equipment	392	6	0.85	1.34
RF peripheral equipm.	410	7	0.68	1.05
LF peripheral equipm.	230	11	0.55	0.79
Container	*	*	*	
ATG2(operation)	83	25	0.79	1.01

* Only two failures occurred in the case of the container so that evaluation does not yet appear appropriate. These failures are naturally taken into consideration within the overall context.

4.3 Assessment of ATG2

4.3.1 Assessment for the Integration Period

The upper confidence limit for the shape parameter ß with a probability of 95% is yielded according to section 2.3 for ATG2 during the integration period as

$$\beta_0 = 0.66 \cdot 1.22 = 0.81 < 1.$$

The hypothesis that the shape parameter of the overall ATG2 reaches at least the value 1 is rejected at the level α = 5%.

The determined shape parameter b = 0.66 indicates that hidden defects remain. Analysis of the shape parameters of the subsystems shows that the hidden defects occur in the case of the subsystem processor and peripheral equipment, having a shape parameter significantly less than 1. The remaining 3 shape parameters do not differ significantly from 1, so that random failures may be assumed.

The MTBF values relating to the subsystems processor, RF and container all lie at the same order of magnitude. The average failure-free time accruing to the subsystem LF, however, is only half as long and thus particularly affects the MTBF value of ATG2 for the period observed.

4.3.2 Assessment for the Operational Period

The hypothesis that the shape parameter of all the failure-free times in the case of ATG2 reaches at least the value 1 is tested with the aid of the upper confidence limit $ß_0$ for the shape parameter ß.

The hypothesis is not rejected at level α = 5%, since $ß_0 > 1$:

$$\beta_0 = 0.79 \cdot 1.28 = 1.01$$

The determined shape parameter b = 0.79 thus does not contradict the assumption that the ATG2 failures are random after delivery to the user.

At the subsystem level, the conditions for processor and RF peripheral equipment are the same. The shape parameter for the LF peripheral equipment, however, is significantly smaller than 1, that is, early failures would still be detected on normal operation.

The MTBF values of the subsystems processor and RF peripheral equipment are approximately twice as high as the MTBF of the LF peripheral equipment. The reliability of ATG2 for the operational period under observation thus depends to a particularly high degree on the subsystem LF peripheral equipment.

The subsystem container could not be considered separately, due to the small number of failure reports.

4.4 Time Curve of Failures

Over the entire period observed (August 1976 to November 1978), 22 failures occurred on the subsystem processor and peripheral equipment of ATG2, the points in time of which (x) are plotted in Fig. 8. The operating time since the occurrence of the first failure and some pertinent dates are plotted on the abscissa, for purposes of better representation the sum of the failure reports since August 1976 is chosen as the ordinate.

It is noticeable that failures occurred at large intervals up to June 1977 and as of October 1977, and, in contrast, in rapid succession between July 1977 and September 1977. It can be concluded from this that the warmer season of the year might have caused the rash of failures. Since the ATG features its own air-conditioning plant, it must be investigated whether the plant is efficient enough, or whether it broke down during this period of time.

Fig.8 likewise shows the points in time of the 8 failures on the subsystem container, to which the air-conditioning plant belongs (o). The first 7 of these 8 failures concern the air-conditioning plant, due to which the temperature in the ATG rose considerably. The cumulation of the processor failures over this period can thus be considered rather definitely as a consequence of the air-conditioning plant breakdowns.

5. EVALUATION OF THE ATG3 FAILURE REPORTS

12 failures occurred on ATG3 over the 10 months under observation during the integration period. Due to the low number, evaluation by subsystems is impossible. Evaluation was conducted according to section 3.2.1 for the overall system, this yielding an MTBF = 122 h and a shape parameter b = 1.52.

The lower confidence limit for the shape parameter with a probability of 95% is yielded for ATG3 at n = 11 failure-free periods as

$$\beta_u = b/F_b = 1{,}52/1{,}43 = 1{,}06 > 1.$$

The hypothesis that the shape parameter of the overall ATG3 is compatible with the value 1 is rejected at level α = 5%. The shape parameter so determined indicates that the failures are not due to hidden defects. A certain degree of series production maturity has thus been attained.

6. CONCLUSION

The values determined for the three test systems are summarized for purposes of comparison.

Table of MTBF-Values [h]:

Test System	ATG1	ATG2		ATG3
Mode of Operation	Operation	Integration	Operation	Integration
Processor	210	255	392	-
RF	351	215	410	-
LF	(1164)	122	230	-
Container	-	233	-	-
Overall MTBF	80	64	83	122

The first obvious fact is the extremely high MTBF value for the subsystem LF of ATG1. This LF peripheral equipment is a prototype on which many modifications were effected which are not shown in failure reports. The real MTBF values for this subsystem and for the overall ATG1 are thus less than the values indicated.

Comparison with ATG2/Operation shows that the remaining values for ATG2 are better than for ATG1. Due to the many modifications introduced for ATG2, the latter can also be considered as a virtually new development. The values determined lead to the assumption that the experience gained in the course of developing ATG1 stood ATG2 in good stead.

Comparing integration-operation in the case of ATG2 shows better values in the operational case. This leads to the conclusion that the test unit was well checked out during integration and weak points eliminated. Due to the weak confidence level of the ATG3 value, comparing ATG2(integration)-ATG3 is permitted only with reservations. However, results tend to show that ATG3 features greater reliability in the integration phase than ATG2. This can again be explained by the increasing experience.

Let us at this point compare the ATG with test equipment for the anti-tank weapon system HOT.(Lienau, 1977) assessed the MTBF according to MIL-HDBK 217 B (Department of Defence, 1974) for the above equipment. Utilizing the data for similar assemblies and extrapolating in view of the greater complexity of ATG3 yield an MTBF value of 400 h for ATG3. This value is of the same order of magnitude as that determined in sec.5. For this reason, the extremely complex assessment for the HOT test equipment, as performed in accordance with MIL-HDBK 217B, does not appear to be necessary for the ATG.

Table of Shape Factors:

Test System	ATG1	ATG2		ATG3
Mode of Operation	Operation	Integration	Operation	Integration
Processor	0.64 (<1)	0.55 (<1)	0.85	-
RF	0.58 (<1)	1.02	0.68	-
LF	0.45 (<1)	1.21	0.55	-
Container	-	0.75	-	
Overall shape factor	0.57 (<1)	0.66 (<1)	0.79	1.52(>1)

The shape factors show similar conditions as the MTBF values:

Whilst all shape factors for ATG1 are significantly less than 1, thus indicating hidden defects, all values except for b_{LF} are compatible with 1 for ATG2/operation. The maturity has increased, the failures are in general random ones.

At system and subsystem level, no uniform tendency is seen between integration and operation of ATG2. A clear reliability growth, however, appears to be the case between ATG2 (integration) and ATG3.

The following findings were gained in the course of this investigation:

1. The theoretically derived method for life investigations with the aid of the Weibull distribution is applicable in practice.
2. Applying the method yields results compatible with the anticipated reliability and availability.
3. Applying the method leads to the detection of weak points
4. Knowledge of the weak points directly influences the development with the objective of lowering the failure rate.
5. Presentation of the values ascertained in practice can increase the awareness of the developer of problems relating to reliability.
6. The simplest representations (see section 4.4) can sometimes reveal interconnections.
7. The reliability of electronic systems is increased by lowering the ambient temperature.

All these points contribute to increasing the reliability of a system and to obtaining empirical, explicit and informative values.

Bibliography

Department of Defence, 1974, "Reliability Prediction of Electronic Equipment", Military Handbook 217 B;
Deutsche Gesellschaft für Qualität, 1975, "Das Lebensdauernetz", DGQ 25;
Graf/Henning/Wilrich, 1974, "Statistische Methoden bei textilen Untersuchungen", Springer-Verlag;
Lienau, H., 1977, "Zuverlässigkeitsabschätzung HOT-Prüfgerät", MBB intern;
Molter, H.-H., 1978, "Zuverlässigkeitsuntersuchung ATG-KORMORAN", MBB intern;
Verband der Automobilindustrie, 1976, "Zuverlässigkeitssicherung bei Automobilherstellern und Lieferanten", VDA.

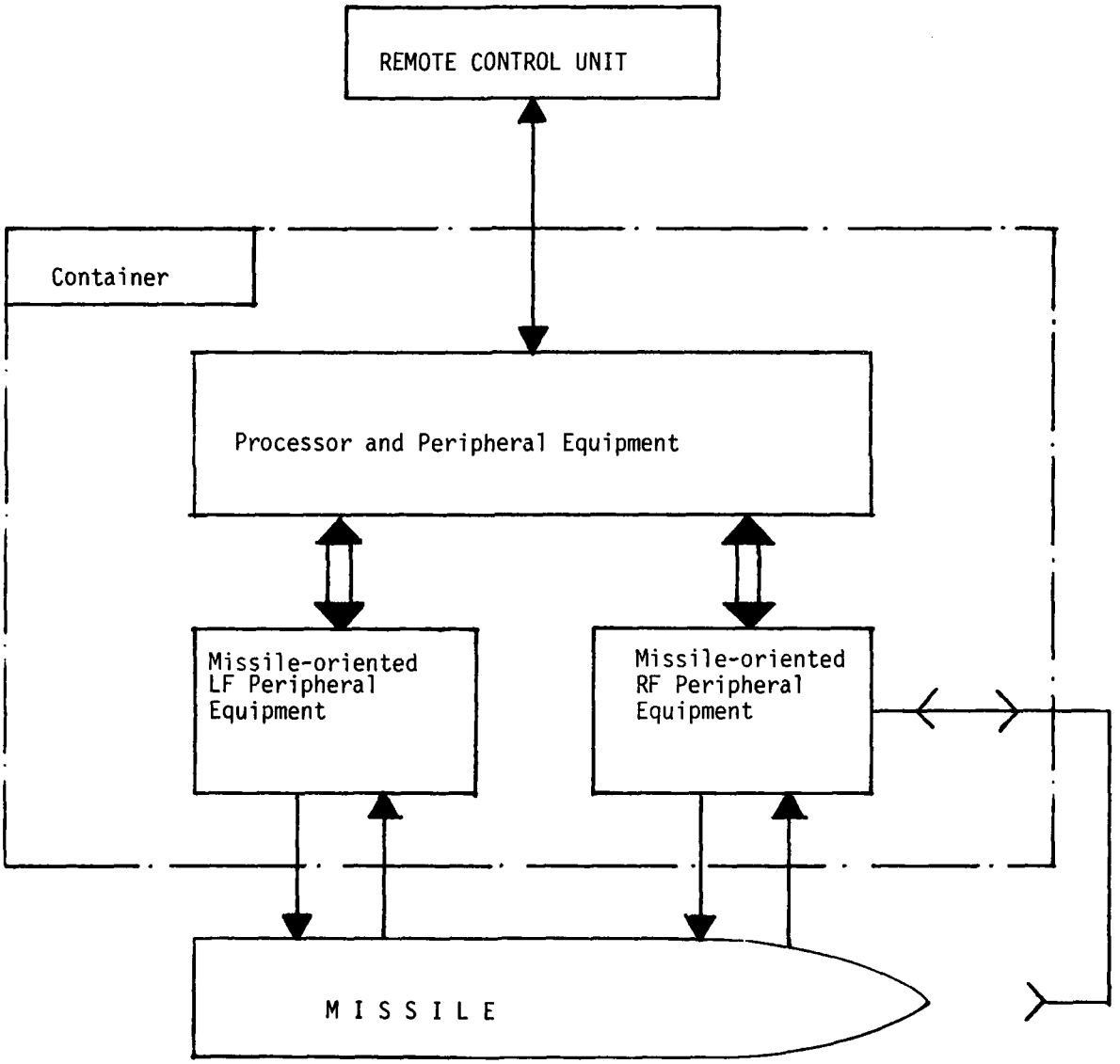


Fig.1 Basic ATG configuration

Messerschmitt-Bölkow-Blohm GmbH
Unternehmensbereich Apparate

STÖRMELDUNG – STÖRBEHEBUNG
FICHE D'INTERVENTION
FAILURE REPORT

Nummer / Numéro / Number 1
S M - -
Projekt / Matériel / Project 2

Verteiler / Diffusion / Distribution

Hersteller / Constructeur / Manufacturer 6
Benennung / Désignation / Designation 3
Zuordnung / Affectation / Ref. design. 7
Sachnummer / Référence Fabricant / Part Number 4
Eingebaut / Monté / Integrated ☐ Ja / Oui / Yes ☐ Nein / Non / No
In Baueinheit / Serien-Nr. / Sur Ensemble / N° de Série / In unit / Serial N° 8
Lfd. Nr. in LLK / N° dans Fiche Suiv. / Running N° in Log sheet 9
Los- / Serien-Nr. / N° de Lot / Série / Lot / Serial-N° 5

Störung festgestellt bei / Incident constaté en / Failure detected during 10
☐ Entw.-Versuch / Essai / Engin. Test ☐ Abnahme-, Güteprüfung / Recette / Acceptance Test ☐ Integration ☐ EMV-Prüfung / Ess. EMC / EMC-Test
☐ Kontrolle / Contrôle / Inspection ☐ Qualifikationsprüfung / Ess. de Qualif. / Qualif. Test ☐ Montage ☐ Sonstige / Autres / Others:

Umwelt / Environnement / Environment 11
☐ Bei Raumtemperatur / A l'Ambiance / Ambient ☐ Während / Pendant / During ☐ Nach / Après / After

Beschreibung der Störung / Description de l'Incident / Failure description
Fehlerschlüssel / Code du Défaut / Failure code 12

Vermutete Ursache / Cause probable / Possible cause of failure 13
Datum / Date Name / Nom Abt. / Serv. / Dept.

Sofortmaßnahme / Action immédiate / Quick action 14
Datum / Date Name / Nom Abt. / Serv. / Dept.

☐ Keine weiteren Maßnahmen notwendig / Pas d'autres actions necessaires / No further actions required 15a

☐ Untersuchung notwendig / Expertise demandée / Analysis required 15b
Durchzuführen von / A exécuter par / To be executed by:

☐ Behebung notwendig / *Réparation demandée / Corrective action required* 15c
Durchzuführen von / A exécuter par / To be executed by:

Entscheidung (weitere Behebung der Störung) / Décision / Decision 15
Datum / Date Name / Nom Abt. / Serv. / Dept.

Untersuchungsergebnis / Résultat de l'Expertise / Analysis results 16
Datum / Date Name / Nom Abt. / Serv. / Dept.

Durchgeführte Behebungsmaßnahmen / Intervention / Corrective action 17
Datum / Date Name / Nom Abt. / Serv. / Dept.

Die Störung betrifft / L'Incident concerne / This failure concerns 18a
☐ Nur diese Baueinheit / Ce Matériel seul / This item only
☐ Alle Baueinh. dieses Typs / Tous Mat. de ce Type / All items of this type
☐ Sonstige / Autres / Others:

☐ Bauabweichungsantrag / Demande de Dérogation / Deviation 18b
☐ *Unterlagen-Änderung / Modificatio. Dossier / Drawing change*

Weitere Verwendung / Utilisation / Further use 18c
☐ Unbeschränkt / Illimitée / Unrestricted
☐ Beschränkt / Limitée à / Restricted to
☐ Sonstige / Autres / Others:

Entscheidung (Verwendung, Folgemaßnahmen) / Décision / Decision 18
Datum / Date Name / Nom Abt. / Serv. / Dept.

6 2030/377

Fig.2 Failure report form

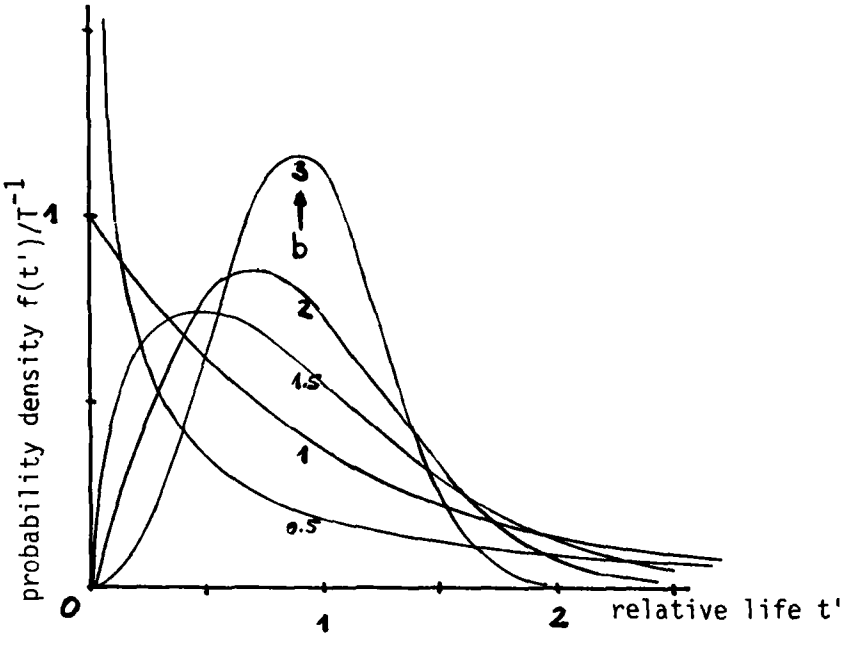


Fig.3 f(t') for various values of b, $t' = (t - to)/T$

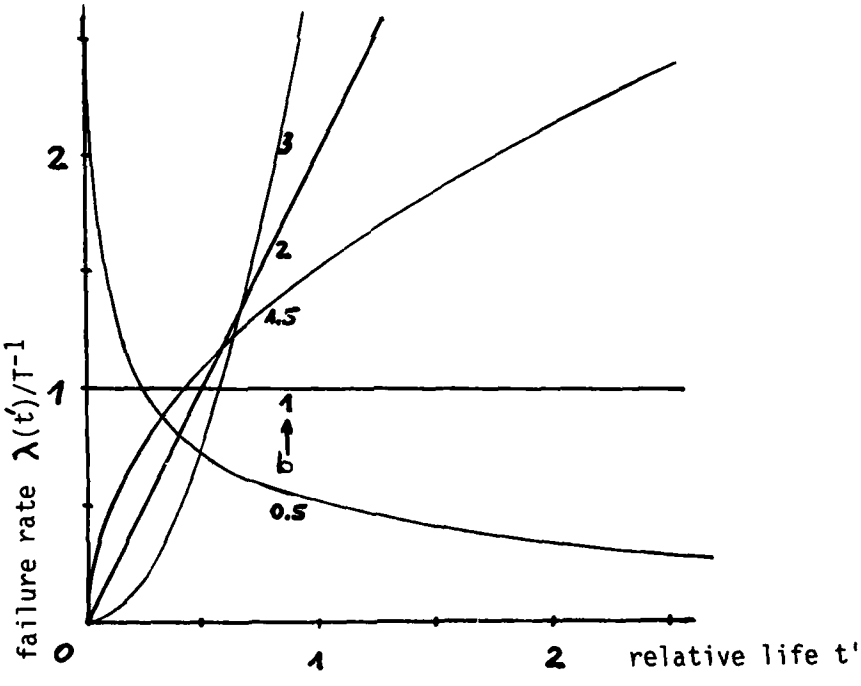


Fig.4 λ(t') for various values of b, $t' = (t - to)/T$

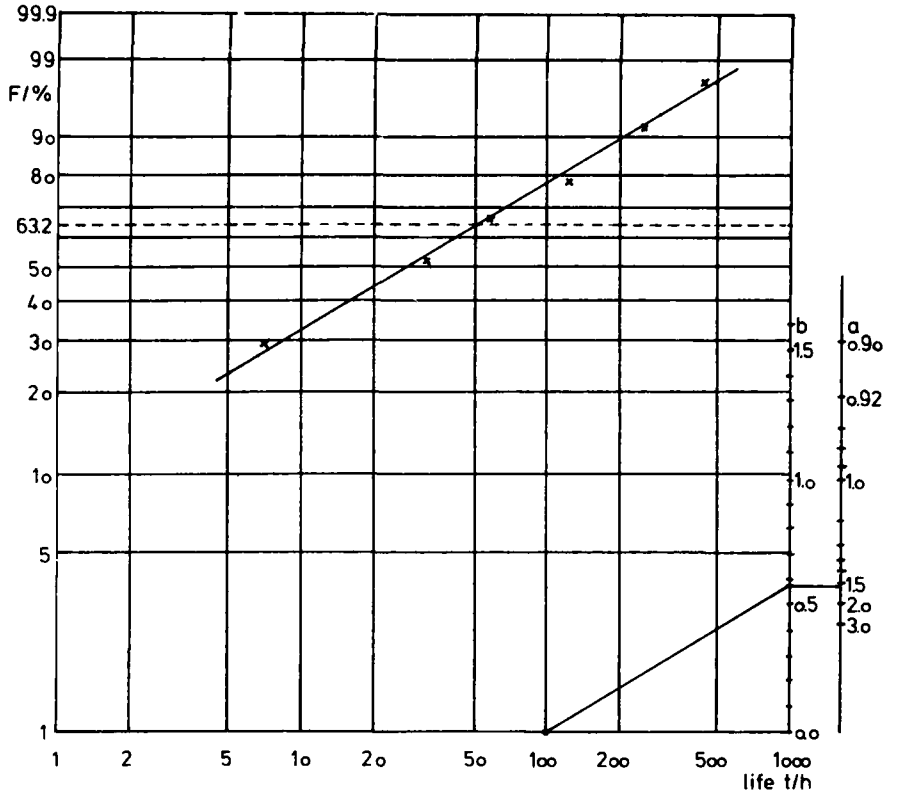


Fig.5 The Weibull-distribution, ATG1

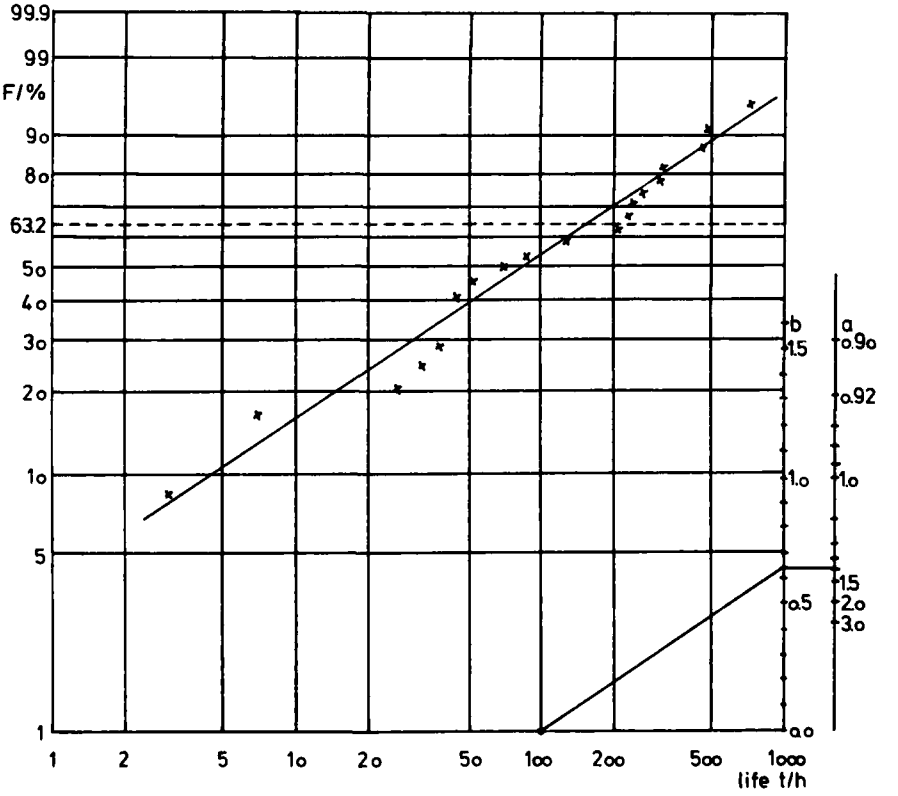


Fig.6 The Weibull-distribution, processor ATG1

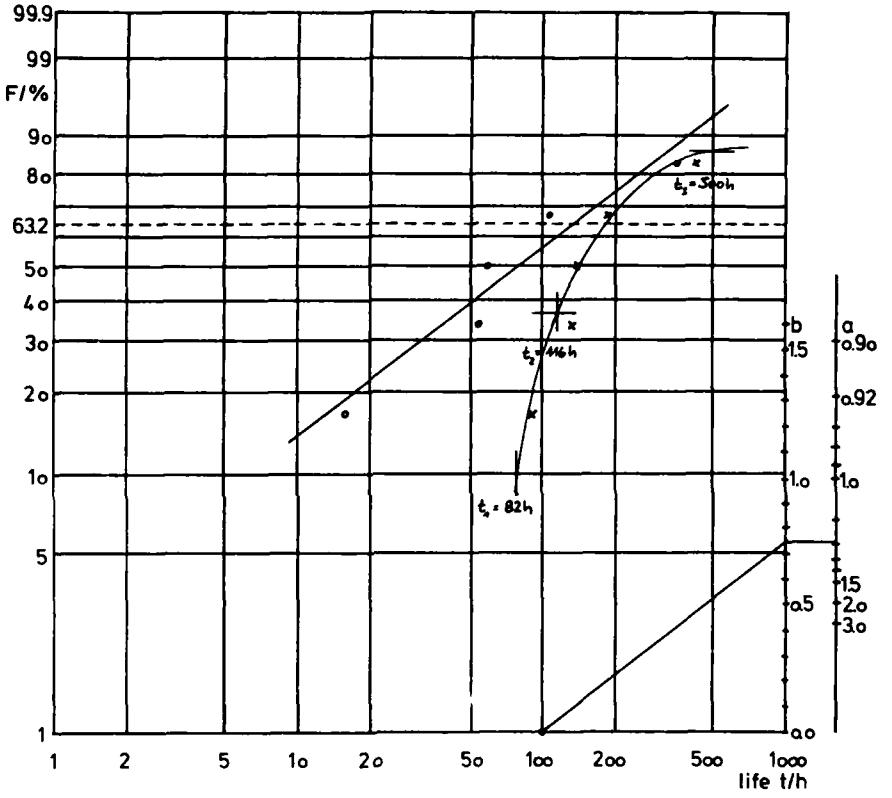


Fig.7 The Weibull-distribution, container ATG2

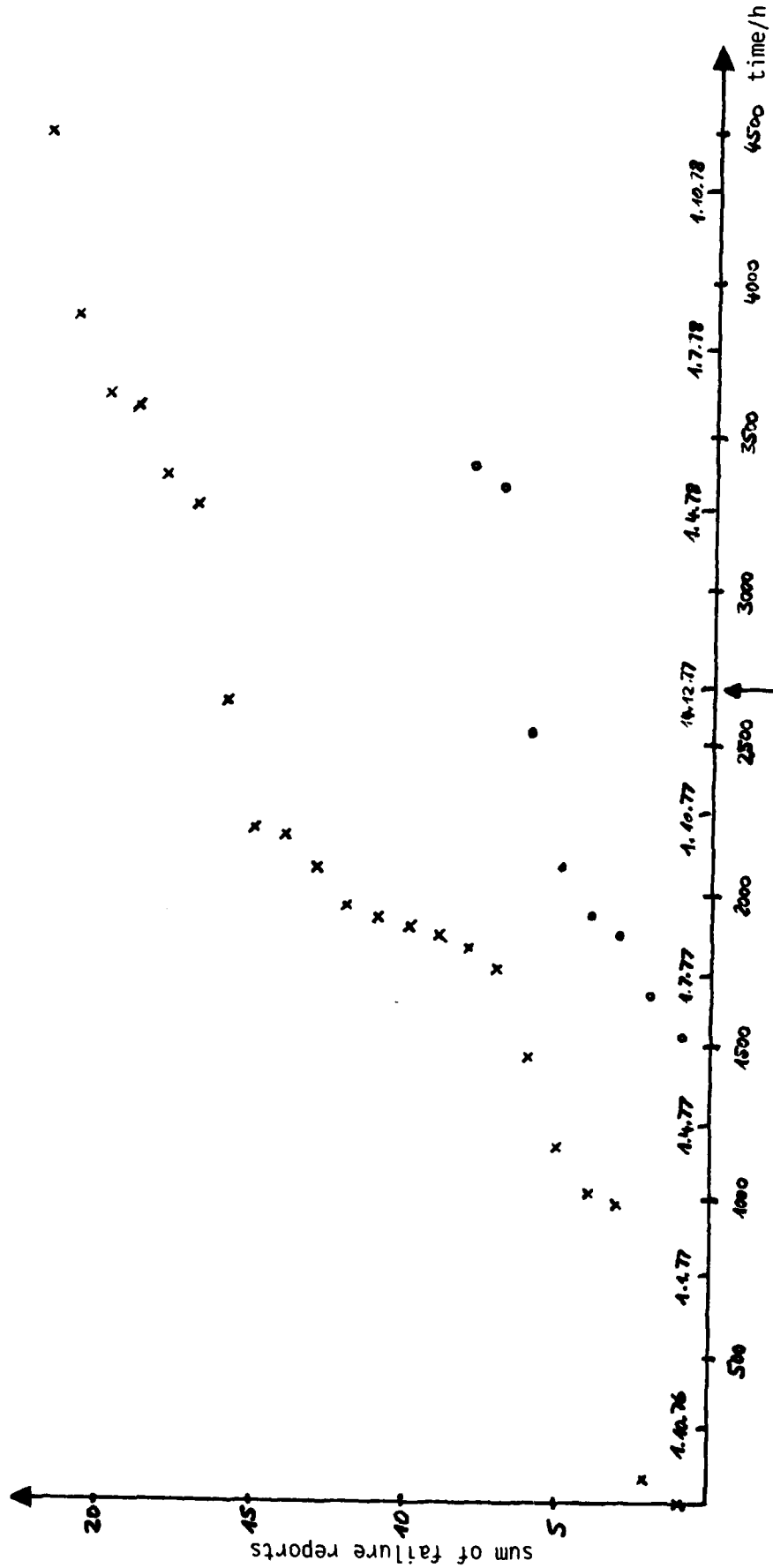


Fig.8 Time curve of failures, processor (x) and container (o), ATG2

DISCUSSION

P.D.T.O'Connor, UK

I would like to say how this paper shows how useful the Weibull method can be. I think it contrasts with the previous paper which was trying to analyse similar failure data but not using this method.

I don't believe that the Weibull method should be used for plotting more than one failure mode on one analysis otherwise all the failures within one ATG, the central limit theorem, will just mean that your shape parameter will gradually convert towards a value of 1. You should really consider one type of failure or at the very most a small number of common types of failure, e.g. integrated circuit failures.

I think the ATG should be seen as a black box.

Author's Reply

I think that it is correct to make such an analysis.

APPLICATION OF THE LOGNORMAL DISTRIBUTION TO CORRECTIVE MAINTENANCE DOWNTIMES

Professor Melvin B. Kline
Naval Postgraduate School
Monterey, California

LCDR Ronny Almog
Israeli Navy

SUMMARY

The effectiveness of a system depends not only on its ability to meet its specified performance requirements, but also on its ability to perform when needed, for the duration of its assigned missions, and for its operational lifetime. The technical disciplines concerned with these time-related system characteristics are reliability, maintainability, and logistics. These are related mathematically by the concepts of availability, dependability, and operational readiness.

The usual mathematical formulation of availability assumes an exponential distribution for failure and repair times. It has been shown empirically that such an assumption is valid only for a limited class of components and systems with respect to reliability and almost never for maintainability. In fact, there appears to be overwhelming evidence that the lognormal distribution is the "best" descriptor for corrective maintenance repair times. Military standards for prediction and demonstration of maintainability generally are based on the assumption of the lognormal distribution.

The objectives of this research were (1) to verify that the lognormal distribution is a suitable descriptor for corrective maintenance repair times, (2) to estimate the percentage error caused in assuming an exponential distribution for availability and maintainability calculations when in fact the distribution is lognormal, (3) to test the lognormal and exponential distributions against mechanical and other non-electronic systems since the current data base is primarily on electronic systems, (4) to test these distributions for systems and equipments in which new technologies in microcircuitry and computation are used to increase reliability and decrease diagnostic time, and (5) to determine expected ranges of the principal distribution parameters for different classes of equipment.

Approximately 20 sets of existing maintainability demonstration and field data were analyzed using probability plots and Goodness-of-Fit tests to determine the appropriate distribution and distribution parameters.

The preliminary results of the research are reported in this paper.

1. INTRODUCTION

In an attempt to improve *operational availability*, the U.S. Department of Defense has put increasing emphasis on the reliability, maintainability, and logistic supportability of its systems and equipments. In the early 1970's, the Chief of Naval Material stressed reliability and maintainability (R&M) as systems engineering disciplines of equal importance as performance in design trade-offs. He established an office reporting directly to him to review all system acquisitions for R&M. The current F/A-18 aircraft program has a significant design effort in reliability, maintainability, and logistic support.

For reasons not delineated in this paper, *inherent availability* (sometimes called *intrinsic availability*) rather than operational availability is the system effectiveness measure most often specified. The definition of inherent availability includes only corrective maintenance active repair times, omitting preventive maintenance times and administrative and logistic supply delay times. In its steady-state form, inherent availability can be expressed by the equation:

$$A_I = \frac{MTBF}{MTBF + MTTR} \qquad (1)$$

where

MTBF = Mean Time Between Failures
MTTR = Mean Time to Repair

Although the availability equation is easily derivable from calculus using assumptions of an exponential distribution for failure and repair times, the steady-state term given

above can be applied without making any assumptions on the distributions. Most theoretical papers and many applied papers are written using exponential distribution assumptions for both failure and repair. While this is sometimes correct for reliability, it is not valid for maintainability since a repair time distribution must start with a value of zero and not with its maximum value at time $t = 0$. No repairs can be made in zero time.

The logarithmic normal (lognormal) distribution has been shown empirically to be a better descriptor for repair times. It is characterized by a value of zero at $t = 0$, rises to its maximum value in a reasonably short time, and gradually decreases towards zero as repair time increases. This is exactly what is desired for corrective maintenance--to minimize downtime. For the lognormal distribution, the *logarithm of the random variate* is normally distributed. It is characterized by the probability density function

$$f(x) = \frac{1}{\sigma x\sqrt{2\pi}} e^{\frac{-(\ln x - \mu)^2}{2\sigma^2}}, \quad x > 0 \qquad (2)$$

where $\mu = \overline{\ln x}$

and $\sigma^2 = \text{var}(\ln x)$.

The lognormal distribution has properties which make it convenient to use. A more detailed description of the lognormal distribution is given in References 1 and 2. The values of the significant parameters are shown in Figure 1.

2. SYSTEMS AND DATA ANALYZED

Most of the repair times used in the analysis come from maintainability demonstration reports of essentially electronic systems/equipment. None come from field data. Although we desired to analyze data from mechanical systems also, we were unable to obtain such data. The systems/equipments analyzed and their sources are given in Table I. They range from 1950's-1960's vintage systems representative of primarily analog, vacuum tube, discrete component design to some 1970's vintage systems using digital, transistor/microelectronics designs with extensive built-in test and modular replacement maintainability design features.

In some cases, the source reports include a description of the tests used during the maintainability demonstration. It was necessary to carefully screen the reports in analyzing specific repair times that varied significantly from expected results in order to remove anomalies which biased the data. The data sources include the elements of active repair time-diagnostic time (localization and isolation), remove and replace time, and verification and checkout time.

TABLE I
Systems/Equipments Analyzed

No.	Description	Source Reference
1	AN/TRC-87 Communications Transceiver	5
2	Quick Reaction Capability Radar	5
3	AN/GSA-51 Back Up Interceptor Control System	5
4	AN/FPS-80 Tracking Radar	5
5	AN/TPS-39(V) Radar Surveillance System	5
6	AM/3949-GR Radio Frequency Amplifier	5
7	AN/ARC-164(V) Radio Set	6
8	AN/ASN-131 Airborne Navigation System--Inertial Measurement Unit (IMU) Interface Electronics Unit (IEU)	7
9	HARPOON Ship Command Launch Control Set	8
10	Defense Communication System Satellite Control Facility Interface System (DSIS)	9
11	USASATCOMA Communication Subsystem (Contingency Configuration)	10
12	USASATCOMA Communication Subsystem (Nodal Configuration)	11
13	Continental Air Defense Command Ground Data System (User Display Segment)	12
14	Strategic Air Command Ground Data System (User Display Segment)	13
15	SAMSO 46 FOOT TT&C Antenna	14
16	NADC Digital Television Projection Unit	15
17	USASATCOMA HT/MT Terminal	16
18	National Military Command System Ground Data System (User Display Segment)	17
19	US Army Electronics Command AUTODIN Memory/Memory Control Equipment	18

3. METHODS FOR DATA ANALYSIS

Two different techniques were used to analyze the data--(1) plotting on probability paper and (2) use of statistical goodness-of-fit tests--in order to verify the assumed distribution model.

3.1 Probability Plotting

Plotting the data points on probability paper is quite simple and does not require complicated calculations or the use of statistical tables. According to Hahn and Shapiro (Reference 3), "Probability plotting is a subjective method in that the determination of whether or not the data contradict the assumed model is based on a visual examination, rather than a statistical calculation." The only calculation needed is that of the cumulative frequencies (expected values) of the ordered sample observation. As stated by Aitchison and Brown (Reference 1):

> It is usually worth while to submit data to some kind of graphical scrutiny as a preliminary to any more detailed analysis. By so doing we may eventually save much time and labour and even have suggested what form the more elaborate analysis should take; moreover we may obtain, for those measures in which we are interested, provisional estimates which will both serve our purpose until more accurate values may be obtained and also provide a check on subsequent calculations. For the lognormal distribution we are fortunate in having a quick and, with experience, fairly accurate graphical method of analysis; this method is facilitated by the use of a special type of graph paper--logarithmic probability paper.

For the lognormal distribution plots, logarithmic (normal) probability paper was used. For the exponential distribution plots, chi-square probability paper (two degrees of freedom), which represents the exponential distribution, was used.

3.2 Statistical Tests for Assumed Distributions

There are a number of statistical tests available to determine the validity (or invalidity) of an assumption of a particular distribution from a sample of observed data. One that is often used is the chi-squared goodness-of-fit test (Reference 3). Another test, due to Shapiro and Wilk (Reference 4), called the W-test, has been shown by them to be effective for testing the assumption of the normal and lognormal distributions. They have also developed a test for the exponential distribution called the WE test. Hahn and Shapiro (Reference 3) gives details of all three tests and their application. (Because published tables used in the W and WE tests are limited to sample sizes up to 50 and 35, respectively, these tests, when needed, were limited to those cases that did not exceed such sample sizes.)

In assessing the results of the goodness-of-fit tests, one rejects the assumed distribution if the probability of obtaining the test statistic is below some arbitrarily selected criterion, often 5 or 10%. If the probability of obtaining the test statistic is above the reject criterion, one can only say that the data provides no evidence that the assumed distribution is inadequate, and thus, by presumption, is adequate for our purposes.

The chi-squared test has several disadvantages. The number of equiprobable cells used is arbitrary (the number of data elements per cell should be at least 5). Thus for small samples, there is correspondingly a small number of cells and fewer degrees of freedom. In addition, the information concerning sign and trend of discrepancies is ignored. These factors might explain the indication (Reference 4) that the power of the W-test is higher than that of the chi-squared test. For this study, our reject criterion was 10% for the chi-squared test and 5% for the W-test.

3.4 Application Procedure Used in This Study

The following procedure was used for analyzing the data:

a. The data was plotted on lognormal probability paper and the "best fit" line drawn.

b. A chi-squared goodness-of-fit test was performed for both the lognormal and exponential distributions, using a computer program prepared for the analysis.

c. A W-test was used to test the distribution assumption when the sample size permitted.

d. In those few cases where the data analysis indicated close results for both distribution assumptions, or where the exponential distribution appeared to be appropriate, a plot of the data on chi-square (two degrees of freedom) probability paper was made.

The computer program prepared for the analysis makes use of appropriate routines from the International Mathematical and Statistical Library (IMSL) for the chi-squared test. The program calculates from the sample data such parameters as the mean, variance, and percentiles (in this case the 50th, 90th, and 95th) for the exponential and lognormal distributions, which are defined in the program. It also computes the percentage difference for each parameter for comparison purposes, and it is used to compute and print out the approximate frequencies (expected value of the ordered observations) for plotting purposes.

The major output from the program for each set of data is a summary of results as shown in Table II. A sample lognormal plot is shown in Figure 2.

TABLE II
Summary of Results for: AN/GSA - 51 (BUIC) System (90 Repair Times)

Sample Size = 90 Sample Mean = 20.431 Standard Dev = 17.068

	EXPONENTIAL	LOGNORMAL	ERROR
PARAM1	λ = 0.049	μ = 2.730	
PARAM2		σ^2 = 0.573	
MEAN	20.431	20.419	0.06 %
50-TH PERCNT	14.162	15.334	7.64 %
90-TH PERCNT	47.044	40.459	16.28 %
95-TH PERCNT	61.206	53.251	14.94 %
CHI-SQUARE [CS]	25.600	7.600	EQUIPROBABLE CELLS
DEG OF FREED [ν]	16	15	18
$P[\chi^2_{1-\alpha}(\nu) \geq CS]$	0.5992E-01	0.9388E 00	

WHERE α = LEVEL OF SIGNIFICANCE

4. ANALYSIS RESULTS

Table III summarizes the results of the statistical test analysis. These results are based on the output of the computer program. The results from the probability plotting and calculated percentiles are not presented here.

4.1 Results of Tests for Exponential and Lognormal Distributions.

Most of the sets of data show that the lognormal distribution is an adequate descriptor for corrective maintenance repair time. From Table III, the assumption of the exponential distribution is rejected in 17 cases while the assumption of the lognormal distribution cannot be rejected in 15 cases. In 3 cases both assumptions are rejected and in one case both assumptions cannot be rejected. Only in one case is the lognormal distribution rejected and the exponential distribution cannot be rejected. The cases in which the lognormal distribution is not accepted are cases 5, 9, 11 and 17.

The following discussion refers to those cases which either do not satisfy the underlying assumption or are of special interest as their results are different from the others and/or point out some interesting issues.

4.1.1 Set No. 3 - AN/GSA-51

This case represents the successful cases in which the lognormal distribution appears to be a good fit. Figure 2 shows a plot of the data on lognormal probability paper which results in a straight line, except for small deviations which can be attributed to randomness in the lower level. The line goes from the origin as it should be expected theoretically. The histogram presented in the source paper (Reference 5) shows the shape of a lognormal distribution.

4.1.2 Set No. 4 - AN/FPS-80

This case was presented as an unsuccessful one in the source paper (Reference 5) due to inexperienced technicians and the need for adjustment factors to the repair times during the demonstration. That conclusion is based on the histogram which is presented in the paper. Indeed, one might reach such a (wrong) conclusion when relying solely upon a histogram which, by its nature (data combined into intervals resulting in loss of information) cannot be used to "test" an assumption on the distribution that fits the data.

All the tests (chi-squared, W, probability plotting) conducted for this case show that the lognormal distribution is a good assumption and with high levels of significance (0.71 and 0.92 for the W and the chi-squared tests, respectively).

The result of the chi-squared test for the exponential distribution shows that, had it been tested separately, one would fail to reject it with a level of significance of 0.15. The reason for this result may be that many of the repair times are concentrated at the lower level (around the 10 minute value). Therefore, in cases like this one, careful analysis must be made and more than a single test has to be performed in order to test the assumption.

TABLE III

Summary of Results

Set No.	Sample Size (n)	Goodness-of-fit Tests (level of significance)			Percentage Error (*) on the Mean (minutes)			Accept/Reject		NOTES
		Chi-squared		W				R[α(χ²)<0.1] R[α(W)<0.05]		
		EXP	LGN	LGN	M_{LGN}	M_{EXP}	E(%)	EXP	LGN	
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)=|(6)-(7)|/(6)	(9)	(10)	(11)
1	59	0.04	0.23	--	18.3	18.7	2.0	R	A	
2	20	0.006	0.27	0.06	25.1	25.4	0.9	R	A	
3	90	0.06	0.94	--	20.42	20.43	0.06	R	A	
4	45	0.15	0.92	0.71	80.9	78.1	3.5	A	A	
5	75	1×10^{-24}	1×10^{-8}	--	11.3	11.1	1.4	R	R	(1)
6	38	4×10^{-6}	0.32	0.07	29.0	28.5	1.7	R	A	(2)
7	50	2×10^{-6}	0.13	0.06	22.6	22.4	0.9	R	A	
8a	21	0.005	0.094	0.25	20.3	20.2	0.6	R	A(W)	
8b	21	0.04	0.71	0.87	71.7	70.7	1.5	R	A	
9	44	7×10^{-4}	0.053	<0.01	53.1	56.2	5.9	R	R	
10a	25	0.006	0.11	0.05	16.4	17.4	6.2	R	A	(3)
10b	25	0.11	0.11	0.42	20.9	20.8	0.2	A	A	(4), (5)
11	50	6×10^{-8}	0.004	<0.01	9.3	10.0	7.0	R	R	(6)
12	50	3×10^{-4}	0.41	0.06	11.2	11.5	2.8	R	A	
13a	50	0.002	0.15	0.58	48.3	48.2	0.24	R	A	
13b		1×10^{-8}	0.016	0.07	72.1	72.3	0.35	R	A(W)	
14a	37	0.096	0.45	0.03	54.7	50.4	7.8	R	A,R(W)	
14b		0.063	0.35	0.26	155.7	154.0	1.1	R	A	
15	50	0.014	0.17	0.85	54.3	52.0	4.2	R	A	
16	22	3×10^{-4}	0.34	0.43	19.3	19.0	1.5	R	A	
17	50	0.395	0.006	<0.01	19.9	17.0	--	A	R	(7)
18a	39	0.086	0.047	0.85	42.9	41.6	3.0	R	A(W)	
18b		0.05	0.15	0.85	44.1	43.3	2.0	R	A	
19	33	0.03	0.36	0.98	32.5	32.4	0.4	R	A	

(*) The lognormal mean is $M_{LGN} = e^{\mu + \frac{1}{2}\sigma^2}$ where μ and σ^2 are defined in equation (2). The exponential mean is $M_{EXP} = \frac{1}{\lambda}$ where $\hat{\lambda} = \frac{1}{\bar{x}}$. The percentage error when using the sample mean (assuming an exponential distribution) instead of the logarithmic mean is $\frac{|M_{LGN} - M_{EXP}|}{M_{LGN}} \times 100$.

NOTES:

(1) A chi-squared test for normality resulted with probability of less than 0.005.

(2) The original sample size was 57, which included 19 actions for the same fault (replacement of transmitting tube).

(3) A WE test resulted with rejection (less than 0.05).

(4) A WE test resulted with acceptance (more than 0.10).

(5) A histogram of 5 minute intervals shows a rough lognormal distribution.

(6) The assumption made in the demonstration report is that the repair time is log-normally distributed.

(7) The percentage error is not significant as a result of the validity of the exponential distribution in this case.

4.1.3 Set No. 5 - AN/TPS-39(V)

This case was presented as a definite violation of the lognormal characteristics. The reasons, as given in the paper (Reference 5), are attributed to the size of the equipment. It is suggested there that the normal distribution should be considered as an adequate descriptor because the repair times for small equipment are short and have small variations around a mean value. The histogram in the paper indicates "almost" a normal distribution. However, a chi-squared goodness-of-fit test for normality rejects this assumption (with a significance level of 0.005), as well as did the tests for the lognormal and exponential distributions. The lognormal probability plot shows a curve with clustered data points to which a straight line cannot be fitted.

4.1.4 Set No. 6 - AM/3949-GR

In this case, the histogram including all 57 repair times was bi-modal (Reference 5) and, therefore, does not fit either distribution assumption. One third of the repair times were for a single fault, replacement of the transmitting tube. Filtering out these 19 data points resulted in a histogram in the paper which appears lognormal. Indeed, our tests show this to be a valid assumption.

4.1.5 Set No. 8 - AN/ASN-131

This case points out an interesting issue concerning corrective maintenance repair times. By differentiating between organization (set 8a) and intermediate (set 8b) repair levels, it appears that intermediate level repair times follow the lognormal distribution better than the organizational level repair times. This might be due to randomness in this specific case. (The W-test and lognormal plot, for the organizational level, do not indicate that the assumption should be rejected.)

Some conclusions may be derived from this case: (1) that the sample size is too small for a chi-squared test, but is enough for a W-test, (2) that the W-test is less sensitive to deviations around the mean value, (3) that a single test may not be sufficient, and (4) that the lognormal distribution is an adequate model for both organizational and intermediate repair levels.

Figure 3 shows the data plotted on lognormal probability paper for both levels of repair.

4.1.6 Set No. 10 - DSIS-SCF

In this case, 50 tests were conducted, 25 for on-line repair and 25 for off-line repair. The equipment has much redundancy which allows on-line repair by "reconfiguring" the system by patching.

For the on-line repair times (set 10a), the chi-squared test rejects the assumption for the exponential distribution but not for the lognormal. For the off-line repair times (set 10b), the chi-squared test does not reject either distribution and, according to our reject criteria, just barely "accepts" both at the same level of significance.

Since the chi-squared test statistic was identical for both distributions for the off-line repair case, both a WE-test and a W-test were then run. This resulted in acceptance of both distributions. A histogram of 5 minutes intervals indicates a roughly lognormal distribution which, together with the higher level of significance of the W-test, and the probability plot (Figure 4(a)) shows that the lognormal distribution is still a "better" descriptor.

Although the chi-squared test resulted in the same significance level for the lognormal distribution for both on-line and off-line repair times, the W-test and the probability plot (Figure 4) both indicate a better fit for the off-line repair time.

4.1.7 Set No. 11 - Communications Subsystem—"Contingency Configuration"

In this case both distribution assumptions would be rejected based on our criteria. A plot of the data on lognormal probability paper (Figure 5) shows indeed that any attempt to fit a straight line to the data points would result in deviations all along the line. This case represents the unsuccessful cases in which the lognormal distribution is found to be an inadequate fit to the repair times.

[The reasons behind this phenomenon, in this and other cases, are subject to further analysis in this study which is not yet completed.]

4.1.8 Sets No. 13, 14 and 18 - User Display Segments

In these cases, the demonstration reports (References 12, 13 and 17) include separate repair times--"inherent" and "achieved." The "achieved" repair time includes additional time required for obtaining test equipment, tools, spare items and maintainability information during the demonstration tests.

The validity of the lognormal distribution, when differentiating between "inherent" (sets 13a, 14a, and 18a) and "achieved" (sets 13b, 14b and 18b) repair time is not quite obvious. For "inherent" repair time, the assumption for the lognormal distribution is not rejected, by both tests, in Case 13, but it is rejected by the W-test in Case 14 and by the chi-squared test in Case 18. For "achieved" repair time, this assumption is rejected by the

chi-squared test in Case 13, but it is not rejected by either test in Case 14 and Case 18. The assumption for the exponential distribution can be rejected in all cases.

These results and lognormal probability plots show that both "inherent" and "achieved" repair time can be assumed to be lognormally distributed.

4.1.9 Sets No. 17 and 19 - HT/MT Terminal and Autodin Memory/Memory Control Equipment

These two cases are discussed not because they violate the assumption of lognormality (in fact the first one does) or because they show, without any doubt, that the lognormal distribution fits the data (as does the second one), but because their tests results would have been expected in the opposite way. In the demonstration reports (References 16 and 18), what is assumed for one system is what should have been assumed for the other, insofar as the appropriate distribution for repair time is concerned. The assumptions in the reports were based on the nature of the equipment, rather than on statistical tests. Probability plotting used in these reports have been found in our study to be inaccurately interpreted (Case 17) and incorrect (Case 19).

In Case 17, the lognormal probability plot included in the report does not show a straight line; most of the deviations are in the lower level (Figure 6). But, despite this, it is concluded in the report that the lognormal distribution fits the data. Both the chi-squared and W-tests reject this assumption, while the chi-squared test and an exponential probability plot (Figure 7) show that the repair time appears to be exponentially distributed, a phenomenon to be investigated in further analysis.

In Case 19, it is shown in the report that an exponential distribution fits the data. There is an error in the way the exponential plot was made in the report. The report explains that "...(the exponential distribution) frequently occurs when repair techniques include diagnostics which have clustered running time and component replacement times which are constant." [Reference 18]. As correct as this statement might be, the statistical tests conducted by us for this case and the probability plots (Figures 8 and 9) do not support it. The exponential distribution is found to be an inadequate model while the assumption of lognormality is accepted.

Figures 8 and 9 readily illustrate the capability of probability plots to give a quick indication of the suitability of a distribution.

4.2 Error in Inherent Availability Caused by Assuming an Exponential Distribution

An additional point that should be pointed out is related to the assumption of the exponential distribution when calculating the inherent availability of a system. From Table III, it can be seen that the percentage error due to this assumption, instead of using the lognormal mean, is low, and, thus, its effect on the value of the availability or on the accept/reject criterion during a maintainability demonstration is not significant.

Since for a high availability what is desired is a high MTBF and a low MTTR, equation (1) can be rewritten as

$$A_I = \frac{1}{1 + \frac{MTTR}{MTBF}} \tag{3}$$

where

$$\frac{MTTR}{MTBF} << 1.$$

In a practical case, MTTR is of the order of one hour while MTBF is of the order of 100 to 1000 hours. Thus, MTTR/MTBF $\approx$ 0.01 to 0.001. Therefore, an error of a few percent in MTTR by assuming an exponential distribution, instead of a lognormal distribution, will have negligible effect on availability. Furthermore, the convenience in using the sample mean, instead of the lognormal mean, despite the error, justifies such a minor deficiency.

It should be also noted that the same effect is not true in the case of the percentiles where the results of our analysis show errors of as much as 25% in the median values and 50% in the upper percentiles.

5. CONCLUSIONS

From the data analysis conducted in this research, we conclude that the lognormal distribution is a good descriptor of the distribution of corrective maintenance repair time. Fifteen of the nineteen cases from maintainability demonstrations of radically different designs tend to show that, with an acceptable level of significance, this assumption cannot be rejected. Similarly, the data analysis shows that the assumption of an exponential distribution should be rejected in seventeen of the cases.

The percentage error in the MTTR when assuming an exponential distribution instead of a lognormal distribution, as a matter of convenience, for calculating system availability

has been found to be small. Other than the one case in which the exponential would not be rejected and the lognormal would, all cases have an error less than 10% and thus will not have any significant effect on availability.

We have found that the methods used in the analysis complement one another. Because of differences among the sets of data and their accuracy, this enabled us to arrive at more meaningful conclusions in some cases.

Histograms, frequently used by some investigators, were found to be helpful to some extent. But in a histogram alone, there is often a loss of information which may lead to wrong conclusions. On the other hand, probability plots are very useful in determining the suitability of a particular distribution and estimating its percentile, and sometimes density, parameters. We did not find any significant difference when estimating the lognormal distribution parameters (50th, 90th, or 95th percentiles) from a probability plot instead of calculating them from the data. The average error is 2% for the median and 5 to 7% for the upper percentiles.

As regards continued research, three areas of special interest will be considered. The first one is mechanical equipment repair times, an area in which, so far, we have been unable to obtain maintainability demonstration data. In this case, the remove/replace or repair actions may be of larger magnitude to the extent of exceeding diagnostic time, and the lognormal assumption may not be valid.

The second area is related to the increasing use of digital techniques in electronic equipment with increasing use of automatic fault detection and diagnosis and built-in test. Coupled with the increasing use of microelectronics, the reduction in diagnostic and repair times is already showing MTTR's approaching ten minutes.

The third area is the investigation of those cases in which the lognormal distribution is rejected in order to discover the underlying reasons therefor.

6. ACKNOWLEDGEMENTS

We wish to acknowledge the assistance given us by members of the maintainability assurance branch of the Engineering Services Division of Ford Aerospace and Communications Corporation. They provided us with detailed reports on maintainability tests on approximately half of the systems analyzed and in discussions of their tests.

The work reported herein was supported in part by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research. LCDR Almog participated in the research as a student at the Naval Postgraduate School.

REFERENCES

1. Aitchison, J. and Brown, J. A. C., The Lognormal Distribution, Cambridge University Press, England 1957.

2. Kline, M. B., "Application of the Lognormal Distribution to Maintainability Design," Logistics Spectrum, Journal of the Society of Logistics Engineers, Vol. 1, No. 2, pp. 13-20, Winter 1967-68.

3. Hahn, G. J. and Shapiro, S. S., Statistical Models in Engineering, John Wiley and Sons, Inc., New York, 1967.

4. Shapiro, S. S. and Wilk, M. B., "An Analysis of Variance Test for Normality (completed samples)," Biometrika, Vol. 52, Nos. 3 and 4, pp. 591-611, Dec. 1965.

5. Coppola, A. and Pettinato, A. D., "RADC Case Histories in R&M Demonstration," Proceedings, 1966 Annual Symposium on Reliability, pp. 395-408, IEEE, New York, 1966.

6. Worster, M. B., Reliability/Maintainability Assessment and Demonstration Test Report on AN/ARC-164(V) Radio Set, Government and Industrial Division, The Magnavox Company, Fort Wayne, Indiana, 2 January 1974.

7. Taylor, J. H., AN/ASN-131(SPN/GEANS) Maintainability Assessment and Demonstration Final Report, Avionics Division, Honeywell, Inc., St. Petersburg, Florida, 26 September 1977.

8. West, E. J., Final Report--Maintainability Demonstration for Harpoon Ship Command Launch Control Set--FF 1052 Class Ship Configuration, RMA-22-75, Naval Ship Weapon Systems Engineering Station, Port Hueneme, California, September 1975.

9. Defense Communications System/Satellite Control Facility Interface System (DSIS) Maintainability Demonstration Report, TR7650, Engineering Services Division, Ford Aerospace and Communications Corporation, Palo Alto, California, 16 November 1977.

10. U.S. Army Satellite Communications Agency Communication Subsystem Program Maintainability Demonstration Report (Contingency Configuration), WDL-TR-5226, Western Development Laboratories, Philco-Ford Corporation, Palo Alto, California, 5 September 1973.

11. U.S. Army Satellite Communications Agency Communication Subsystem Program Maintainability Demonstration Report (Nodal Configuration), WDL-TR-5228, Western Development Laboratories, Philco-Ford Corporation, Palo Alto, California, 5 Sept. 1973.

12. Continental Air Defense Command Ground Data System Maintainability Demonstration Report (User Display Segment), WDL-TR-4701, Appendix F, Western Development Laboratories, Philco-Ford Corporation, Palo Alto, California, 3 December 1972.

13. Strategic Air Command Ground Data System Maintainability Demonstration Report (User Display Segment), WDL-TR-4997, Western Development Laboratories, Philco-Ford Corporation, Palo Alto, California, 9 June 1972.

14. U.S. Air Force Space and Missile Systems Organization 46 Foot Tracking, Telemetry, and Command Antenna Reliability and Maintainability Status Report, WDL-TR-4820, Western Development Laboratories, Philco-Ford Corporation, Palo Alto, California, 10 January 1972.

15. Naval Air Development Center Digital Television Projection Unit Maintainability Demonstration Report, Letter Report, Engineering Services Division, Ford Aerospace and Communications Corporation, Palo Alto, California, February 1976.

16. U.S. Army Satellite Communications Agency Heavy Transportable/Medium Transportable Terminal Program Maintainability Demonstration Report, WDL-TR-5059, Western Development Laboratories, Philco-Ford Corporation, Palo Alto, California, 9 October 1972.

17. National Military Command System Ground Data System Maintainability Demonstration Report (User Display Segment), WDL-TR-4998, Western Development Laboratories, Philco-Ford Corporation, Palo Alto, California, 15 June 1972.

18. U.S. Army Electronics Command AUTODIN Memory/Memory Control Equipment Replacement Project Maintainability Demonstration Report, Letter Report, Engineering Services Division, Ford Aerospace and Communications Company, Palo Alto, California, 20 March 1978.

THE LOGNORMAL DISTRIBUTION

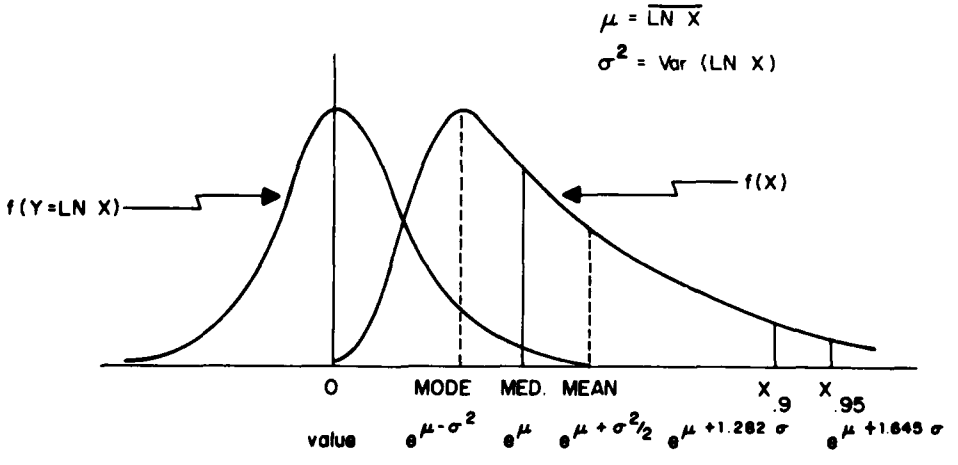


FIGURE 1 - THE LOGNORMAL DISTRIBUTION

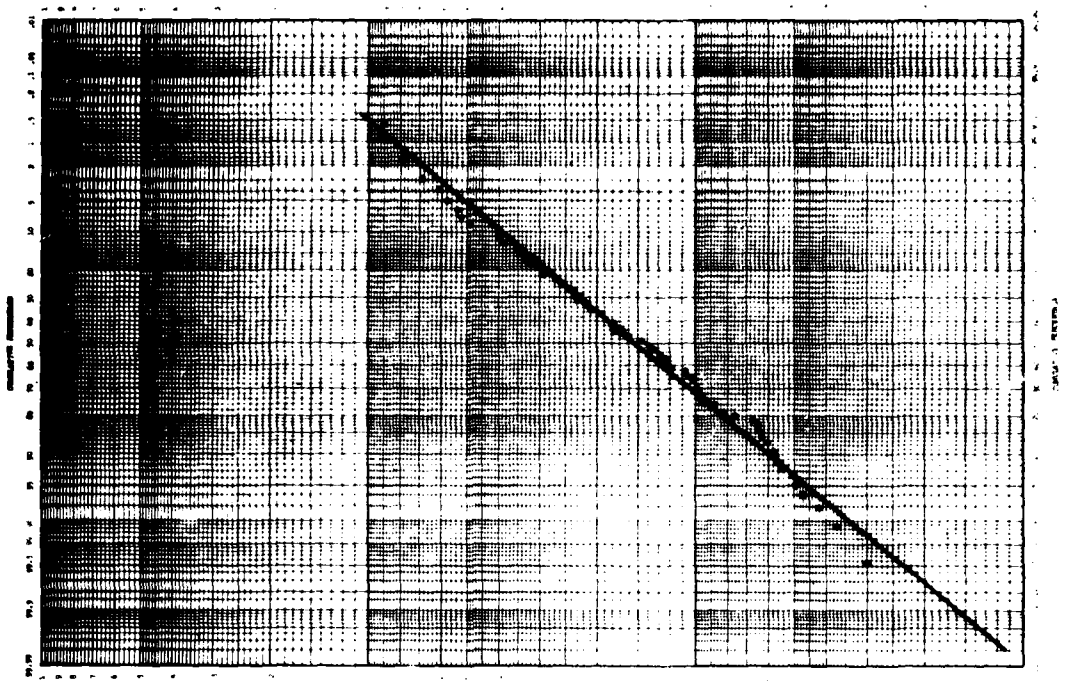


FIGURE 2 - AN/GSA-51
90 REPAIR TIMES

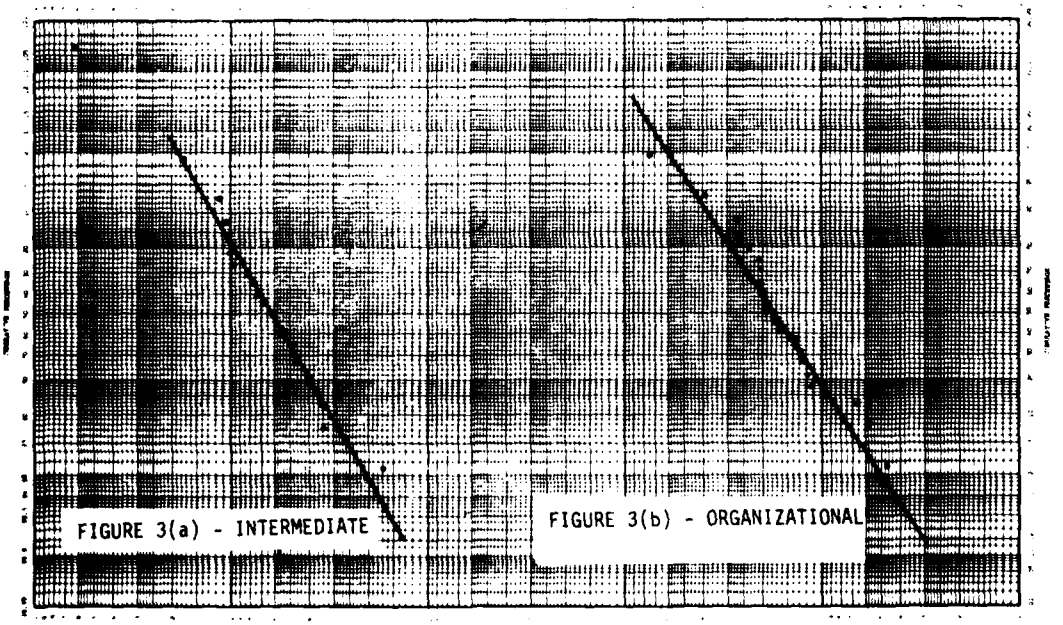


FIGURE 3 - AN/ASN-131
21 REPAIR TIMES

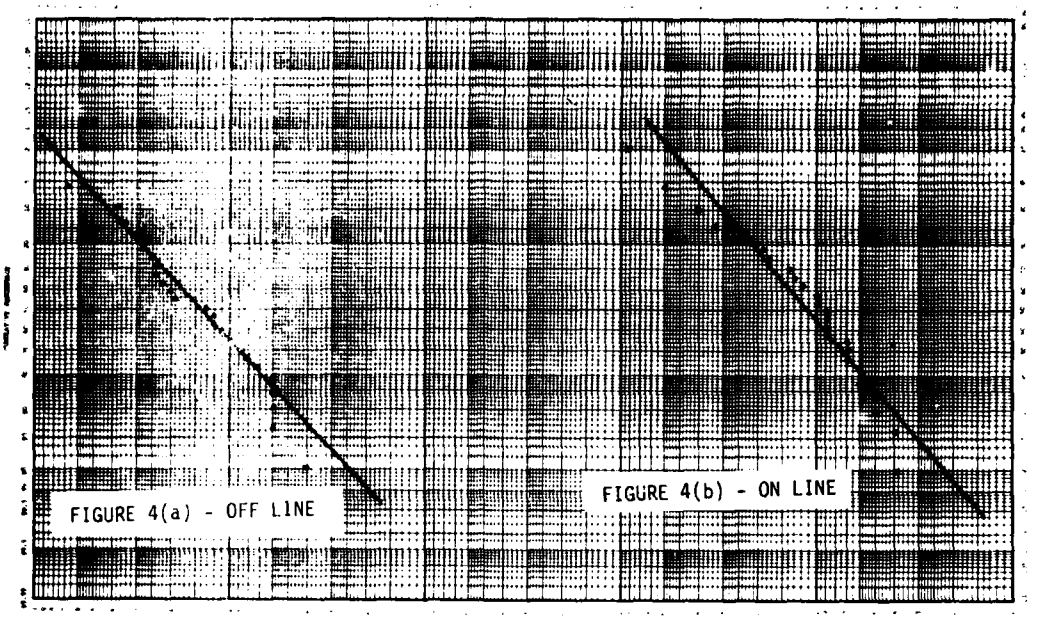


FIGURE 4 - DSIS-SCF
25 REPAIR TIMES

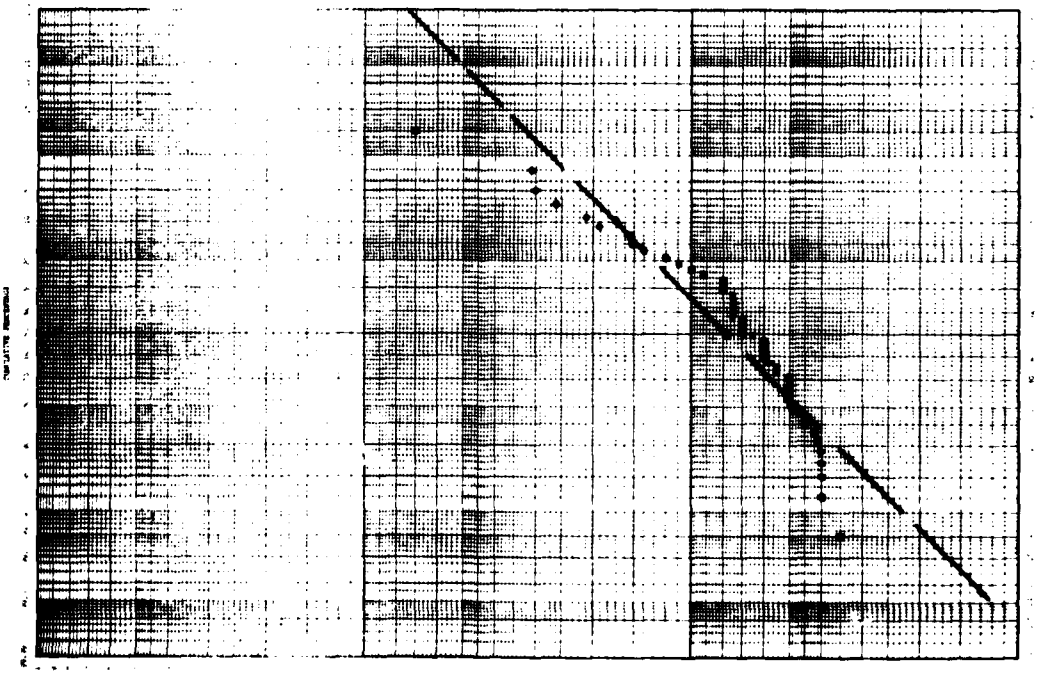


FIGURE 5 - USASATCOMA COMMUNICATION SUBSYSTEM
(CONTINGENCY CONFIGURATION)

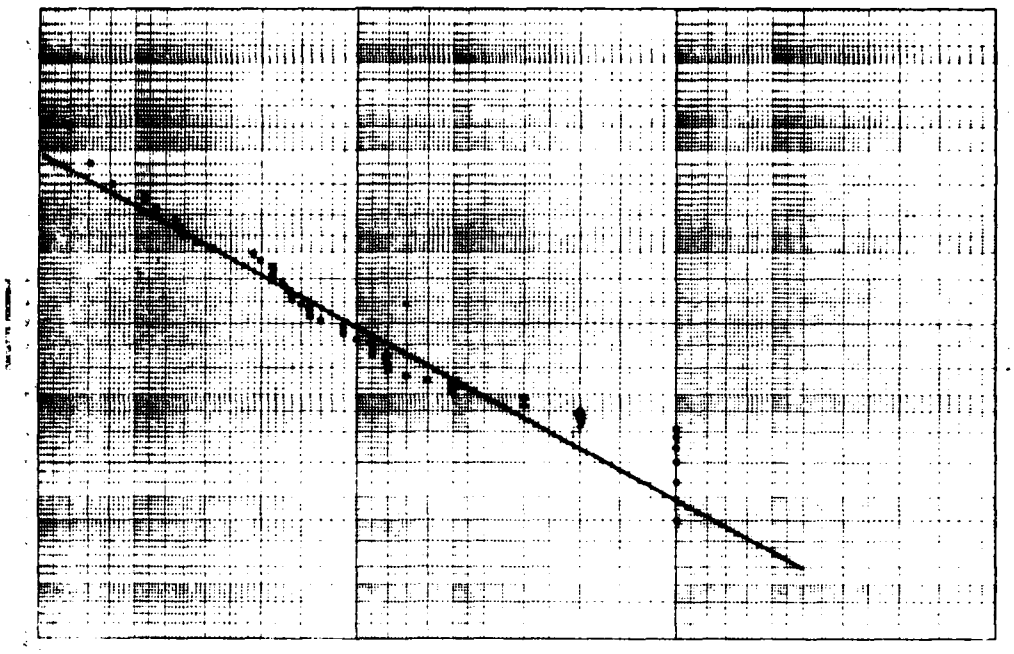


FIGURE 6 - HT/MT TERMINAL
50 REPAIR TIMES

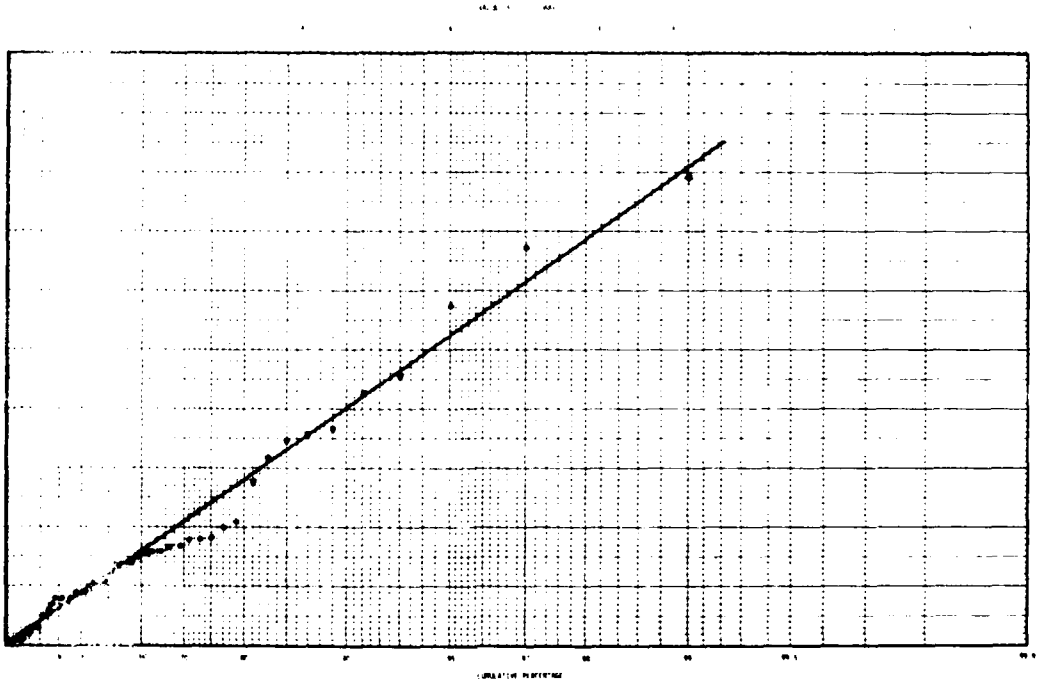


FIGURE 7 - HT/MT TERMINAL
50 REPAIR TIMES

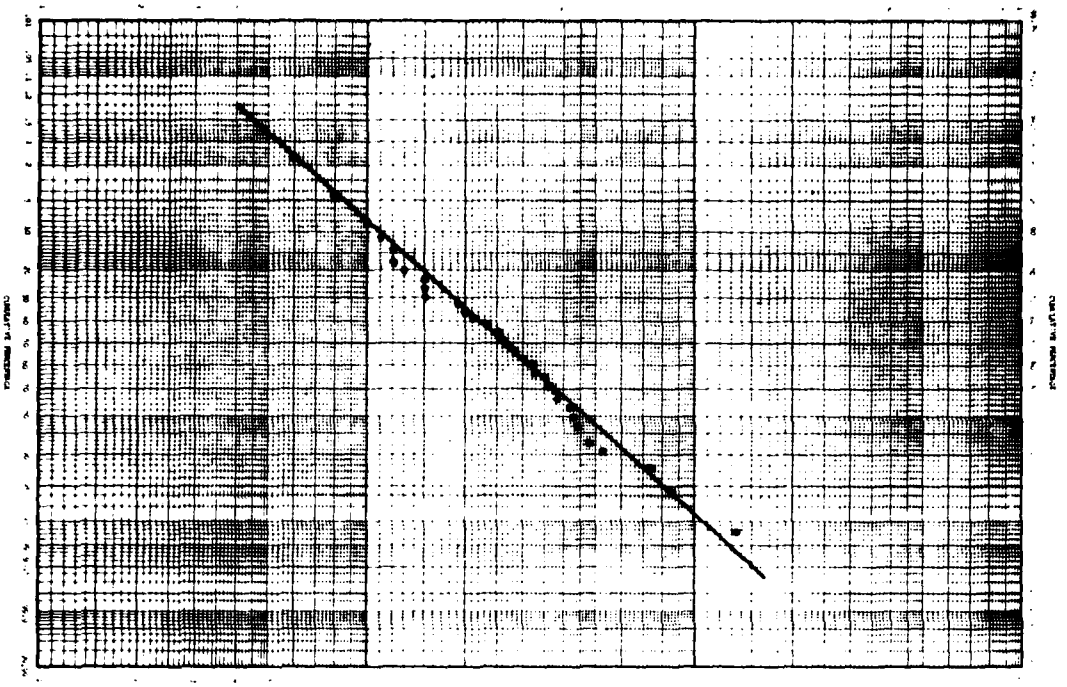


FIGURE 8 - AUTODIN MEMORY/MEMORY CONTROL
33 REPAIR TIMES

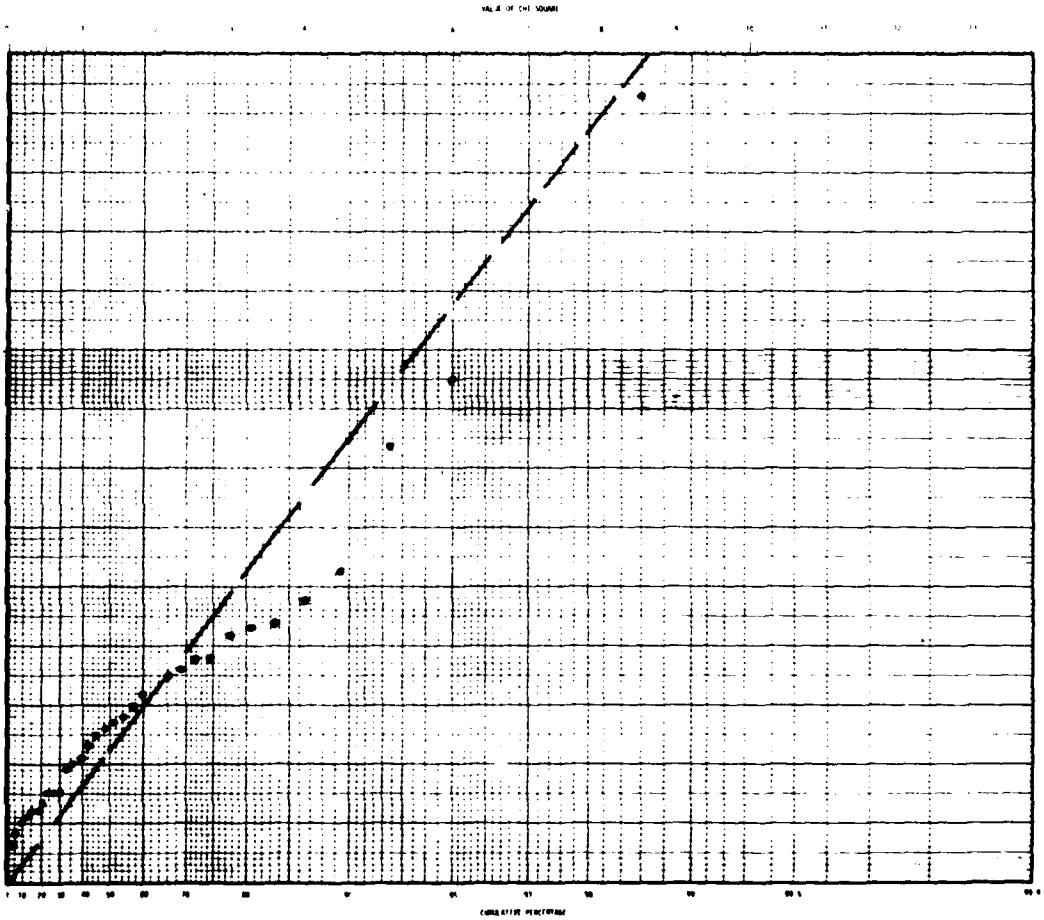


FIGURE 9 - AUTODIN MEMORY/MEMORY CONTROL
33 REPAIR TIMES

DISCUSSION

C.J.P.Haynes, UK

Is it not correct that for analytical availability models, given that the distribution of number of failures per time period is Poisson, then the distribution of numbers out of service can be fully defined from the defect arising rate and the MTTR the distribution of time to repair is not needed (Palm's Theorem)?

Author's Reply

I don't think I fully understand your question, and I am not familiar with the theorem you mentioned. If we assume only one failure and repair at a time, then we must be concerned with the repair time distribution just as we are about the failure distribution. The distribution of numbers out of service is quite a different thing than the time to repair.

B.G.Peyret, Fr

Les durées de réparations interviennent dans la majorité des cas par leur moyenne $E[T_R] \triangleq MTTR$

Donc pour calculer numériquement MTTR, à partir des résultats

$$(T_R)_1, (T_R)_2, (T_R)_3 \ldots\ldots\ldots\ldots (T_R)_n$$

on sait que, si n est grand (σ 30) $\frac{1}{n}\frac{\Sigma}{i}(T_R)_i$ est un bon estimateur de MTTR inconnu et l'on peut proposer

l'intervalle de confiance associé.

Question: Pourquoi a-t-on besoin de raccorder l'échantillon $T_{R1}, \ldots\ldots\ldots\ldots T_{Rn}$ à une distribution a priori expon, log normal etc?

Author's Reply

One can calculate the MTTR of course from data. However, if one wants to predict, allocate and perform design test demonstrations, it is necessary to assume a distribution. In the case of active repair time, where the log-normal distribution is assumed, one needs two parameters which are normally taken as the mean and an upper percentile or the median and an upper percentile. In the case of the lognormal distribution, the median is identically the geometric mean which, in this case, is a better measure of central tendency than the arithmetic mean.

RELIABILITY MANAGEMENT OF THE AVIONIC SYSTEM OF A MILITARY STRIKE AIRCRAFT

A.P. White
J.D. Pavier

EASAMS Ltd.,
Park Street,
CAMBERLEY, Surrey, UK

SUMMARY

The System Management techniques to achieve the reliability requirements for the Avionic System of the PANAVIA TORNADO aircraft are described. The method of apportionment of these requirements to each of the constituent parts of the system is explained. The aims, cost-effectiveness and experience to date of reliability demonstrations are outlined.

1. INTRODUCTION

EASAMS Ltd. is a systems company, which provides operational analysis, system studies, and system design and integration in avionics and other fields.

A decade ago, EASAMS undertook the design and integration of the Navigation and Tactical System of the highly successful Nimrod MR Mk. 1 aircraft.

In 1969, the Royal Air Force, the West German Air Force and Navy and the Italian Air Force, pooled their requirements to define a new high performance military strike aircraft. Due to the various roles it would have to perform this aircraft became referred to as the Multi-Role Combat Aircraft (MRCA). EASAMS has been undertaking the design and integration of the avionics system for the MRCA, and this paper is based on EASAMS' experience on this project.

2. MRCA

The PANAVIA MRCA, now known as the TORNADO, will soon be entering service. It is a two-seater, twin-engined, swing-wing aircraft, which is being produced in two main versions:

- Interdiction/Strike Version (IDS)
- Air Defence Variant (ADV)

This paper is primarily concerned with the IDS version, but similar principles apply for the ADV.

2.1 The Main Roles of the IDS Version

The main roles of the IDS version are:

a) Ground Support
b) Interdiction and Strike
c) Air Superiority
d) Reconnaissance.

2.2 The Operational Environments

The operational environments under which the aircraft would have to operate are:

a) Day or night
b) All weather
c) Defended areas
d) Variety of targets.

3. THE MRCA ORGANISATION

Figure 1 shows the international organisation for the MRCA project. The interests of the defence ministries of the UK, the Federal Republic of Germany (FRG) and of Italy are represented by the NATO MRCA Management Organisation (NAMMO). The NATO MRCA Management Agency (NAMMA) placed a contract with PANAVIA for the design and development of the MRCA aircraft. The partner companies of PANAVIA are BAe (Warton), MBB (Munich) and Aeritalia (Turin). PANAVIA then placed a contract with EASAMS to undertake the following functions:

a) Co-ordinating design responsibility for the avionics system

b) Technical authority for avionics equipment development.

EASAMS placed subcontracts with ESG (Munich) and SIA (Turin). The avionic system companies EASAMS, ESG and SIA are the authorities for technical direction of the avionic equipment suppliers, while the PANAVIA partner companies BAe, MBB and Aeritalia are the respective commercial authorities.

4. RELIABILITY INFLUENCES

Although international collaboration on such a project must inevitably cause many problems there has been a consistent and high level of effort devoted towards achieving the reliability requirements. The main causes appear to have been:

a) Sharing the experience of previous projects of the national services and the determination by the defence ministries to apply adequate planning and financial support to the reliability activities.

b) The size of the combined production orders has helped to justify costs devoted to reliability activities.

c) The monitoring by NAMMA throughout the project of the reliability progress and the encouragement to PANAVIA to ensure that the reliability requirements are achieved.

d) Fixed price contracts onto the equipment suppliers.

e) Money allocated to reliability activities has been preserved and applied to that purpose.

5. APPLICATION OF THE CUSTOMER'S RELIABILITY REQUIREMENTS

5.1 NAMMA - PANAVIA

The Customer in this context means the air forces of the three nations as represented by NAMMA. The technical requirement document included in the contract from NAMMA to PANAVIA states the mission reliability for the whole MRCA weapon system. The weapon system comprises the aircraft together with the essential ground support services. The aircraft itself is divided into:

a) The Airframe

b) The Aircraft Instruments and Control System

c) The Engines

d) The Guns and Armaments

e) The Avionic System.

5.2 PANAVIA - EASAMS

PANAVIA determined an apportionment from the NAMMA requirement in order to place a mission reliability requirement onto EASAMS for the avionic system. EASAMS then had the following tasks:

a) Assess the feasibility of achieving the avionic system mission reliability

b) Apportion the reliability requirement onto each avionic equipment

c) Prepare and negotiate the reliability parts of each equipment supplier's contract

d) Monitor and control the development progress of each equipment supplier to ensure that the reliability requirements are achieved and demonstrated.

6. INITIAL DETERMINATION OF RELIABILITY REQUIREMENTS

6.1 Relative Complexity of the Equipment Types

The MRCA avionic system comprises about 35 different types of equipment. (The exact number depends on the particular air force fit.) Each of these equipments has a different level of complexity and hence a different reliability potential. The relative complexity varies from that of a simple equipment with 100 components to a complex equipment with 10,000 components. In addition there is a range in the complexity of the individual components.

6.2 Market Survey

During an early phase of the MRCA project, before the process of equipment supplier selection, a market survey was carried out. This consisted of preparing and sending our Preliminary Technical Requirements to many electronic firms in the UK, the FRG and Italy to determine those which would be interested in and capable of undertaking the design and development of any of the equipments concerned. In the reliability part of these Preliminary Technical Requirements each firm was requested to state the numbers of the different types of components which their proposed design would contain and the results of a reliability prediction using standard component rates. The component failure rates were copied from the Royal Radar Establishment Report No. 244(1). Many of the prospective equipment suppliers considered that these failure rates were pessimistic and submitted reliability estimates based upon their own component failure rates. However, for a fair apportionment it was essential that a uniform basis should be used.

6.3 Equipment Reliability Analysis

When all the replies to the Preliminary Technical Requirements had been received they were assessed to ensure that each proposed design had a reasonable chance of meeting the performance requirement. The reliability estimates in the replies were then checked against the prescribed standard component failure rates, and the estimates were then averaged for each equipment type. The results gave a measure of the relative complexity of each equipment type necessary to meet the performance design requirements.

6.4 Apportionment of Reliability Requirements to each Equipment Type

The first step towards the apportionment of reliability requirements was to make an analysis of the mission reliability. This will be described later under the heading Reliability Model.

The results of the equipment reliability analysis were compared with the avionic system mission reliability specified by PANAVIA. The reliability estimates for all the avionic equipments were then adjusted by an equal factor to achieve the mission reliability. These apportioned equipment reliability values were subsequently used for the avionic equipment design specifications for the development contracts.

7. COST EFFECTIVENESS OF CONTRACTUAL RELIABILITY DEMONSTRATION

7.1 Contractual Aspects

At an early stage in the MRCA project it was proposed that the contracts onto the avionic equipment suppliers for both the development and the production phases should include reliability values as contractual requirements. This means that the reliability must be specified in a way which can and will be proven by test. Each supplier's contract was negotiated as a 'fixed price' and the costs of the reliability demonstration tests were regarded as a valid part of this fixed price. However, if the equipment design failed the demonstration, the supplier would be required to carry out redesign and resubmit to test without increase of the cost limits of the negotiated contract. Although the cost reserved for the reliability demonstrations formed a significant part of each supplier's contract it was expected that there would be a cost saving overall as a result of increased reliability of the equipment when in service use. NAMMA therefore requested considerations of the cost implications of reliability testing, initial and life cycle.

7.2 Cost Implications on Procurement

The requirements for the reliability demonstrations of the MRCA avionic equipment were taken from the American document MIL-STD 781B(2).

The cost of a reliability demonstration comprises:

a) The design test samples
b) The environmental test facility
c) The test equipment to operate and check the equipment under test
d) The engineering effort involved in setting up and running the test.

When carrying out this analysis in 1971 calculations were, of necessity, based on very limited factual information. Where the data was vague there was a tendency to form pessimistic assumptions.

7.3 User Costs

The benefit of a contractual reliability requirement is that User costs were expected to be lower due to the equipment showing a higher reliability on entry into service use. On the basis of experience expressed in a UK Ministry of Defence report it was expected that equipment that did not have to comply with a contractual reliability demonstration would enter service use with 20% of the predicted reliability

and that, resulting from in-service experience and consequent design changes, would increase to 90% of the predicted value within six years. Conversely, where a contractual reliability demonstration applied, the equipment would enter service with 45% of the predicted reliability and would increase to 90% of this value in one year. This is presented in Figure 2. When equipment entering service use has a higher reliability the following results can be expected:

a) Fewer aircraft would be required to achieve the same mission effectiveness. This would reduce not only the prime aircraft costs but also the operation costs.

b) The reduction in the number of equipment failures would reduce the maintenance costs and the amount of spares holding to effect these repairs.

c) Most of the reliability design improvements would have been effected by the supplier, thus reducing the number and application of modification kits during service use.

7.4 Cost Comparison

When the detailed cost estimates were made as outlined above the expected cost saving in service use came to 1.8 times the cost of including reliability as a contractual requirement. This was sufficient to justify such a policy even allowing for some uncertainty in the data used.

The method of calculating the reliability demonstration costs had applied rigidly MIL-STD 781B Test Plan II to each equipment. The effect of this is that the demonstration costs for some of the less complex and hence more reliable equipments would represent a very high proportion of the development and production costs for those equipments. Furthermore a proportional reduction on the reliability of those equipments would have a less significant effect upon the system reliability. This indicates therefore that there should not be reliability demonstrations on some of the less complex equipments.

A further comparison with the User Costs shows that the cost-effectiveness could be four times rather than 1.8 times.

Following the estimates which were made back in 1971, it was decided that a reliability demonstration should be applied to 30 out of the total 47 MRCA avionic equipments.

The production phase reliability demonstration techniques and costs are currently being resolved. After some years in-service use, the benefits of the contractual reliability demonstrations will be revealed. It is strongly recommended to the Ministry of Defence departments of each of the three nations that the in-service cost-effect of these reliability demonstrations should be measured and compared with the procurement costs; this would help to justify on a future project the level of attention to reliability activities that there has been on the MRCA project.

8. STANDARD RELIABILITY REQUIREMENTS FOR EQUIPMENT SUPPLIERS' CONTRACTS

8.1 Reliability Content of Equipment Specifications

Following the cost-effectiveness study of reliability demonstrations it was initially planned to require all new avionic equipments with MTBFs less than 4,000 hours to be demonstrated in the Development Phase. For cost saving reasons the demonstrations of some of the less important equipments were deferred until the Production Phase and treated as Options, but Development Phase reliability demonstrations were required for 30 equipments.

A 'model specification', defining standard requirements for every avionic equipment, was written, into which any quantitative or other requirement applicable to a particular equipment could be inserted. The first basic task in the formulation of standard reliability requirements was to define unambiguously the terms 'defect' and 'failure'.

Having defined the terms 'defect' and 'failure', and assuming that defects and failures are random, it remained to specify the conditions under which the equipment is used before a quantitative requirement can be imposed. Three different reliability requirements were in general standard:

a) Defect rate installed in aircraft, for ground and flight operational conditions

b) Failure rate installed in aircraft, for ground and flight operational conditions

c) MTBF under a defined Test Plan and demonstration conditions of MIL-STD 781B.

The first two of these, although of most interest to the Customer are difficult to enforce contractually, because it is difficult to prove to the supplier that the particular conditions in the aircraft and methods of use did not exceed the specified limits for the equipment. On the other hand, a reliability demonstration is conducted under carefully controlled conditions at the Supplier's premises. However, reliability demonstrations are long and expensive tests, and the results are statistical; there are always chances that a sub-standard equipment will pass and that an above standard equipment will fail.

8.2 Test Requirements

The standard conditions adopted for reliability demonstrations are shown in Figure 3.

The four main types of stress on the equipment, namely temperature cycling, vibration, ON-OFF switching and voltage cycling are not intended to simulate conditions in the aircraft but to provide a universal standard against which equipment could be assessed, and it was expected that similar results to fully operational service would be obtained.

Probability Ratio Sequential Tests (PRST) of MIL-STD 781B were used in general, rather than the alternative Fixed Length Tests. A typical PRST plan, shown in Figure 4, has three basic parameters:

a) Supplier's risk - the probability that a 'good' equipment will fail

b) Purchaser's risk - the probability that a 'poor' equipment will pass

c) Discrimination ratio - the MTBF ratio of a 'good' to a 'poor' equipment.

'Relevant' failures, that is, failures attributable only to the equipment under demonstration, are plotted in relation to the applicable test plan and eventually an accept or reject decision is reached. Failures due to external causes or occurring outside the demonstration, e.g. during burn-in, are termed 'non-relevant' and are not counted.

Test plans of MIL-STD 781B cover a range of risks and discrimination ratios. The lower the risks and discrimination ratio required, the longer the test plan if expressed in multiples of MTBF. Although Test Plan II of MIL-STD 781B was initially proposed as a basis, it was not possible to use Test Plan II for high MTBF equipments with the available number of models in the required timescale. The following table shows typical test plans chosen for ranges of MTBF:

MTBF Range	Test Plan	Risks	Discrimination Ratio
Up to 500 hours	II	20%	1.5 : 1
500 to 1,600 hours	IV	20%	2 : 1
1,600 to 4,000 hours	IVA	20%	3 : 1

The test plans were chosen so that demonstrations were estimated to be completed in one year with three models being tested simultaneously. One year was considered to be sufficient time for either a demonstration to reach an accept decision at the truncation point, or for a reject decision to be reached, the equipment modified and the second demonstration to reach a decision at the expected decision point.

The possibility of early failures causing pessimistic reliability demonstration results was reduced by allowing the equipment models to undergo a period of 'burn-in' before the start of the demonstration. MIL-STD 781B allows any period of burn-in to be carried out on the demonstration models or none at all, provided that the same burn-in is applied to all other models. However, a minimum burn-in requirement of 48 hours ON-time under reliability demonstration conditions was imposed on models to be delivered, which effectively limited the burn-in on demonstration models to 48 hours.

8.3 Commercial Aspects

The main purpose of the requirement for reliability demonstrations was to encourage suppliers to make maximum efforts to ensure reliability in the design and manufacturing stages. The price for reliability demonstrations was fixed and was usually related to truncation or expected decision points, so that if a demonstration was passed with a minimum number of failures, the difference between the actual cost and fixed price received was profit to the Supplier. Conversely, if a supplier had several reject decisions before reaching a final accept decision, a loss could be incurred, apart from his expense and time in the redesign of the equipment and resubmission for demonstration.

9. RELIABILITY MODEL

Having defined the reliability requirements for all the avionic equipments, an estimate was made to determine whether the Customer's overall requirement could be met. This involved a calculation of the mission reliability of the avionic system from the defined mission and the failure rates of the equipments used. For this purpose a standard mission was chosen out of the many possible missions this multi-role aircraft could perform. The chosen mission represented a planned strike against an enemy ground target and was assumed to be a peacetime mission simulating a wartime situation. The mission was divided into eight phases as follows:

a) Phase 1: Check of all equipment by aircrew. Any failure of an equipment during this phase would imply failure of the mission.

b) Phase 2: Take-off

c) Phase 3: Outward flight over friendly territory

d) Phase 4: Outward flight over hostile territory

e) Phase 5: Attack

f) Phase 6: Return flight over hostile territory

g) Phase 7: Return flight over friendly territory

h) Phase 8: Approach and landing.

The requirements were therefore:

i) Phase 2: Manual Control

ii) Phases 3-5: Automatic flight control and terrain following

iii) Phase 6: Manual or automatic flight control and terrain following

iv) Phase 7: Manual or automatic flight control

v) Phases 3-6: Accurate navigation

vi) Phases 2-8: Communication with ground and between aircrew

vii) Phase 8: Visual or instrumented approach

viii) Phases 4-6: Defensive aids

ix) Phases 2-8: IFF.

For every required function in each phase the necessary equipments are listed for all prime and acceptable reversionary modes. An example of this is shown in Figure 6. A reversionary mode may be selected, but this usually involves degraded accuracy or increased aircrew work load.

The overall mission reliability can be calculated from a series of logic equations by means of a computer program originated by MBB(3). The program can accommodate factors which may be applied to equipment failure rates to represent different modes of operation in different phases, and it also allows for the possibility of components failing in phases previous to the one in which they are used.

The possibility of failures occurring in equipments which are not being used at the end of a mission is allowed for by applying a factor in Phase 1. All detected failures are required to be repaired between missions, but undetected failures which occur during a mission are not detected until Phase 1 of the next mission.

10. PROGRESS OF EQUIPMENT RELIABILITY DEMONSTRATIONS

10.1 Allowance for Design Improvements

In order to fulfil a contractual requirement to pass a reliability demonstration, a Supplier must:

a) Design the equipment

b) Build it

c) Test it under all conditions including RDT conditions

d) Identify any design weaknesses

e) Modify the design

f) Repeat c) to e) until no weaknesses are thought to remain

g) Carry out the reliability demonstration.

Ideally there should be more preliminary testing than formal reliability demonstration testing. However, because of the extremely tight time schedules allowed for development and qualification, there was often insufficient time left for adequate preliminary testing. Many suppliers therefore requested that some design improvements should be allowed to be made during the demonstration phase. This was agreed and an addition made to the rules for reliability demonstrations contained in MIL-STD 781B, to the effect that a failure once classified as relevant, may be re-classified to non-relevant with the Purchaser's approval. Minimum conditions for such approval to be given are:

i) Sufficient test data has been accumulated and presented to the Purchaser to assure that the corrective action proposed is effective in eliminating the failure mode.

ii) The above referenced corrective action (which may be design, part or production process change), is incorporated in all equipment of the lot from which the reliability test sample was drawn, or on those equipments selected by the Purchaser.

This additional rule sometimes allowed a demonstration to continue after a reject decision, if corrective action could be undertaken and shown to be effective. For a reclassification to be approved the Purchaser has to be convinced that if the corrective action had been applied to the equipment before the demonstration started, that failure would not have occurred. It was therefore considered that the reclassification rule does not reduce the effectiveness of the reliability demonstrations, and would combine a reliability improvement programme and a reliability demonstration in the same test, thus saving time and ultimately producing a more reliable design.

10.2 Results

Table 1 shows the results of reliability demonstrations to date.

TABLE 1

RESULTS OF RELIABILITY DEMONSTRATIONS TO DATE

Total number of avionic equipments	47
Total number of reliability demonstrations required	30
Number of demonstrations started to date	26
Number of demonstrations which have reached accept or reject decisions	21
Number of first time accept decisions	13
Number of accept decisions after initial reject decisions	3
Number of accept decisions without any failure	3
Number of accept decisions where failures were reclassified	13
Number of demonstrations in progress where failures have been reclassified	5

Many design improvements have resulted from using this procedure. As shown in Table 1 it has been possible to apply reclassification to most of the demonstrations. The three demonstrations which had no failures were all of equipments with MTBFs above 1,800 hours. This leads to the conclusion that demonstrations of less complex equipments may not be cost-effective. It is considered that the major stresses of temperature cycling, vibration and on-off switching have been effective in showing up weaknesses in the equipment. Voltage cycling is considered less effective, because the equipments generally have power input stabilisers, and only these parts of the equipments experience the voltage changes. The direction of vibration required by MIL-STD 781B (normal to the majority of the printed circuit boards or cards) led to equipments being mounted in the chamber on their sides or ends. For most of the TORNADO equipments, this inconvenience was considered unjustified, because the boards were small and well-supported, and the stress on the board connectors would have been greater with a vibration direction in the plane of the boards.

The additional stresses of random vibration and moisture required by the more recent standard MIL-STD 781C(4), which is based on actual mission profile environments, are not considered cost-effective at present. To summarise, from experience of working on TORNADO, it is considered that reliability demonstrations according to MIL-STD 781B, with processes for reclassification of failures as a result of corrective action, are effective in ensuring a reliable design of equipment.

11. MAINTENANCE OF STANDARDS DURING PRODUCTION

11.1 Batch Sampling Tests

Once an equipment has passed its Qualification tests, including the reliability demonstration, its design is considered to have satisfied the requirements of the development contract. Although it is statistically possible for an equipment to pass a reliability demonstration with design weaknesses, it is more likely that subsequent unreliability of equipment in Production is due to manufacturing deficiencies or bad batches of component parts, rather than problems in design.

Some method of demonstrating that qualification standards are maintained in Production was initially thought clearly necessary, and batch sampling tests were envisaged. To reduce the requirement for test facilities to the level required for qualification demonstrations, and for reasons of economy, a low sampling rate (10% reducing to 2%) was proposed. Another cost saving proposal was to treat these batch sampling tests as options which would not be taken up, if experience of the equipment in use were so far satisfactory.

11.2 Alternative Tests

It soon became evident, however, that this type of test would not be very effective in detecting a deterioration in standards, because of the lengthy time interval between the start of any such deterioration and the results of any subsequent batch sampling test. Alternative methods of checking the maintenance of standards were considered and the most cost-effective method was considered to be the testing of every equipment produced under demonstration conditions for a short period, typically 30 hours. The defects found would be plotted cumulatively against time, in relation to a test plan, as shown in Figure 5.

If the staircase defect-time plot crosses the 'Corrective Action Required' line, or if pattern defects are present, the Supplier must take corrective action. Typically, after every 150 equipments tested, if the 'Corrective Action Required' line has not been crossed, the plot is restarted at zero. The Purchaser reserves the right to stop the acceptance of the equipment at any time that a 'Corrective Action Required' situation exists.

This type of testing, known as Production Reliability Assurance Testing (PRAT) has the main advantage of quick indication if anything is going wrong. As it is similar to an extended burn-in, it will require minimum extra facilities and cause minimum interference to the Supplier's production process.

A disadvantage of this type of testing is that the demonstration period takes place on every equipment just after the burn-in period. If the burn-in does not completely cover the period of 'infant mortality' defects, the results of PRAT will be pessimistic. Objections by suppliers to PRAT on these grounds are usually overcome during negotiations by allowing for the possible extra defects in the risks associated with the chosen PRAT plan. PRAT has now been negotiated with nearly every supplier and is now being applied to the first Production equipments.

12. CONFIRMATION OF RELIABILITY ACHIEVEMENT

Reports of defects occurring during TORNADO flight trials are being analysed and assessed. Any apparent patterns are notified and corrected, and the overall avionics defect rates per flying hour are calculated for six monthly periods.

These calculated defect rates are considered pessimistic because:

a) An early standard of equipment is used

b) Interfacing and compatibility of equipments is not fully developed

c) Ground tests are conducted on equipment installed in aircraft. During these tests equipment accumulates defects but no flying hours.

However, results over the last four six-monthly periods show a steady decrease in defect rates which indicates that the target will be achieved, as shown in Figure 7.

13. CONCLUSION

Throughout the TORNADO programme the Customer, PANAVIA organisation and the equipment suppliers, have all worked vigorously to achieve the reliability requirements for the avionic system. As a result, the avionic equipment designed has proved capable of maintaining performance to stringent specification requirements under severe environmental conditions with a low failure rate, thereby ensuring that the TORNADO Weapon System will have a high probability of mission success. The main factors influencing the achievement so far have been the high level of European co-operation at all levels, the decision to impose reliability demonstrations and to allow for corrective action during these tests.

As may be expected, however, various problems have had to be overcome, and the following in particular have caused difficulties.

a) Equipment contractors were instructed to subcontract parts of their equipment to other suppliers, often in different countries, as part of national work-share agreements. This has led to communication delays and problems in determining liabilities.

b) In some cases an equipment had too low a level of complexity for an effective reliability test programme to be performed independently. It would have been better in such cases, where applicable to the system design, to include it with another equipment and to make one supplier responsible for both equipments.

c) Some equipments are not common to all aircraft but were selected differently for different national requirements. Due to the lower production requirement for 'national fit' equipments, the relative cost to achieve the required reliability is greater.

d) Suppliers were authorised to begin manufacture of Production units before all development tests have been completed. This led to priority being given to Production manufacture rather than to further reliability improvement.

It is recommended that, for similar projects in the future, the relevant authorities should appreciate those difficulties and should, where possible, avoid them by arranging the organisation and management of the project accordingly.

14. SYMBOLS USED

^{o}C = degree Celsius

g = local acceleration of free fall

h = hour.

15. ACKNOWLEDGMENTS

Opinions expressed in this paper are those of the authors but the experience and information have been derived from work on the TORNADO project. The authors thank EASAMS Ltd. for permission to present this paper.

A. P. White, J. D. Pavier
EASAMS, 18 October 1978.

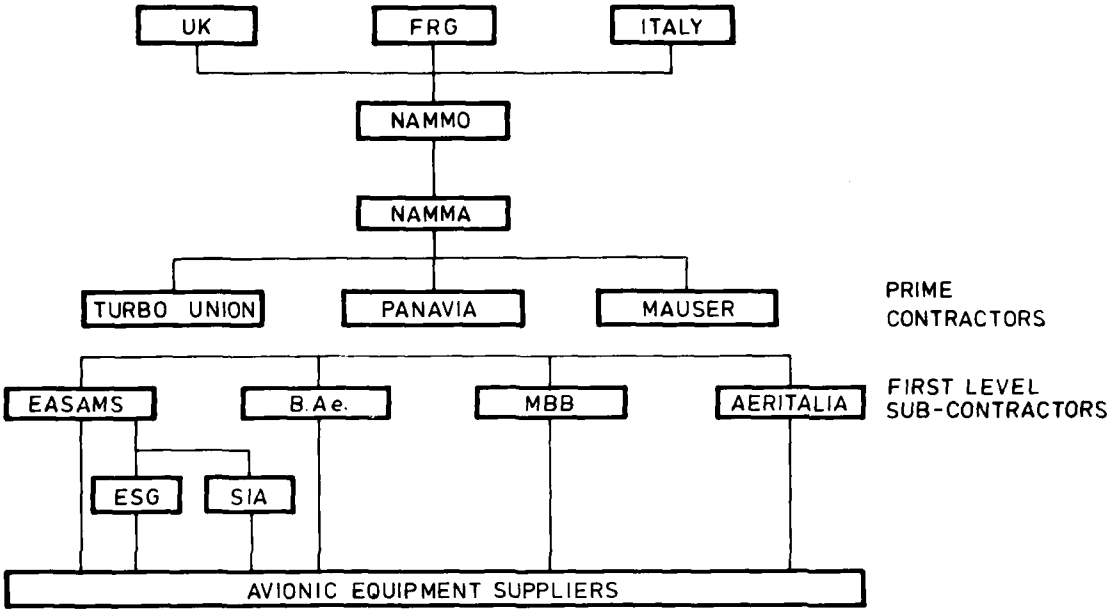
16. REFERENCES

1. Royal Radar Establishment, Report No. 244, 1967, "Part Failure Rates - Electronic Equipment Installed on High Performance Military Aircraft".

2. US Military Standard, MIL-STD 781B, 1967, "Reliability Tests: Exponential Distribution".

3. Kühnlein, H., and Trötsch, E., 1971, "Ablauf und Planungsforschung", Band 12, Heft 1, 27-41.

4. US Military Standard, MIL-STD 781C, 1977, "Reliability Design Qualification and Production Acceptance Tests: Exponential Distribution".



B.Ae BRITISH AEROSPACE
SIA SOCIETA ITALIANA AVIONICA
MBB MESSERSCHMITT-BÖLKOW-BLOHM
ESG ELEKTRONIK-SYSTEM-GESELLSCHAFT

Fig. 1 MRCA ORGANISATION.
AVIONIC SYSTEM AND EQUIPMENT

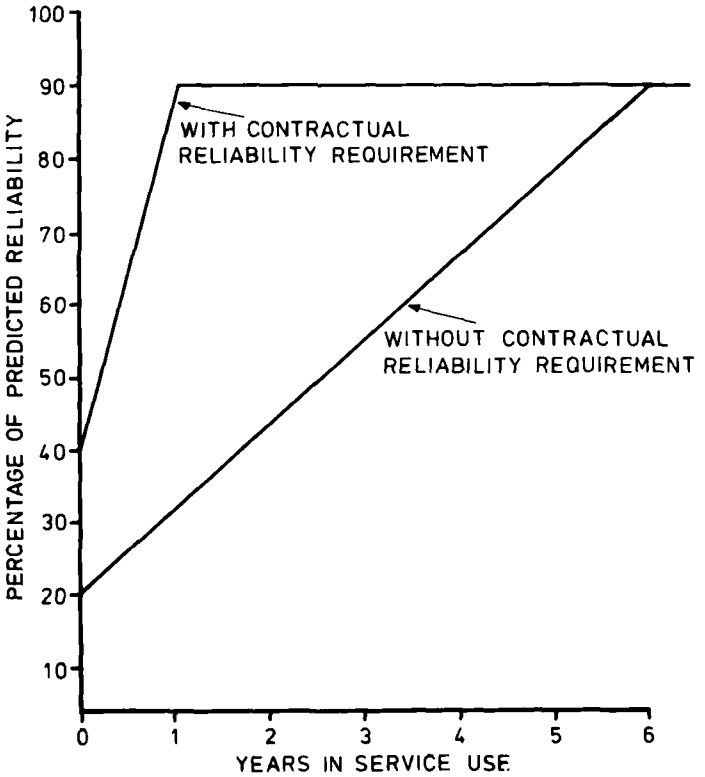


Fig. 2 IN-SERVICE RELIABILITY.
EFFECT OF CONTRACTUAL RELIABILITY REQUIREMENTS

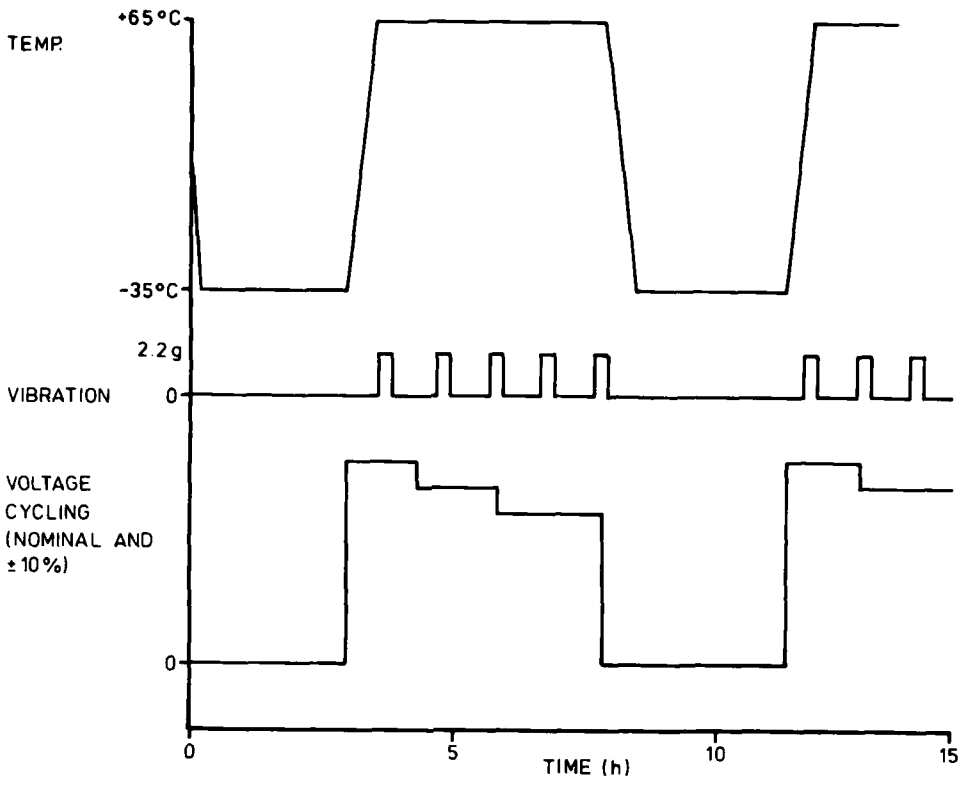


Fig. 3 TYPICAL CONCURRENT STRESSES FOR RELIABILITY DEMONSTRATIONS

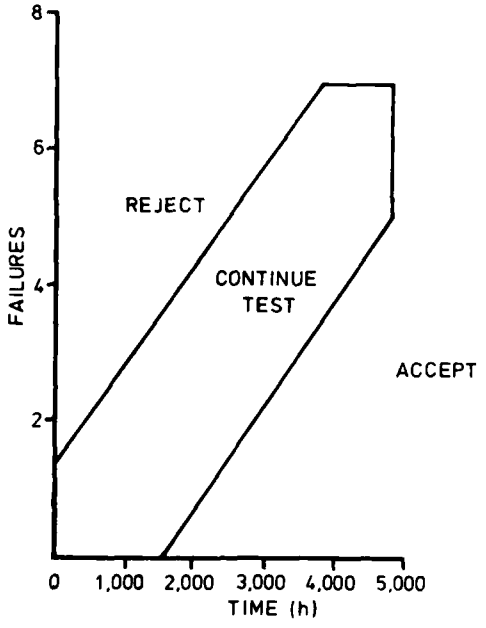


Fig. 4 TYPICAL PRST PLAN OF MIL-STD 781B

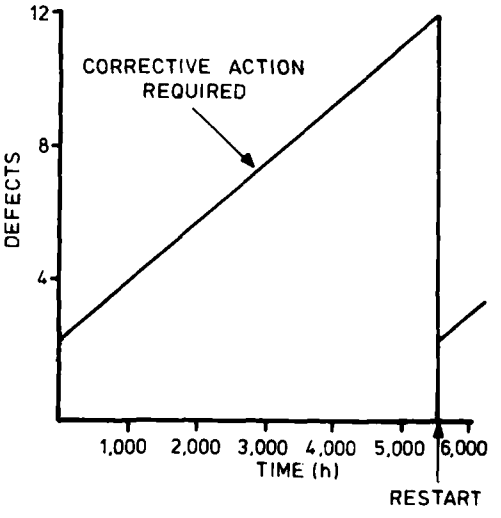


Fig. 5 TYPICAL PRAT PLAN

FUNCTION: FLIGHT CONTROL

PHASE: TAKE-OFF

PRIME MODE EQUIPMENTS: INERTIAL NAV — SAHR — MAIN COMP — AIR DATA COMP — HUD

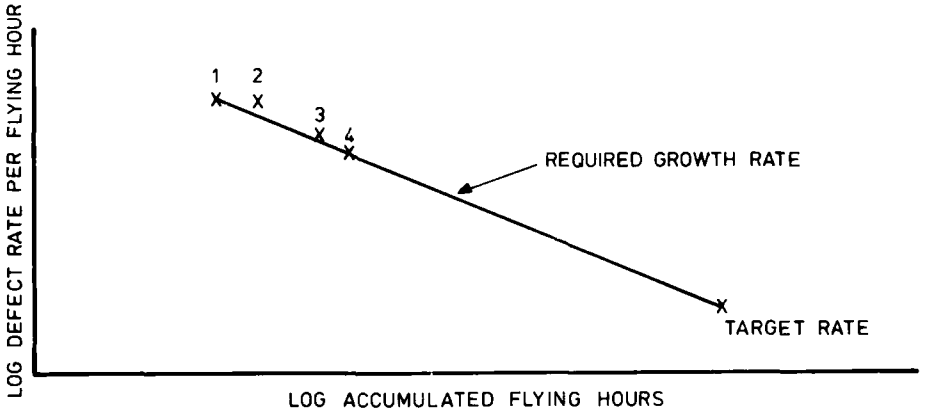
REVERSIONARY MODES 1) INERTIAL NAV — MAIN COMP — AIR DATA COMP — HUD

2) SAHR — MAIN COMP — AIR DATA COMP — HUD

COMBINATION: (INERTIAL NAV / SAHR) — MAIN COMP — AIR DATA COMP — HUD

LOGIC EQUATION: MA1 = (6.200 + 6.205) * 6.224 * 6.201 * 6.215

Fig.6 EXAMPLE OF DERIVATION OF LOGIC EQUATION INPUT TO COMPUTER MODEL



1. DEFECT RATE UP TO 31.8.76
2. DEFECT RATE 1.9.76 TO 28.2.77
3. DEFECT RATE 1.3.77 TO 31.8.77
4. DEFECT RATE 1.9.77 TO 28.2.78

Fig.7 OBSERVED DEFECT RATE OF AVIONIC SYSTEM

DISCUSSION

A.Andrews, UK

(1) Does not the fact that you allow manufacturers to discount a failure for which a fix is introduced during the test, introduce the risk that the modification itself, which may possibly have undesired side effects, is not subject to the rigours of the test.

(2) You mentioned that environmental testing was not carried out on TORNADO equipment because it would not be cost effective due to the high cost of the capital investment. Is this really the case, or is it that funds were simply not available at a late stage in the program?

Author's Reply

(1) Sufficient test experience is always required before a failure can be reclassified to non-relevant (and thus discounted) as a result of design corrective action. This test experience can often be accumulated during the remainder of the demonstration. However, if corrective action is applied towards the end of a demonstration, we would certainly insist on extra testing to prove that the fix is effective and that it has not introduced undesirable side-effects.

(2) I would expect that testing to Mil-Std-781B would reveal a large proportion of the design weaknesses in an avionic equipment. Although testing to Mil-Std-781C may provoke a few more failures, I am not convinced that these few extra failures would justify the high cost of the test facilities required by Mil-Std-781C, in a program such as TORNADO.

INTRODUCTION TO SOFTWARE RELIABILITY - A KEY ISSUE OF COMPUTING SYSTEMS RELIABILITY

Günter Heiner
AEG-TELEFUNKEN, Forschungsinstitut
Abt. N14V3, Postfach 1730, D-7900 Ulm, FR Germany

SUMMARY

The paper presents an introduction to the problems of computing systems reliability with special emphasis on software reliability and gives a survey of the fundamental approaches to achieve reliable software.

Starting from a definition of the basic terms and a classification of various error types the concept of software reliability is examined. The problems involved in treating hardware and software reliability on common terms are discussed. For the purpose of illustrating the concept of software reliability, two simple reliability models are presented.

The basic methods of attaining reliable computing can be divided into the two complementary approaches of fault-avoidance and fault-tolerance. With regard to software reliability fault-avoidance is still the predominant approach. In this category the two classes of constructive and analytical methods are summarized and discussed. Constructive methods facilitate error-free software construction and provide for a good testability of the software. Analytical methods are used for software validation. The main techniques are proving, static and dynamic analysis. Especially for systems meeting with very high safety requirements, there is need for software fault-tolerance. Two fundamental approaches to fault-tolerance by program diversity are presented and discussed.

1. INTRODUCTION

In the past the reliability of the computer itself, i. e. the hardware, was predominant, and it was only for the hardware that the reliability could be quantified satisfactorily. Until now there is no acknowledged model of system reliability for computing systems in which the interaction of the hardware, software and human operation is taken into consideration. This is all the more remarkable as the ratio between hardware and software in terms of cost has drastically changed in the past few years (80 % software in 1973 vs. 25 % in 1960 [RAMAMOORTHY, C.V. et al., 1974]). In quite a number of large scale systems, more than 50 % and even up to 75 % of the total software cost were incurred for maintenance only, i. e. after the system was made available for use [MYERS, G.J.,1976].

It is obvious that the high cost of software is largely due to reliability problems.

2. BASIC CONCEPTS AND TERMS

Regarding computing systems reliability and, in particular, software reliability, we can note a confusion in terminology and a lack of consistency in the definition of the terms used. The confusion is largely due to the fact that people involved in development and application of computing systems employ subjective, i. e. very distinct, approaches to assess the reliability of the systems.

Hence treating terms and definitions in some detail is necessary for a thorough discussion of the topic "software reliability".

2.1 COMPUTING SYSTEMS RELIABILITY

If possible, we will use the same reliability notions for computing systems as for general technical systems. There are, however, special aspects in
- the interaction of the hardware, software, and human operation,
- the enormous number of discrete system states, and
- the extremely high rate of state changes. Compared to other physical systems, the changes in state are, for all practical purposes, "inertialess" [GOLDBERG, J., 1975].

The result is that faults can spread quite quickly to the overall system. On the other hand, however, faults can remain "hidden" from the user for a long period and are very difficult to reproduce.

In the following sections, the function and fault terms will be defined and the sources of faults will be investigated.

2.11 The Concept of Function

In analogy with IEC TC 56 (list of basic terms), the reliability of a computing system is understood as being its ability to perform the required functions under stated conditions for a stated period of time.

The performance of the required function in the sense of the definition given means that the programs are executed "correctly", i. e. in conformity with the intended specifications [FREY, H.,1977].

The correct function presupposes the following individual aspects [AVIZIENIS, A., 1975]:
1. correct hardware;
2. correct software;
3. correct execution of the computer programs in the presence of hardware failures.
   This, in turn, presupposes:
   - that the results are not falsified as a consequence of failures;
   - that the programs are not altered as a consequence of failures ("program integrity");
   - that the data are protected against undesired changes, e. g. as a consequence of physical faults or improper intervention ("data integrity");
   - that the execution time of any program does not exceed the specified limit;
   - that the storage capacity provided for every program remains above a specified minimum value.

2.12 The Concept of Fault

To achieve reliable systems, we must know about the sources of faulty behaviour and the various types of fault. As a basis for further investigation the three terms "fault", "failure", and "error" will be defined.

A fault is understood quite generally to be any attribute which adversely affects the reliability of the system [FREY, H., 1977] or, in other words, which might cause it to malfunction.

This definition leaves open the question of
- whether an observed unit with a fault, e. g. a component or a whole system, was already defective at the beginning of the period of observation, and
- whether it is still performing its function.

In contrast, the occurrence (in a dynamic sense) of one or more faults in an observed unit not defective at the beginning of operations is designated a failure when the unit under observation can no longer perform its function.

Error has several meanings and it should be used with care. The definitions applied in this context are [FREY, H., 1977]:
- a human mistake or omission or
- the (fault-caused) difference between the actual and the desired output of a computing system.

The sources of faulty behaviour are very numerous. They can,however, be roughly divided into three classes of faults [HEINER, G., 1979] (cf. fig. 1).

"Logical faults" are faults in the structure realized which result from false reasoning. They can arise during design (e. g. incomplete specifications, the wrong algorithm, wrong circuit) or during implementation (e. g. programming error or faulty wiring).

"Physical faults" are due to uncontrolled physical processes caused by age or the environmental conditions. They can arise during implementation (e. g. cold soldering) or during operation (e. g. failure of a transistor).

"Operation faults" are impermissible and inadequate human interventions in the operation (e. g. wrong input).

Software and operator faults are characteristic of computing systems. The fact that they can be numerically significant is shown by
- the general experience that in the first years of operation after starting up most of the errors are software errors, and
- the experience gained with some special systems where malfunction was mostly caused by improper operation.

Software errors arise due to the incomplete formation of a sequence of conclusions and the incomplete understanding of all the consequences. This insufficiency expresses itself in
- incomplete and/or inconsistent specifications and
- errors in the process of transforming a problem solution into a program.

Various investigations, e. g. [BOEHM, B.W., 1975], [ENDRES, A., 1975], indicate that more than half of the software errors are design errors (specification errors included). Reasons might be:
- the problem has not been understood,
- the designer did not know how to solve the problem,
- the algorithm was wrong or insufficient.

The implementation errors (about 40 %) are mostly due to
- inadequate methods and
- wrong use of methods (especially programming languages).

Please note that we did not regard errors arising during error correction, because their sources are the same as the ones already discussed. However, error correction itself is very error-prone [KOPETZ, H., 1976]: the probability of successful correction of only one instruction at the first go is less than 50 %.

2.13 The Problems of Using Common Terms for Hardware and Software with Regard to the Fault Process

As depicted in fig. 1, software errors are exclusively logical faults made during the design or implementation stage. That means there are no "software failures". Proceeding from the - generally common - assumption that the hardware is not defective when starting up, hardware faults are physical faults caused by failures which occur during use (cf. fig. 2).(From the point of view of its behaviour, a logical fault of the hardware can be equated with a software error.)

A hardware fault either makes itself felt immediately at the time of failure or, in the case of a latent fault, not until a later operating state which arises due to the occurrence of a triggering event (shown in fig. 2 as "malfunction when called on"). A software error does not make itself felt until a faulty instruction is run through or faulty data are used during execution of the program.

In this behaviour - which can be describe as being stochastic in the case of extensive software or real-time systems - the software error resembles a hardware failure.

The probability of the program "running across a fault" which corresponds to the probability of hardware failure depends on
- how many errors there are in the program and data,
- how the errors are distributed in the program and data,
- how the program is run through (therefore the impression of "load-dependence").

Here, however, it is possible to see a basic difference from the hardware. The main random variable for describing the stochastic behaviour is the time of failure for the hardware, and the data flow for the software, or, to say it more precisely: the system state vector which depends on the sequence of the inputs made.

Whereas, for hardware, prediction of reliability data corresponds to the state of the art for quite some time already, due to the fact that reliability models are available and the necessary parameters are generally known, software reliability prediction is at present still very dubious. An estimate can be made by statistical tests and by debugging evaluation which will be discussed later.

2.2 SOFTWARE RELIABILITY

The similarities between the software fault process and the hardware failure process showed that the behaviour of software is not purely deterministic as might be expected theoretically. Because of this apparently stochastic behaviour and because of our experience that generally the behaviour of software is not predictable, the term "software reliability" is justified.

The ambiguous behaviour of software is partly reflected by two basic concepts that make up the notion of software reliability [FREEMAN, P., 1976], [NAUR, R., 1977]: correctness and robustness.

A definition of the term "correct" has already been given: in accordance with the intended specification. The word "intended" points out that meeting the stated specification is not sufficient for correctness if the specification is wrong. The less demanding definition ("in accordance with the stated specification") is widely used and is suitable for verification purposes. However, if there is a specification error and the programs satisfy the faulty specification, the software cannot be regarded as being correct. It could only be denoted as being "relatively correct" with respect to the specification.

Correctness alone is not sufficient; reliable software implies robustness, i. e. the ability to withstand unexpected demands. This implies, for example, that the programs will properly handle inputs out of range or in different format than defined, without degrading the performance of functions not dependent on the non-standard inputs [BOEHM, B.W., et al., 1976].

Though the notion of robustness has become quite popular in recent literature, it should be underlined that it is not a term to be very happy with. The definition is not sufficiently precise and therefore not very helpful for practical purposes. Furthermore it should be attempted to minimize the unexpected demands, e. g. during the specification stage, by making provisions against invalid data inputs.

3. SOFTWARE RELIABILITY MODELS

Until now software reliability has been understood as a common notion implying several properties. It has not been understood as a measure. Quantification is difficult because the concept of software reliability is not sufficiently analogous to that of pure hardware reliability, and reliability measures are not easily translatable. Without going further into this subject, two simple reliability models shall be sketched.

Fig. 3 illustrates the mapping of input data into output data. We will assume that
- there is a systematic dependence of error occurrence on the input data,
- there are neither hardware faults nor operator errors.

If the software is correct, or if the program parts executed for a specific input data set i_C do not contain errors, the output data o_C correspond to the required function

$$o_C = F(i_C)$$

Only for input data sets out of a specific domain E, an error will be encountered

$$o_E \neq F(i_E).$$

A simple model for a software error rate is given by [MACWILLIAMS, W.H., 1973] and [KOPETZ, H.,76]:

$$\lambda = r \cdot \sum_{j=1}^{n} p(i_j) \cdot e(i_j)$$

where: r = input rate (number of operations per time unit)
$p(i_j)$ = probability of occurence of the input data set i_j
$e(i_j)$ = binary error performance for the input data set i_j: 0 if performance is error-free; 1 otherwise
n = number of input data sets.

The error rate can be applied to reliability theory as in hardware considerations. Although the model is quite clear and supports understanding of the problems of software reliability, it remains purely academic since p and e cannot be quantified in practice due to the vast input data domain $I = \{i_1, i_2, \ldots, i_n\}$.

A model which can be used for reliability prediction has been proposed by Shooman [SHOOMAN, M.L., 1973]. The model is based on the debugging effort τ (in terms of debugging time) involved in removing errors (see fig. 4). We will assume that
- no new errors are introduced, and
- the number of machine language instructions I_T is constant.

The error model represents the normalized number of errors ε_r remaining in a program after debugging:

$$\varepsilon_r(\tau) = \frac{E_T - E_d(\tau)}{I_T}$$

where: E_T = number of total errors in a program at the start of debugging
E_d = number of errors debugged:

The time-dependent reliability model

$$R(t) = e^{-c \cdot \varepsilon_r(\tau) t}$$

where: c = constant of proportionality (indicating "instruction processing rate"),

represents the probability of a residual error being encountered during operation of the system. It is assumed that this probability is proportional to the probability that any randomly chosen instruction contains an error.

There are many objections to these models which make their validity very dubious, for example:
- the assumptions are generally too simplified,
- quantitative factors are difficult to determine,
- models based on testing are rarely transferrable from one software project to another, since the testing process is rather undisciplined. These models can be used, however, in advance of a project, to estimate the amount of testing required to achieve a specified reliability goal [MUSA, J. D., 1975]

Finally, software reliability does not depend only on the number of errors remaining in the programs, but on the impact that errors have on the system users.

4. APPROACHES TO SOFTWARE RELIABILITY

There are numerous methods available for meeting reliability requirements and they can be assigned to the two basic approaches of
- fault-avoidance and
- fault-tolerance.

The two approaches are complementary, and the total resources allocated to attain the required reliability may be divided between fault-avoidance and fault-tolerance. Though experience and analysis have shown that a balanced allocation of resources between the two approaches has led to the highest reliability of computing [AVIZIENIS, A., 1975], fault-avoidance has been and is still the predominant choice in software.

4.1 FAULT-AVOIDANCE

Fault-avoidance is the approach in which the reliability is assured by a-priori elimination of faults or the causes of faults, i. e. before the system has been put into the intended operation. Hence fault detection and fault correction during testing are included in the notion of fault avoidance.

Since in practice it is not possible to guarantee the absence of software errors, the goal of fault-avoidance is the reduction of the unreliability to an acceptably low degree.

The methods used to avoid software errors can be grouped into two classes (fig. 5):
- the "constructive" methods which, when applied during the "construction" of the software, are supposed to largely prevent errors and assure good testability,
- the "analytical" methods which validate the intermediate results of a development activity or the implemented programs as the final results, i. e. methods which provide evidence of software reliability.

A high degree of reliability is only achieved by applying both methods.

Please note that figure 5 shows the development process in a very simplified way. Every development step applying a constructive method should be followed by a step applying an analytical method in order to verify the intermediate result. If an error can be detected and corrected at an early development stage, this will cause considerably fewer problems and cost than later on (cost analysis in [BOEHM, B.W., 1977]).

4.11 Constructive Methods

For the reasons stated above, the constructive methods are applied in every project stage of software development. Fig. 6 shows the main steps in the development. As a rule, the course taken will not be as straightforward as indicated, but will also contain backtracks, for example, when the incompleteness of the specification is noticed during the design stage.

The intentions of the future user of the computing system are the starting-point. Almost always, the users have imprecise notions of what they want out of the system.

A systematic approach to the requirements analysis is called requirements engineering [REES, R.K.D., 1977]. However, there is still no solution for the method of expressing the requirements, because common languages like English and computer languages like FORTRAN are not fit for this task.

Deriving a specification out of the requirements is particularly critical since, as a rule, the proof of satisfaction of the specification substitutes the proof of correctness. On the other hand, specification errors (incompleteness, inconsistency, imprecision) are especially frequent and probably cannot be avoided in the case of new developments.
These errors can be reduced and the validation facilitated by using a uniform formalized specification language on every program level [PARNAS, D.L., 1972]. For the purpose of, for example, consistency and completeness checking certain languages have been created in machine analyzable form [REES, R.K.D., 1977].

Errors in the design phase are often due to gaps in the delimitation of the task areas covered by a team of programmers. The team should be structured on the concept of the "chief-programmer team", and the individual members should be assigned clearly defined tasks [BENSON, J.P., 1973].
The program itself should be designed "top-down" by successively refining the system and grouping functions into modules.
Both strategies, top-down design and chief-programmer team, can be regarded as aspects of hierarchical structuring and of modularization, i. e. decomposition for reducing complexity.

The method of "structured programming" has proved a success in implementation. With this method, tree-like structures are produced which consist of blocks having only one entry and one exit. Such structures can be produced with less susceptibility to errors and are easier to check. Fig. 7 shows the three basic elements of structured programming: sequence, alternatives and iteration, and with them every program can be written if construction of subprograms is allowed. A program structured in such a way is GOTO-free and thus easier to unterstand because the order in which its instructions are written corresponds to the order in which the instructions are executed during the computing process [DAHL, O.J., et al., 1972].

To a great extent, modern programming languages such as PL/I and PASCAL contain language elements which permit structured programming [JENSEN, K. and WIRTH, N., 1975]. In addition there are also processors, e. g. for FORTRAN, by means of which structured programming is possible.

Ehrenberger and Taylor have described guidelines for designing and constructing safety related user programs [EHRENBERGER, W. and TAYLOR, J.R., 1977]. Boehm provides a checklist which can be used as a guideline for programming, and compliance can be checked automatically [BOEHM, B.W. et al., 1976].

High level programming languages facilitate reliable programs on account of the following features:
- they are more problem-oriented, they allow use of mathematical notation and thereby reduce errors arising in the process of transforming a problem solution into a program;
- one statement will replace many assembler instructions, and therefore it is less error-prone as experiences have shown [CORBATO, F.J., 1969];
- the variety of data types in high level languages enables the compiler to consistency checking [ACM SIGPLAN, 1977].

The programmers themselves do the debugging, starting at the lowest level ("bottom-up"). Here the most important aid is an efficient compiler which analyses the syntax and the structure.

A comprehensive treatment of the constructive methods (and of testing as well) is given in [MYERS, G.J., 1976].

4.12 Analytical Methods

The analytical methods are primarily concerned with the validation of the reliability of the software system or of individual programsafter they are written and debugged.

This validation can be carried out in three basically different ways (cf. fig. 8):
- proving
- static analysis
- dynamic analysis (testing).

Proving (program verification): [ELSPAS, B. et al., 1972], [RAMAMOORTHY, C.V. et al., 1974], [LONDON, R.L., 1975], [ENDRES, A., 1977], [KRÜGER, G. and NEHMER, J., 1977]

Proving provides evidence on the basis of strictly mathematical methods that a program for all the input data satisfying the input specification
- will terminate and
- will compute output data which are specified functions of the input data.

Thus proving provides abstract evidence that an overall problem solution has been obtained.

Disadvantages are:
- the method is at present only fit for small programs with low complexity,
- the difficulty of providing a complete formal specification,
- the extent of the proof, which often takes longer than the program to be verified (remedy: automatic proof).

Static Analysis [RAMAMOORTHY, C.V. et al., 1974], [GOODENOUGH, J.B. and GERHART, S.L., 1975], [HUANG, J.C., 1975], [MYERS, G.J., 1976], [ENDRES, A., 1977], [GEIGER, W. and VOGES, U., 1977], [KRÜGER, G. and NEHMER, J., 1977], [OKROY, K. and KERSKEN, M., 1977], [EHRENBERGER, W., 1978]

The static (program) analysis examines the structure of the program and the actions the program is to perform. This is done by an analysis of the source code (manually or automatically) without the program being run.

The aim of the analysis is:
- to uncover semantic and structural errors which, for instance, have not yet been detected by the compiler, and, in particular,
- to select the test data for a minimum number of test runs,

since static program analysis as the sole means of validation does not suffice.
Like proving, static analysis benefits from applying the techniques of structured programming.

Dynamic Analysis (Testing)

Dynamic analysis is the process of software testing consisting of driving the program with the devised test data, evaluating the outputs and comparing them with the results to be expected on the basis of the specification (cf. fig. 8).

The crucial point in testing is the selection of test data.
There are two distinct approaches:

- Systematic testing (path testing) [RAMAMOORTHY, C.V. et al., 1974], [HUANG, J.C., 1975], [MYERS, G.J., 1976], [GEIGER, W. and VOGES, U., 1977], [EHRENBERGER, W., 1978], [TAYLOR, J.R. and VOGES, U., 1978]
  uses systematically derived test data (as a rule by means of a static analysis).
  The test data are selected in such a way that, for example, all the program paths or all the branchings and instructions respectively are covered at least once.
  Nevertheless some kinds of errors cannot be detected:
  - errors which do not falsify the correct result with certain test data,
  - errors causing the omission of program paths.

Consequently, attempts are made to supplement systematic testing by

- statistical testing [EHRENBERGER, W., 1978], [TAYLOR, J.R. and VOGES, U., 1978]
  in which case the test data are random data.
  The success of this method does not depend on a preceding structural analysis, but it depends all the more on the validity of certain assumptions of error distribution within the programs.
  Further problems are the large amount of testing required and the difference between the input pattern during testing and during actual use.

In contrast with proving, testing provides concrete evidence that at least a partial problem solution has been obtained.
Both methods are complementary in that the strengths and weaknesses can be mutually compensated for.

Joint use of complementary methods like
proving and testing or
systematic and statistical testing
seems to be the most promising approach to validation.

4.2 FAULT-TOLERANCE

After this short survey of the constructive and analytical methods applied to software development, we must state that there is no way of ensuring that a fault-avoidance approach will completely succeed in preventing the occurrence of errors.

The second, complementary approach is termed the "fault-tolerance" approach, where the reliability is ensured by the use of protective redundancy. In software, the redundancy required is not a simple replication of programs, since software errors, which are purely logical errors, would be present in the copies, too. In order to cope with logical errors, diversity must be provided, i. e. different means of performing the required functions.

Two fundamental redundancy principles, static and dynamic redundancy [AVIZIENIS, A., 1978], which are well known in hardware have their counterparts in software.

Static Redundancy (Masking)

In this case, multiple independently generated programs are operated concurrently, possibly on different computers. Comparison or majority voting is employed to detect or correct errors (fig. 9).

Two methods can be distinguished:
- accomplishing the same task by distinct and "independent" programming teams ("multiple" or "n-version programming") [CHENG, L. and AVIZIENIS, A., 1978] and
- applying fundamentally different solutions for the same task ("diverse programming")

Problems are:
- there is no knowledge of the extent to which common mode errors will arise,
- voting or comparison must be accomplished by verified software or (for safety related application) fail-safe hardware,
- synchronization of the output signals is necessary,
- results obtained by diverse numerical calculations can deviate.

Dynamic Redundancy (Error Detection and Recovery)

In the case of dynamic redundancy, the principle of standby-redundancy is transferred to software. Randell has proposed a program structure, known as "recovery block", for error detection and recovery (fig. 10) [RANDELL, B., 1975], [HECHT, H., 1976], [ANDERSON, T., 1977].

A recovery block consists of a conventional procedure which is provided with a means of error detection (a programmed acceptance test) and one or more standby spares, the additional alternates (prodedures 2 up to n in fig. 10). All of the alternates are diversely programmed statement lists. The recovery block is executed by performing each alternate in turn, starting with the conventional procedure (no. 1), until for some alternate the acceptance test is satisfied. However, if the last alternate fails to pass the acceptance test, then the entire recovery block is regarded as having failed. Recovery is then attempted at the level of the software module in which the failed recovery block is embedded.

As in the case of static redundancy the problem of common mode errors remains. Furthermore, the acceptance test may be quite difficult to accomplish and may become a reliability problem itself (though the example of applying the recovery block technique to a sorting program is very evident and is quoted with pleasure).
However, the recovery block can be an efficient and inexpensive tool for attaining a partial fault-tolerant ("gracefully degraded") solution which might be adequate for a wide number of applications.

5. CONCLUSION

In the preceding survey of software reliability techniques, numerous methods for meeting reliability requirements have been outlined.

Although there are many promising techniques, many problems still have to be solved.
A major cause seems to be that the development of reliability theory and practice hardly keeps up with the rapidly increasing use of software and with the still growing complexity of computing systems. Furthermore, there are too many academic solutions with a too limited range of use.

Significant individual problems are:
- Most system specifications are incomplete, inconsistent or just plain wrong.
- There is a lack of standards for design and documentation. Concepts are available, but deserve broader application: [KATZENELSON, J., 1971], [EHRENBERGER, W. and TAYLOR, J.R., 1977], [LAUBER, R., et al., 1978].
- The validation methods are not yet adequate for being as widely applied as they should be.

The lack of knowledge and experience in the field of software contributes to the fact that an integral treatment of system reliability which includes the hardware, software and human factors has not been achieved until now.

6. REFERENCES

ACM SIGPLAN, 1977, "Selected Papers from the ACM Conference on Language Design for Reliable Software",Comm. ACM 20,8, pp. 539-595

ANDERSON, T., 1977, "Software Fault-Tolerance: A System Supporting Fault-Tolerant Software", INFOTECH State of the Art Report on Software Reliability, Vol. 2, Maidenhead (UK), pp. 1-14

AVIZIENIS, A., 1975, "Architecture of Fault-Tolerant Computing Systems", Digest of Papers FTC-5, Paris, pp. 3-16

AVIZIENIS, A., 1978, "Computer Systems Reliability: An Overview", INFOTECH State of the Art Report on System Reliability and Integrity, Vol. 2, Maidenhead (UK), pp. 23-35

BENSON, J.P., 1973, "Structured Programming Techniques", IEEE Symp. Computer Software Reliability, New York, pp. 143-147

BOEHM, B.W., 1977, "Software Requirements and Design Aids", INFOTECH State of the Art Report on Software Reliability, Vol. 2, Maidenhead (UK), pp. 30-48

BOEHM, B.W., BROWN, J.R. and LIPOW, M., 1976, "Quantitative Evaluation of Software Quality", Int. Conf. Software Engineering, San Francisco, pp. 592-605

CHENG, L. and AVIZIENIS, A., 1978, "N-Version Programming: A Fault-Tolerance Approach to Reliability of Software Operation", Digest of Papers FTCS-8, Toulouse, pp. 3-9

CORBATO, F.J., 1969, "PL/I as a Tool for System Programming", Datamation 15

DAHL, O.J., DIJKSTRA, E.W. and HOARE, C.A., 1972, "Structured Programming", Academic Press, New York/London

EHRENBERGER, W., 1978, "Systematische und statistische Verfahren zur Gewinnung von Zuverlässigkeitskenngrößen für Programme", VDI-Berichte Nr. 307, Düsseldorf, pp. 71-77

EHRENBERGER, W. and TAYLOR, J.R., 1977, "Recommendation for the Design and Construction of Safety Related User Programs", Regelungstechnik 2, pp. 46-53

ELSPAS, B., LEVITT, K.N., WALDINGER, R.J. and WAKSMAN, A., 1972, "An Assessment of Techniques for Proving Program Corectness", Comp. Surv. 4, 2, pp. 97-147

ENDRES, A., 1975, "An Analysis of Errors and their Causes in System Programs", Int. Conf. Reliable Software, Los Angeles, pp. 327-336

ENDRES, A., 1977, "Analyse und Verifikation von Programmen", Oldenbourg Verlag, München/Wien

FREEMAN, P., 1976, "Software Reliability and Design: A Survey", Design Automation Conf., San Francisco, pp. 484-494

FREY, H. (Ed.), 1977, "Glossary of Terms and Definitions Related to the Safety of Industrial Computer Systems", Purdue Europe TC7, Working Paper No. 132

GEIGER, W. and VOGES, U., 1977, "Eine Prüfstrategie für sicherheitsrelevante Prozeßrechner-Software", Informatik Fachberichte, Fachtagung Prozeßrechner, Springer Verlag, Berlin, pp. 479-488

GOLDBERG, J., 1975, "A Survey of the Design and Analysis of Fault-Tolerant Computers", Reliability and Fault Tree Analysis, SIAM, Philadelphia, pp. 687-731

GOODENOUGH, J.B.,and GERHART, S.L., 1975, "Toward a Theory of Test Data Selection", IEEE Trans. Software Engineering 1,2, pp. 156-173

HECHT, H., 1976, "Fault-Tolerant Software for Real-Time Applications", Comp. Surv. 8,4, pp. 391-407

HEINER, G., 1979, "Zuverlässigkeitsprobleme bei Rechensystemen", Zuverlässigkeit von Rechensystemen, Fachberichte und Referate,Band 9, Oldenbourg Verlag, München/Wien, pp. 43-70

HUANG, J.C., 1975, "An Approach to Program Testing", Comp. Surv. 7,3, pp. 113-128

JENSEN, K. and WIRTH, N.,1975, "PASCAL User Manual and Report", Springer Verlag, New York/Heidelberg/Berlin

KATZENELSON, J., 1971, "Documentation and the Management of a Software Project - A Case Study", Software Practice and Experience 1, pp. 147-157

KOPETZ, H., 1976, "Softwarezuverlässigkeit", Hanser Verlag, München

KRÜGER, G. and NEHMER, J., 1977, "Methoden zur Steigerung der Zuverlässigkeit von PR-Systemen", Prozeßautomatisierung im Wandel der Zeit, Fachberichte Messen-Steuern-Regeln, Band 1, Springer Verlag, Berlin, pp. 504-539

LAUBER, R., KONAKOWSKY, R. and REINSHAGEN, K.-P., 1978, "Structured Documentation Method for Safety-Related Computer Controlled Systems", Preprints 7th IFAC World Congress, Helsinki, pp. 739-746

LONDON, R.L., 1975, "A View of Program Verification", Int. Conf. Reliable Software, Los Angeles, pp. 534-545

MAC WILLIAMS, W.H., 1973, "Reliability of Large Real-Time Control Software Systems", IEEE Symp. Computer Software Reliability, New York, pp. 1-6

MUSA, J.D., 1975, "A Theory of Software Reliability and Its Application", IEEE Trans. Software Engineering 1, 3, pp. 312-327

MYERS, G.J., 1976, "Software Reliability - Principles and Practices",Wiley, New York

NAUR, R., 1977,"Software Reliability", INFOTECH State of the Art Report on Software Reliability, Vol. 2, Maidenhead (UK), pp. 243-252

OKROY, K. and KERSKEN, M., 1977, "Anwendung und Automatisierung der Analyse von Prozeßrechnerprogrammen", Informatik-Fachberichte, Fachtagung Prozeßrechner, Springer Verlag, Berlin, pp. 312-323

PARNAS, D.L., 1972, "A Technique for Software Module Specification with Examples", Comm. ACM 15, pp. 330-336

RAMAMOORTHY, C.V., CHEUNG, R.C. and KIM, K.H., 1974 "Reliability and Integrity of Large Computer Programs", Lecture Notes in Computer Science, Vol. 12, Springer Verlag, Berlin, pp. 86-161

RANDELL, B., 1975, "System Structure for Software Fault Tolerance", Int. Conf. Reliable Software, Los Angeles, pp. 437-449

REES, R.K.D. (Ed.), 1977, "Software Reliability", INFOTECH State of the Art Report, Vol. 1: analysis and bibliography, Maidenhead (UK)

SHOOMAN, M.L., 1973, "Operational Testing and Software Reliability Estimation during Program Development", IEEE Symp. Computer Software Reliability, New York, pp. 51-57

TAYLOR, J.R. and VOGES, U., 1978, "Use of Complementary Methods to Validate Safety Related Software Systems", Preprints 7th IFAC World Congress, Helsinki, pp. 731-737

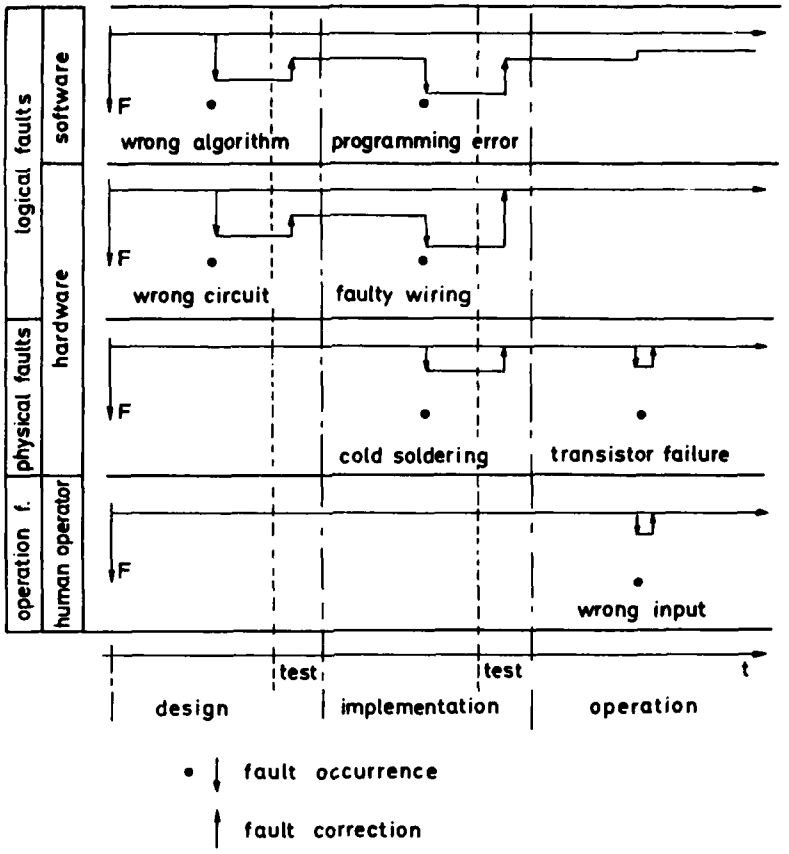


Fig.1 Fault classes

	hardware (no faults on start - up)	software
fault class	physical fault	logical fault
fault occurs ...	during use	during elaboration
fault becomes effective ...	immediately at the time of failure or as malfunction when called on	as malfunction when called on
main random variable for describing stochastic behaviour	time of failure	data flow (system state vector)
prediction of reliability data	state of the art	?

Fig.2 Comparison of hardware and software with regard to the fault process

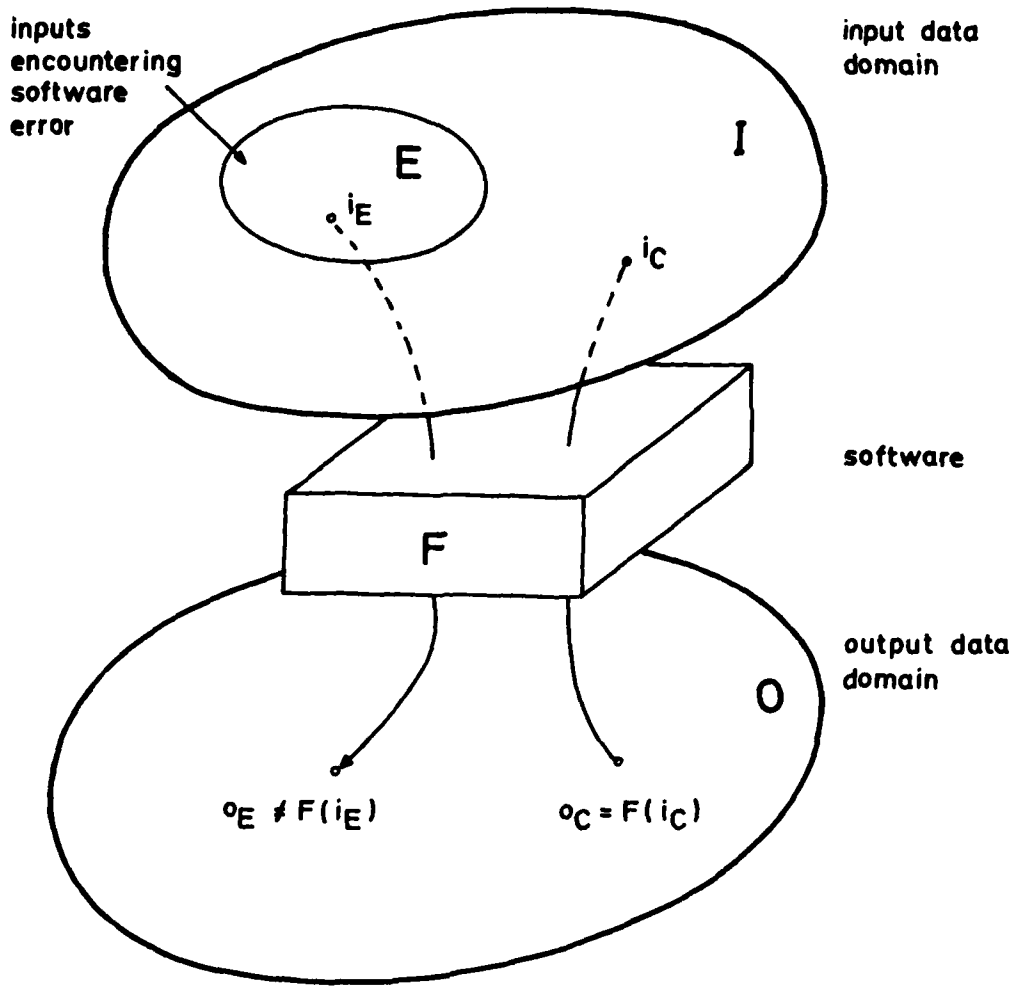
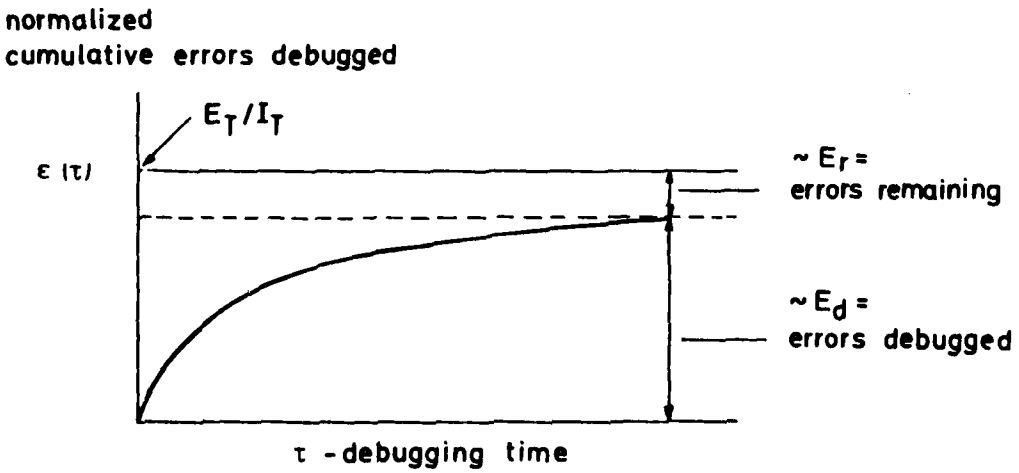


Fig.3 Software error model



$$\varepsilon_r(\tau) = \frac{E_T - E_d(\tau)}{I_T}$$

Fig.4 Cumulative errors debugged versus debugging time (Shooman)

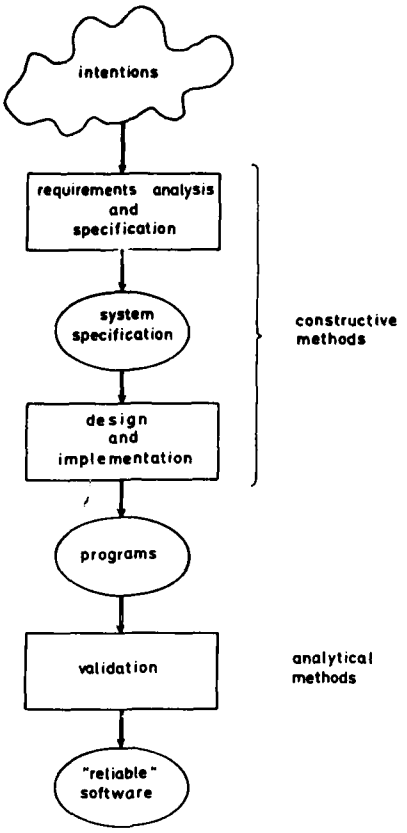


Fig.5 Reliable software development (very simplified)

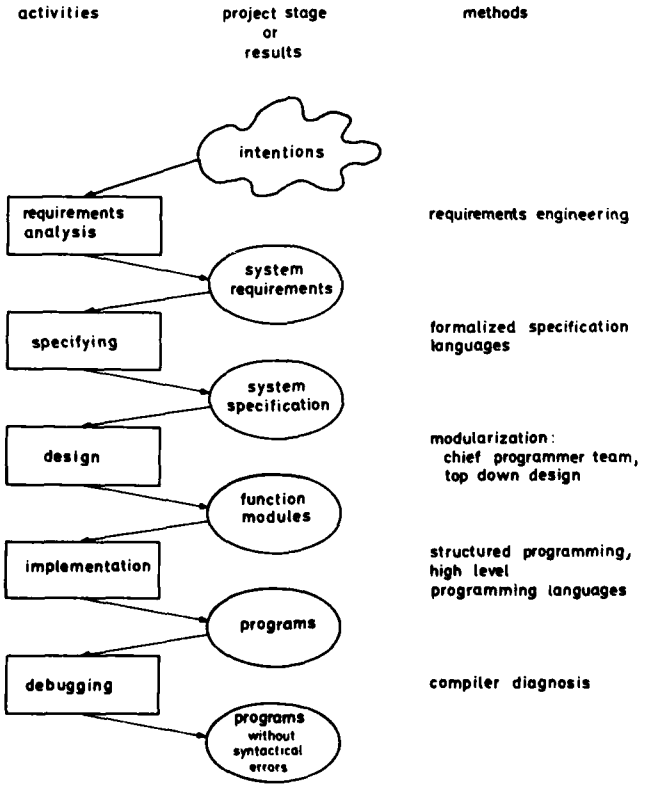


Fig.6 Constructive methods in reliable software development

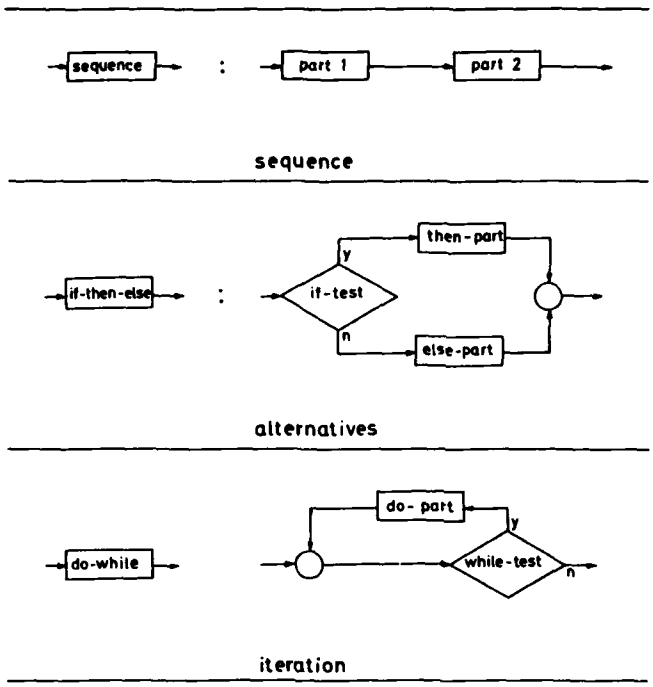


Fig.7 Basic elements of structured programming

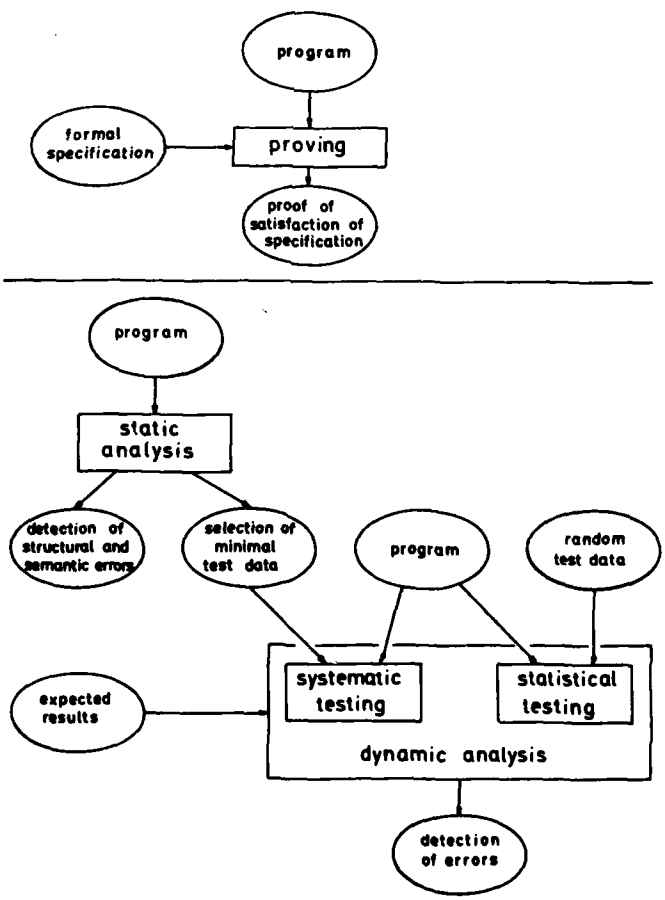


Fig.8 Validation methods

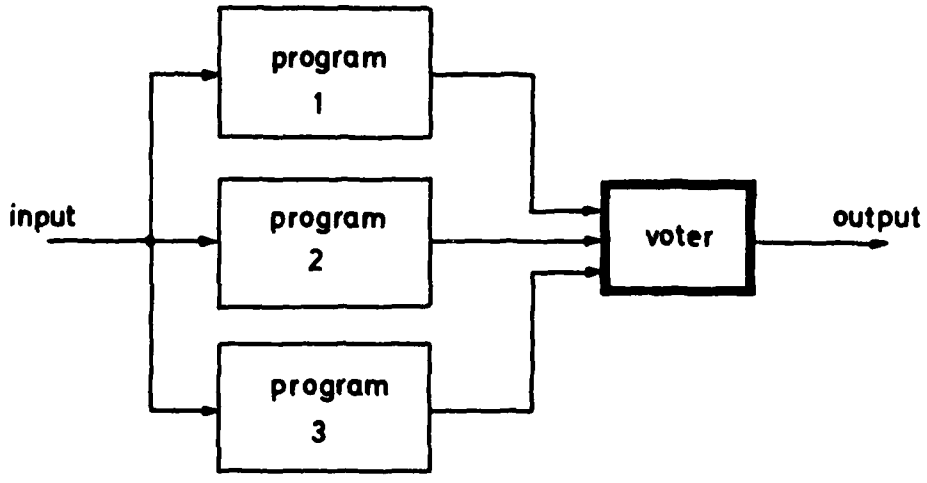


Fig.9 Software fault-tolerance
by diverse 2 out of 3-redundancy

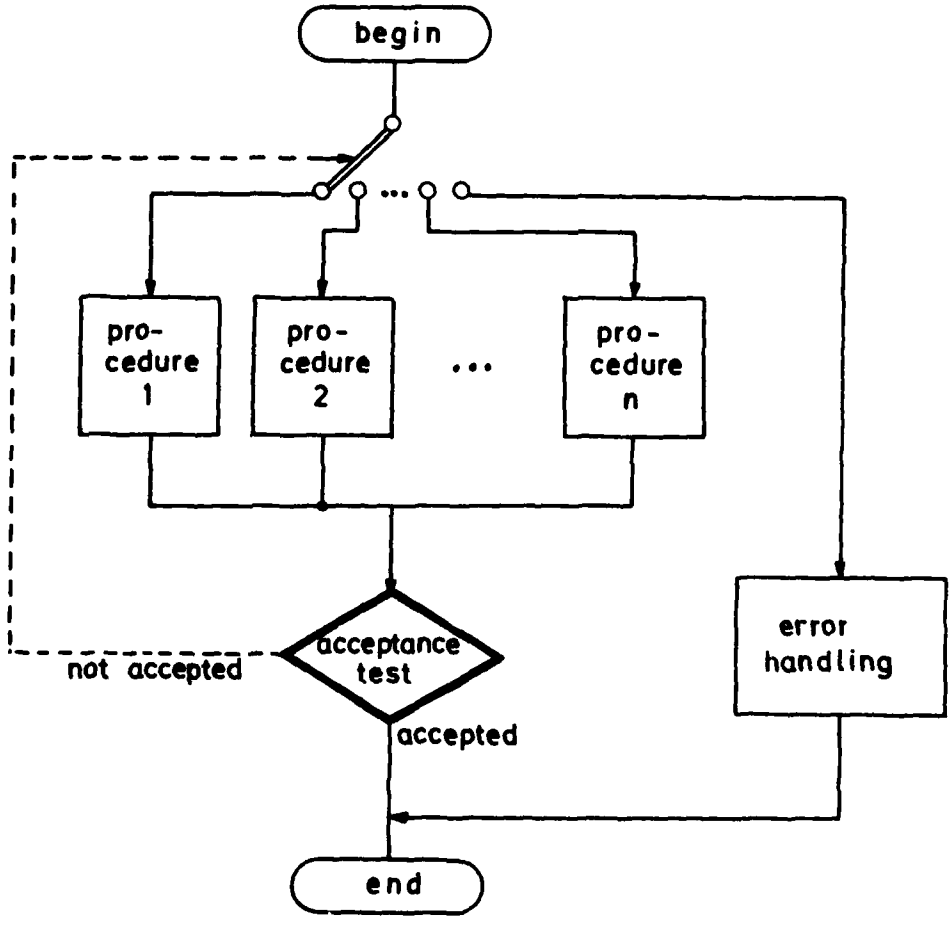


Fig.10 Software fault-tolerance
by error-recovery-block technique

SOFTWARE RELIABILITY - UNDERSTANDING AND IMPROVING IT

L. Mackie

Software Department, Marconi Radar Systems Limited

Chelmsford, Essex, England.

SUMMARY

This paper is in three parts. First, ideas regarding Software Reliability are examined so that we may understand its nature. Definitions are given of Software, its Quality, Errors and Software Reliability. Second, a few realities of Software Development are selected as the most important for consideration here. These are, "The Hardware/Software/People System", Software Requirements Specifications, Software Testing, Detection and Correction of Errors, and Time necessary for Software Development. In the author's view, they lead to a set of individually simple (in principle if not in practice) evolutionary steps which, if taken together, will provide a basis for producing more reliable Software at reduced cost. The third part details these steps which in summary are to write Software Functional Requirements in a rigorous form of natural language, to simplify the problem structure and preserve it in the software, to seek to eliminate and not just improve error-prone processes in Software Development, to design for testing and make tools, to use more hardware to permit all the above, and lastly to allow the time to do the Software Development job properly.

1. DEFINITIONS

The term "Reliability" has meaning for most of us. Something (or someone) that is "reliable" can be depended upon to perform a task,

a) whenever we want it done, and

b) to our satisfaction.

The performance need not be flawless. If it is flawless, human nature being what it is, we will find something to criticize even if it has to be the performer!

Clearly, it would be meaningless to discuss reliability without defining "the task" and what constitute "errors"; and in the case of software reliability, what software is. This is where we must start.

What is Software?

To answer this question, let us consider Software as part of a System (see Fig. 1). The diagram shows People and Hardware reacting to and using Real World situations and information to achieve objectives in a changed Real World. The general capabilities of the Hardware and the People are called upon by specific instructions which must be followed if we intend to achieve the system's objectives. Both sets of instructions are necessary, and each is dependent on the other.

Hence our definition:

"Software is that set of instructions to Hardware and to People using it in the Real World, so that the System as a whole attempts to achieve defined objectives with information (and sometimes, materials)."

Note: The purpose of a new System is to change the existing Real World. How do we predict our future needs, and how well can we do so? These and other points are discussed in Part 2 - "REALITIES OF SOFTWARE DEVELOPMENT" - of this paper.

The Task and Software Quality

British Standard (BS 4778) defines Quality as "The totality of features and characteristics of a product or service that bear on its ability to satisfy a given need". This can be summarized in one word, namely "DUTY". I believe that this definition is completely adequate for software.

To explain this view which may not be widely shared in the Software world, let us compare a Transport Service and an on-line Banking System. The defined Quality of the Transport Service includes the area covered, the period covered, the comfort, in fact all sorts of attributes which are too numerous to mention here. All these are expectations of service. The higher our expectations, provided these are defined (and achievable) or undefined but reasonable, the higher the required Transport Service Quality. Sadly, all too often our expectations are not realized in practice - and the Quality is not achieved.

So also with the on-line Banking System. Is it local, countrywide or international, 5 days/week or non-stop, with a cash withdrawal service time not exceeding 3 minutes, etc? Again there are very many other defined attributes to the service. Its Quality will increase with its Duty.

The efficiency, modularity and other characteristics of the Software may not contribute to meeting the required Quality. Increasing the efficiency often decreases the maintainability (which may be included in the Quality definition). It is the Quality definition that determines whether the above characteristics matter. However, the testability will almost invariably matter.

A priori allowance for human and hardware errors, and particularly the host of exceptions to the normal in the real world, all these add greatly to Software Quality, and especially its cost. The psychological barrier to simplification discourages discrimination in such matters thereby generally leading to unnecessarily high quality (N.B. the physically dispersed detailed definition of the Software Task exacerbates this problem - see Key Point C2 and the paragraph preceeding it for discussion of why physical dispersion occurs).

Errors

The occurrence of an error will cause failure of an objective which, if considered part of the Software Task, is a failure to achieve Quality. This is an instance of Un-Reliability. We have Unreliable Software of a given Quality, and not Reliable Software of a lesser Quality since our normal expectation is that the error will be corrected. Some errors are permanently tolerable (- in effect, a lower Quality is thereby defined). Others may be totally unacceptable.

Undefined but reasonable expectations are not ruled out of the definition of a system's Quality. There is ample precedent for this attitude in everyday contractual/retailing situations. So why should Software be an exception to this practice? Whether or not it is reasonable to expect reasonable system behaviour when operating outside its specified limits depends on the impact of the unreasonable behaviour (e.g. air traffic control vs. payroll systems).

Software Reliability

The difficulties of defining Quality levels and Error states should illustrate why it is difficult to measure Software Reliability without making some over-simplifications. In my view, it is more important at this time to improve Software Reliability than to measure it with a rather arbitrary yardstick. Nevertheless, Myers (1976) gives a practical definition:

"Software Reliability is the probability that the software will execute* for a particular period of time without a failure, weighted by the cost to the user of each failure encountered."

Reliability is measured and indicates the extent to which the defined Quality has been achieved.

*(My note: i.e. the system user instructions will be used and the software programs will run in the system hardware all operating in the real world).

Thus objectives need to be classified according to cost of failure, the software reliability measured for each class, and then aggregated for the system. In practice it happens, in that frequent trivial failures are more acceptable than infrequent catastrophe.

2. REALITIES OF SOFTWARE DEVELOPMENT

The Hardware/Software/People System

This System has the following important features which are often overlooked:

a) Stochastic components are present with differing stochastic behaviour (i.e. people, repaired hardware, repaired software).

b) It has 2nd and Higher Order effects which may be obscure, slow to manifest, and quite unexpected.

c) The existing Real World before the coming of the System is precisely what we wish the System to change. But how well do we understand the Real World? It is complex and continuously changing, of its own accord and also in anticipation of the new System. Our perception of it necessarily lags the facts.

This difficulty must grow exponentially with the System scope.

d) The System's objectives (i.e. requirements) are intended to satisfy our future needs. In predicting our future needs, I suggest that in practice we use modelling in some form. We use past experience to decide upon our first model which is then "understood" only by exercising it to see how it behaves. It is this feed-back which leads us to change the model, and it takes time.

Our modelling is inevitably incomplete, mainly because we are unable to conceive all the circumstances that the system will encounter in practice. Even some that we do conceive we fail to apply to the model. This leads to implicit system requirements, of two kinds, known as "Human Expectations"; those which are anticipated but unstated, and those which are unanticipated and defined only through hindsight. It may be a matter of luck when the system's behaviour happens to satisfy these human expectations. And where these expectations are violated, they can not always be ignored in the commercial situation of system vendor and purchaser, just because they were not specified in advance. Table 1 shows some typical methods for Modelling.

KEY POINTS

The above features of the Hardware/Software/People System should lead us to recognize that,

A1. To specify all the system requirements before Software Implementation is probably impossible.

A2. As there is an inevitable need to iterate, we must seek ways to improve iteration.

A3. Software Development has been a process with inherently low predictability. In many areas, it still is, and particularly for those engaged in the specialized and complex one-off system, without significant changes in approach, it is likely to remain so.

Software Requirements Specifications

These should cover at least,

a) System Inputs and Outputs, and the Functional Requirements relating them,

b) Performance (Timing, Accuracy, Availability)

c) Loading and Limit conditions, and Over-load actions

d) Hardware environment

etc.

Generally, by far the largest and most complex of these sections will be the Functional Requirements relating the System Inputs and Outputs. This section, the "Functional Requirements", is procedural in nature (i.e. given a Real World Situation, react accordingly; produce this Output from these Inputs in this way). Traditionally this has been written in natural language prose, with all the attendant freedom and ambiguity that prose offers. Diagrammatic/tabular presentation is often included. It is unusual to find any accompanying Glossary of Terms and an Index (not a contents list). Why is this, when we certainly would criticize their absence from a text-book? I think that this is one of the all too frequent instances of the Software Industry ignoring well-established experience.

Consider now the objectives of the Requirements Specification. The obvious one is Definition of the task. But is it not common practice during Software Development for the Requirements Specification also to be used for the introduction of new-comers to the System? At that stage, it is the only document purporting to define what the system has to do.

Thus, by default, it is used for the Induction of people and indeed, sometimes it is written with this second objective in mind. The new-comer is presented with it to read, "... it's all there ..." he is told. Analysis of these two objectives and the methods best suited to their achievement shows very simply why, in general, Requirements Specifications fail in both objectives.

DEFINITION (of TASK)		INDUCTION (of PEOPLE)	
Needs to be	* COMPLETE	Needs to be	* PEOPLE-ORIENTED
	* PRECISE	Aims to	* MOTIVATE
	* CONCISE	and	* EQUIP

Induction of People requires that they should be treated as such. First we should talk with them and show them the project, and not isolate them with a pile of paper. Whilst inducting, one simplifies. Often one will deliberately mis-lead to convey a point. Correction then reinforces the truth. Repetition and various styles of presentation need to be used to suit all the parties involved.

Clearly, what is necessary and desireable for Induction militates against Definition.

KEY POINTS

B1. Separation of Definition and Induction will permit success in each.

B2. Induction needs a separate people-oriented process which should be free from the stringent needs of Definition.

B3. A Requirements Specification aimed solely at Definition will almost certainly fail for Induction, but will succeed with properly inducted people.

Now consider Definition. A procedure involving objects is hardly meaningful without our knowledge of the objects themselves. They are distinguishable by their origin, attributes and the purposes to which they are put. The lack of a glossary (or dictionary) does not necessarily imply absence of such knowledge. But its lack hinders assimilation, understanding and use of such knowledge, and encourages inconsistent referencing to objects.

Failures in Definition come through incompleteness and imprecision, in addition to being precisely wrong! To encourage completeness, a catalogue of the Real World situations and objects - the System Inputs and Outputs - and their states, provides a necessary check-list. The blockage to production of such a check-list is, I believe, psychological. Passive objects by themselves are far less interesting than our own activity in their manipulation.

Iteration during understanding of the paper model is helped by Conciseness in Definition, and hindered if Conciseness is taken to an extreme. Clearly, there is an optimum to be found.

What happens to gaps in the Definition? Gaps get filled by decisions which are, at one extreme, by agreement between the Customer and Supplier's authorized representatives, or at the other extreme by default - to do nothing; or between these extremes, for example, undisclosed and taken unilaterally by a programmer. I am, of course, discussing what the System's behaviour is, and not how it achieves that behaviour. Regardless of whether the behaviour is a contractual requirement or at the Supplier's discretion, and acceptable or not, it is very rare for the full definition of the System's behaviour ever to be accumulated in the Requirements Specification. I suggest that the main reason for this is the inherent difficulty involved which is anyway greatly exacerbated by the amorphous nature of prose. Thus, much of the System's behaviour is not discovered until it has been built and extensively used.

KEY POINTS

The use of natural language prose for Definition of the Functional Requirements precludes,

C1. any significant improvement in achievement of Definition before Software Implementation.

C2. complete aggregation of unfolding Definition during Implementation.

C3. automatic conversion of the Definition into Hardware instructions (e.g. executable program code).

Software Testing

This is not a trivial task. It is simply not practical to test all the possible System behaviour before real-life operation because of,

a) the sheer bulk of effort involved and,

b) the need for real-life operation to enable certain situations and user expectations to manifest.

In the Software Industry, there is no agreed principle for testing. The extremes of "top-down" and "bottom-up" together with various hybrids all have their advocates. Even the purpose of Software Testing is disputed. To some, it is to verify achievement of software quality. Whilst to others, it is to prove the very opposite by the discovery of errors. Each argument is valid from very practical viewpoints. The Customer and Supplier both need to know that the Software works, whereas those responsible for achieving error-free software will tend to fail to discover errors if they aim to demonstrate their absence.

Cure or Prevention of Errors? One fact is clear, namely that as the System scope increases, so also does the proportion of its possible behaviour that will necessarily remain untested before real-life operation. The principle of "Cure" is not relevant to such untested software. "Prevention" is the only alternative available to use for improving its reliability. Since the rate of testing strongly depends upon the extent to which the software works, then improvements in Error Prevention will markedly increased our ability to test more software. Study of the Software Development process reveals many error-prone practices which today are simply a legacy of the past. Good reasons can be given to explain why the present situation exists. But I find it difficult to find reasons as to why it necessarily must persist.

Testability of Systems. Regardless of how the set of tests is decided, the System Testability plays a crucial part in their application. "Testability" is a term here embracing our ability to CREATE the test data, apply the test REPRODUCIBLY, RECORD the results and DIAGNOSE any faults that may arise. It is instructive to examine Software Systems under development for these attributes.

For example, with ever increasing use of volatile output (i.e. displays), just how recordable is this and what design features (hardware/software) are embodied to allow recording of display output? And consider Diagnosis. For hardware one can use probes generally of sufficient discrimination and data rate to be practical.

For software, particularly in real-time systems as commonly designed, there is often no practical method of extracting the required diagnostic information and yet continue to run at full-speed. The alternative of slowing the system down may not be possible without severely altering its logical behaviour.

Improvement of Reliability (i.e. achievement of Quality) is made difficult and sometimes impossible by poor testability. So can we improve testability easily?

The lack of testability in many systems is not because the necessary design features are unknown technology. I believe it is because they are simply overlooked. Many of the hardware and software tools are available. Others we are capable now of building. It is not always clear why restrictive and labour intensive methods are chosen in preference to more cost-effective methods. I suggest that this is yet another example of the Software Industry ignoring other well-established experience.

Detection and Correction of Errors

The relationship between consecutive stages of a development project is that of Specifier and Implementor. A specification error will be an error of intent. Only when the Implementor has incurred cost, either in analysing the task in depth or implementing it as specified, will the potential/real result be observed and recognized as a specification error. But the onus of proof is on the Implementor to demonstrate the absence of implementation errors. Such proof therefore increases the Implementor's costs beyond those incurred in his or the Specifier's recognition of the error.

An ever increasing body of information evolves as development proceeds. The sum total ramifications of a decision made at one stage therefore grows as successive stages are passed. Because of the "fan-out-phenomenon", and interactions between decisions themselves, the growth rate itself increases.

For the above reasons, the cost of error correction grows by orders of magnitude the later the stage of development in which it is detected and corrected.

Time Necessary for Software Development

Commerce/Industry at large is prone to unrealistic delivery time-scales and the Software Development Industry is notoriously no exception. For a Software Development Project consider the relationship between the variables, TIME, QUALITY and RELIABILITY. Obviously they are related, but constraints are placed on them individually. For TIME, "We need the system by ...", for QUALITY, "The System has to do ... , last for n years ... , and we'll be maintaining it ...", and for RELIABILITY, "it must not crash more than once a day".

The constraints themselves are related in only a very loose way, or not at all. The commercial/political function relating them is hardly mathematical, or if it is, then it is not readily comprehensible. Thus it is not surprising that parties to the project are prone to implying an unreal relationship between Time, Quality, and Reliability (i.e. through incompatible constraints).

In Software Development, the most important job to have time to be done properly, is the first - Defining the Requirement. Traditionally, a 30 month project time-scale will be split 6 : 15 : 9 to Design, Implement and Commission. The "Design" phase will have perhaps 2 months strictly devoted to spelling out the Requirement. This is a well proven recipe for disaster. It is simply not long enough,

a) to get at the detail - even once, let alone to iterate

b) for the Customer and the Supplier to appreciate fully the implications of the job

c) to structure the job to allow a Quality/Time trade-off.

Why not trade-off the Reliability? Our experiences in daily life tell us to go for less that works, rather than more that doesn't. To go for Reliability forces us, rightly in my view, into a Quality/Time trade-off, always remembering that the Time required for a given Quality depends upon the degree of automation involved.

Software Development short-cuts are full of boomerangs. Hurl problems away and sure enough, they'll come back.

---

3. SOME RECOMMENDATIONS FOR IMPROVING SOFTWARE RELIABILITY

3.1 Software Requirements Specifications

The consequences of the current, natural language approach have already been examined. Thus I believe it is essential for the Functional Requirements to be defined using a formal language which must not be restricted in the range of systems/applications for which it is suitable.

The language must be able to accommodate ever-changing perceptions of the real world (e.g. hierarchical in detail view, discrete and continuously variable properties, etc.).

System Description Languages satisfying some of the above criteria have been in use for some years, now on an increasingly widespread basis. Some are computer-based. Others are being developed so that a purely conceptual model (of the system being specified) can be built, free form any constraints of the implementation solution or indeed, of the system description language itself.

The recommendation can be implemented in two stages. First, go formal. Second, go computer-based.

3.2 Eliminate unnecessary error-prone processes

The generality of this appears to be about as useful as "being against sin". However, this is not so. If the principle is applied with determination, we discover very many examples of unnecessary practice lingering on. Some may illustrate this point sufficiently.

Programming Languages and Data Types. In progressing from early High Level Languages, the ability was developed to declare a data item to have a type such as "pence" or "day-in-the-year", instead of using the general numeric type, INTEGER. This additional specification can therefore be used so that only data of like type can participate in certain manipulations. Inconsistencies can be readily detected - a useful feature. Those less familiar with software will recognize this idea simply as "making sure that units are consistent".

But what has been left in as an error-prone process in some languages is that the data type etc. declarations for a given piece of data have to be essentially repeated for each and every procedure using that piece of data. Methods exist now where this repetition is avoided, but they are infrequently used.

Control of Concurrent Access to Data in Real-Time Systems. It has long been the practice that programs wishing to access simultaneously shared data are designed to allow in some way for conflict between reading and writing of the data. The allowance is sometimes difficult to consider comprehensively (i.e. in all its ramifications) and is certainly error-prone.

3.3 Design for Testing and Develop Tools

This has already been considered somewhat in Part 2 of this paper. The financial justification for this is obvious, particularly with Systems Testing in the case of a Manufacturer/Supplier of Hardware-Software Systems. Analysis of what is involved in improving systems Testability shows that often, comparatively small investment in test hardware and software will pay for itself many times over in the saving of commissioning effort and elapsed time (and therefore the financing costs).

3.4 Minimize Structural Complexity. Preserve the Problem Structure

Ever-increasing complexity is a major difficulty facing the Software Industry. Some of it is inescapable with Hardware/Software/People Systems. Some of it is induced by Users' increasing expectations of such systems. Other complexity is self-induced by Software Development; reasons are found for structuring the Implementation solution differently from that of the Requirement. Often, the reasons have no commercial justification when analysed in a context broader than the purely technical.

So complexity is immediately introduced with the correlation of the two structures. If more hardware is necessary to preserve the problem structure, the ever-decreasing ratio of hardware to software costs provides ample justification. If physical constraints at first sight rule out more hardware, the reaction should be to look away from the software for a solution (i.e. even return to the problem or look again at the hardware).

3.5 Use more hardware as necessary

This should at least be an attitude of mind, if not a matter of practice, both for the Implemented System and for its development. It is a necessary step for aspects of some of the above recommendations t. be feasible. If it is adopted as fully as possible, I believe it will permit the fundamental improvement to Software Development that is widely held to be necessary. It is bound to be commercially justified.

3.6 Allow Time to do the Job Properly

The major break-through necessary to implement this recommendation is the acceptance by parties to a Software Development Project that generally, they do not know the TIME - QUALITY - RELIABILITY relationship. This is true of Customers, not just Suppliers.

SUMMARY OF RECOMMENDATIONS

D1. Adopt a formal jargon-free language for the Functional Requirements aspect of Software Requirements Specifications to achieve definition and clarity.

D2. Eliminate unnecessary error-prone processes.

D3. Design for Testing and develop tools.

D4. Minimize structural complexity. Preserve the problem structure.

D5. Use more hardware as necessary for all the above.

D6. Allow time to do the job properly.

. .. '.', "Software Reliability : Principles & Practices", Wiley - Interscience, New York, U.S.A.

SOME TYPICAL METHODS FOR MODELLING
(illustrative, not definitive)

	FORM of MODEL	METHOD of EXERCISING IT
PAPER	Mental sketch of system component(s) + Rough notes	*Thinking and discussing. *Rudimentary recording of circumstances postulated. *Conclusions memorized.
	Requirements Specification in free-format conventional prose with diagrams	plus *Reading, drawing inference (possible wrong). *Majority of circumstances postulated with conclusions embodied in the specification (sometimes ambiguously).
	Formal System Description Language	plus *Tracing through clearly defined paths. *Actual behaviour explicity shown. *Possible automatic detection of inconsistencies and omissions (ONLY with respect to what has been included).
HARDWARE	Prototype	*Manual/Automatic stimuli in an environment close to/far from realistic operation. *Observation, with possible automatic recording of results for performance analysis
	Full Scale Operational System	THIS IS INCLUDED HERE BECAUSE: Typically, 50% of Software Life Cycle Cost is so-called "Maintenance" which includes minor enhancements and even major changes, not just correction of implementation errors.

Table 1

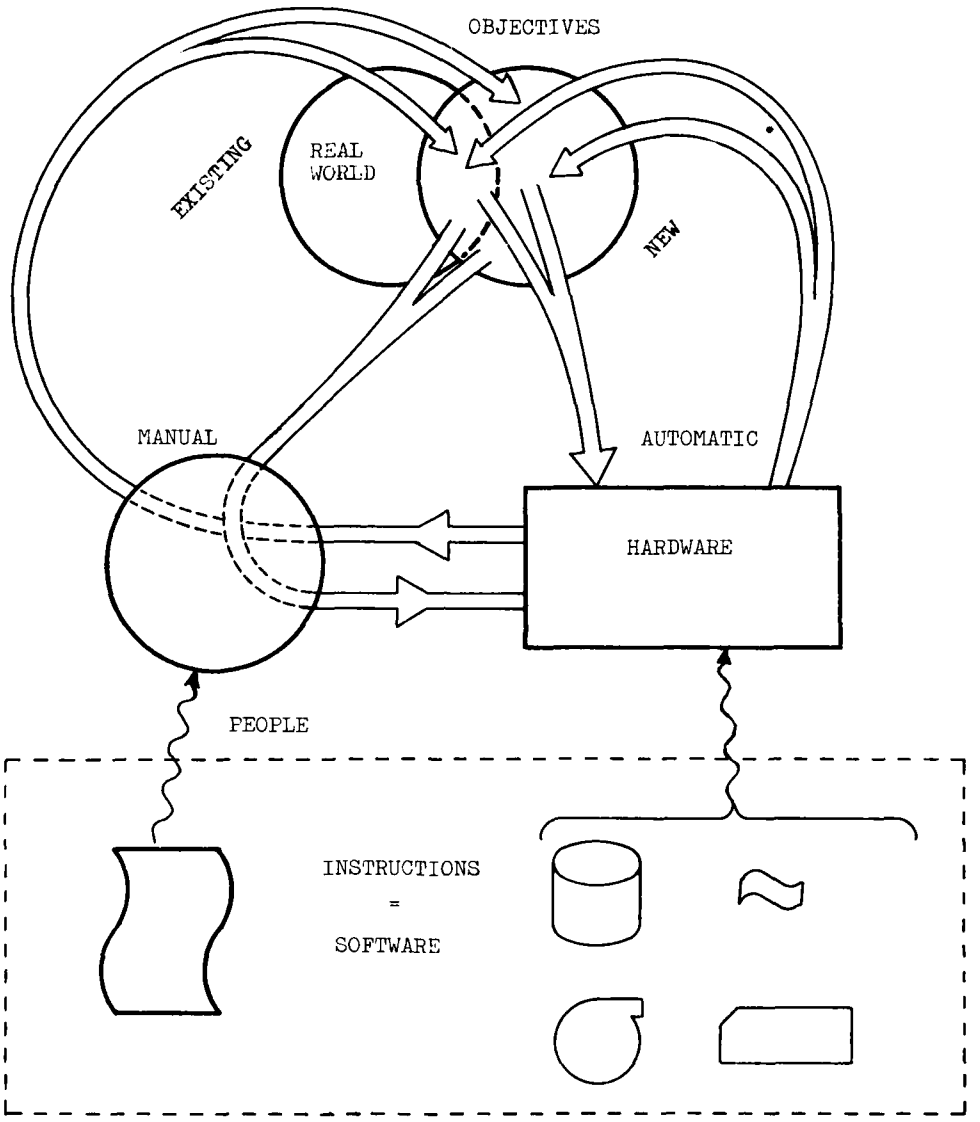


Figure 1 The Hardware/Software/Pecple System

DISCUSSION

T.L.Regulinski, US

With reference to your 4 d ii) recommendation i.e. to eliminate unnecessary error-prone processes:

Most people would agree that to eliminate unnecessary processes would be relatively easy but it is extremely hard in view of the languages that we use e.g. the nesting generally used in advanced FORTRAN 4 would probably account for upward of 40% of errors experienced, yet to eliminate nesting using FORTRAN 4 is extremely difficult.

On the other hand Pascal has a built-in system. However, I am unaware of any language which is devoid of error-prone processes. An improvement in one area usually means degradation in another, unfortunately.

At the moment an advanced language is being developed in the US by two contractors to lead to one language which is at least numerally deficient of error-prone processes. Your comment please.

Author's Reply

You are quite right that to eliminate some error-prone processes will create others. My essential point is that perhaps you shouldn't be doing what you are. My first slide urged you to look at the software development process as a process and try to understand why things are done, if, in fact, they need to be done at all. In my paper I have been looking at requirement specifications and I tried, through showing the distinction between definition and induction, to see what the implications are, and if we can treat induction as a separate process then we are free to tackle definition in the way which best suits its needs. Therefore you can start to use a formal language. If you recognise that we can legitimately use such a language then you have to look at the requirement specification which is the paper model, and get this into the system. The first thing is to be able to process the formal language, to recognise the syntax and semantics, you therefore have to be capable of processing a dictionary.

Then you must provide an environment around the procedural statement to enable you to put the statement into the system without touching it. A fairly recent approach in the UK states that you have to access data using proper data access methods and their use is built into the procedural statement. So, as a user of the system, if you specify a requirement with this approach you have to take the lid off the box and put in data access methods. They have been designed by a system architect.

There is however an approach we can adopt where you literally take the procedural statement in a formal language and put it into the system.

Formal Methods for Achieving Reliable Software

by Jack Goldberg

Director, Computer Science Laboratory
SRI International
Menlo Park, CA, U.S.A.

SUMMARY

Future requirements for reliable avionic systems will require a level of confidence in program correctness that cannot be achieved by present programming methods, which are based on informal design and program testing. Newly developed formal methods for specification, design and validation offer the prospect of achieving the required level of confidence. We present an example of the SRI Hierarchical Development Methodology, taken from the current SIFT fault-tolerant computer development.

1. Introduction

We will attempt to establish that future requirements for reliable avionic systems will require a major advance in programming methodology in the direction of increased formalism (i.e., mathematical rigor) in specification, design and validation. We will also attempt to establish that a useful degree of formalism will soon be feasible in engineering practice.

In Section 2, we discuss the need for new methods to cope with the complexity of critical real-time systems, in order to achieve high reliability and to reduce life cycle costs.

In Section 3, we present some general concepts about formal methods and give an example of the SRI Hierarchical Development Methodology (HDM) taken from the executive system of the SIFT fault-tolerant computer. In Section 4 we compare formal methods with alternatives and consider the prospects for introducing formal methods into practice.

2. The Need for Greatly Improved Software Methods

2.1 Reliability and cost objectives.

The Federal Aviation Administration requires that failures capable of causing a fatal event in the flight of a commercial aircraft be "extremely improbable." For the flight-critical computer subsystem within the avionic system of such an aircraft, this requirement has been translated into a specification that the probability of failure in a ten-hour flight be less them 10^{-9}. The complexity of future flight-control computations dictate that the computer be digital, and the reliability level of current digital components dictates that the computer have a fault-tolerant organization. Currently, two experimental fault-tolerant digital computers, FTMP[Hopkins 78] and SIFT[Wensley 78], are aiming at that level of reliability assuming that faults occur only in hardware; that is, the software must be perfect! For the present, this assumption is unrealistic, to say the least.

As low-cost hardware becomes more pervasive in avionic systems, there will be a growing need for very high software reliability either to achieve adequate safety or to reduce maintenance costs. In addition, maintaining the software itself over its life-time has become a major cost factor in digital systems, with estimates of 40-70% of initial development costs. Recent work in hardware fault-tolerance has generated good understanding of the problem, which justifies the hope that hardware reliability soon will be achievable to a reasonable level. When that happens software will stand as the single greatest source of system unreliability and cost [Goldberg 73]. Software faults are essentially the results of errors in thought. Unlike hardware, there are no well-understood physical mechanisms that will allow failure prediction. Furthermore there is not the physical continuity of state (such as in physical structures or analog electronic circuits) that might justify the use of a finite set of tests to extrapolate behavior over all likely inputs. Unfortunately for the objectives of reliability and cost, even a modest quantity of software is capable of enormous complexity, given the infinite number of possible sequences of input data and combinations of internal state variables. These characteristics combine to render software not only unreliable, but unpredictable.

2.2 Aspects of Software Complexity

Programs are, intrinsically, one of the most complex of human artifacts. Their intrinsic complexity is compounded by the need for precise communication among those involved in the production of a software system. Broadly, these consist of buyers, specifiers and implementers. Probably the most serious source of trouble in a software system is its specification, which is produced by a specifier in response to a buyer's stated or implied requirements. Specifications, when they exist at all, are notoriously ambiguous, incomplete, and inconsistent. Typically, they are given in the form of natural text, often in fragments, and with major gaps, wherein the program is allowed to serve as its own implicit specification. Clearly, with an incomplete and imprecise specification, the door is open to arbitrary program behavior, for which no objective decision about correctness is possible.

Even in the case where a full and careful natural-text specification is presented to the implementers, their efforts toward a correct implementation may be inhibited by the existence of subtle conditions that can lurk in the huge combinatorial space of typical programs. This is a common problem for ordinary programs, and is even more serious for real-time programs, in which timing relationships among concurrent processes usually introduce additional complexity. Such conditions are very difficult for designers to visualize, and constitute unpredictable hazards for correct operation. Yet another opportunity for error is found in the need for the need for communication among a team of implementers about functions that

Software complexity, both intrinsic and production-induced, makes it virtually impossible to test programs for correctness sufficiently to achieve the levels of reliability quoted previously. The only way to do that is to construct tight logical arguments (e.g., proofs) that have sufficient generality to comprehend all possible input histories. Even for less demanding applications, the cost of finding programming mistakes through testing grows rapidly after the obvious mistakes are found, and becomes a serious obstacle to the construction of large real-time programs.

Program complexity is also responsible for present high software life-cycle costs. Most programs are written in a way that makes it very costly to analyze the effects of making changes. This is compounded by the generally poor state of program description, which makes it costly and error-prone for a programmer (even the original author) to determine what the program was supposed to do in the first place.

To summarize, real-time programs produced by current methods suffer from ambiguous specifications and incorrect implementations. Their complexity makes it virtually impossible to achieve confidence in their correctness by program testing, and makes program maintenance a very costly activity.

3. Formal methods for software development.

3.1 General Concepts.

Over the last five years, researchers have responded to the problem of software complexity by developing programming methods that have a more rigorous mathematical foundation than existing methods. Most of the work has had the long-range goal of making it possible to argue about the correctness of programs within the framework of some mathematical theory. Not surprisingly, the formal methods that have evolved from this work have been found to simplify and to add precision to software, independently of the long-range provability goal.

The key ideas of the new methods are the use of formally-specified abstractions (e.g., abstract data types and abstract computational processes) and hierarchical structure. These correspond directly to methods for dealing with multiple levels of detail in other engineering disciplines, but with much greater rigor, in order to cope with the enormous and complex data space of programs.

The use of abstract-hierarchical specification has four main benefits for program development and verification:

1. it helps a programmer to decompose a design into parts that are small enough so that he can understand them

2. it helps to hide or postpone details of implementation and thus increases the programmmer's ability to control design decisions

3. it helps to set precisely defined interfaces on program segments so as to bound and isolate the effects of program changes

4. it helps to structure and simplify the process of proving that a system's specification and its implementation are consistent.

A variety of formal methods have been developed within this general conceptual framework. In the next section we present a summary of one of these, The Hierarchical Development Methodology (HDM), developed at SRI by L. Robinson[Robinson, Levitt 77, Robinson 77].

3.2 The SRI Hierarchical Development Methodology (HDM)

HDM is intended to be a general-purpose methodology for formal specification, design implementation and verification of software. It consists of

1. a general structural model for digital systems and a procedure for building systems to conform to the model,

2. a set of languages and language processors for use in specifying and implementing system elements and their inter-relations, and

3. an approach for use in proving the consistency of specifications and implementations.

HDM unifies and formalizes ideas initially proposed by Dijkstra [Dijkstra 68] (hierarchical structure and abstraction), Parnas [Parnas 72] (modularity and formal specification) and Floyd [Floyd 67] (program verification).

In HDM a system is realized as a linear hierarchy of abstract machines, sometimes called levels (figure 1). The top level is called the user-interface, and it is composed of functions that are relevant to the user's application. The bottom level is called the primitive machine, and it is usually composed of some standard, available functions whose performance is compatible with system performance needs. These operations are those of the hardware on which the system runs, and/or the constructs of a programming language that are made available to the system developer to hide irrelevant features of the hardware. The remaining levels are called intermediate machines. The nature of the intermediate machines must be determined (i.e., invented) by the designer. At each level, a formal specification describes the functional behavior of a virtual machine (returned values for all input combinations), without reference to the way in which this behavior is accomplished. That is, a formal specification allows the details of implementation to be hidden. Formal specifications thus provide natural interfaces for the efforts of a programming team. Each specification also implies a family of systems having different implementations for the same functions.

The realization of a machine, (illustrated in Figure 2) is a two step process. First, the abstract

data structures of a machine i (i ~= 1) are represented by those of the next lower-level machine, i-1 by means of a mapping from lower-level states to upper-level states. Second, each of the operations of a machine i (i ~= 1) is implemented as a program in terms of the operations of machine i-1. The collection of implementations for all machines excluding the primitive machine constitutes the system implementation. A machine is sometimes decomposed into simpler units called modules. There is no preferred order of design for the intermediate levels. Rather, a set of levels is developed over several stages, proceeding from a set of qualitatively described data structures to a set of precisely defined functions and inter-level mappings.

In HDM the modules that comprise a level are functionally specified using the language SPECIAL[Roubine 77],which is strictly a specification language. The bodies of the modules may be implemented using almost any programming languages. Because of the structure given by specifications and the use of hierarchy,the implementation language may be very simple. The experimental language ILPL has been developed as the prototype of a simple programming language that conforms to the underlying computational model of HDM. ILPL is as yet unimplemented. Other appropriate languages are Modula, Pascal, and Euclid. These are available, and can be used with some minor restrictions. SPECIAL is supported by a set of interactive computer tools. It is in current use by at least five research and development teams in the U.S. and Europe. Some examples of its application (thus far, only for system specification and design) are

- A hardware-fault-tolerant operating system (SIFT)[Wensley 78]

- A provably secure operating system (PSOS)[Robinson 75]

- A real-time tactical operating system (RTOS)[Feiertag 79]

- A flight-control software module [Boebert 77]

The basic method of proof of correctness in HDM consists of the construction of a chain of reasoning, linking all levels, that shows that the abstract program at each level correctly implements the functions specified at the next higher level. Each link in this chain involves two steps, i.e., (1) generation of a set of logical formulas that, if true, imply the consistency of a program segment and its specification, and (2) proof of the truth of these formulas. Due to the large number of formulas that are required to prove a significant system, some computer assistance is required. The first step is relatively easy to mechanize. The second is a tedious intellectual task that cannot be fully mechanized. Several powerful computer tools have been developed to ease this task, but significant human involvement is required in proving particularly difficult formulas. Perhaps the most powerful existing proving tool is the system developed by Boyer and Moore at SRI [Boyer, Moore 79]. Some of the difficult programs that have been proved correct by this system are: a simple optimizing expression-computer, a fast string-searching algorithm, a mechanical theorem prover for propositional calculus, and an arithmetic simplifier.

Current work at SRI is aimed at integrating HDM and the Boyer-Moore theorem prover. The goal is to make it possible to use HDM to structure a large system into a simply-connected, hierarchy of modules whose sizes and inter-relationships are within the scope and power of the theorem prover.

3.3 Example of formal specification, structuring and proof.

This section contains two examples of formal programming methods. The first illustrates the use of the SPECIAL specification language to specify a simple buffer module, and the second illustrates an approach to the verification of a fault-tolerant avionic computer.

3.3.1 Specification of a "stack" module

Figure 3 gives a specification for the external behavior of an unbounded push-down-stack, written in a simplified, informal version of the SPECIAL language. The basic model used is a certain kind of finite-state machine, which contains data structures and operations. The data structures are defined by a set of state definition functions, and can be accessed only through operations. The operations are externally callable. They may modify the state and/or return a state-dependent value. These state-functions and operations have certain characteristics, derived from their nature as specifications:

- State-functions: the observed state may be one of a set of defined states or it may be (as yet) undefined

- State operations: if none of a set of defined exception conditions apply, a change of state may occur, and a value may (or may not) be returned. The statement of state-change effects is a list of constraints on the state of the machine that must be satisfied after the operation is completed. The list is a conjunction of individual constraints, with no notion of sequential evaluation, i.e., the net effect is defined in a non-procedural way. Furthermore, the constraints may not necessarily have a unique state solution.

The specification in Figure 3 starts with data structures Access, which contains the elements of the stack, and Size, which holds a value giving the present number of stack elements. The operations are Top, which returns the value of the top-most element (unless the stack is of size 0), Push, which has the effects that the new i-th item is the old i-th item, that the new size is 1 more than the old size, and that the data denoted by taking the new size as an index to the newly placed element, and Pop, which has the effect that the new size is one less than the old size, that the new i-th item is the same as the old i-th item, up to one-less than the old size, and which returns a value indexed by the old size.

These specifications are given in terms of a particular abstract data structure. In a hierarchically structured system, the functions specified will be implemented by programs that act on data structures at a lower level. These structures may be of quite a different nature, e.g., an array or a tree, at the convenience of the implementer. The specifications may seem to give excessive detail for such a simple system, but it is precisely in the area of "obvious" details that many errors and ambiguities occur in ordinary programs. The several categories of data called for in SPECIAL are intended to help organize

the enumeration of these tedious and seemingly obvious details.

3.3.2 Proof of the SIFT computer design

We will present a sketch of formal structuring and proof (due to L. Lamport and R. Shostak) taken from the SIFT (Software Implemented Fault Tolerance) avionic computer [Wensley 78, Shostak 77]. Details of the formal specification will be omitted.

The general approach to system structuring being used in the SIFT Computer development is illustrated in figure 4. A system is conceived and specified with three major levels of functionality: User Functions, Software Functions, and Hardware Functions. All the levels represent views of the same system, taken with different degrees of detail. Formal specifications are written that give precise statements of the function provided at each level. The user specification is implemented by a requirements hierarchy, typically in the form of set-theoretical models. The software specifications are implemented by a software hierarchy, typically in the form of abstract programs. The hardware specifications are assumed to be implemented conventionally, typically with some degree of hierarchy.

Figure 5 illustrates three corresponding views of SIFT. The User View (5a) (in this case, the view given to the systems programmer) displays an adaptive voting system that serves a set of tasks (e.g., a set of flight-control programs). Tasks are dispatched and executed, redundantly and in parallel, on several processors (the number may vary according to the critically of the tasks). The results are combined (by voting) to generate intermediate results and outputs. Discrepancies among results are analyzed and are used dynamically to determine the assignment of tasks those the processors that are deemed to be fault-free. The software view (5b) displays a five-level software hierarchy: SIFT virtual macnines (including task-dispatcher), Local Executives (one per virtual machine), Global-Local Reporting, Global Executive (acutally a set of programs acting in unison) and Application Software. The Hardware view (5c) displays a set of conventional, avionic processor-memory pairs, each with its own clock and power supply, interconnected by a fault-tolerant buffering and interconnection system.

In order to verify that the design is correct, it is necessary to show that the user view is correctly implemented by the software, and that the software is correctly implemented by the hardware. The general scheme for doing this is illustrated in Figure 6. The upper two levels are shown as models in the form of state-sets and state-transitions, such as a Markov Model or a Scheduler-Dispatcher. The lowest level (in this example) is shown as a software object, such as a global executive program.

The proof consists of a chain of arguments about the properties of each level and the relations between the levels. Logical arguments about properties of a system must, of course, be based upon axioms appropriate to the system -- for example, the property that a task-processor scheduler is correct, would take as an axiom that the processors are properly synchronized. Part of the power of hierarchical proof derives from the fact that the axioms used in proving properties at one level are exactly the properties that must be proven at the next lower level.

Figure 7 gives a somewhat more detailed hierarchy of models for SIFT together with the properties that must be proven at each level.

The proof of SIFT is still in a formative stage, but the approach described here appears to be theoretically sound. Since the SIFT software system is much smaller than conventional operating systems, we believe that the proof will not contain serious difficulties.

4. A Future View of Software Validation

The formal approach described in the preceding section is still immature. It is being used by highly talented teams for certain advanced system developments, with very promising results; however, not enough experience has been gathered to predict its cost and effectiveness. On the other hand, various existing approaches to software validation have some proven and perhaps unique benefits.

Figure 8 suggests a framework for comparing various approaches to system validation. The major approaches compared are: Static Analysis of Design, including formal and informal methods, and Dynamic Analysis, including design-based simulation and physical testing. It is clear that for the goal of design correctness, formal static analysis provides the highest confidence and physical testing the least. All methods, however have unique benefits. For example, physical testing is needed to validate assumptions made in the design about the properties of a physical implementation; simulation is valuable for confirming to the user that his stated requirements truly reflect his needs; and informal static analysis (or "walk-throughs") is usually very cost-effective for discovering simple errors in design.

In the future, we may expect to see some mixture of all of these methods. As formal methods become more established, the other methods will be reduced to providing their unique contributions, rather than, as presently, attempting to cover all aspects of the validation function.

5. Acknowledgements

The ideas and results described are the product of many members of the Computer Science Laboratory. The author particularly appreciates the helpful conversations with Karl Levitt, Brad Silverberg, Larry Robinson, Peter Neumann, and Leslie Lamport.

The ideas derived from work supported by NASA Langley Research Center under contract NAS1-13792 and Navy Ocean Systems Center under contract N00123-76-C-0195. The support and encouragement of Mr. Nick Murray and Mr. Linwood Sutton are gratefully acknowledged.

REFERENCES

[Boebert 77] BOEBERT, W.E., KAMRAD, J.M., RANG, R.E.
Analytic Validation of Flight Hardware.
Technical Report 77SRC63, Systems and Research Center, Honeywell, September 1977.

[Boyer, Moore 79]
BOYER, R., and MOORE, J STROTHER.
A Computational Logic.
Academic Press, 1979.

[Dijkstra 68] DIJKSTRA, E.W.
Complexity Controlled by Hierarchical Ordering of Function and Variability
Report on a Conference on Software, Engineering, NATO, 1968.

[Feiertag 79] FEIERTAG, R., LEVITT, K.N., MELLIAR-SMITH, P.M.
A Real-Time Operating System for Tactical Applications.
Technical Report, SRI International, 1979.

[Floyd 67] FLOYD, R.W.
Assigning Meanings to Programs, pages 19-32.
In Mathematical Aspects of Computer Science 19, (J. T. Schwartz, Editor), Amer. Math. Soc., 1967.

[Goldberg 73] GOLDBERG, J.
Proceedings of a Symposium on the High Cost of Software
SRI International, Menlo Park, CA, 1973.

[Hopkins 78] HOPKINS, A.L. JR., SMITH T.B.,III, and LALA, J.H.
FTMP-A Highly Reliable Fault-Tolerant Multiprocessor for Aircraft.
Proceedings IEEE 66(10):1221-1239, October 1978.

[Parnas 72] PARNAS, D.L.
A Technique for Software Module Specification with Examples.
Comm. ACM 15(5):330-336, May 1972.

[Robinson 77] ROBINSON, L., LEVITT, K.N., NEUMANN, P.G., and SAXENA, A.R.
A Formal Methodology for the Design of Operating System Software.
In Current Trends in Programming Methodology, Vol. I, (R.T. Yeh, Editor), Prentice-Hall, 1977.

[Robinson 75] ROBINSON, L., LEVITT, K.N., NEUMANN, P.G., SAXENA, A.R.
On Attaining Reliable Software for a Secure Operating System.
Sigplan Notices 10(6):267-284, June 1975.

[Robinson, Levitt 77]
ROBINSON, L. and LEVITT, K.N.
Proof Techniques for Hierarchically Structured Programs.
Comm. ACM 20(4):57-67, April 1977.

[Roubine 77] ROUBINE, O.M., and ROBINSON, L.
The SPECIAL Reference Manual.
Technical Report CSL-45, SRI International, 1977.
SRI Project 4828, Contract N00123-76-C-1095.

[Shostak 77] SHOSTAK, R.E., et. al.
Proving the Reliability of a Fault-Tolerant Computer System
Proc. 14th IEEE Comput. Soc. Int. Conf., IEEE, 1977.

[Wensley 78] WENSLEY, J.H., LAMPORT, L., GOLDBERG, J., GREEN, M.W., LEVITT, K.N., MELLIAR-SMITH, P.M., SHOSTAK, R.E., and WEINSTOCK, C.B.
SIFT: Design and Analysis of a Fault-Tolerant Computer for Aircraft Control.
Proceedings IEEE 66(10):1240-1254, October 1978.

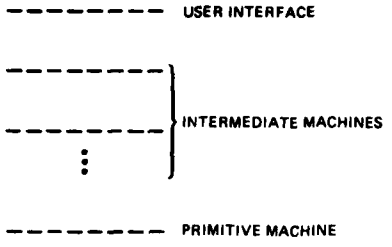


FIGURE 1 MULTI-LEVEL ABSTRACT MACHINE HIERARCHY IN HDM

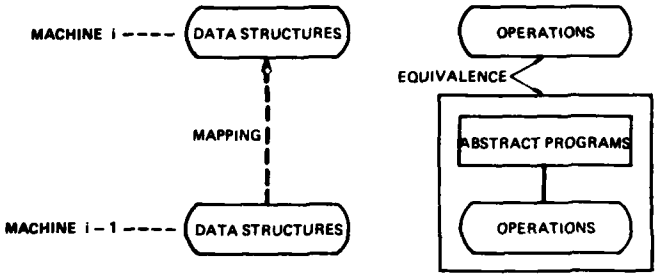


FIGURE 2 ABSTRACT-MACHINE IMPLEMENTATION IN HDM

Data Structures:

Access(integer) → elem;
intially undefined.

Size( ) integer;
intially 0

Operation:

Top( ) elem;
exceptions empty: Size = 0;
effects none;
returned value = Access(Size).

Push(elem);
exception none;
effects 'Size = Size + 1;
'Access(i) = Access(i) for 1 ≤ i ≤ Size;
'Access(Size + 1) = elem;
returned value none.

Pop( ) elem;
exceptions empty: Size = 0;
effects 'Access(i) = Access(i) for 1 ≤ i ≤ Size - 1;
'Size = Size - 1;
returned value = Access(Size).

FIGURE 3 SPECIAL SPECIFICATION OF PUSH-DOWN STACK

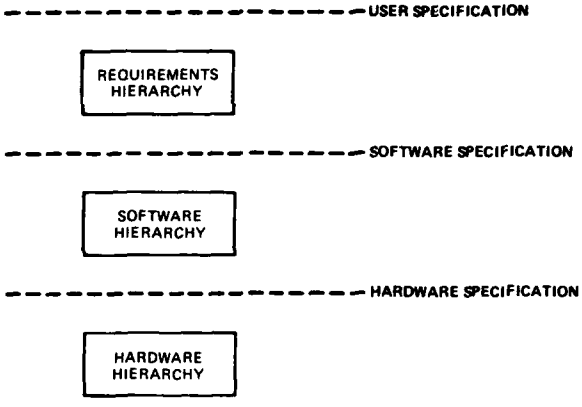
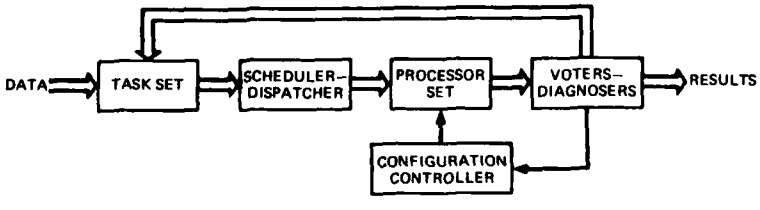
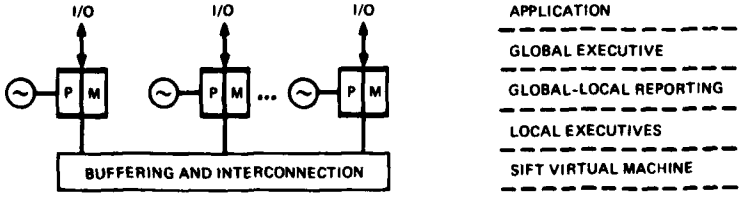


FIGURE 4 GENERAL SYSTEM STRUCTURE



(a) SIFT: USER VIEW



(c) SIFT: HARDWARE VIEW

(b) SIFT: SOFTWARE VIEW

FIGURE 5 USER, SOFTWARE AND HARDWARE VIEWS OF THE SIFT COMPUTER

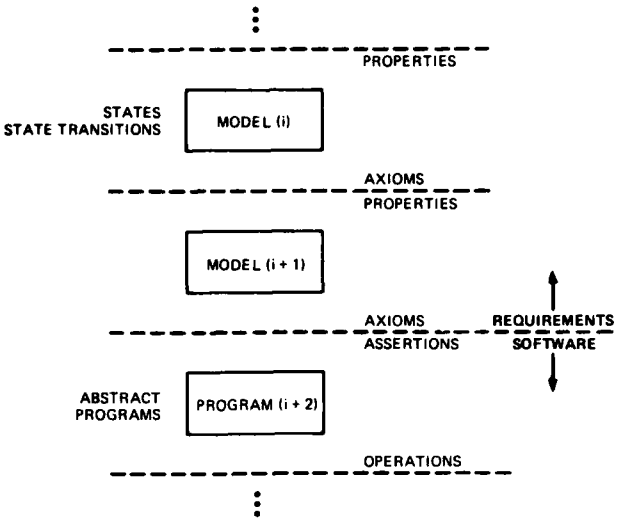


FIGURE 6 SCHEME FOR HIERARCHICAL PROOF

HIERARCHICAL PROOF OF DESIGN CORRECTNESS

SYSTEM VIEW	STATE VARIABLES	PROPERTY
MARKOV RELIABILITY MODEL	NUMBER OF GOOD PROCESSORS NUMBER OF OBSERVED FAULTS	THERE ARE ENOUGH GOOD PROCESSORS TO SUPPORT VOTING
ALLOCATION MODEL	PROCESSORS X TASKS	THE GOOD PROCESSORS ARE ASSIGNED TO THE RIGHT TASKS
SYNCHRONIZATION MODEL	TASK SCHEDULE PROCESSOR X PROCESSOR-COMMUNICATION	PROCESSORS ACT AND COMMUNICATE AT THE RIGHT TIME
EXECUTIVE PROGRAM	PROGRAM VARIABLES BUFFER STATES	PROGRAM CODE IS CONSISTENT WITH SYNCHRONIZATION MODEL
HARDWARE	MEMORY & REGISTER CONTENTS BUS STATE	MACHINE INSTRUCTIONS SUPPORT EXECUTIVE PROGRAMS
LOGIC	LOGIC VARIABLES	LOGIC IMPLEMENTS INSTRUCTIONS

FIGURE 7 HIERARCHICAL VIEWS AND PROPERTIES FOR PROOF OF SIFT DESIGN CORRECTNESS

APPROACH	BENEFITS	WEAKNESSES
STATIC ANALYSIS		
• FORMAL PROOF	• HIGH CONFIDENCE	• ASSUMES CORRECTNESS OF REQUIREMENTS • HIGH TECHNOLOGY
• INFORMAL REVIEW ("WALK-THROUGH")	• ECONOMY	• MAY MISS SUBTLE FAULTS
DYNAMIC ANALYSIS		
• SIMULATION	• EXTENSIVENESS	• ASSUMES PERFECT REALIZATION
	• VALIDATES REQUIREMENTS	• CANNOT COVER ALL CASES
• PHYSICAL TESTING	• VALIDATES REALIZATION	• COSTLY • VERY POOR COVERAGE

FIGURE 8 APPROACHES TO SOFTWARE VALIDATION

DISCUSSION

T.L.Regulinski, US

You asserted that you are not aware of any models dealing with software maintenance cost. Is that correct?

One of my graduate students at the Institute has developed a modified Raleigh model for the development of software LCC. I strongly detect that there is an implication that the maintenance cost of the model may be difficult to derive. From your experience would you please comment on this?

Author's Reply

No, I don't think it will be difficult to derive. Our problem is that there is not sufficient attention paid to this. There are many organizations presently and in the past which have carried out maintenance activities but there is not enough awareness of the cost importance. Therefore the past data has been of a very poor quality but data is available now which must be sought out and interpreted. This is not academic, but I have estimates of maintenance costs in a total life-cycle as being somewhere in the range of 70%. This is therefore a profitable business.

M.B.Kline, US

We have a serious semantics problem when discussing such things as software reliability and maintainability. When we talk about software reliability it is different to hardware reliability, as Mr Heiner points out. As far as I can see, there is no time degradation or wear-out in software (no bathtub curve equivalent), but rather software reliability is a design phenomenon, reduction of residual program logic errors. Thus, what we are really talking about is software design assurance through test and evaluation. In some ways it is like the hardware reliability growth concept of test, analyse and fix, in this case error reduction.

Since software does not fail or degrade in time as a result of use, software maintenance or maintainability has no significance. We do not do corrective or preventive maintenance on software. Rather it is my belief that when people talk about software maintainability they are really talking about software configuration management, i.e. how do we redesign the software as a result of hardware or operational design change. I think it is essential that we understand and use proper terminology so that confusion is reduced. We cannot communicate if we do not pay attention to semantics.

Let us not allow software to follow blindly our notions of hardware reliability and maintainability. Let us keep separate the things we do in design and development (including its test and evaluation) from the things that happen during operation after the system is produced. Certainly we wouldn't talk of hardware test and redesign for failure to meet performance criteria as hardware reliability.

All three papers cause me to comment. I am quite disturbed that software is making the same mistakes as hardware in assuming that maintainability is the same as reliability.

There is no bathtub curve to software, software reliability is really what we do to renew as best we can errors and error paths during the design. Software reliability entails designing and testing to reduce logistics errors as much as possible.

With software maintainability it seems to me that we are really talking about software configuration management.

Author's Reply

If you take a program with n number of errors, with time n should be reduced, consequently the reliability ought to be improved.

W.Ehrenberger, Ge

Could you please give an example of a proof mentioned in your paper, and perhaps also an example of strict top down design?

Author's Reply

Proof:
+ reference (Wensley 78), Last section (plan for proof)
+ reference (Boyer, Moore 79) (Theory and Practice of Program Proving)
+ Feiertag R.F. (paper on Proof of Security Properties)
Proc. ACM Symposium on Operating Systems 1978

Examples of designs
+ reference (Robinson 75)
+ reference (Robinson 77)
+ Spitzen, J. et al. (Approx: "An Example of Hierarchical Design") Communic. ACM, a Fall 78 issue.

F.S.Stringer, UK

A new problem arising within complex systems are the interactions, many of which may be unpredictable, when a software program is modified.

Can the author please comment on how this problem should be tackled to prevent dangerous situations developing when programs are modified?

The advent of flight critical avionic systems is appropriate to this issue.

Author's Reply

In integrated (non-distributed) systems, the establishment of previously specified levels of virtual functionality can (1) serve to hide changes in the implementations of the function, and (2) serve to provide a well-defined base of support for new functions.

The problem is more difficult in distributed systems

One issue is the prevention of deadlock in communication due to faulty logic. Our understanding of this issue is improving, but there can be higher-order forms of deadlock resulting, e.g. from circular references to data that may be created accidentally by a modification. Another kind of change-induced fault is improper synchronization, e.g. changing the state of a variable too early or too late for proper reading by an external processor.

Another problem is the loss of consistency among multiple versions of some data. There are more problems here than solutions. One general approach is to organize each processor hierarchically, so that the flow of control needed to link remote data can be traced easily. This should be combined with a strong doctrine for defining the time of occurrence of system events.

QUANTITATIVE ASSESSMENTS OF SOFTWARE RELIABILITY

J. - C. RAULT (1) - G. MEMMI (2) - S. PIMONT (3)

(1) IRIA - Domaine de Voluceau, 78150 LE CHESNAY, France

(2) ECA AUTOMATION, 315 Bureau de la Colline 92213 SAINT-CLOUD, France

(3) THOMSON-CSF, Paris, France.

SUMMARY

After stressing current need for quantitative measures of software reliability for prediction, monitoring and a posteriori assessment purposes, the present paper provides a categorization and a description of those approaches leading to practical applications. Three main approaches to quantitatively assessing software reliability are identified :

- Models adapted from classical reliability theory wherein reliability is expressed in terms of software error rate.
- Models based on sampling techniques applied to the error domain or to the input data domain ; here reliability is related respectively to an estimate of the number of residual errors and to the probability of not using those input data leading to software failures.
- Models based on program complexity measures; their basic principle is attempting to discover correlations between complexity measures and the most likely number of errors made during programming.

Underlying assumptions and areas of application are indicated. It is concluded to the existence of methods of practical interest and of data that might help to understand how often, when, where, and why programmers introduce software errors and how software errors may be detected and corrected.

1. THE HIGH COST OF SOFTWARE AND THE RATIONALE FOR METRICS

The high cost of software today is a well recognized fact of which developers of computer-based systems are fully aware. For this reason a stringent need for better design methodologies and guides for developing and certifying software products is felt among users, designers and vendors.

Past and present efforts to fulfill this need have covered the following two main areas :

- design techniques and methodologies for the attainment of reliable software
- a posteriori improvement of software : here it is attempted to control quality and reliability in order to alleviate deficiencies and limitations of design methodologies. Such control may proceed by two approaches :
  - proof of correction wherein validation is part of design ; as of today techniques for proof of correction are still far from being applicable to actual industrial environments.
  - testing and debugging procedures ; here validation lies outside the design phase.

Concurrently with effort concerning the above two approaches, but to a lesser extent, studies have been conducted for about ten years in order to measure the confidence or the reliability to be placed on programs ; here assessment is quantitative and no longer qualitative.
Both users and designers of software products have a need for techniques which are conducive to quantitative assessment of software reliability as well as the prediction and monitoring of software quality and/or reliability.

On the one hand, users should know how much confidence can be put into the use of a given piece of software in terms of availability, reliability and security ; usually, however, evaluation is based only on the program source code and a set of tests provided by designers plus a priori knowledge of future use of the program. This information is used mainly in a qualitative and subjective manner. However, following the practice that has been well established for hardware, measures of test exhaustivity and, subsequently, of reliability, are preferred to mere assertions of goodwill or self-confidence on the part of software product designers. Users would also like to be certain, through a formal, reproducible and quantitative procedure, that quality and reliability are built into the products they are acquiring.

On the other hand, designers need to know the prevailing factors that induce reliability. Enforcing the associated procedures in the design process and choosing rationally among the available design methodologies aid the attainment of reliable software. Moreover, in the context of a given design process, the use of metrics to monitor the progress of quality and reliability are invaluable guides to taking the appropriate actions and ascertaining the completion of the different design phases.

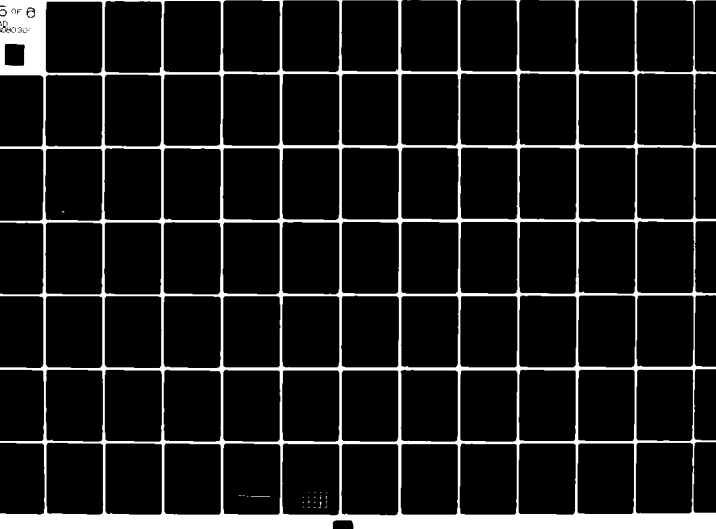
The following sections will focus on the available metrics that can be used in the quest for quantitative methods of assessing software reliability, be it for prediction, monitoring or a posteriori evaluation.

A shorter and earlier version of this paper will apprear in the Special Issue on Software Reliability of the IEEE Transactions on Reliability (1979).

AD-A080 301 ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT--ETC F/G 9/5
AVIONICS RELIABILITY, ITS TECHNIQUES AND RELATED DISCIPLINES.(U)
OCT 79 M C JACOBSEN

UNCLASSIFIED AGARD-CP-261 NL

5 OF 6
AD
A080301



2. APPROACHES TO SOFTWARE RELIABILITY METRICS

The concept of reliability assessment concerns a quantitative expression of the confidence which can be placed on a program. In fact, absolute measures are not known and are, in our opinion, meaningless ; quantitative and relative guidelines make more sense. For this reason, according to the chosen measures and underlying principles, the approaches that have been used fall into one of two general categories :

- . those resulting from extensions to software of techniques used for assessing hardware reliability and test coverage ;
- . those which are specific to software taking into account the internal complexity of programs, their dynamic behavior and possibly psychometric data on the programming activity.

The following presentation is structured according to the above categorization and is based on a detailed study (Memmi G. and Rault J. - C., 1979).

2.1. Extension of hardware-based techniques

Knowledge of long established techniques in the hardware field suggests two main classes of reliability assessment techniques for software :

1 - those techniques which are direct applications of conventional reliability theory ;

2 - those techniques derived from the procedures for quantitatively assessing efficiency and coverage of hardware testing sequences.

Both types of techniques have been greeted with occasional scepticism by software specialists. However, we feel that a carry-over from hardware to software is feasible. This is well substantiated by the practical results obtained so far. For this reason, we will present both of the above types of techniques while attempting to stress their respective advantages and limitations.

Note : With a lack of standard terminology, "error" and "fault" will be used with the following definitions :
Error : Discrepancy between actual and expected results when executing a program
Fault : The cause of an error.

2.1.1. Application of conventional reliability theory

2.1.1.1. Principles:
This approach, which has been investigated for about ten years by former hardware reliability specialists, basically consists of applying, to software, the concepts and modelling techniques of conventional reliability theory (for instance see Barlow R.E., 1965). Accordingly, reliability is defined as the probability that a program works without error during a given time span on the machine for which it has been intended and under specified environmental conditions. Corresponding reliability models indicate the following reliability measure :

$$R(t) = \exp\left[-\int_0^t z(t)\,dt\right]$$

where z (t) is the estimated or measured error rate ; this gives the mean time between failures :

$$MTBF = \int_0^\infty R(t)dt$$

About 15 reliability models along these lines have been proposed and some of them thoroughly investigated and then applied to actual software projects. These models differ mainly in the assumptions underlying the proposed expressions of the error rate.
While a transfer for technology from hardware to software is natural and valid as far as concepts and vocabulary (error rate, MTBF and the like) are concerned, it has been argued that important and irreconcilable differences are observed in the nature and types of faults (no wear-out or load factors are known for software for instance). In particular, software faults have a design origin, most hardware faults have a physical origin. For these reasons, some people strongly object to these reliability models. Although the above observations are reasonable and are based on presently available statistics and taxonomical studies regarding software faults, probabilistic reliability models are nevertheless justified by the same statistics that evidence the random pattern of manifestation of software faults.
The reliability models can be applied to the prediction of final reliability (or of the amount of effort necessary to reach a given reliability) and to the a posteriori assessment of reliability of a software product. Practical use of the reliability models requires a record of the testing history, e.g., the number and description of input test data sets, the number and time of occurrence of detected errors, the cumulative testing time, the time to correct each fault, and so forth. The unknown parameters of the models are then obtained through statistical estimates based either on data collected during past projects, or on data collected in the early stages of the testing phase.
We will review several of the proposed reliability models without attempting to be exhaustive.

2.1.1.2. Examples of some reliability models :
A - Conventional hardware reliability models
a) constant error rate :

$$z(t) = \lambda$$

this model hardly matches experience
b) Weibull error rate (Wagoner, W.L., 1973) :

$$z(t) = \alpha\beta t^{\beta-1}$$

this model takes into account a decreasing rate of error often observed in the first phase of software testing. Considering that the error detection and correction process is discrete in essence, these two models have been discarded in favor of discrete laws which permit the modeling of a decreasing error rate.

B - Z. Jelinski and P. B. Moranda models (Forman E. H., 1977 - Jelinski Z., 1972, 1973 - Moranda P. B., 1975, 1977 - Sukert, A. N., 1977). The assumptions for these models are as follows :
a1 : errors are randomly detected
a2 : error rate is constant between the detection of two subsequent errors
a3 : errors are detected independently of each other
a4 : faults are corrected as soon as they are detected
a5 : correcting one fault does not introduce additional faults
a6 : error rate is proportional to the number of remaining faults.
These assumptions lead to :

$$z_i = K (N - n_{i-1})$$

where :
n_{i-1} : the cumulative number of errors detected during the first (i-1) intervals (measured in days, weeks, or CPU hours).
N : the total number of faults (constant according to a5)
z_i : the error rate during the i^{th} testing interval.
The unknown parameters N and K may be estimated ($\hat{N}$ and $\hat{K}$) according to conventional criteria : maximum likeiihood or least squares.
Relations and discussions of several practical cases of applications of this model may be found in the open literature ; there is a good agreement between experimental data and theoretical predictions.
C - G. J. Schick and R. W. Wolverton models (Moranda P. B., 1977b, Schick G. J., 1972, 1978 - Sukert A. N., 1977).
This model departs from the previous model only in assumption a6 which becomes here :
a6' : during the i^{th} testing interval, error rate is proportional to the number of remaining faults and to the time, x, spent since the start of the i^{th} testing interval.
Accordingly, the expression for the error rate is :

$$z_i (x) = K (N - n_{i-1}) x$$

The estimators for N and K may be determined according to conventional criteria such as maximum likelihood or least squares.
Modifications to this model have been proposed by several authors (Schick, G. J., 1978 - Sukert, A. N., 1977).
Practical applications of this model have been described (Schick, G. J., 1978 - Sukert, A. N., 1977). However, the basic assumption (a6') still remains to be justified.
D - M. Shooman model (Moranda P. B., 1977b - Schick, G. J., 1978 - Shooman M. L., 1972, 1973, 1975a).
This model relies on the same assumptions as that of Z. Jelinski and P. B. Moranda ; however, it introduces other factors and different estimating procedures for model parameters. The error rate is expressed as :

$$z (t) = \frac{C}{I} (N - n (\tau))$$

Where :
I = number of instructions in the program ; this number is assumed to be constant over the testing period
n (τ) = cumulative number of detected errors
τ = cumulative debugging time
t = cumulative time of execution
Letting $\frac{C}{I} = K$ and $n (\tau) = n_i$; this corresponds to the Z. Jelinski and P. B. Moranda model.
N and K are estimated by expressing the error rate in two different ways at a given time.
E - J. D. Musa model (Musa J. D., 1975)
This model can be considered as an elaboration over the previous three models. Improvement lies in accurate measurement of time (CPU and calendar) and in actual distinctions among faults an errors.
To use this model, one needs to measure very carefully the number of programmers working for how long and the amount of CPU time spent every day. The complete model gives estimates of execution time and ealendar time remaining until a reliability goal is attained. It is described in detail, along with a supporting computer program, in (Musa J. D., 1975).
F - N. F. Schneidewind model (Schneidewind, N. F., 1975)
This model differs form the previous ones. The analysis does not account for errors separately but refers to the number of errors detected within a fixed time interval. The underlying assumptions are :
a1 : error occurrences are distributed according to a Poisson law
a2 : the mean number of errors detected by testing interval decreases exponentially (rate $\alpha(t) = \alpha \exp(-\beta t)$)
a3 : error rate is proportional to the number of remaining errors.
Accordingly, the number of detected errors at time t is :

$$n (t) = (\alpha/\beta) [1 - \exp (-\beta t)]$$

α and β are estimated according to the criteria of maximum likelihood.
As only easy to get data are necessary and associated computations are straightforward, this model is easy to use. There has not been much application of this model to date.
G - Other models :
Limitation of space does not permit the discussion of the other models which are less known and have been, to our knowledge, less used than the models briefly described above ; among these models are the Markovian models (Estep J. G., 1973 - Goel A. L., 1978 - Littlewood B., 1975 - Okumoto K., 1978 - Trivedi A. K., 1975) and those described in (Barlow R. E., 1969 - Corcoran W. J., 1954 - Hecht H., 1977 - Weiss K. K., 1956).

2.1.1.3. Practical use :

As of today, a definitive choice among the various hardware reliability-based models is premature, however, in order to aid their potential users, we propose to put these models into three categories according to their respective degree of validation.

In the first category are the models of Jelinski-Moranda, Schick-Wolverton and J. D. Musa which apparently have experienced the highest level of validation. Published comparative results indicate a preference for the Jelinski-Moranda model.

In the second category are models such as those of M. Shooman and N. F. Schneidewind, easy to implement and of practical interest, which have been sufficiently studied for considering their actual use. The third category corresponds to those models not sufficiently investigated for the practitionners to use them ; theoretical and experimental investigations remain to be performed.

2.1.2. Extension of hardware test efficiency measurement to software

2.1.2.1. Principles :

Schematically, preparation of test programs for digital systems involves one of two basic approaches :

- A deterministic approach in which, through direct synthesis or fault analysis, input sequences detecting a given set of assumed faults are determined.
- A probabilistic approach in which the system under test is paralleled with a reference system simultaneously exercised by random input sequences.

In both approaches, the main problem is in assessing the thoroughness of the testing sequences with respect to both possible input sequences and possible faults.

However, due to necessary simplifying assumptions, the assumed list of faults and input sequences actually used for testing purposes are only subsets of respectively real-life and operational conditions. Therefore, an exact assessment of thoroughness is generally infeasible. In short, whatever approach is used, testing is, for practical and economical reasons, restricted to a double sampling in both the input data domain and the fault domain. Software measurement does not escape this situation. Accepting the fact that sampling techniques must be used, there are three different types of models for software reliability assessment which are as follows :

- a model wherein reliability is defined as the maximum likelihood estimate of the number of residual faults. This estimate is obtained by fault analysis (the fault analysis of hardware) on a set of artificial faults purposely inserted into the program under test.
- a model wherein, input test sets are selected randomly according to an operational utilization profile ; reliability is then defined as the probability of using input data that produce correct results.
- a model wherein a two-step programming technique is used. It provides an analysis of the assessment of error coverage.

These three models are examined in more detail below.

2.1.2.2. Sampling in the fault domain - H. D. Mills model (Mills, H. D., 1970) :

For this model, reliability is measured as the maximum likelihood estimate of the number of faults in a program. The principle here is that one inserts into the program to be tested a set of j known artificial faults "typical" (nature, origin, frequency of occurrence, etc. ; here one may resort to available fault statistics) of the unknown expected faults (that is, indigenous faults). The number (N) of indigenous faults is estimated from the number of artificial (a) and indigenous (i) faults detected after a given amount of testing. It is proved that (ij)/a is a maximum likelihood estimate for N. The testing procedure is based upon the confidence to be placed in the statistical test of the assumption that the program contains no more than a given number D of faults. The associated confidence level is $c = j.(j+D+1)^{-1}$. Accordingly, the test procedure consists of the following steps :

- It is assumed that there are less than D indigenous faults in the program
- j artificial faults are seeded (j is a function of a given confidence level)
- the program is tested until :
  - j artificial faults are detected before obtaining D+1 indegenous faults (true hypothesis)
  - D+1 indigenous faults are detected before detecting j artificial faults (false hypothesis).

More elaborate test procedures may be devised following other ways of estimating the confidence level (Tausworthe R. C., 1977).

Implementation of this technique is straightforward. However, attainment of high levels of confidence requires seeding a fairly large number of artificial faults. Consequently, its use is recommended at the end of the testing phase where the number of residual faults is small.

2.1.1.3. Sampling in the input domain - E. C. Nelson model (Craig, C. R., 1974 - Nelson, E. C., 1973 - Llyod D. K. et al., 1977). Here reliability is measured as follows : if after ℓ tests e errors are detected, reliability is estimated as $R = 1 - e/\ell$. The input data space, although finite, may be so large that practical treatment is not feasible. Therefore, it is partitioned into a small number of r disjoint subsets such that the probability that each subset is presented to the program is known.

Let $S_1, S_2, \ldots, S_r$ be the subsets of the partition and $P(S_i)$ their probabilities of occurrence.

The $P(S_i)$, $i = 1, \ldots, r$ represent the operational profile. Each subset S_j may be partitioned into two disjoint subsets S'_j and S''_j where S'_j consists of all the points of S_j leading to a correct execution of the program and S''_j of all those causing an error. The reliability is then :

$$R = \sum_{j=1}^{r} P(S'_j) = 1 - \sum_{j=1}^{r} P(S''_j)$$

If input data points are randomly chosen from each S_j and the program is run $\ell = \sum_{j=1}^{n} \ell_j$ times with f_j errors observed, then an estimate of R is :

$$\hat{R} = 1 - \sum_{j=1}^{r} (f_j)/n$$

$\hat{R}$ is unbiased if sampling is proportional to the probability of the input data set, i.e., if $\ell_j = nP(S_j)$. The variance of R is then :

$$\mathrm{Var}(R) = \ell^{-1} \sum_{j=1}^{n} P(S''_j)\, P(S'_j)/P(S_j)$$

This model makes no assumption as to the occurrence of errors in time, but it requires the definition of an operational profile and, therefore, the knowledge of the future use of the program.
Nelson's method has been further elaborated to take into account not only the operational profile defined by the user, but also a dynamic profile determined by the set of logic paths in the program.
This method is based on a simple definition of reliability and does not require specific assumptions regarding the time distribution of error occurrences : on the other hand, it requires an accurate definition of the input test data sets. It is easy to implement, however, and is consistent with other tools for program analysis and testing.

2.1.2.4. Simultaneous sampling in the input and fault domains (Girard E., 1973 - Rault, J.-C., 1973)
This third technique is based on the assumption that a two-step programming technique is used, that is, in the first step, specifications and algorithms are validated in a language L_1 by a first group of designers ; at the end of the first step one obtains a first version of the program to be developed. In the second step, a second version is written in a different language L_2 and possibly by a second group of designers ; in this step, efficiency rather than functional validation is at stake. Description of such a two-step programming technique may be found in (Rault, J.-C., 1973). A two-step programming methodology lends itself naturally to a transfer, from hardware to software, of techniques for generation of test sets and assessment of their coverage. As a matter of fact, the first version acts as a simulator for the second version. Subsequently, error simulation and statistical comparison testing are made feasible. References (Girard E., 1973 - Rault J.-C. , 1973) provide a discussion of the two testing techniques as applied to software and an analysis of the assessment of error coverage which they provide.

2.1.2.5. Other sampling modes :
Sampling techniques may be extended to other measures of program testing thoroughness ; along these lines one may consider sampling in program instructions (Basin S. L., 1974), execution paths, modules, and, more generally, in the various structural entities into which a program can be decomposed.

2.1.2.6. Practical use :
In our opinion, sampling techniques are sufficiently formalized for their use in actual software projects. These techniques require, on the one hand, a good knowledge of the structure and intended use of programs and, on the other hand, a good communication between programming teams and quality control groups. Sampling techniques should be preferably used at the end of the testing phase in addition to hardware-based reliability models.

2.2. Techniques specific to software

The basic principle of these techniques, which are more recent than the previous ones, is to relate reliability to measures of program complexity. Although the effects of program complexity on the total cost of software products have been known for years, it is only recently that formal definitions and associated measures of program complexity have emerged. The various measures proposed so far may concern :

- structural complexity based on static analysis of program graphs
- textual complexity based on a static analysis of program source texts
- structural complexity with respect to execution behavior.

The first two correspond to static analysis while the third one requires both static and dynamic analyses.

2.2.1. Structural complexity (Hansen W.J., 1973 - Mc Cabe T.J., 1976 - Myers G.J., 1977 - Zolnowski J.M., 1977 - Zweben S.H., 1977)

Here, program complexity is measured in terms of characteristic parameters of the graph derived from the program text. Several different parameters may be considered ; the most common being the cyclomatic number which is formally related to the count of decision instructions in the program. Available conventional tools may be used for obtaining such parameters automatically (Miller E.F., 1977 - Reifer D.J., 1977).
As of today, correlation between structural complexity and error rate has been collected on few examples.

2.2.2. Textual complexity - application of the software science theory (Fitzsimmons A.B., 1978 - Halstead M., 1977)

Here, reliability is measured as a number of residual errors. Intuitively, the number, N, of potential errors made by a programmer is related to the number of opportunities of making an error, i.e., the number W of mental discriminations. M. Halstead has established a formal theory which relates W to a set of measurable parameters of programs (such as numbers of operators and operands) ; moreover, the theory indicates that $N = K W^{2/3}$. This theoretical result, which encompasses experimental results from psychometric laws, has been experimentally corroborated by many other independent investigators (Fitzsimmons, A.B., 1978).

2.2.3. Structural complexity with respect to execution behavior (Pimont S., 1976, 1977).

The proposed approach consists of assessing the thoroughness of testing procedures in order to characterize program reliability. The assessment is based on measures of certain classes of paths exercised during the test stage. Moreover, to assess a particular program, the only information that need be supplied is a description of the test input data set. The methodology is general and can be applied to programs in different languages.

A program is said to be verified if, for a given set of tests, it can be shown that every case of interest has been tested. As this end is generally unattainable, we regard a program as being verified if one can prove that all the program - paths have been traversed. Accordingly, one can say that a certain degree of verification is attained with a given set of tests, according to the number of paths actually traversed. This degree of verification, which is a non-decreasing function of the number of tests, can be considered as an assessment of program reliability . The degree of verification, attained through experiments, can be deduced from the images of experiments in the program flow-graph. A practical procedure to perform such an evaluation has been defined (Pimont S., 1976, 1977).

To verify a program (according to the definition stated above), one has to consider all the paths from the program input to any particular block in the program flow-graph (paths emerging from a given block are not relevant). In order to obtain an assessment , one has to group these paths into classes where one such class is the set of all paths ending with a given pair of adjacent edges. Such a pair is subsequently referred to as a "switch". This level of grouping is considered sufficiently accurate and global to make one satisfied with experiments exercising nearly every switch and, conversely, unsatisfied with experiments keeping too many switches unexercised.

The proposed reliability assessment is based on two analyses, performed in parallel, which provide two measurements (experimental and theoretical) that are subsequently compared. Therefore, the proposed approach consists of the following three phases :

- phase 1 : static analysis of the program. The basic process here is to perform a theoretical analysis in order to analyze, for each switch, the paths from the program's input to the switch.
- phase 2 : dynamic analysis fo the program. Here, an experimental analysis is performed in order to determine under what circumstances every switch is exercised during the tests.
- phase 3 : assessment of the program reliability.

This phase corresponds to a comparison of the theoretical and experimental data obtained in the first two phases. A measure of the extent to which each switch is tested is derived from this comparison. Then, in order to grade programs and sets of tests, these local measurements are aggregated. A heuristic function is proposed to perfrom this aggregation.

Since loops can generate very large number of paths, a language is defined in order to limit the length of paths and, subsequently, their number. A suitable language, H, and its associated algorithms, are described in the case of programs written according to the structured programming technique (Pimont S., 1976, 1977). An algorithm computes, in a single pass and for every switch, the number of paths ending with that switch and belonging to the language H. This computations does not require the generation of such paths.

Testing experiments are made using conventional tools which aid in tracing the execution of programs. These experiments are analyzed as follows : a correspondence is defined between each set of experimental paths and a set of words belonging to the language H ; for each switch, a study of this set allows one to establish an equivalence of some words for that switch and to provide experimental data.

In conclusion, it is proposed that the extant to which each switch has been tested can be derived from the comparison of experimental data with theoretical results. Then by assembling all this information, the program's overall reliability can be assessed. The above technique remains to be applied to large scale programs. It would be best applied at the end of the testing phase.

2.2.4. Practical use

Models proceeding from complexity measures are unequally developed. Without any doubt, software science is the most thoroughly investigated one ; it has been tried by several academic and industrial groups which have evidenced a high correlation between theoretical and practical results. Moreover, analyzers have been developed for automatically deriving software science parameters of programs written in different programming languages (CMS (Felty J. L., 1978), FORTRAN (Ottenstein K. J., 1976), JOVIAL, PL/1 (Elshoff, J. L., 1976, 1978 - Tariq M. A., 1977)). It is not premature to recommend use of this technique to those responsible of software quality control.

The two other complexity-based models, structure and behavior, are too recent and require deeper investigation before attempting their introduction into the programmer's arsenal.

3. CONCLUSION : THE STATE-OF-THE-ART

Quantitative measurement of software reliability requires :

- Measurement tools : no universal measurement procedure is presently available but testing tools that might be used for measurement pruposes are available (Miller E. F., 1977-Reifer D. J., 1977);
- Models : several models have been proposed, but most of them seem to be better applied to a posteriori reliability assessment than to reliability prediction. There is a patent lack of data which can be analyzed reliably at the same time by means of several of the above models. Therefore, fair comparisons regarding their accuracy and usefulness are presently difficult to make.

Moreover, only a few of these models have been applied to a sufficiently wide range of application areas.
. Techniques and tools for collecting systematically the experimental data necessary for the design, validation and reliable use of the proposed models. To be of practical value, the data should be accurate, complete, easy-to-get (avoid psychological problems among programmers) and universal, i.e., consistent from one project to the other. These characteristics are required if one wishes to determine correlations between environmental factors and reliability measures. It seems as if we are caught in a vicious circle of two inter-related problems :
- lack of data prevents one from validating the proposed models,
- the absence of validated models prevents one from determining which data (nature and format) should be collected.
Collecting data systematically, even if in a somewhat empirical manner, is undoubtedly a first approach to take. This has already been started by several organizations.
Finally, in spite of the fact that no universal approach to quantitative evaluation of software reliability has emerged, promising techniques do exist as well as a fairly large amount of raw data that might help to understand how often, when, where, and why programmers introduce software faults. Moreover, these techniques may help us to understand how software faults may be detected and corrected. Investigations conducted during the past decade have lead to a set of techniques which remain to be used actually rather than to be extended. Several have been investigated thoroughly enough to be transferred from research laboratories to industry.
Notice that the various techniques described above are not mutually exclusive ; they could be concurrently used along the different phases of software development. As an example a possible scheme would be :
. measure the complexity of modules as they are written (software science) ; the result would be either to reject the program if complexity is higher than a given acceptable level, or to estimate the most likely total number of expected faults, i.e., an estimation of the amount of testing effort to be spent.
. during the testing phase, assess the reliability and control its progress by means of either one of the hardware-based reliability models or sampling in the data domain.
. at the end of the testing phase, assess the efficiency of the test cases that have been used and that might be used again for maintenance purposes ; in this case, one may resort to sampling in the fault domain or to one method for test coverage assessment such as structural analysis with respect to execution behavior (cf. § 2.2.4). An optimal scheme can only be determined for a given programming environment and after sufficient experimentation. However, the relationships among the various tools and the three general approaches described above can be illustrated by the synoptic diagram of figure 1. As a matter of fact, one can envision to extend this diagram to the assessment of the other sixty or so factors (e.g., portability, modularity, testability, maintainability, efficiency, re-useability, clarity, documentation, etc.) which cooperate to software quality. The general arrangement would be a set of integrated analyzers (static and dynamic) centered around conventional utilities such as compilers, assemblers and text-editors ; the input data to these analyzers are either pieces of source code, design and coding rules, or input test cases, their results are aimed either at the programmer (coding errors, violations of standards, design flaws, etc), the project management (project status, testing history, programming productivity, error rate) or those responsible for quality control (some 60 factors to monitor among which reliability).
Those software designers who introduce such techniques as part of their arsenal of tools will have a definite advantage over their competitors in the development of truly reliable software. ACKNOWLEDGEMENT. We would like to thank Professor J.F. Meyer of the University of Michigan for his helpful suggestions and for improving the English text of an early version.

REFERENCES

Akiyama F., 1971, "An example of software system debugging", Proceedings IFIP Congress 1971, North Holland 1972, p. 353-359.

Barlow R.E. and Proschan F., 1965, "Mathematical theory of reliability", J. Wiley, New York.

Barlow R.E., Proschan F. and Scheuer E.M., 1969, "A system debugging model", University of California Berkeley Operations Research Center, Report ORC 69-6.

Basin S.L., 1973, "Estimation of software error rates via capture-recapture sampling, Sciences Applications Inc., Palo Alto, Ca.

Basin S.L., 1974, "Measuring the error content of software", Sciences Applications Inc., Palo Alto, Ca.

Bell D.E. and Sullivan J.E., 1975, "Further investigations into the complexity of software, MITRE Technical Report Number MIR-2874 Volume 2, Bedford Mass..

Brown J.R. and Lipow M., 1975, "Testing for software reliability", Proceedings of the 1975 International Conference on Reliable Software, pp. 518-527.

Corcoran W.J., Weingarten H., and Zehna P.W., 1954, "Estimating reliability after corrective action", Management Science, Vol. 10, no. 4, pp. 786-795, July 1954.

Craig C.R. et al., 1974, "Software reliability study", Report AD-787-784. (see T.A. Thayer, 1978).

Elshoff J.L., 1976, "A numerical profile of commercial PL/1 programs", Software Practice and Experience, Vol. 6, n° 4, pp. 505-525, October 1976.

Elshoff J.L., 1978, "A study of the structural composition of PL/1 programs" ACM SIGPLAN Notices, Vol. 13, n° 6, pp. 29-37, June 1978.

Estep J.G., 1973, "A software availability and reliability model", IBM Corp Rep. 6631x2403 Dept JM1 Safeguard Systems Engineering, Morris Plains, New Jersey ; also Record of the 1973 IEEE Symposium on Computer Software Reliability New York, 1973, p. 101.

Felty J.L. and Davis M.W., 1978, "SPAR (Source Program Analyzer and Reporter) User's Guide", Intermetrics Inc. Report IR-215-2.

Ferdinand A.E., 1974, "A theory of system complexity", Int. J1. of General Systems, vol. 1, pp. 19-33.

Fitzsimmons A.B. and Love L.T., 1978, "A review and evaluation of the Software Science", ACM Computing Surveys, Vol. 10, n° 1, March 1978, pp. 3-18.

Forman E.H. and Singpurwalla N.D., 1977, "An empirical stopping rule for debugging and testing computer software", Journal of the American Statistical Association, vol. 72, n° 360, pp. 750-757, December 1977.

Girard E. and Rault J.-C., 1973, "A programming technique for software reliability", Record of the 1973 IEEE Symp. on Computer Software Reliability, New York, pp. 44-50.

Goel A.L. and K. Okumoto K., 1978, "Bayesian software prediction models - Bayesian software correction limit policies", Rome Air Development Center, RADC-TR-78-155, vol. 4.

Goel A. L. and Okumoto K., 1978, "Bayesian software prediction models-an imperfect debugging model for reliability and other quantitative measures of software systems," Rome Air Development Center, RADC-TR-78-155, vol. 1.

Green T.F., Schneidewind N.F., Howard G.F., and Pariseau R.J., 1976, "Program structures, complexity and error characteristics", Proceedings of the 1976 Symposium on Computer Software Engineering, Polytechnic Press of the Polytechnic Institute of New-York, pp. 139-154.

Halstead M.H., 1977, "Elements of software science", Elsevier.

Hamilton P.A. and Musa J.D., 1978, "Measuring reliability of computation center software", Proceedings of the 3rd International Conference on Software Engineering, 1978, pp. 29-36.

Hansen W.J., 1973, "Measurement of program complexity by the pair", SIGPLAN Notices, Vol. 13, No. 3, pp. 29-33, March 1973.

Hecht H., Sturm W.A. and Trattner S., 1977, "Reliability measurement during software development," AIAA Conference on Computers in Aerospace, pp. 404-412.

Jelinski Z. and Moranda P.B., 1972, "Software reliability research", in Statistical Computer Performance Evaluation, W. Freiberger ed., Academic Press. pp. 465-484.

Jelinski Z. and Moranda P.B., 1973, "Application of probability-based model to a code reading experiment", Record of the IEEE Symp. on Computer Software Reliability, New York, pp. 78-81.

Laemmel A. and Shooman M.L., 1978, "Statistical (natural) language theory and computer program complexity", Rome Air Development Center, RADC-TR-78-4, vol. 2, July 1978.

Littlewood B., 1975, "A reliability model for Markov structured software", Proc. Intl. Conf. on Reliable Software, pp. 204-207, April 1975.

Lloyd D.K. and Lipow M., 1977, "Reliability-Management, Methods and Mathematics", published by the authors (201 Calle Miramar, Redondo Beach, Cal. 90277).

Mc Cabe T.J., 1976, "A complexity measure", IEEE Trans. on Software Engineering, vol. SE-2, no. 4, pp. 308-320, Nov. 1976.

Memmi G. and Rault J.-C., 1979, "Rapport final sur la fiabilité des logiciels temps réel des missions spatiales", ECA AUTOMATION, Paris, January 1979, (ESA Contract n° 3326).

Miller E.F., 1977, "Program testing art meets history", Computer, pp. 42-51, July 1977.

Mills H.D., 1970, "On the statistical validation of computer programs", IBM Corp Federal Systems Division, Rept. FSC 72-6015, July 1970.

Miyamoto I., 1975 "Software reliability in on line real-time environment", Proc. of the International Conference on Reliable Software, pp. 194-203.

Miyamoto I, 1978, "Toward an effective software reliability evaluation", 3rd International Conference on Software Engineering pp. 46-55.

Moranda P.B., 1975a, "Prediction of software reliability during debugging", Proceedings of the 1975 Annual Reliability and Maintainability Symposium, pp. 327-332.

Moranda P.B., 1975b, "A comparison of software error-rate models", 1975 Texas Conference on Computing, pp. 2A-6.1. - 2A-6.9.

Moranda P.B., 1975c "Probability-based models for failure during the burnin phase", Joint National Meeting TIMS-ORSA.

Moranda P.B., 1975d, "Estimation of a priori software reliability", Proceedings of Computer Sciences and Statistics, 8th Annual Symposium on the Interface, J.W. Frane, Ed., Health Sciences Computing Facility, Los Angeles, pp. 364-370.

Moranda P. B., 1977a "Quantitative methods for software reliability measurement", IEEE Computer Society Repository R-77-299.

Moranda P. B., 1977b "A comparison of software error rate models", IEEE Computer Society Repository, R77-300.

Moranda P. B., 1978, Comments on "An analysis of competing software reliability models", by G. J. Schick and R. W. Wolverton, IEEE Transactions on Software Engineering, vol. SE-4, n°2, pp. 104-120, March 1978, IEEE Computer Society Repository R-78.

Mulock R. B., 1969, "Software reliability", Proceedings of the 1969 Annual Symposium on Reliability, pp. 495-498.

Musa J. D., 1975, "A theory of software reliability and its application", IEEE Trans. on Software Engineering, Vol. SE-1, No. 3, pp. 312-327, Sept. 1975.

Musa J. D. and Hamilton P. A., 1977, "Program for software reliability and system test schedule estimations-Program documentation", IEEE Computer Society Repository R77-243.

Musa J. D., 1977, "Program for software reliability and system test schedule estimation -User's guide", IEEE Computer Society Repository R77-244.

Myers G. J., 1977, "An extension to the cyclomatic measure of program complexity, SIGPLAN Notices, pp. 61-64, Oct. 1977.

Nelson E. C., 1973, "A statistical basis for software reliability assessment", TRW Software Series, TRW-SS-73-02, March 1973.

Okumoto K. and Goel A. L., 1978, "Bayesian software prediction models-Classical and Bayesian inference for the software imperfect debugging model", Rome Air Development Center, Rapport RADC-TR-78-155, vol. 2, July, 1978.

Okumoto K. and Goel A. L., 1978, "Bayesian software models availability analysis of software systems under imperfect maintenance", Rome Air Development Center Rapport RADC-TR-78-155, vol. 3, July 1978.

Ottenstein K. J., 1976, "A program to count operators and operands for ANSI-FORTRAN modules, Computer Sciences Department, Purdue University, West Lafayette, Ind., Report, CSD-TR 196, June 1976.

Ottenstein L. M., 1978, "Further validation of an error hypothesis, Software Engineering Notes, vol. 3, n° 1, pp. 27-28 January 1978.

Pikul R. A. and Wojcik R. T., 1976, "Software effectiveness : A reliability growth approach", Proc. MRI Symp. Comp. Soft. Eng., April 1976, pp. 531-546.

Pimont S. and Rault J.-C., 1976, "A software reliability assessment based on a structural and behavior analysis of programs", Proc. 2nd Intl. Conf. on Software Engineering, San Francisco, pp. 486-491.

Pimont S. , 1977, "Une évaluation de la fiabilité du logiciel s'appuyant sur une analyse de la structure et du comportement dynamique des programmes", Thèse de 3ème cycle, Institut de Programmation de l'Université de Paris VI, January 1977.

Rault J.-C., 1973, "Extension of hardware fault detection models to the verification of software", in "Program test methods", W. Hetzel ed., Prentice-Hall, pp. 255-262.

Reifer D. J., and Trattner S., 1977, "A glossary of software tools and techniques", Computer, pp. 52-60, July 1977.

Rubey R. J. and Hartwick R. D., 1968, "Quantitative measurement of program quality", Proc. ACM National Conference, 1968, pp. 671-677.

Schick G. J. and Wolverton R. W., 1972, "Assessment of software reliability", Proceedings of the 11th Annual Meeting of the German Operations Research Society, Hamburg, 6-8 September 1972.

Schick G. J. and Wolverton R. W., 1978, "An analysis of competing software reliability models", IEEE Trans. on Software Engineering, Vol. SE-4, No. 2, pp. 104-120, March 1978.

Schneider V. B. and Halstead M. H., 1978, "Further validation of the software science programming effort hypothesis", NBS Technical Symposium on Tools for Improved Computing in the 80's, June 1978.

Schneider N. F., 1972, "An approach to software reliability prediction and quality control", Proc. FJCC 1972, pp. 837-847.

Schneider N. F. and Green T. F., 1975, "Simulation of error detection in computer programs", Proceedings of the Symposium on the Simulation of Computer Systems, National Bureau of Standards, pp. 101-105.

Schneidewind N. F., 1975, "Analysis of error processes in computer software", Proc. Intl. Conf. on Reliable software, Los Angeles, April 1975, pp, 337-346.

Schneidewind N. F., 1977, "The use of simulation in the evaluation of software", Computer, avril 1977, pp. 47-53.

Shooman M. L., 1972, "Probabilistic models for software reliability prediction", in Statistical Computer Performance Evaluation, W. Freiberger, ed. , Academic Press, pp. 485-502.

Shooman M. L., 1972, "Probabilistic models for software reliability prediction", 1972 International Symposium on Fault-Tolerant Computing, pp. 211-215.

Shooman M. L., 1973, "Operational testing and software reliability estimation during program development", Proc. 1973 IEEE Symp. on Computer Software Reliability, pp. 51-57, April-May 1973.

Shooman M. L., 1975a, "Software reliability measurement and models", Proceedings of the 1975 Annual Reliability and Maintainability Symposium, pp. 485-491.

Shooman M. L. and Bolsky M. I., 1975b, "Types, distribution and test and correction times for programming errors", Proc. of the International Conference on Reliable Software, pp. 347-357-.

Shooman M. L., 1976, "Structural models for software reliability prediction", Proc. 2nd International Conference on Software Engineering, October 1976, pp. 268-280.

Shooman M. L., and Laemmel A., 1977, "Statistical theory of computer programs , information content and complexity", COMPCON Fall 1977, pp. 341-347.

Sukert A. N., 1977, "A multi-project comparison of software reliability models", AIAA Conf. on Computers in Aerospace, pp. 413-421, Nov. 1977.

Sukert A. N., 1977, "An investigation of software reliability models", Proceedings of the 1977 Reliability and Maintainability Symposium, pp. 478-484.

Tal J. and Barber G. H., 1977, "Development and evaluation of software reliability estimators", Proceedings of the Hawaii International Conference on System Sciences, January 1977.

Tal J. and Barber G. H., 1977, "Estimators for software reliability", 1977 Joint Automatic Control Conférence, pp. 1067-1072.

Tariq M. A., 1977, "Programs to count operators and operands for PL/1 and FORTRAN modules, M. S. Thesis, Middle Eastern University, Ankara.

Tausworthe R. C., 1977, "Standardized development of computer software", Prentice-Hall, pp. 308-317.

Thayer T. A., Lipow M., and Nelson E. C., 1978, "Software Reliability - a study of large project reality", TRW Technology Series, Vol. 2, North Holland Publishing Company.

Trivedi A. K. and Shooman M. L., 1975, "A many-state Markov model for the estimation and prediction of computer software performance parameters", Proceedings of the International Conference on Reliable Software, April 1975, pp. 208-220.

Wagoner W. L., 1973, "The final report on a software reliability measurement study", The Aerospace Corp. Report no. TOR-0074, (412)-1, Aug. 15, 1973.

Weiss H. K., 1956, "Estimation of reliability growth in a complex system with a Poisson-type failure", Operations Research, Vol. 4, No. 5, pp. 532-545, Oct. 1956.

Williamson O. L., Dorris G. C., Ryberg A. J. and Straight W. E., 1970, "A software reliability program", Proceedings of the 1970 IEEE Symposium on Reliability, pp. 420-428.

Willman H. E. et al, 1977, "Software systems reliability-a Raytheon project history", Raytheon Company, Final Technical, Report, RADC-TR-77-188, June 1977 (AD-040992).

Zolnowski J. M. and Simmons D. B., 1977, "Measuring program complexity", COMPCON Fall 1977, pp. 336-340.

Zipf G. K., 1935, "The psycho-biology of language - an introduction to dynamic philology", MIT Press 1965 (First edition Houghton Miffin, 1935).

Zweben S. H., 1977, "Study on the physical structure of algorithms", IEEE Transactions on Software Engineering, vol. SE-3, n° 3, May 1977, pp. 250-258.

A larger bibliography is available on request from J. - C. Rault (IRIA-Domaine de Voluceau-Rocquencourt-78150 LE CHESNAY, France).

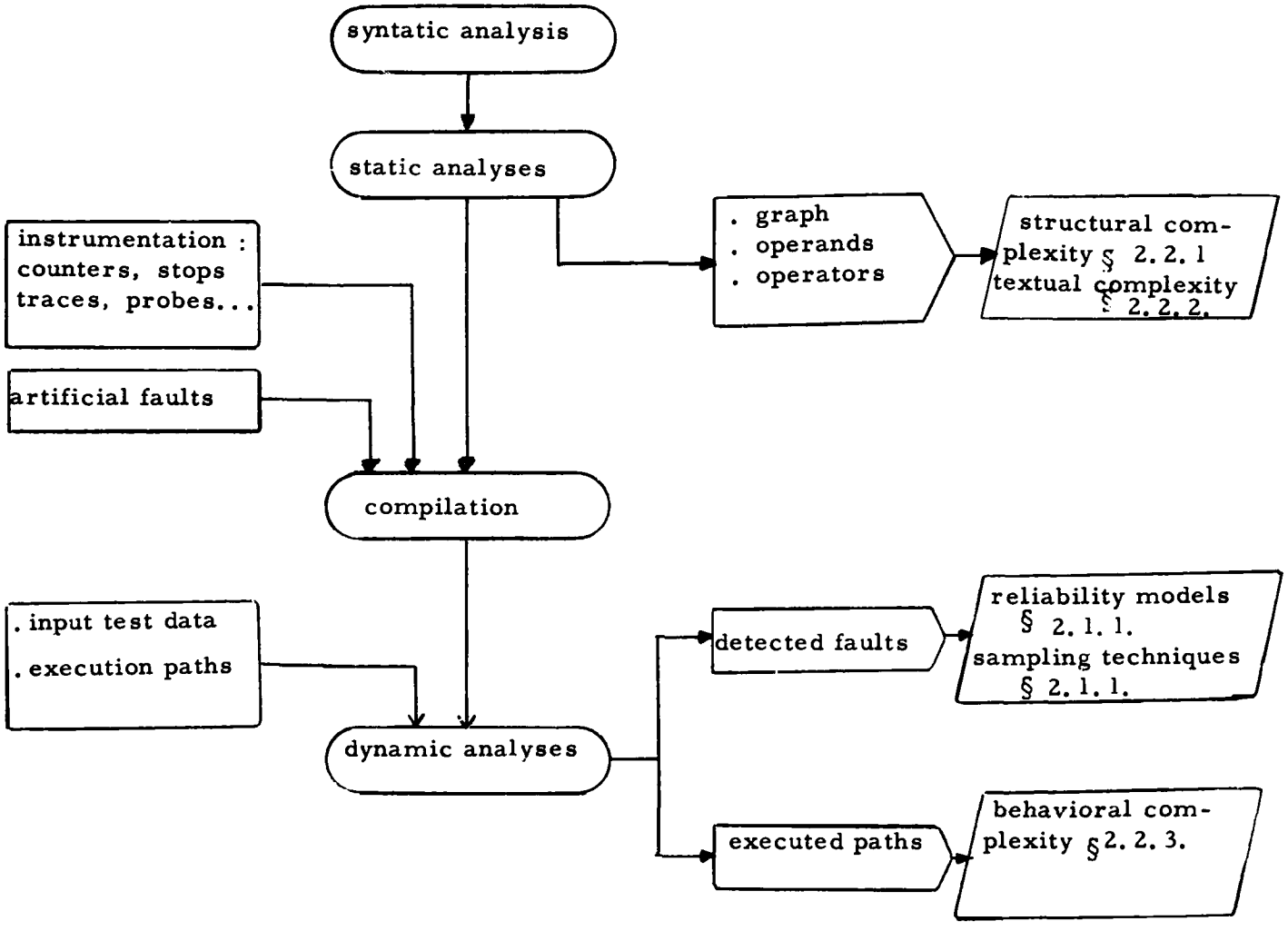


Figure 1. Analysis tools and reliability measurement techniques

An Analysis of Software Reliability Prediction Models

Alan N. Sukert
Rome Air Development Center (AFSC)
Griffiss Air Force Base, New York 13441 USA

Summary

From Aug 1974 to May 1978 a study to validate several mathematical models for predicting the reliability and error content of a software package against error data extracted from the formalized testing of four large Department of Defense software development projects was undertaken by Rome Air Development Center (RADC), Griffiss AFB NY USA. This paper will describe the results of this empirical study for three such models, the Jelinski-Moranda, Schick-Wolverton and a modified Schick-Wolverton, using both Maximum Likelihood (ML) and Least Squares (LS) methods for estimating model parameters. Model predictions are compared on a total project, functional and error severity basis. Model predictions are also compared on an errors/day and errors/week basis for defining model time intervals. From this analysis conclusions are drawn concerning the application of these models, with the principal conclusion being that model predictions should be begun when all modules in a system are ready for testing.

1. INTRODUCTION

The past several years have seen the formulation of numerous mathematical models for predicting the reliability and error content of a software system, for the purpose of permitting better tracking of software developments by providing a software manager with more detailed information regarding the status of his development. As examples, models assuming an exponential distribution of time to detect errors have ranged from early ones, such as Shooman's (Shooman, M. L., 1973), to more complicated ones, such as Musa's execution-time model (Musa, J. D., 1975). These models have been experimentally tested against available software error data by the model developers to demonstrate the applicability of each model (Wolverton, R.,1978). However, due to the limited availability to model developers of multiple software error data bases there remains serious doubt on the part of potential model users as to the general applicability and accuracy of these models.

To obtain knowledge about the applicability of these software reliability models, and to obtain better confidence in their predictions, an effort has been underway since 1974 at RADC to analyze the predictions of several software reliability models against error data obtained during the formalized testing of several large DOD and NASA software developments. This paper will present an empirical analysis of the results of this study. First, the study itself and the models and projects used will briefly be described. Then, model predictions will be presented and analyzed with some conclusions offered as to model applicability and some possible avenues for further model testing and analysis.

2. MODEL DISCUSSION

The initial goal of this in-house study was to analyze as many software reliability models as possible, using as many software error data sets as possible. As the study evolved, it became very apparent that the limiting factor was data availability. The data available to RADC consisted of data extracted from Software Problem Reports (SPRs) that were filled out by the various contractors during the formal test phases whenever a software error was detected. The only time measurements given on these forms was in terms of calendar days; thus no CPU time data was available. Since models such as Shooman's and Musa's required CPU time data, they could not be analyzed. Other models, such as Reliability Growth, were eliminated for a similar lack of the necessary data. The models examined were the Jelinski-Moranda, Schick-Wolverton, Modified Schick-Wolverton, Jelinski-Moranda Geometric De-Eutrophication, and a Modified Geometric De-Eutrophication.

Initially, predictions from these five models were analyzed against data from a large DOD command and control project on a total project basis using ML estimates for model parameters (Sukert, A. N., 1976). Next, software error data from three additional DOD projects were analyzed against the three non-geometric models, since they predicted the number of initial errors while the geometric models did not. Both ML and LS estimates for model parameters were used due to a lack of convergence of the ML estimates for some model parameters. Model predictions were obtained on both a total project basis, and error criticality basis, i.e. on the basis of a contractor-assigned assessment of the degree each error impeded execution of a test case or prohibited demonstration of a requirement (Sukert, A. N., 1977). Finally, model predictions on a functional subsystem, i.e. the contractor-assigned functional class (e.g. I/O, logical) to which the module each error was discovered in belonged, were included (Sukert, A. N., 1978).

A brief discussion of the assumptions of the three models used follows. Table 1 contains definitions of the hazard functions describing each of the three models. (Sukert, A. N., 1976) contains a more detailed description of the models and their estimate equations.

2.1. Jelinski-Moranda Model

The basic assumptions of this model are:

1. The amount of debugging time between error discoveries has an exponential distribution with an error discovery rate proportional to the number of remaining errors.
2. The failure rate between errors is constant.

Lipow's modification (Thayer, T. E., 1976) to the original formulation (Moranda, P., 1975) to allow for more than one error detected per debugging time interval was used in this study.

2.2. Schick-Wolverton Model

The basic assumption of this model is:

1. The error discovery rate is proportional to the number of errors remaining and the time spent in debugging.

The original form of the model (Wolverton, R., 1972) was again modified to permit more than one error per debugging time interval.

2.3. Modified Schick-Wolverton Model

Lipow (Thayer, T. E., 1976) suggested modifying the basic Schick-Wolverton model as follows:

1. The error discovery rate is a constant during a time interval and is proportional to the number of errors remaining, the total time previously spent in debugging, and an "averaged" error search time during the current debug interval.

Since the particular form of the hazard function for this model leads to an indeterminate form for the mean time to failure, no LS estimate was obtainable for model parameters.

Table 1. Model Equations

Model	Error Discovery Rate
Jelinski-Moranda	$z(t_i) = \phi[N - n_{i-1}]$
Schick-Wolverton	$z(t_i) = \phi[N - n_{i-1}]t_i$
Modified Schick-Wolverton	$z(t_i) = \phi[N - n_{i-1}][T_{i-1} - t_i/2]$

where: ϕ is the error discovery rate proportionality constant
N is the number of initial errors
n_i is the cumulative number of errors found through the ith debugging interval
t_i is the length of the ith debugging interval
T_i is the sum of t_j, j=1 to i

Note: z(t) is the probability of an error being discovered in a given time interval (t, t+dt) given that no error has occurred previously to time t.

3. PROJECT DISCUSSION

In this section a brief description of each of the four projects, from which the error data analyzed was obtained, is given. To maintain anonymity the Projects are referred to as Projects 1, 2, 3 and 4.

3.1. Project 1

This project (Willman, H. E., 1977) was a real-time control system for a land-based radar system written mostly in JOVIAL/J3, with the Executive and a few of the application modules written in Assembly. The error data for this project was obtained from the formal testing of all the project software, including the Executive. Formal testing began with Build Integration, where the modules were tested together with the system executive and system data base. Upon successful completion of this testing a build was formed, which then was passed on to Acceptance testing. After completion of Acceptance testing the build entered Operational Demonstration, where a series of increasingly demanding mission profiles designed to exercise the system and evaluate its response were run. It is important to note that this system was a demonstration model. i.e. it was only designed to demonstrate that a system meeting the user's requirements could be designed and built. It was never intended to become operational.

Project 1 software was developed using both top-down and bottom-up techniques (Kessler, M., 1975) and in a modular fashion. For example, module specifications were derived from the top-down starting with the system-level requirements. System integration was performed in incremental builds to check the interrelationships among the software modules and with the hardware. Dummy modules and drivers were used for testing those modules not part of a given build.

3.2. Project 2

Project 2 (Thayer, T. E., 1976) was a command and control system written in JOVIAL/J4. The software was developed in a series of modifications with each modification governed by a separate set of requirements and developed independently. The software was developed functionally, i.e. the project was divided into work units responsible for different functions. Testing of each modification was conducted in five phases starting with Development testing by the development personnel to demonstrate specific functional capabilities, test data extremes, etc. Formal testing began after Development testing with Validation and Acceptance testing. Validation testing was performed by an independent test group at the subsystem level and demonstrated the approved software performance and requirements. Acceptance testing ran a subset of the Validation tests to demonstrate specific requirements. After this testing the software underwent final Integration testing by an independent group. This Integration testing demonstrated that the applications software correctly interfaced with the operating system and system support software. Data used in this study was from the formal testing of the Project 2 applications software only.

3.3 Project 3

Project 3 (Thayer, T. E., 1976) was a large command and control project written in JOVIAL/J4. Structurally and procedurally, Project 3 was developed similarly to Project 2, with the four test phases described previously. However, the Project 3 software underwent an Operational Demonstration test phase following Integration testing, which was designed to demonstrate software functional requirements in an operational environment, using an operational data base. The data obtained from this project was from the four formal test phases (Validation, Acceptance, Integration, Operational Demonstration) of the applications software.

3.4. Project 4

This project (Fries., M. J., 1977) was a large avionics software application program written in JOVIAL/J3B and Assembly. The software consisted of five major functional areas in the operational software and two in the simulation software. Testing of this software began with Module Verification testing performed by the developer of each module. Once this testing was finished, the module was released for formal testing. Formal testing began with Inter-Module Compatability testing where the software was checked against its functional requirements as a total unit, and which was done by the software development group. After completion of this testing the software system was given to an independent system test group for Systems Validation testing, where acceptance testing for quality control purposes was performed. The data obtained for this project was from the two formal test phases and is from both the operational and simulation software for the first two versions (called blocks) of the software system.

Table 2 contains a summary of the four projects.

Table 2. Project Characteristics

	Project 1	2	3	4
Language Used	JOVIAL/J3 Assembly	JOVIAL/J4	JOVIAL/J4	JOVIAL/J3B Assembly
Size*	86780(J) 49900(A)	96931(J)	115346(J)	40640(J) 84065(A)
No. of Modules	109	173	249	69
Application	Land-Based Radar Control	Command & Control	Command & Control	Avionics
Operate Mode	Real-Time	Batch	Batch	Real-Time
Formal Testing Phases	Build-Integration Acceptance Operational Demonstration	Integration Validation Acceptance	Integration Validation Acceptance Operational Demonstration	Inter-Module Compatability Systems Validation

* - JOVIAL sizes are in number of lines of delivered source code; Assembly sizes are in number of machine instructions

4. MODEL RESULTS

In preparing the data for model input, it became apparent that some assumptions had to be made in order to use the models. First, since the data had been extracted from SPRs, the only date known was the date the corresponding SPR for each error was opened. It was decided that for consistency this would be considered the error occurrence date for each error. It was reasoned that in the vast majority of cases an SPR was filled out for a particular error the same day it occurred (subsequent discussions with project developers essentially verified this). It was further reasoned that since each SPR had to undergo the same formal configuration process, in general every SPR would take approximately the same time through this approval process. However, since in practice there are delays in closing SPRs due to priorities and test schedule demands, using the closing date for model purposes was considered impractical because of the inherent bias prioritizing placed on this data. Finally, it was assumed that the day the first SPR was opened corresponded to the first day of testing.

All data used in this study was restricted to those errors that resulted in a change to the software itself. The reason for this was that although unquestionably documentation errors are important and should be considered along with the other types of software errors, inability to interpret the Project 2 and 3 documentation errors forced the arbitrary decision to eliminate all documentation errors, thus avoiding confusion in interpreting model predictions.

The data was organized into errors per day and errors per week. The reason for this was that it was desirous to see how the use of different time frames for the data affected model predictions. Further, based on the Phase I analysis it was discovered that model predictability was affected by the start date of model prediction; i.e. in Phase I both the start date of testing and the date all modules were ready for testing were used to begin predictions. The differences in predictability were significant enough to extend this analysis to the other three projects. Since the data was subdivided functionally for all four projects, it was decided to use the date of the first SPR and the latest date that a subsystem began testing as the two dates for comparison. Thus all results will be given for both starting dates. Finally, operational data was available for Project 3 only. Thus all remarks made concerning model

predictability for Projects 1, 2 and 4 are based on conversations with project developers and on relative comparisons.

Although both N and ϕ were estimated for each model, all results will be presented in terms of the predicted number of remaining errors, i.e. the number of predicted initial errors N minus the number of errors found to date. All standard error figures given are those for the initial number of error predictions and thus the reader is asked to keep this in mind when comparing these figures. The "Day" and "Week" columns correspond to the project error data being input into the various models using errors/day and errors/week, respectively. Lastly, for notation purposes the models will be denoted as follows:

JM - Jelinski-Moranda model with parameters estimated by ML method
LJM - Jelinski-Moranda model with parameters estimated by LS method
SW - Schick-Wolverton model with parameters estimated by ML method
LSW - Schick-Wolverton model with parameters estimated by LS method
MSW - Modified Schick-Wolverton model with parameters estimated by ML method

4.1. Total Project Comparison

Tables 3-6 present that total project predictions for the five models for Projects 1-4, respectively. Note that in all the following tables the numbers in "()" are the standard errors for the initial number of error predictions, while the numbers in "[]" are the number of errors found to date for each project, i.e. the number of errors detected from the start date of model predictions until the end of formal testing of that software package.

Table 3. Project 1 Predictions

	Predicted Remaining Errors From 1-2-73 [1853]		From 3-6-73 [1769]	
Model	Day	Week	Day	Week
JM	724(91)	696(87)	499(62)	480(60)
SW	26(7)	156(21)	14(5)	108(16)
MSW	14(5)	13(5)	8(4)	7(4)
LJM	54(8)	89(10)	51(8)	87(10)
LSW	8821(271)	1110(43)	8814(278)	984(40)

Table 4. Project 2 Predictions

	Predicted Remaining Errors From 10-14-71 [212]		From 1-17-72 [189]	
Model	Day	Week	Day	Week
JM	---	---	72(29)	66(27)
SW	---	---	39(15)	56(23)
MSW	82(31)	75(29)	5(4)	3(3)
LJM	140(18)	57(10)	99(14)	46(9)
LSW	1998(189)	750(60)	1315(144)	330(31)

Table 5. Project 3 Predictions

		Predicted Remaining Errors From 6-1-73 [2191]		From 7-28-73 [1307]	
Model	Remain Errors	Day	Week	Day	Week
JM	198	1288(163)	1179(148)	233(34)	198(31)
SW		955(113)	1289(165)	193(29)	220(35)
MSW		42(9)	25(8)	10(4)	1(3)
LJM		88(10)	23(7)	81(10)	27(7)
LSW		2155(66)	685(32)	1322(52)	505(28)

Table 6. Project 4 Predictions

	Predicted Remaining Errors From 5-22-73 [1877]		From 8-27-74 [1509]	
Model	Day	Week	Day	Week
JM	---	---	650(92)	591(84)
SW	---	---	185(26)	625(90)
MSW	830(2E3)	6328(1E3)	27(7)	22(7)
LJM	176(15)	74(10)	116(12)	63(9)
LSW	9682(245)	2202(70)	234(17)	1103(115)

As can be seen from Tables 3-6, there is a considerable difference in predictability when the different start dates for model prediction are used, and this difference shows up for all four projects. For example, for Projects 2 and 4 the ML estimation procedure failed to yield a finite solution (failed to converge) for the number of remaining errors for the JM and SW models using the date testing started as the start date for model predictions, while these two models did converge when the date all models were ready for testing was used as the start date. Notice the significant drop in remaining error predictions in many instances between the use of the two different start dates. For instance, for Project 4 there is a factor of 100 drop in the remaining error predictions for the MSW model when 8-27-74 is used, as opposed to 5-22-73, as the start date. The same is true for the MSW remaining error predictions for Project 2, although here there is only a factor of 10 difference. However, there are instances, particularly for the LJM and LSW models, when there is no significant difference between the remaining error predictions using the two different start dates. Thus it would appear from the total project predictions that the difference in start dates affects the ML parameter estimations more than the LS parameter estimations. Note, though, that in almost all cases the standard error of the initial error predictions is less when the date all modules were ready for testing is used.

The daily versus weekly model predictions do not offer as convincing a pattern. In most cases there does not appear to be a significant difference between using the day and week as the time interval, although the LJM and LSW models do in some cases show some nontrivial differences. For example, for Project 2 using the 10-14-71 start date the remaining error predictions for the LJM model were 140 using the day and 57 using the week. The LSW model predicted 1315 using the day and 330 using the week for Project 2 with a 1-17-72 start date. However, in some cases the "weekly" prediction was greater than the "daily" prediction. For example, the LJM model predicted 51 remaining errors using the day and 87 errors using the week for Project 1 with a 3-06-73 start date. There was also no general pattern in the differences in standard errors. For example, for Project 1 with a 1-2-73 start date, when the day and week were used the standard error for the JM model went from 91 to 87, the SW model standard error went from 7 to 21, and the MSW standard error stayed at 5, respectively.

For the projects studied, on a project versus project basis the LSW model predicts much higher values than the other models, while the MSW model generally predicts lower values that the other models. From the actual remaining error count for Project 3 it is clear that those using 7-28-73 are much more accurate than those using 6-01-73, with some predictions being almost "too good". Since Project 2 and 3 are both command and control projects written by the same contractor, one might expect that the same pattern of model predictability holds for both projects. From Table 4 it does appear that using 1-17-72 as the start date gives realistic predictions for all models except the LSW model, while for the 10-14-71 start date only the MSW and LJM models give reasonable predictions. One would hope that the actual number of remaining errors was closer to the "1-17-72" predictions than the "10-14-71" predictions. Project 1, being a variation of a command and control project, would also hopefully give the same pattern of model predictability, and from Table 3 it appears that this is so. For Project 4, since it is an avionics development and thus significantly different from the other three projects, one would be interested in any differences in model predictability. From Table 6, one can note that the same general pattern appears with respect to the difference between the two start day predictions. However, it is interesting to note that the Project 4 predictions seem to be higher overall than those for the other three projects. The notable exception is the LSW model predictions for the 8-27-74 start date. This does suggest (at least for the limited data available) that the ML parameter estimates might not be as accurate for avionics software predictions as the LS estimates, while just the reverse holds for the Project 1-3 predictions (when the estimation procedure converged). Obviously more testing is needed to verify this hypothesis

4.2. Criticality Predictions

Tables 7-9 present model predictions by criticality category for Projects 1-3. No criticality data was available for Project 4. Also, the "Unclassified" errors for Project 3 are errors that were not assigned a category. They have been included for completeness. Finally, the "Improvement" errors for Project 1 are those errors that resulted from attempts to "improve" the system.

As can be seen from Tables 7-9, just as for the total project basis the model predictions are generally lower for the case when the date all modules are ready for testing is used as the start date than when the start of testing was used as the start date. The main exception is the LSW model predictions for Project 1. The same can be said of the standard errors of the error predictions. There is no three project pattern between the "day" and "week" predictions. For example, for Project 1 with a 1-2-73 start date, the criticality predictions for all five models for the "week" are less than or equal to the "day" predictions. However, for the medium category in Project 1 with a 1-02-73 start date the SW and LJM model predictions are higher for the "week" than for the "day". Note, however, that except for the Project 3 LSW predictions the "week" predictions for Projects 2 and 3 were lower than the "day" predictions.

As a means of project comparison, consider the "critical" error predictions for Projects 1-3. For Project 1, one would certainly expect that after several months of testing there should not be 110 critical errors left in the system. Thus it would appear that the JM, SW, and MSW predictions are more "reasonable" than the LJM and LSW predictions. However, since as stated above this project was for "demonstration" purposes, it is conceivable that many more errors were left in the system than would have been if this system was to be operational. However, certainly the LJM prediction of 206 remaining critical errors for the 1-2-73 start date is not what one would want after completion of testing.

For Project 2, the fact that only 57 errors were found but the minimum prediction for the 10-14-71 start date was 60 (for the MSW model using the week) could cause one to assume that these particular predictions are unreasonable. On the other hand, the MSW predictions for the 1-17-72 start date are realistic, if possibly somewhat high. Note the consistent lack of convergence of the JM and SW models. More will be said of this point later. Note also that the LJM and LSW predictions are still unreasonable even for the 1-17-72 start date.

For Project 3, 71 "critical" errors were found after system delivery. Using this as a guide, it would appear that the JM and SW predictions for the 6-01-73 start date, especially using the week, are fairly accurate. The same can be said of the LJM predictions for the 7-28-73 start date, especially using the day. The MSW predictions are very low for both start dates, while the LSW predictions are generally high for both start dates. We don't see the clear pattern for Project 3 that was evident for Projects 1 and 2, in that depending on the start date the LJM and LSW predictions are fairly accurate. Since Project 3 is by far the larger of the four projects in terms of number of errors but smallest of the four in terms of total test time period, this might suggest that for the smaller projects or for projects with very long test periods, the MSW model and the JM and SW models should give realistic criticality predictions, whereas for the large project developed over a short time period the LJM and LSW models might give better predictions.

4.3. Subsystem Predictions

Tables 10-13 contain the subsystem predictions for Projects 1-4, respectively. Note that these subsystems are functionally oriented; however, only for Project 1 was the exact function of each subsystem known. This limits much of what one can say about these predictions, since the type of function will almost certainly affect the predictability. Thus any comments made will be limited to general conclusions except for Project 1.

From Table 10, we see that with the exception of the LSW model the predictions for the 3-06-73 start date, and the standard errors of these predictions, are less than the corresponding predictions and standard errors for the 1-02-73 start date. Also there is no general pattern to the "day" versus "week" predictions. For example, the JM predictions for the data manipulation subsystem with a 1-02-73 start date is 169 using the day and 160 using the week; for the test driver the JM predictions are 40 using the day and 48 using the week. Note that generally the SW and MSW predictions are lower than those of the other three models (maybe too low to be realistic). If we focus on the logical and mathematical

Table 7. Project 1 Criticality Predictions

Crit. Category	Err. Found	Pred. Start Date	Mdl	Predicted Remaining Errors Day	Week
Critical	104	1-2-73	JM	17(9)	17(9)
			SW	0(1)	0(1)
			MSW	0(1)	0(1)
			LJM	206(36)	110(18)
			LSW	2660(180)	1291(746)
	101	3-6-73	JM	10(6)	10(6)
			SW	0(1)	0(1)
			MSW	0(1)	0(1)
			LJM	211(36)	114(19)
			LSW	2731(174)	1323(502)
Medium	1403	1-2-73	JM	547(79)	526(76)
			SW	22(6)	100(16)
			MSW	11(4)	10(4)
			LJM	20(5)	87(10)
			LSW	7926(301)	594(30)
	1327	1-2-73	JM	400(58)	384(56)
			SW	13(5)	72(13)
			MSW	7(4)	6(4)
			LJM	199(16)	84(10)
			LSW	7939(316)	497(27)
Low	77	1-2-73	JM	73(59)	67(53)
			SW	13(8)	13(9)
			MSW	0(1)	0(1)
			LJM	169(45)	90(19)
			LSW	1930(145)	908(477)
	77	3-6-73	JM	31(20)	29(18)
			SW	28(39)	1(2)
			MSW	0(1)	0(1)
			LJM	182(47)	99(20)
			LSW	2164(159)	1059(950)
Improve	269	1-2-73	JM	165(61)	144(51)
			SW	2(2)	3(3)
			MSW	2(2)	2(2)
			LJM	232(26)	93(12)
			LSW	3961(193)	1901(145)
	264	3-6-73	JM	94(31)	83(27)
			SW	0(1)	0(1)
			MSW	1(2)	0(1)
			LJM	240(27)	100(13)
			LSW	4145(196)	1945(150)

Table 8. Project 3 Criticality Predictions

Crit. Category	Err. Found	Pred. Start Date	Mdl	Predicted Remaining Errors Day	Week
Critical [71]++	803	6-1-73	JM	129(24)	91(19)
			SW	16(6)	91(19)
			MSW	0(1)	0(1)
			LJM	105(12)	43(8)
			LSW	227(18)	613(34)
	382	7-28-73	JM	15(6)	6(5)
			SW	0(1)	6(5)
			MSW	0(1)	0(2)
			LJM	84(11)	39(8)
			LSW	100(12)	360(28)
Medium [3]++	570	6-1-73	JM	---	---
			SW	---	---
			MSW	502(145)	415(117)
			LJM	102(12)	28(7)
			LSW	1905(94)	494(32)
	493	7-28-73	JM	---	---
			SW	---	---
			MSW	55(15)	35(12)
			LJM	84(11)	28(7)
			LSW	1079(60)	382(28)
Low [3]++	33	6-1-73	JM	6(6)	2(3)
			SW	0(1)	0(2)
			MSW	0(2)	0(2)
			LJM	78(20)	41(11)
			LSW	623(160)	326(65)
	19	7-28-73	JM	0(2)	0(2)
			SW	0(1)	0(2)
			MSW	0(2)	0(2)
			LJM	79(27)	39(13)
			LSW	606(30)	111(23)
Unclass [121]++	785	6-1-73	JM	160(30)	111(23)
			SW	13(5)	111(23)
			MSW	2(2)	0(4)
			LJM	105(32)	31(7)
			LSW	228(18)	590(34)
	413	7-28-73	JM	10(5)	2(4)
			SW	2(3)	2(4)
			MSW	0(2)	0(3)
			LJM	91(12)	33(8)
			LSW	221(20)	313(25)

Note: ++ - Actual Number of remaining errors

subsystems, which should have been heavily tested, we note that the SW, MSW , and LJM models predict what could be considered "reasonable" values for the number of remaining errors. The same holds true for the other eight subsystems, with the overall indication being that the SW and MSW models give the "best" overall predicted values. From Table 11, we see that for Project 2 the JM and SW models, when they converge, give realistic predictions. The MSW predictions, with the exception of subsystem G, are also reasonable, but they are probably low. Given the number of errors found for each subsystem, the LJM and LSW models appear to predict larger than expected values. This might, therefore, indicate that for this type of project - small with about a 1 year test time, the ML estimates may give more accurate subsystem predictions than the LS estimates. Note that the predictions for all five estimates follow the general pattern of predicting lower values for the later start date, with the exception of the LJM and LSW predictions for the Database. Interestingly, for this project the predictions for all five models follow a pattern where the "week" predictions are consistently lower than the "day " predictions; certainly more consistently than for Project 1. Finally, the System and Jovial predictions are given for completeness, but since only 1 error was found one should not expect much of the model predictions.

To analyze the Project 3 predictions from Table 12, the actual remaining errors for the subsystems are as follows:

Subsystem A - 23
Subsystem B - 19
Subsystem C - 70
Subsystem D - 21
Subsystem E - 9
Subsystem F - 15
Subsystem G - 23
Subsystem H - 12
Database - 0
Compool - 0

From Table 12 we see that in general the JM and SW model predictions are very good. The MSW predictions are in general low, while the LSW predictions are high. The LJM predictions are high in some cases, as

Table 9. Project 2 Criticality Predictions

Category	Errors Found To Date	Prediction Start Date	Model	Predicted Remaining Errors Day	Week
High	57	10-14-71	JM	---	---
			SW	---	---
			MSW	249(622)	126(193)
			LJM	121(27)	60(13)
			LSW	1176(186)	417(74)
	50	1-17-72	JM	---	---
			SW	---	---
			MSW	13(10)	8(7)
			LJM	87(20)	49(11)
			LSW	458(116)	294(49)
Medium	87	10-14-71	JM	---	---
			SW	---	---
			MSW	45(28)	38(24)
			LJM	135(25)	65(12)
			LSW	1521(70)	663(85)
	80	1-17-72	JM	38(24)	35(23)
			SW	14(9)	23(14)
			MSW	2(3)	1(2)
			LJM	76(16)	48(10)
			LSW	993(350)	188(27)
Low	68	10-14-71	JM	---	---
			SW	---	---
			MSW	6(5)	5(5)
			LJM	116(22)	63(13)
			LSW	1194(144)	597(87)
	59	1-17-72	JM	2(3)	2(3)
			SW	1(2)	1(2)
			MSW	0(1)	0(1)
			LJM	48(12)	43(10)
			LSW	786(186)	78(15)

for Subsystem E, while they are good in others, as for Subsystem A. Note that for Project 3 the predictions using the 7-28-73 start date are lower than those using the 6-01-73 start date, but not as consistently as for Projects 1 and 2. Also note that except for some of the SW model predictions the "week" predictions are lower than the "day" predictions. Interestingly, for Subsystem E the JM and SW models converge for the 6-01-73 start date but not for the 7-28-73 start date. This is different from previous model behavior and needs more investigation as to its cause.

From Table 13, we note the widespread convergence problems for the JM, SW, and MSW models. Note the large predictions for the LJM and LSW models when the other models failed to converge. Given the long test time and sparse error density for this project, this might indicate that on projects where testing is not "uniform" the LS estimates may be desirous to use. We also see that as in the previous cases the predictions using the 8-27-74 start date are much lower and more realistic, given the number of errors found, than using the 5-22-73 start date. Also, as for Project 2, there is the consistent pattern of the "week" predictions being lower than the "day" predictions.

From the above it appears that for the subsystem predictions, as was the case for the criticality and total project predictions, using the date when all modules are ready for testing gives generally more realistic predictions than using the date testing actually starts. Also, for the subsystem predictions the "day" versus "week" predictions showed a fairly consistent pattern for Projects 2-4 of the "week" predictions being lower than the "day" predictions, as was the case for the Project 1-3 criticality predictions.

5. CONCLUSIONS

In presenting these results, attempts have been made to draw general conclusions about model predictions. Since no totally consistent patterns have evolved in most cases, general conclusions are difficult. However, since in most cases Project 1 appears to deviate from patterns that are dominant for the other projects, and since Project 1 was never intended to become operational while the other three were, one could eliminate Project 1 and make conclusions on the basis of model predictions for the other three Projects. In formulating the following conclusions, I have done this only with respect to the subsystem and criticality predictions, since the pattern was so dominant for the Project 2-4 predictions.

Before stating any conclusions, however, a few words are needed about the non-convergence of the JM and SW models (and in some cases the MSW model). It had been hypothesized (Sukert, A. N., 1976) that the reason for this non-convergence was the the large number of both errors and time intervals in the data the models were applied against. However, the small number of errors and time intervals in the Project 2 data, for which the ML estimation procedure failed to converge, somewhat negates this hypothesis. However, there does appear to be a pattern of non-convergence for those data sets where the error density, i.e. the number of errors found per unit time, is oscillatory, with large time intervals between some of the detected errors. This is especially true for Project 4 and some of the large criticality categories. Thus it does appear that a significant factor in determining the convergence of the ML estimates, since the LS estimates always converged, is the rate of error detection. This would seem reasonable, since all three basic models implicitly assume a constant level of testing. An oscillatory

Table 10. Project 1 Subsystem Predictions

Subs.	Mdl	Predicted Remaining Errors From 1-2-73 Day	From 1-2-73 Week	From 3-6-73 Day	From 3-6-73 Week
Data	JM	169(52)	160(49)	106(31)	100(30)
Manip	SW	3(3)	5(3)	0(1)	2(2)
[338]*	MSW	3(3)	3(3)	2(2)	2(2)
[325]+	LJM	211(25)	67(10)	230(26)	59(10)
	LSW	4229(163)	1811(120)	4375(165)	1762(119)
Test	JM	40(15)	48(18)	47(20)	68(30)
Driv	SW	7(4)	16(7)	13(7)	30(13)
[205]*	MSW	1(2)	2(2)	2(2)	3(3)
[166]+	LJM	45(9)	70(11)	42(9)	69(11)
	LSW	2987(571)	1235(101)	2628(409)	1022(93)
Logic	JM	280(183)	262(166)	139(67)	134(64)
[185]*	SW	8(5)	10(5)	1(2)	1(2)
[182]+	MSW	4(3)	4(3)	2(2)	2(2)
	LJM	194(29)	24(6)	208(31)	20(6)
	LSW	2722(106)	1312(179)	3085(119)	1484(204)
Math	JM	320(114)	301(105)	190(59)	180(55)
[364]*	SW	24(8)	30(10)	10(5)	16(6)
[355]+	MSW	593)	4(3)	2(2)	2(2)
	LJM	224(23)	52(9)	233(24)	43(8)
	LSW	4216(255)	1799(111)	4430(259)	1823(114)
Data	JM	302(124)	300(124)	157(32)	155(52)
base	SW	662(423)	829(619)	75(23)	81(25)
[304]*	MSW	6(4)	6(4)	3(3)	3(3)
[300]+	LJM	38(8)	83(12)	32(8)	74(11)
	LSW	3345(276)	1285(91)	3783(391)	1552(107)
Conf	JM	137(78)	126(71)	114(65)	104(58)
Test	SW	36(16)	31(14)	31(14)	28(13)
[149]*	MSW	4(3)	3(3)	3(3)	2(3)
[139]+	LJM	38(9)	50(10)	139(20)	43(9)
	LSW	2199(504)	1190(126)	2156(443)	1148(127)
I/O	JM	109(96)	95(78)	45(29)	36(22)
[85]*	SW	68(49)	83(64)	11(7)	11(7)
[84]+	MSW	2(3)	2(3)	1(2)	0(1)
	LJM	56(16)	77(16)	138(30)	58(13)
	LSW	1351(200)	470(87)	1749(275)	740(128)
Con-	JM	72(33)	71(33)	44(19)	43(19)
trol	SW	1(2)	2(2)	0(1)	0(1)
[151]*	MSW	0(1)	0(1)	0(1)	0(1)
[147]+	LJM	218(37)	96(15)	224(37)	103(16)
	LSW	3086(130)	1434(295)	3178(128)	1480(352)
Com-	JM	---	---	---	---
pool	SW	---	---	---	---
[45]*	MSW	18(15)	14(13)	9(8)	6(6)
[45]+	LJM	142(43)	66(16)	128(39)	53(14)
	LSW	1420(223)	491(116)	1205(175)	58(17)
Micro	JM	1(2)	1(2)	0(1)	0(1)
code	SW	0(14)	0(4)	0(2)	0(2)
[27]*	MSW	0(3)	0(3)	0(2)	0(2)
[26]+	LJM	150(51)	93(27)	150(51)	93(28)
	LSW	1440(143)	766(113)	1439(136)	759(109)

Note: * - From 1-2-73 + - From 3-6-73

Table 11. Project 2 Subsystem Predictions

Subs.	Mdl	Predicted Remaining Errors From 10-14-71 Day	From 10-14-71 Week	From 1-17-72 Day	From 1-17-72 Week
A	JM	---	---	68(268)	56(191)
[27]*	SW	---	---	---	275(3E3)
[18]+	MSW	7(8)	4(5)	2(4)	2(4)
	LJM	93(35)	52(15)	55(27)	38(15)
	LSW	758(176)	339(86)	183(183)	180(54)
B	JM	---	---	7(6)	6(7)
'55]*	SW	---	---	0(1)	0(4)
[50]+	MSW	13(10)	8(7)	0(1)	0(6)
	LJM	105(23)	57(12)	37(10)	28(8)
	LSW	972(65)	426(79)	895(63)	348(62)
C	JM	---	---	12(9)	9(8)
[63]*	SW	---	---	1(2)	1(3)
[56]+	MSW	17(12)	13(10)	0(2)	0(2)
	LJM	105(22)	63(13)	87(18)	34(9)
	LSW	1092(64)	561(106)	945(96)	349(55)
D	JM	---	---	1(3)	1(3)
[8]*	SW	---	---	0(2)	0(2)
[7]+	MSW	6(16)	2(6)	0(2)	0(3)
	LJM	63(488)	33(26)	51(84)	31(24)
	LSW	416(103)	186(81)	321(76)	170(105)
Data	JM	3(29)	3(26)	0(2)	0(2)
base	SW	0(3)	1(9)	0(2)	0(2)
[2]*	MSW	0(3)	0(3)	0(2)	0(2)
[2]+	LJM	2(4)	16(38)	39(109)	25(60)
	LSW	2(3)	68(70)	202(133)	115(130)
E	JM	---	---	2(3)	2(3)
[19]*	SW	---	---	2(3)	2(3)
[19]+	MSW	6(8)	5(7)	0(2)	0(2)
	LJM	90(40)	57(20)	69(29)	33(12)
	LSW	709(98)	388(263)	566(121)	194(68)
F	JM	---	56(192)	0(2)	0(2)
[17]*	SW	---	---	0(2)	0(8)
[16]+	MSW	1(2)	0(2)	0(2)	0(2)
	LJM	79(29)	35(12)	54(24)	36(13)
	LSW	586(276)	207(61)	483(165)	236(73)
G	JM	---	---	---	---
[19]*	SW	---	---	---	---
[19]+	MSW	---	---	---	---
	LJM	96(28)	60(18)	81(24)	46(14)
	LSW	786(125)	419(134)	629(170)	310(87)
Sys-	JM	25(E19)	25(E5)	25(E5)	25(E5)
tem	SW	25(E5)	25(E19)	25(E19)	25(E19)
[1]*	MSW	25(E5)	25(E19)	25(E19)	25(E19)
[1]+	LJM	0(0)	0(0)	0(0)	0(0)
	LSW	0(0)	0(0)	0(0)	0(0)
Jov-	JM	25(E19)	25(E5)	25(E19)	25(E5)
ial	SW	25(E5)	25(E19)	25(E5)	25(E19)
[1]*	MSW	25(E5)	25(E19)	25(E5)	25(E19)
[1]+	LJM	0(0)	0(0)	0(0)	0(0)
	LSW	0(0)	0(0)	0(0)	0(0)

Note: * - From 10-14-71 + - From 1-17-72

error detection rate density would certainly tend to negate this assumption. However, more research is needed to verify this.

From the above analysis, then, the following general conclusions can be drawn:

1. Clearly it is better to use the date all modules are ready for testing to begin model predictions than the date testing actually begins. This pattern was almost universally consistent among all the predictions.

2. For "command and control" projects such as projects 2 and 3, it appears that the ML estimates, when they converge, give more reasonable and accurate estimates than the LS estimates.

3. For "avionics" projects such as Project 4, it appears that the LS estimates are more reasonable and accurate than the ML estimates. However, this conclusion is somewhat suspect due to the non-convergence problems for the Project 4 data.

4. For the criticality and subsystem predictions, it appears that using the week as the time interval gives more reasonable predictions than using the day as the time interval.

5. Overall, the MSW model gives the best results. However, this is a relative comparison, since many of the MSW predictions were not very good.

Table 12. Project 3 Subsystem Predictions

Subs.	Mdl	Predicted Remaining Errors From 6-1-73 Day	From 6-1-73 Week	From 7-28-73 Day	From 7-28-73 Week
A	JM	69(22)	45(15)	2(3)	1(2)
[273]*	SW	5(4)	30(10)	0(1)	1(2)
[159]+	MSW	0(1)	0(1)	0(1)	0(2)
	LJM	64(10)	53(9)	38(8)	43(9)
	LSW	1892(134)	396(32)	1248(115)	274(29)
B	JM	99(41)	68(27)	55(31)	35(19)
[201]*	SW	15(7)	52(21)	28(15)	29(16)
[111]+	MSW	4(3)	1(2)	3(3)	0(3)
	LJM	92(13)	49(9)	81(14)	36(8)
	LSW	1492(144)	337(31)	896(121)	229(28)
C	JM	86(20)	71(18)	57(21)	36(15)
[523]*	SW	32(9)	56(15)	53(20)	36(15)
[206]+	MSW	1(2)	0(2)	3(3)	0(1)
	LJM	118(13)	43(8)	97(14)	39(8)
	LSW	406(28)	491(32)	388(37)	282(27)
D	JM	49(29)	40(23)	39(35)	23(19)
[98]*	SW	22(12)	26(15)	54(54)	30(26)
[52]+	MSW	2(3)	0(2)	2(3)	0(2)
	LJM	102(17)	50(10)	80(18)	42(10)
	LSW	810(113)	202(26)	587(202)	229(37)
E	JM	104(54)	101(52)	---	---
[155]*	SW	18(10)	101(52)	---	---
[69]+	MSW	0(4)	0(4)	23(19)	22(19)
	LJM	56(10)	39(8)	57(12)	31(8)
	LSW	1408(143)	303(31)	86(17)	173(26)
F	JM	23(13)	20(12)	51(59)	33(32)
[100]*	SW	1(2)	5(5)	7(6)	7(7)
[46]+	MSW	0(2)	0(3)	1(3)	0(3)
	LJM	35(9)	38(8)	31(10)	30(8)
	LSW	1288(423)	353(42)	669(437)	237(40)
G	JM	209(79)	161(57)	37(15)	26(12)
[285]*	SW	217(83)	161(57)	15(7)	26(12)
[169]+	MSW	6(4)	2(3)	1(2)	0(2)
	LJM	114(14)	47(8)	87(13)	41(18)
	LSW	741(57)	444(35)	1089(100)	276(28)
H	JM	---	---	39(11)	32(10)
[507]*	SW	---	---	25(8)	32(10)
[463]+	MSW	45(12)	38(11)	0(2)	0(2)
	LJM	122(13)	39(8)	98(12)	36(7)
	LSW	2310(161)	751(45)	513(34)	369(27)
data	JM	1016(E5)	81(141)	---	---
base	SW	38(43)	26(29)	---	369(E4)
[42]*	MSW	6(6)	2(4)	19(26)	10(16)
[26]+	LJM	62(17)	36(10)	61(19)	19(7)
	LSW	576(261)	213(40)	303(116)	209(49)
Com	JM	---	7(20)	1(3)	1(3)
pool	SW	---	41(366)	1(3)	1(3)
[7]*	MSW	1(3)	0(2)	0(2)	0(2)
[6]+	LJM	49(39)	25(14)	15(13)	23(15)
	LSW	217(144)	60(31)	187(110)	77(55)

Table 13. Project 4 Subsystem Predictions

Subs.	Mdl	Predicted Remaining Errors From 5-27-73 Day	From 5-27-73 Week	From 8-27-74 Day	From 8-27-74 Week
A	JM	---	---	703(295)	597(230)
[466]*	SW	---	---	329(97)	784(352)
[457]+	MSW	---	---	21(7)	18(7)
	LJM	706(22)	567(12)	122(14)	73(10)
	LSW	6240(21)	2747(20)	177(18)	790(47)
B	JM	---	---	---	---
[370]*	SW	---	---	---	---
[237]+	MSW	1519(857)	748(423)	120(45)	447(200)
	LJM	598(23)	88(12)	141(18)	83(11)
	LSW	4340(281)	1415(87)	1857(289)	1097(70)
C	JM	---	---	160(171)	149(155)
[94]*	SW	---	---	107(91)	157(168)
[87]+	MSW	---	---	8(5)	5(5)
	LJM	311(40)	107(17)	121(23)	59(11)
	LSW	2921(42)	1268(73)	910(801)	384(48)
D	JM	---	---	43(17)	39(15)
[195]*	SW	---	---	4(3)	15(7)
[195]+	MSW	---	---	0(1)	0(1)
	LJM	422(27)	114(15)	67(11)	66(10)
	LSW	3919(23)	1725(75)	2244(301)	174(19)
E	JM	---	---	60(20)	52(18)
[286]*	SW	---	---	23(9)	39(14)
[257]+	MSW	---	---	2(2)	2(2)
	LJM	250(28)	115(14)	118(16)	74(11)
	LSW	4599(32)	2007(131)	2209(200)	468(38)
F	JM	---	---	60(23)	56(22)
[271]*	SW	---	---	21(9)	28(12)
[191]+	MSW	915(460)	504(308)	2(2)	1(2)
	LJM	232(25)	114(14)	96(14)	73(11)
	LSW	4177(42)	1899(109)	1988(1E3)	243(25)
G	JM	396(272)	348(220)	17(10)	16(10)
[213]*	SW	---	---	8(6)	7(6)
[85]+	MSW	11(6)	10(6)	0(1)	0(1)
	LJM	191(22)	108(14)	74(15)	60(12)
	LSW	2433(569)	1318(105)	1287(678)	94(16)

It should be stated at this point that all of the above conclusions have been based upon work performed on data available to RADC. This data is limited in both the number of different projects encompassed and the categories of data available. Thus it is possible that as more data becomes available the conclusions drawn by this work will be modified or even totally altered. Clearly this points out the need for more actual project data with the necessary information and sample size to examine a variety of different software error prediction models.

The above conclusions have also been based on a limited "pragmatic" analysis of results. Obviously more detailed analysis is needed. There are several questions pertaining to the reasons for non-convergence that must be fully investigated. More data is needed to clear up some of the model predictive behavior that was inconsistent in the predictions for Projects 1-4. Several patterns, such as the apparent greater accuracy of the ML estimates for some criticality predictions, need to be further investigated. Finally, better ways are needed to statistically determine the accuracy of model predictions. Hopefully, current efforts to develop confidence intervals for these models will help in this task.

6. REFERENCES

1. Fries, M. J., Apr 1977, "Software Error Data Acquisition", Boeing Aerospace Co., RADC-TR-77-15.
2. Kessler, M. & Tinanoff, N., Mar 1975, "Structured Programming Series, Vol. I, Programming Language Standards", IBM Corp., RADC-TR-74-300, pp. 2-2 - 2-4.
3. Moranda, P., Nov 1975, "Probability-Based Models for the Failures During Burn-In Phase", Joint National Meeting ORSA/TIMS, Las Vegas NV.
4. Musa, John D., Sep 1975, "A Theory of Software Reliability and Its Application", IEEE Transactions on

Software Engineering, Vol. SE-1, No. 3, pp. 312-327.
5. Shooman, Martin L., May 1973, "Operational Testing and Software Reliability Estimation During Program Development", Proc. of 1973 IEEE Symposium on Computer Software Reliability, Los Angeles CA, pp. 51-57.
6. Sukert, Capt Alan N., Aug 1976, "A Software Reliability Modeling Study", RADC-TR-76-247.
7. Sukert, Alan N., Nov 1977, "A Multi-Project Comparison of Software Reliability Models", Proc. of AIAA "Computers in Aerospace" Conference, Los Angeles CA, pp. 413-421.
8. Sukert, Alan N., Nov 1978, "A Four-Project Empirical Study of Software Error Prediction Models", Proc. of IEEE Computer Society's Second International Computer Software & Applications Conference, Chicago IL, pp. 577-582.
9. Tal, Jacob, Dec 1976, "Development and Evaluation of Software Reliability Estimators", University of Utah, Report No. SRL-76-3.
10. Thayer, T. A. et al, Aug 1976, "Software Reliability Study", TRW Systems Group, RADC-TR-76-238.
11. Willman, H. E. et al, Jun 1977, "Software Systems Reliability: A Raytheon Project History", Raytheon Co., Bedford Laboratories, RADC-TR-77-188.
12. Wolverton, R. E. & Schick, G. E., Sep 1972, "Assessment of Software Reliability", TRW Systems Group, TRW Software Series Report No. TRW-SS-72-04.
13. Wolverton, Ray W. & Schick, George J., Mar 1978, "An Analysis of Competing Software Reliability Models", IEEE Transactions on Software Engineering, Vol. SE-4, No. 2, pp. 104-120.

DISCUSSION

H.S.Balaban, US

It is disturbing to see orders of magnitude differences in the prediction among the three models. It is even more disturbing to see order of magnitude differences for the same model, same data with the only variable being the estimation procedure (ML or LS). Is it possible to allocate the "blame" to the models, data and estimation procedures.

Author's Reply

The "blame" is a function of the data and the models. A just completed study by Hughes Aircraft for RADC indicated that the data sets exhibited behaviour in several instances, such as increasing failure rates, that clearly violate model assumptions. Also, since the data was historical and collected after the fact, there were many "holes" in the data due to insufficiencies, omissions, etc. However, the models themselves are equally to "blame". For example, there are implicit assumptions, such as a constant level of testing and the non-allowance of errors generated during error correction, that have been shown to be unrealistic with actual developments. Newer models that allow error generation and that assume a Poisson distribution of errors detected seem to provide more accurate and more consistent predictions.

J.C.Robertson, UK

I request some clarification of the time axis used. Test hours run would appear to be the best measure, but days elapsed and weeks elapsed appear to have been used. Can we assume that similar test times occurred each day and for each of the four projects?

Also, when an error was encountered and testing was brought to a stop, was the elapsed time for error rectification discounted?

Author's Reply

The data used was historical in nature and collected after the fact. The only information available from the software problem reports was the date an error occurred or the date an error was fixed.

No data on test run hours was available. The only measure I could use was errors per day, or some multiple of errors per day. The testing was normally continued after an error was detected except in very special cases. Thus no elapsed times for error correction were discounted. Finally, there is no way to determine from the available data if similar test runs occurred each day or for each project.

P.Wust, Ge

The different programs have been programmed using different programming languages (HOL, Assembly L.)

Question: Do the prediction models differ for the different programming languages?

Author's Reply

The models, as formulated, do not differentiate between the type of programming language used. However, the models developed were done so for projects written in HOLs. The current emphasis within the US Department of Defense to require HOLs for all applications, especially in light of the status of the common DOD HOL, makes this distinguishing between HOLs and assembly language unimportant now.

ANALYTICAL SOFTWARE VERIFICATION

W.Ehrenberger / P.Puhr-Westerheide

Gesellschaft für Reaktorsicherheit
8046 Garching, F.R. Germany

ABSTRACT

Among the various methods for verifying the correctness of software, the analytical verification procedures are the most promising ones. The paper first gives an example of a manual program analysis. A FORTRAN subroutine is analyzed, which consists of 166 code lines. The result of the analysis are 61 test runs that have to be performed to show, whether this routine works correctly or not. In part 2, we deal with automatic analysis methods. A short survey is given on the various analysis methods used by automated testing tools. In the overall system level, the different kinds of applied testing strategies are mentioned. Testing in detail requires a convenient embedding of the modules to be tested. shown that testing in the module level can be done due to different methods as static analysis and dynamic analysis. The investigations and experiences are used for the development of an analyzer for the process computer language PEARL. They were financed by the German Federal Government via the project PDV.

1. INTRODUCTION

For verifying the correctness of software several methods have been proposed, discussed and tried. An overview is gained by looking through the IEEE Transactions on Software Engineering and the papers from the related conferences /1,2/. /3/ gives a very short impression on some of the different proposals and the results that can be received from practical application. It is the authors' opinion, that the cheapest way of getting an answer to the question whether a program meets its specification or not, is to analyze it. Program analysis corresponds to the way used for proving the safety of electronic hardware. In that field the related units are decomposed and each of their parts is investigated. In the software branch we have, however, a considerable advantage over our hardware colleagues: we need not worry about changes of the product due to wear out. On the other hand, programs usually perform more complex tasks than hardwired equipment does. Thus the effort to be spent for analyzing may be quite high, depending on the program complexity.

The basic idea of our kind of program analysis is to find out, which input data combinations must be offered to our program in order to test it completely. A complete test is performed, if during program operation only cases can be met, which had already been tested implicitly. One of our goals is to find out a set of test cases, which is nearly minimal. A basic requirement is a complete and detailed functional program specification, that enables us to judge after each test run, whether the gained result was correct or not.

Our paper is divided into two main parts. The first deals with manual program analysis, the second with automatic aids. In the first part we restrict ourselves to give an example of the analysis of a FORTRAN subroutine. This example can be understood by itself. Theoretical background is provided in /5/. /4/ gives also a theoretical introduction and reports on an industrial application of that method, with the help of some automatism for drawing the necessary plans.

In our second part we focus on the possibility to automize program analysis. The major published ideas of that field are discussed and some investigations of our own are added. We finish by reporting on our efforts for developing an analyzer for the german process computer language PEARL.

2. MANUAL PROGRAM ANALYSIS

2.1 Preliminary Remarks

Our program analysis assumes that the analyzed program will be translated correctly by the compiler of the language in which the treated code is written. Should we doubt this, it would be necessary to analyze compiled machine code, what, of course, is harder work than to deal with code in a higher level language. Our analysis also is no formal proof of correctness. It is just an aid to come to a judgement whether the program will fulfill its task on the same basis as such judgements were made on relay systems or on electronic safety systems. We also do not deal with the questions of arithmetic overflow or underflow, which may be quite difficult in some cases. Last not least our analysis by itself does not say anything about the completeness of our analyzed code. Only the final comparison with the specification or the evaluation of the test results will reveal whether the investigated code was complete or whether some tasks were missing.

As already pointed out, program analysis shall help to find the really meaningful test cases; it shall reveal what the program is able to perform. Hence the results of the analysis and all intermediate steps are only another description of the functions of the program itself. Therefore it is important to use some easy comprehensible description of these functions. Taking into account that most persons have a primarily visual perception, we use drawings and some primitive formula symbols that are introducedlater. Our drawings and symbols are

designed to make clear the three most characteristic aspects of any program

- the possible movements of the program counter, i.e. possible branchings
- the possible data movements
- the mixture of both, i.e. how data movements influence branchings and how branchings influence data movements.

Concerning the last point particular care is taken of investigating all dependencies between different parts of a program.

We hope to show the technique best by an example. In order to emphasize the pracitcability of our method we take a complete subroutine, written in FORTRAN. The following example was programmed at our own institute. It was used for performing a statistical test with a reactor protection computer. During programming no particular effort was taken to produce a program that is easy to analyze or well structured. The analysis brought up, that our example contains labels and code parts, that are never used - and the routine had already been executed several 100 000 times during several years.

2.2 Example

Our analysis starts from the program list. We amend this list by adding the program plan to it. The program plan makes obvious which paths through the code are possible and which data movements take place.

On the left hand side, right near the code itself we draw the individual sections (S) and the branchings between them. A section is a code part that is run purely sequentially. The individual sections are numbered. Those that are passed during each program execution are represented by full lines, those that are passed depending on input data get dotted lines. Branchings between sections are indicated similarly. In our program plan the letter "S" before the individual sections was left out. For avoinding ambiguity it was, however, used in the related formulae. In the open literature our "section" is frequently called "basic block". Sections that must be traversed under any input condition are called the "dominators" of the program graph with respect to its end node. All sections that lie between two dominators, including the first of the two, form a building block B.

SEE PROGRAM LIST AND PROGRAM PLAN PART 1

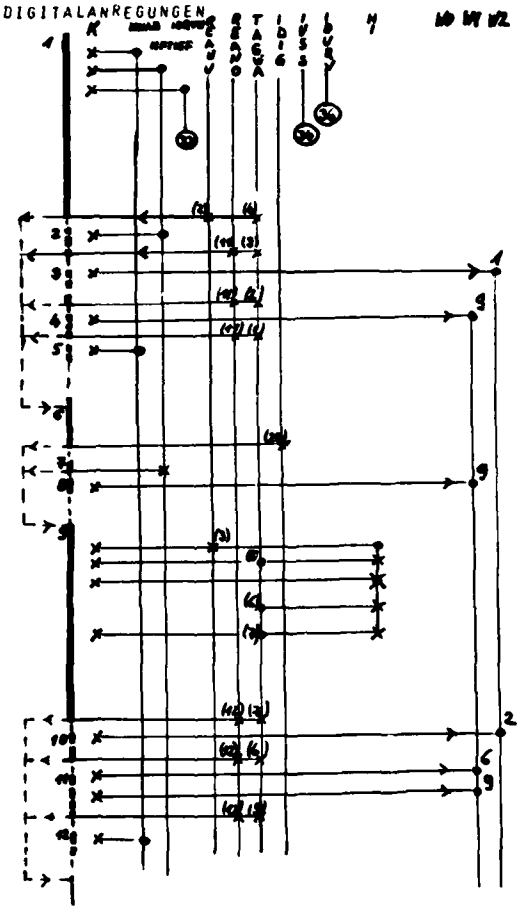
By drawing the
program plan , we find, that not all parts of our code are accessible. The inaccessible parts can be left out from further investigations. We also find, that some labels are not used. They are marked by circles. Our code contains a subroutine called SETBIT (K,i).
We know from the general understanding of the program system that this is a simple assembler routine setting bit i in word K when called.
On the right hand side of our sections and branchings we indicate the data movements. The top of each page gets the names of the variables and arrays used. The leftmost place is reserved for constants (K). We then proceed in the sequence as the related names occur in our code. Hereby we try to place outputvariables on the right hand side. Each variable gets a vertical line, starting from the point where it is used the first time and ending where it is used last. Input variables have their lines from the subroutine's starting point, output variables keep their lines to the RETURN-statement. Movements between data are given by horizontal lines, a "x" marking where the movement starts and a "." marking where it ends. "x" indicates that the related memory place is not changed at this point, "." indicates that its old contents is no longer relevant from this point on. "✖" shows that the old contents is changed. With "(i)" we indicate position i of an array, with "i" we indicate bit i of a word. In the structure plan we draw a data line exactly to the end of a section, if the related data influence the branching. A data line starts from a section, if a constant is transfered.

SEE PROGRAM LIST AND PROGRAM PLAN PART 2

We now can start with the detailed analysis of the building blocks, that leads us to the necessary test cases. Before doing this, we mention that we know from the program system, that

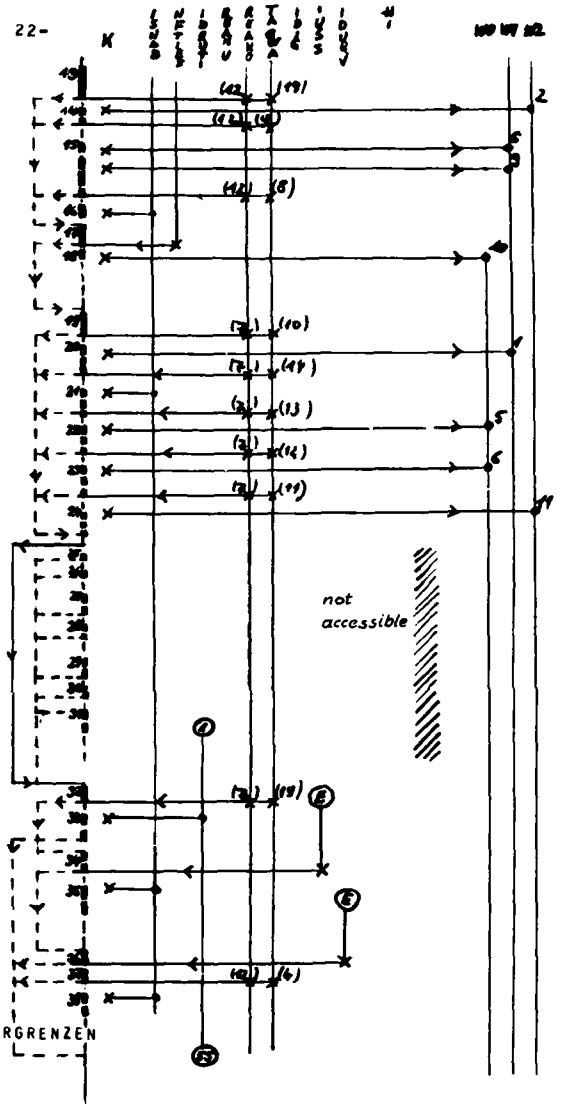
- array TAGWA contains constants, the other inputs depend from the technical process
- WO,W1 and W2 are outputs to the technical process. For some of the first building blocks the meaning of their bits is indicated.

```
1
2
3
4                SUBROUTINE IAFAN(IDIG,IAGWA,REANO,REANU,IUSS,IDURV,W0,W1,W2)
5                DIMENSION IDIG(32),IAGWA(40),REANO(13),REANU(3)
6                INTEGER W0,W1,W2
7          C
8          C     UNTERPROGRAMM ZUR BEHANDLUNG DER ANALOGEN GRENZWERTVERLETZUNGE'
9          C     UND DER MIT ANALOGWERTEN VERKNUEPFTEN DIGITALANREGUNGEN
10         C
11               ISNAB=0
12               NFTIEF = 0
13               IDRUTI = 0
14         C
15         CCCCC ENDE DER VORBESETZUNGEN,
16         C
17         C     NFUD
18         C
19               IF(REANU(2)-IAGWA(4)) 9,9,20
20            9  NFTIEF = 1
21           10  IF(REANU(11)-IAGWA(3))20,12,12
22           12  CALL SETBIT(W2,1)
23               IF(REANO(11)-IAGWA(2))20,14,14
24           14  CALL SETBIT(W1,9)
25               IF(REANO(11)-IAGWA(1))20,16,16
26           16  ISNAB=1
27         C
28         C     UBKA
29         C
30           20  IF (IDIG(29)) 30,30,25
31           25  IF(NFTIEF)30,30,26
32           26  CALL SETBIT(W1,9)
33         C
34         C     KEDU
35         C
36           30  HI = 0.8*REANU(3)
37               IAGWA(5) = 0.160 + HI
38               HI = 0.080 + HI
39               IAGWA(6) = HI
40               IAGWA(7) = HI - 0.024
41         C
42         C     NFLD
43         C
44               IF(REANU(12)-IAGWA(7))40,31,31
45           31  CALL SETBIT(W2,2)
46           35  IF(REANO(12)-IAGWA(6))40,36,36
47           36  CALL SETBIT(W1,6)
48               CALL SETBIT(W1,9)
49               IF(REANO(12)-IAGWA(5))40,37,37
50           37  ISNAB=1
```



Program List and Program Plan, part 1

```
51          C
52             40 IF(READ0(12)-TAGWA(10))50,41,41
53             41 CALL SETBIT(W2,2)
54             45 IF(READ0(12)-TAGWA(9))50,46,46
55             46 CALL SETBIT(W1,6)
56                CALL SETBIT(W1,9)
57                IF(READ0(12)-TAGWA(8))50,47,47
58             47 ISNAB=1
59             50 IF(NFTIEF)51,51,49
60             49 CALL SETBIT(W0,10)
61          C
62          C     KEDR
63          C
64             51 IF(READ0(7)-TAGWA(15))60,52,52
65             52 CALL SETBIT(W1,1)
66                IF(READ0(7)-TAGWA(14))60,53,53
67             53 ISNAB=1
68                IF(READ0(7)-TAGWA(13))60,54,54
69             54 CALL SETBIT(W0,5)
70                IF(READ0(7)-TAGWA(12))60,55,55
71             55 CALL SETBIT(W0,6)
72                IF(READ0(7)-TAGWA(11))60,56,56
73             56 CALL SETBIT(W2,11)
74             60 GOTO 70
75                IF(READ0(7)-TAGWA(18))70,61,61
76             61 IF(IDIG(28))70,62,62
77             62 CALL SETBIT(W0,5)
78                IF(READ0(7)-TAGWA(17))70,63,63
79             63 IF(IDIG(27))70,64,64
80             64 CALL SETBIT(W0,6)
81                IF(READ0(7)-TAGWA(16))70,65,65
82             65 IF(IDIG(26))70,66,66
83             66 CALL SETBIT(W2,11)
84          C
85          C     SCHNELLSCHLUSS HAUPTWAERMESENKE
86          C
87             70 IF (READ0(7)-TAGWA(19))71,72,72
88             71 IDRUTI = 1
89                GOTO 80
90             72 IF (IUSS) 75,75,73
91             73 ISNAB = 1
92          C
93          C     SCHNELLSCHLUSS DURCHDRINGUNGSVENTI
94          C
95             75 IF (IDURV) 80,80,76
96             76 IF (READ0(12)-TAGWA(4)) 80,77,77
97             77 ISNAB = 1
98          C
99          C     KEFU, ZUERST OBERGRENZEN, DANN UNTERGRENZEN
100         C
```



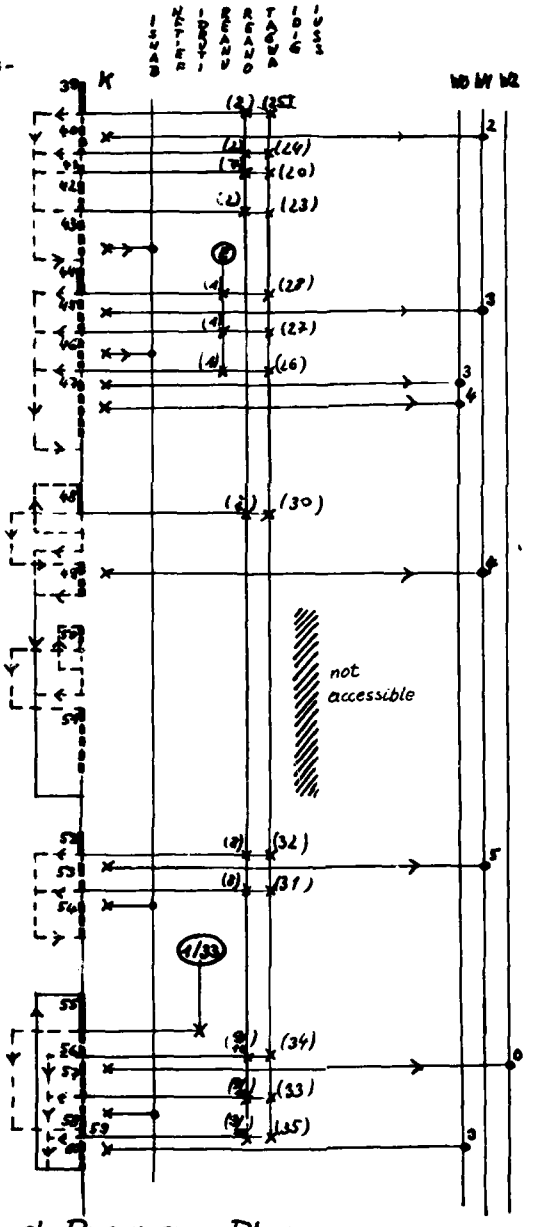
Program List and Program Plan, part 2

We use the following

Nomenclature:

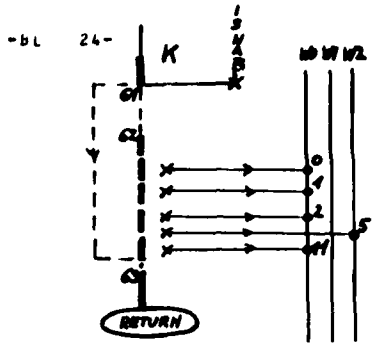
Bi↑	after building block i ran
Si↑	after section i ran
Si↓, Bi↓	section i, building block i do not run
Si↑↑	section i is running
Si → Sj	instruction counter changes from Si to Sj
FIELDi	contents of component i of array FIELD
VAR,i	bit i of variable VAR
|	exclusive or
Si(A,B,...)	in Si a decision or a mapping is influenced by the variables A,B,...
Wi,n or Wi bit n	bit n of variable Wi is set to 1


```
101            80 IF(REANU(2)-IAGWA(25))100,81,81
102            81 CALL SETBIT(W1,2)
103               IF(REANU(2)-IAGWA(24))100,82,82
104            82 IF(REANU(7)-IAGWA(20))100,83,83
105            83 CONTINUE
106            90 IF(REANU(2)-IAGWA(23))100,91,91
107            91 CONTINUE
108            92 ISNAB=1
109         C
110           100 IF(REANU(1)-IAGWA(28)) 101,101,110
111           101 CALL SETBIT(W1,3)
112               IF(REANU(1)-IAGWA(27)) 102,102,110
113           102 ISNAB = 1
114               IF(REANU(1)-IAGWA(26)) 103,103,110
115           103 CALL SETBIT(W0,3)
116               CALL SETBIT(W0,4)
117         C
118         C     FD01 BIS FD04
119         C
120           110 DO 111 I=3,6
121               IF(REANU(I)-IAGWA(30)) 111,113,113
122           111 CONTINUE
123               GOTO 120
124           113 CALL SETBIT(W1,4)
125               GOTO 120
126         C     KOMMT NICHT DRAN WEGEN VERZOEGERUNG
127               DO 114 I=3,6
128               IF(REANU(I)-IAGWA(29)) 114,115,115
129           114 CONTINUE
130               GOTO 120
131           115 CALL SETBIT(W0,8)
132               CALL SETBIT(W0,3)
133               ISNAB=1
134         C
135         C
136         C     SBDR
137         C
138           120 IF(REANU(8)-IAGWA(32)) 130,121,121
139           121 CALL SETBIT(W1,5)
140               IF(REANU(8)-IAGWA(31)) 130,122,122
141           122 ISNAB=1
142               CALL SETBIT(W0,4)
143         C
144         C     KDR1,2
145         C
146           130 DO 139 I = 9,10
147               IF (IDUKUI) 131,131,135
148           131 IF(REANU(I)-IAGWA(34)) 139,132,132
149           132 CALL SETBIT(W2,0)
150               IF(REANU(I)-IAGWA(33)) 139,133,133
151           133 ISNAB=1
152           135 IF(REANU(I) -IAGWA(35)) 139,136,136
153           136 CALL SETBIT(W0,3)
154           139 CONTINUE
155         C
```



Program List and Program Plan, part 3

```
156          900 IF(ISNAB)950,950,920
157        C
158        C     SCHNELLABSCHALTUNG
159        C
160          920 CALL SETBIT(W0,0)
161              CALL SETBIT(W0,1)
162              CALL SETBIT(W0,2)
163              CALL SETBIT (W2,5)
164              CALL SETBIT (W0,11)
165          950 RETURN
166              END
```

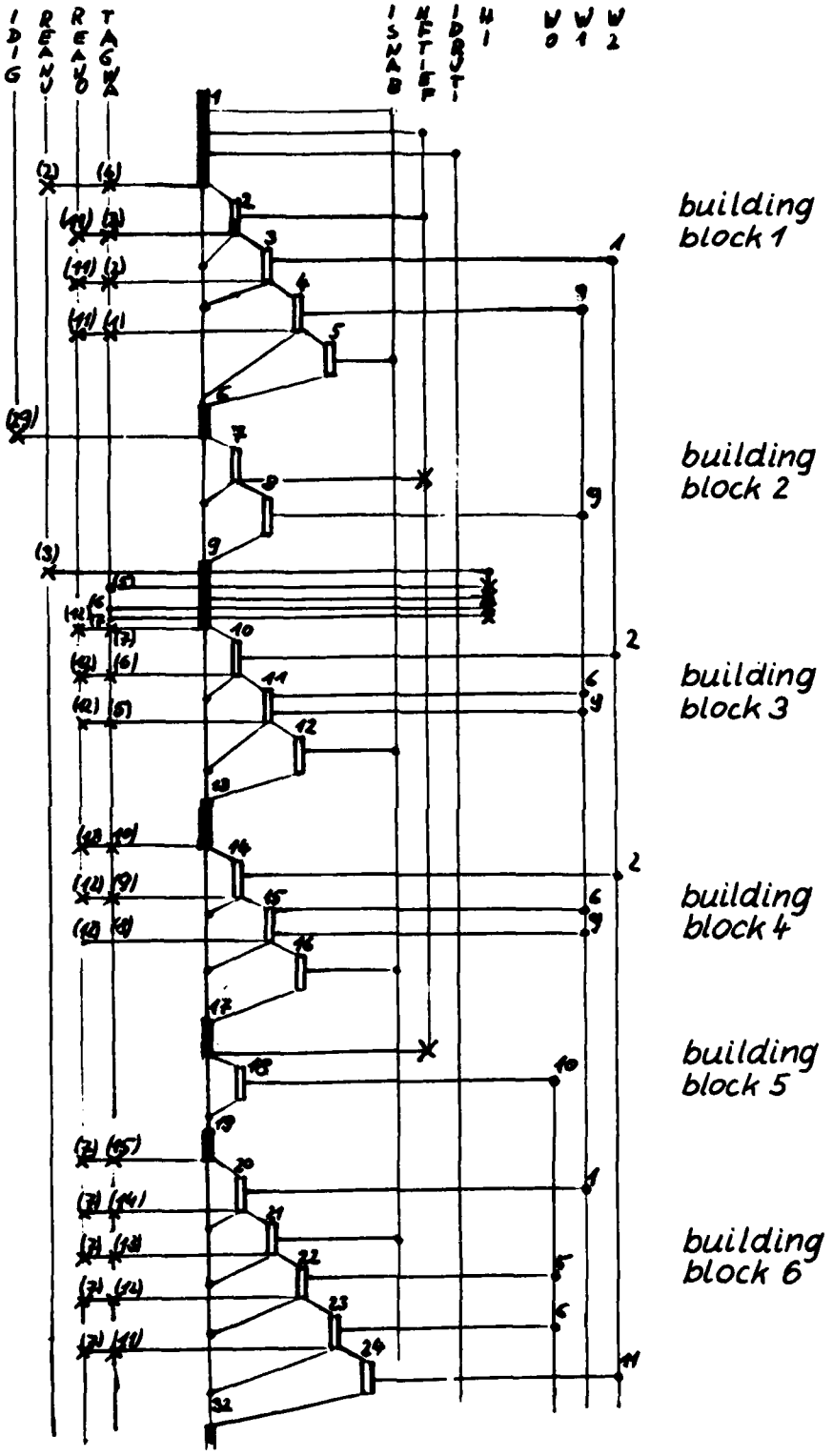


Program List and Program Plan, part 4

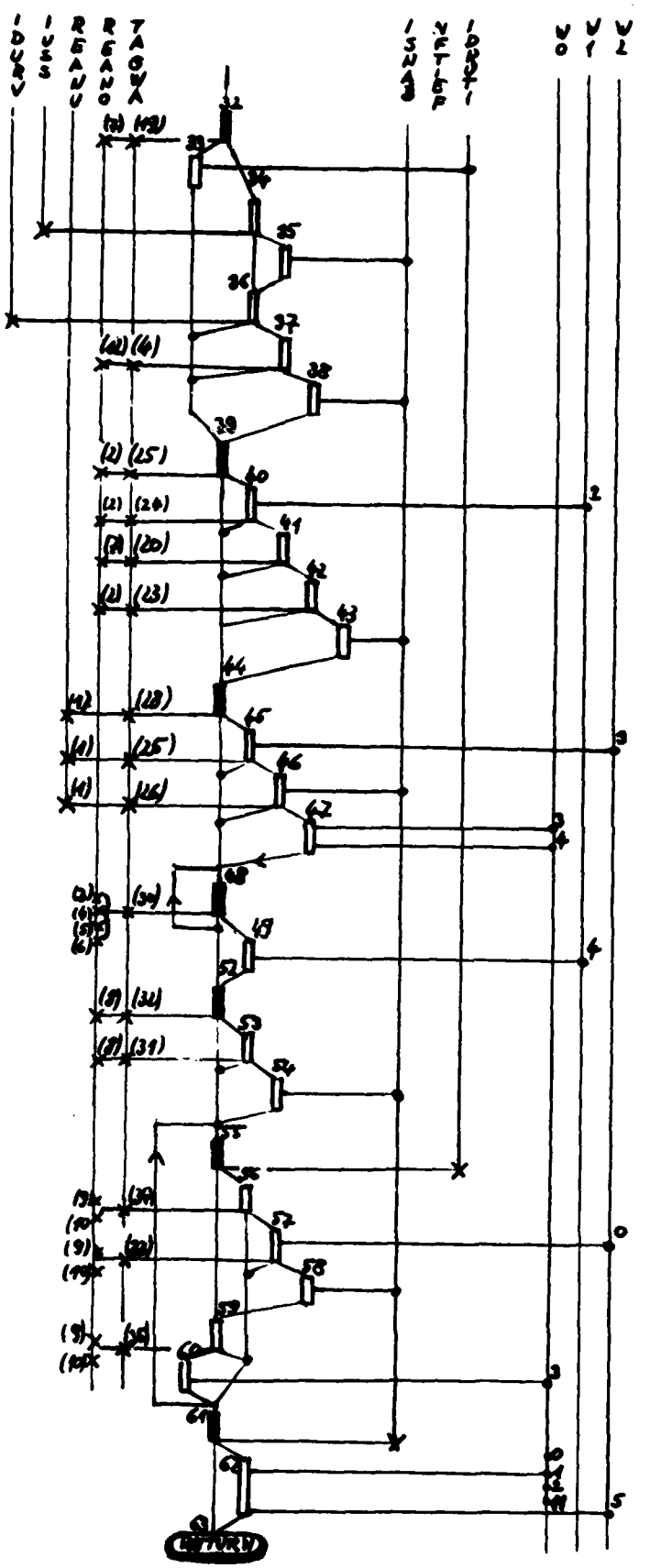
Starting from our program plan we draw our structure plan, which shows the essential relationships more clearly. For further structuring we collect all sections between those which are passed during each run into one building block B.

The block plan is an even rougher representation. It gives an overview over the use of the individual input variables by the individual building blocks. All data movements go from left to right unless indicated differently by an arrow. From this we get some important intermediate results:

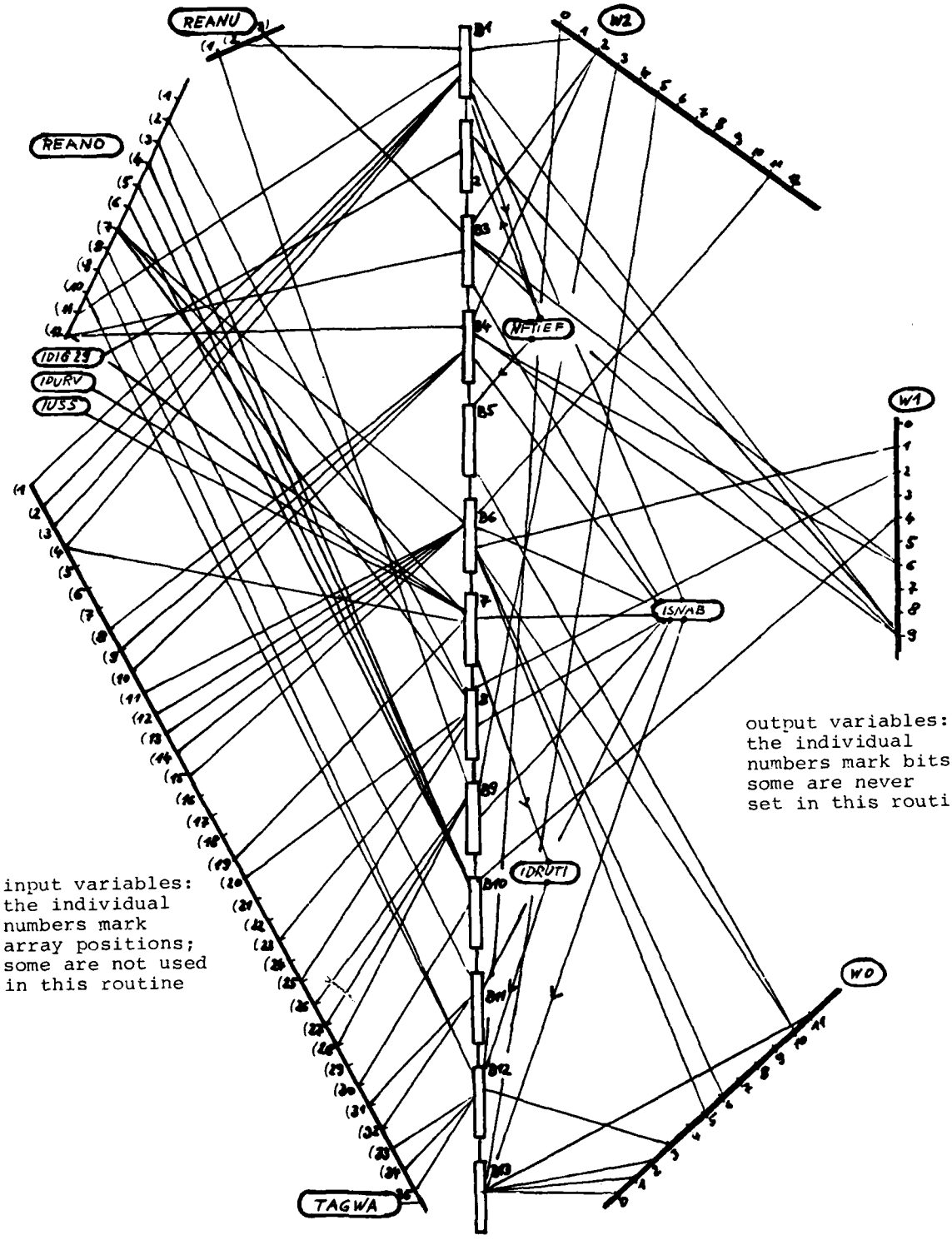
- Apart from couplings through the auxiliary variables NFTIEF, ISNAB and IDRUTI each building block can be tested separately.
- NFTIEF couples B1 with B2 and B5
- IDRUTI couples B7 with B12
- ISNAB couples almost each building block with B13.



Structure Plan, part 1



Structure Plan, part 2



input variables:
the individual
numbers mark
array positions;
some are not used
in this routine

output variables:
the individual
numbers mark bits;
some are never
set in this routine

Plan of Building Blocks

Building block 1

From program list: ISNAB , NFTIEF, IDRUTI intiated with O.

From structure plan: Assignments to W2,1, W1,9 and ISNAB with constants, depending on branchings, additionally assignment to NFTIEF.

From program list: B1 : NFTIEF = O|1,
W2,1 = _|1,
W1,9 = _|1,
ISNAB = O|1, (1.1) _.= as initiated

Structure plan: S1→S2|S1→S6 = S1(REANU2 , TAGWA4)
S2→S3|S2→S6 = S2(REANO11 , TAGWA3)
S3→S4|S3→S6 = S3(REANO11 , TAGWA2)
S4→S5|S4→S6 = S4(REANO11 , TAGWA1) (1.2)

From program list: All decisions depend on differences and comparisons of ≷ O. When = O, branching to the following section.

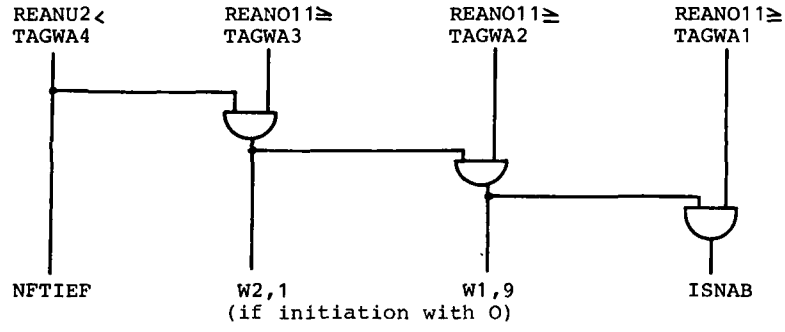
From (1.2), structure plan and program list:

S2↑↑, if REANU2 < TAGWA4
S3↑↑, if S2↑ ∧ REANO11 ≧ TAGWA3
S4↑↑, if S3↑ ∧ REANO11 ≧ TAGWA2 (1.3)
S5↑↑, if S4↑ ∧ REANO11 ≧ TAGWA1

From program list and structure plan:

S2↑, NFTIEF = 1
S3↑, W2 bit1 = 1 (1.4)
S4↑, W1 bit9 = 1
S5↑, ISNAB = 1

From (1.3) and (1.4):



Logic plan 1.1

From structure plan: ISNAB is set in S5, S12, S16, S21, S35, S38, S43, S46, S54, S58.

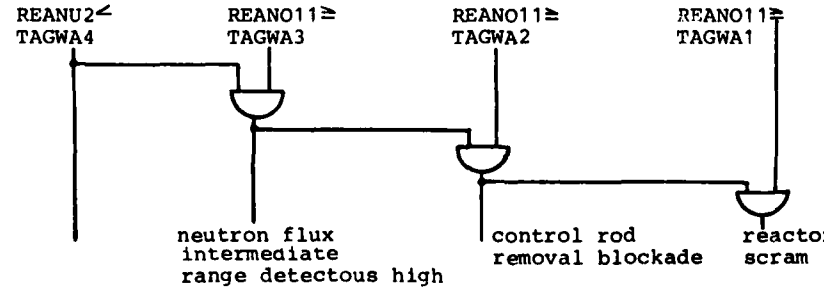
From program list: Assignments all with 1.

Building block 13

S61→S62|S61→S63 = S61(ISNAB)
ISNAB ≧ 1 : S62↑↑
S62↑ : WO,O = 1
WO,1 = 1
WO,2 = 1
WO,11 = 1
W2,5 = 1
ISNAB = 1 ≡ reactor scram (13.1)

Building block 1 continuing

From(13.1) and the meaning of the other bits logic plan 1.1 changes to:



Logic plan 1.2

Test cases:

REANU2 > TAGWA4 ∧ REANO11 ≥ TAGWA3
REANU2 < TAGWA4 ∧ REANO11 < TAGWA3
- " - ∧ " ≥ " ∧ REANO11 < TAGWA2
- " - ∧ - " - ∧ " ≥ " ∧ REANO11 < TAGWA1
- " - ∧ - " - ∧ - " - ∧ REANO11 ≥ TAGWA1

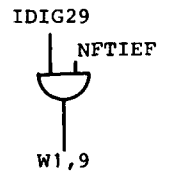
REANU2 < TAGWA4	REANO11 ≥ TAGWA3	REANO11 ≥ TAGWA2	REANO11 ≥ TAGWA1
0	1		
1	0		
1	1	0	
1	1	1	0
1	1	1	1

emply places:
value irrelevant

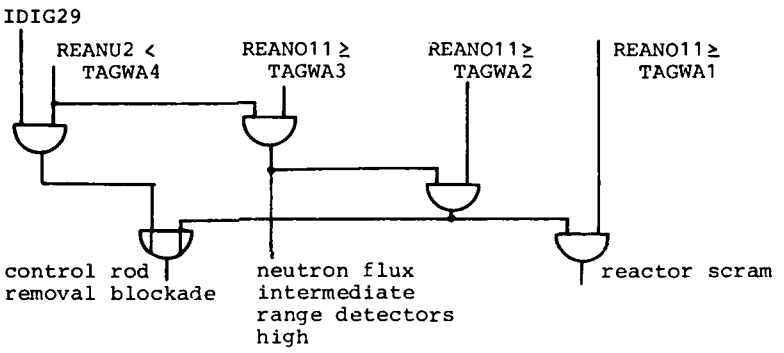
Table 1.1

Building block 2

From structure plan and program list: S7↑↑, if IDIG29 > 0
S8↑↑, if S7↑ ∧ NFTIEF > 0



Logic plan 2.1



connected with logic plan 1.2:

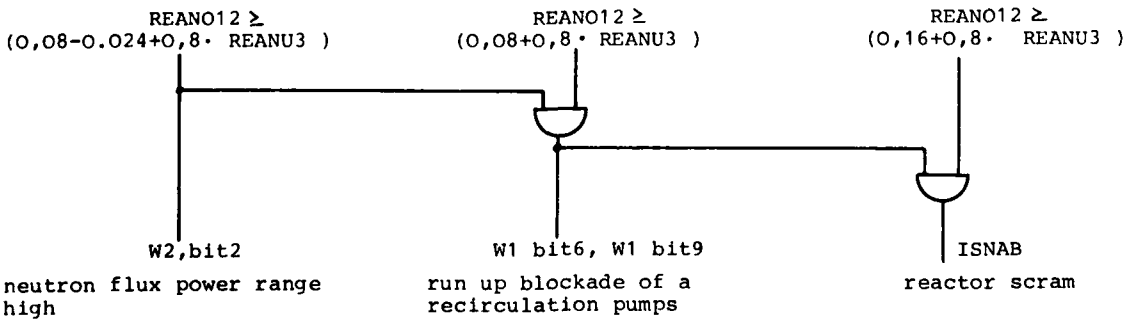
Logic plan 2.2: Neutron flux intermediate range detectors

Building block 3

S9↑↑: TAGWA5 = 0,16 + 0,8· REANU3
TAGWA6 = 0,08 + 0,8· REANU3 (3.1)
TAGWA7 = 0,08 - 0,024+0,8· REANU3

S9→S10∣S9→S13 = S9(TAGWA7 , REANO12)
S10→S11∣S11→S13 = S10(" 6 , ") (3.2)
S11→S12∣S11→S13 = S11(" 5 , ")
S12→S13

S10↑↑, W2,2 = 1
S11↑↑, W1,6,9 = 1 (3.3)
S12↑↑, ISNAB = 1



Logic plan 3.1

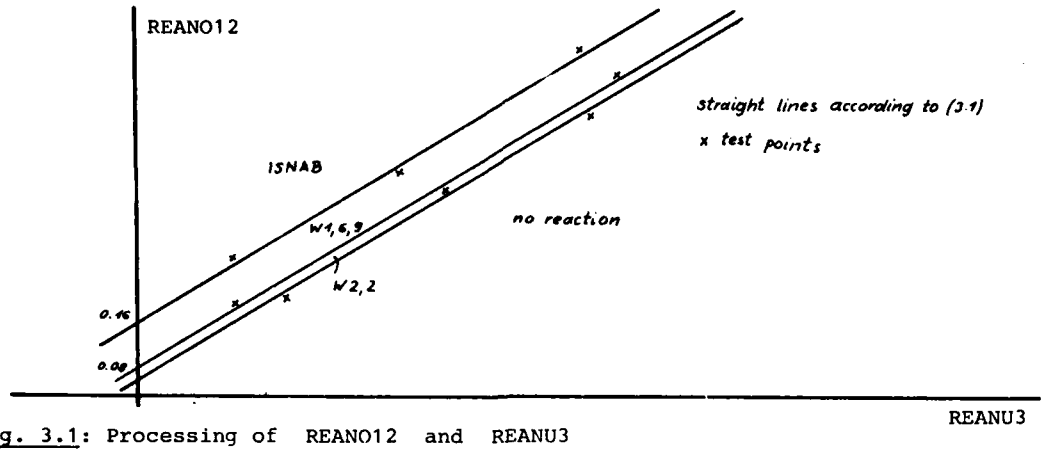


Fig. 3.1: Processing of REANO12 and REANU3

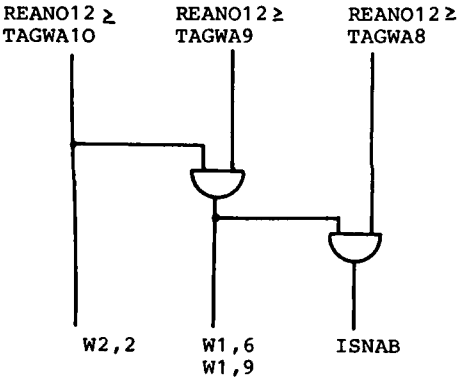
Building block 4

From structure plan

S13→S14 ! S13→S17 = S13(REANO12, TAGWA10)
S14→S15 ! S14→S17 = S14(REANO12, TAGWA9)
S15→S16 ! S15→S17 = S15(REANO12, TAGWA8)
S16→S17

and: S14 ↑ , W2, bit 2
S15 ↑ , W1,6;W1,9
S16 ↑ , ISNAB = 1

From this and program list:



REANO 12 ≥ TAGWA10	REANO 12 ≥ TAGWA9	REANO12 ≥ TAGWA8
0		
1	0	
1	1	0
1	1	1

Table 4.1: necessary test cases

Logic plan 4.1

Building block 5

From structure plan:

S17 → S18 ! S17→S19 = S17(NFTIEF)
S18 ↑↑ : WO, 10

So: if NFTIEF = 1: WO,10

From logic plan 1.1:

REANU2 < TAGWA4

WO,10

Logic plan 5.1

REANU2< TAGWA4
0
1

Table 5.1
test cases

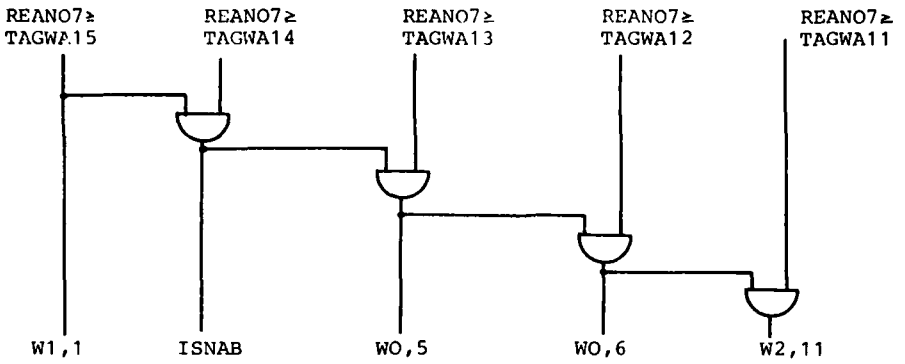
Building block 6

From structure plan:

S19→S20!S19→S32 = S19(REANU7, TAGWA15)
S20→S21!S20→S32 = S20(REANU7, TAGWA14)
S21→S22!S21→S32 = S20(REANU7, TAGWA13)
S22→S23!S22→S32 = S22(REANU7, TAGWA12)
S23→S24!S23→S32 = S23(REANU7, TAGWA11)
S24→S32

S20 ↑ , W1,1
S21 ↑ , ISNAB
S22 ↑ , WO,5
S23 ↑ , WO,6
S24 ↑ , W2,11

From this and program list



Logicplan 6.1

REANO7≥ TAGWA15	REANO7≥ TAGWA14	REANO7≥ TAGWA13	REANO7≥ TAGWA12	REANO7≥ TAGWA11
0				
1	0			
1	1	0		
1	1	1	0	
1	1	1	1	0
1	1	1	1	1

Table 6.1

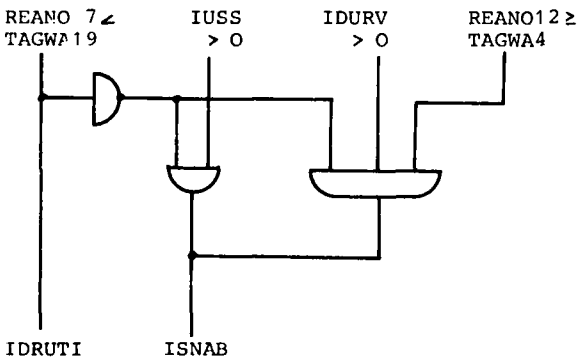
Building block 7

From structure plan

S32→S33|S32→S34 = S32 (REANO7, TAGWA19)
S33→S39
S34→S35|S34→S36 = S34 (IUSS) IUSS input parameter
S35→S36
S36→S37|S36→S39 = S36 (IDURV) IDURV input parameter
S37→S38|S37→S39 = S37 (REANO12,TAGWA4)
S38→S39

S33↑, IDRUTI = 1 from program list
S35↑, ISNAB
S38↑, ISNAB

From this and program list



REANO 7≤ TAGWA 19	IVSS > 0	IDURV > 0	REANO12≥ TAGWA4
1	1	1	1
0	1	0	1
0	0	1	0
0	0	1	1

Table 7.1

Logicplan 7.1

Test of IDRUTI included in B12.

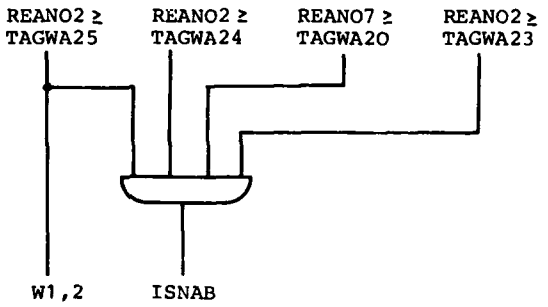
Building block 8

From structure plan

S39→S40|S39→S44 = S39 (REANO2, TAGWA25)
S40→S41|S40→S44 = S40 (REANO2, TAGWA24)
S41→S42|S41→S44 = S41 (REANO7, TAGWA20)
S42→S43|S42→S44 = S42 (REANO2, TAGWA23)
S43→S44

S40↑, W1,2
S43↑, ISNAB

From this and program list



REANO2 ≥ TAGWA25	REANO2 ≥ TAGWA24	REANO2 ≥ TAGWA20	REANO2 ≥ TAGWA23
0			
1	0		
1	1	0	
1	1	1	0
1	1	1	1

Table 8.1

Logicplan 8.1

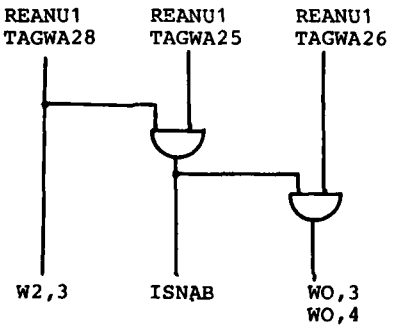
Building block 9

From structure plan

S44→S45|S44→S48 = S44(REANU1,TAGWA28)
S45→S46|S45→S48 = S45(REANU1,TAGWA25)
S46→S47|S46→S48 = S46(REANU1,TAGWA26)
S47→S48

S45↑, W2,3
S46↑, ISNAB
S47↑, WO,3,4

From this and program list



REANU1≤ TAGWA28	REANU1≤ TAGWA25	REANU1≤ TAGWA26
0		
1	0	
1	1	0
1	1	1

Table 9.1

Logic plan 9.1

Building block 10

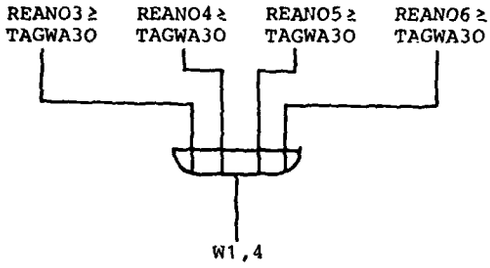
From structure plan and program list:

loop over S48, number of runs 1,2,3 or 4
during the loop REANO3, REANO4, REANO5, REANO6 are read.

S48→S49|S48→S52 | S48 S48 = S48 (REANO3, REANO4, REANO5, REANO6, TAGWA 30)
S49→S52

S49↑, W1,4 S49↑↑, depending on decision in repetition 1, or 2, or 3, or 4.

From this and program list



REANO3≥ TAGWA30	REANO4≥ TAGWA30	REANO5≥ TAGWA30	REANO6≥ TAGWA30
0	0	0	0
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Table 10.1

Logic plan 10.1

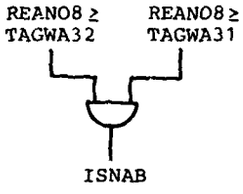
Building block 11

From structure plan:

S52→S53∣S52→S55 = S52 (REANO8, TAGWA32)
S53→S54∣S53→S55 = S53 (REANO8, TAGWA31)
S54→S55

S54↑ , ISNAB

From this program list:



REANO8≥ TAGWA32	REANO8≥ TAGWA31
0	
1	0
1	1

Table 11.1

Logic plan 11.1

Building block 12

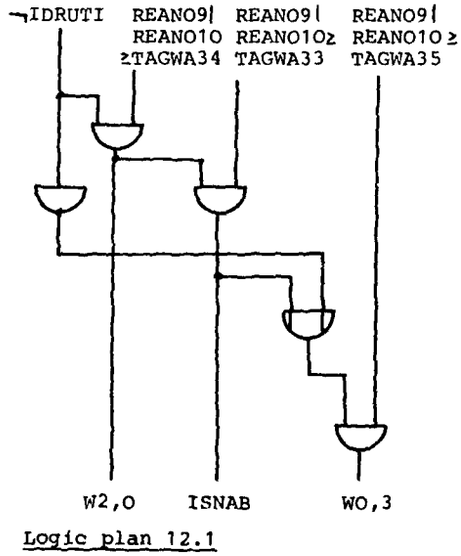
From structure plan and program list

Loop from S60 to S55, 2 runs constantly.
1st runs ≙ I = 9 concerns REANO9
2nd runs ≙ I = 10 concerns REANO10

Inner part of the loop:

S55→S56∣S55→S59 = S55(IDRUTI)
S56→S57∣S56→loopend = S56(REANO9, REANO10,TAGWA34)
S57→S58∣S57→loopend = S57(REANO9, REANO10,TAGWA33)
S58→S59
S59→S60∣S59→loopend = S59(REANO9, REANO10,TAGWA35)
S60→S61
S57↑ , W2,0
S58↑ , ISNAB
S60↑ , WO,3

From this

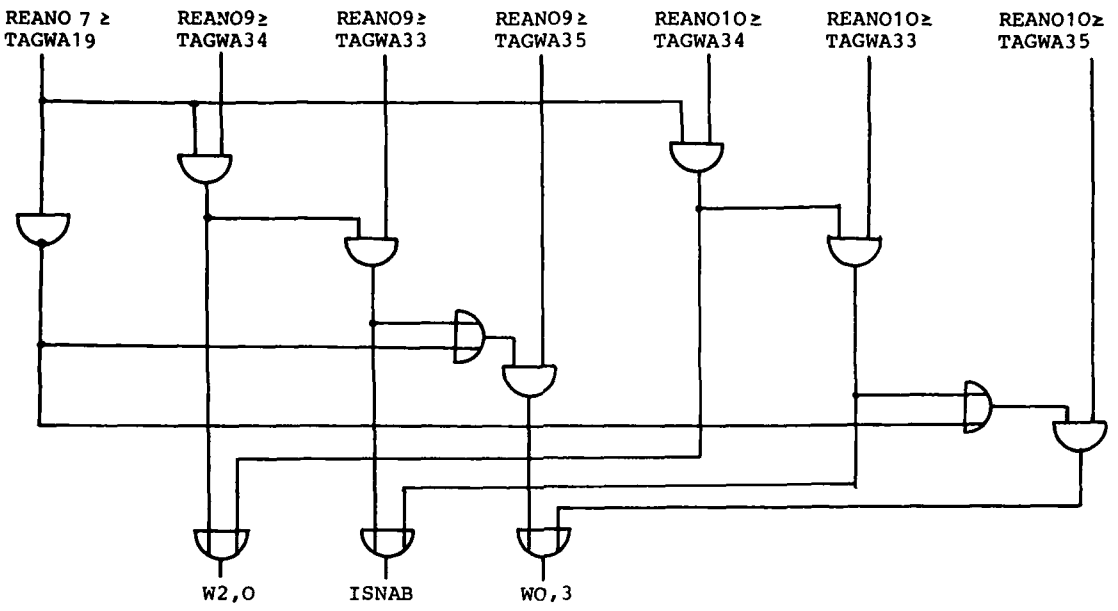


Logic plan 12.1

From B1 and B7: IDRUTI = {0,1}

From structure plan: the two loop repetitions perform a logic "or".

From this, from logic plan 7.1, logic plan 12.1 and program list:



Logic plan 12.2

REANO7 ≥ TAGWA19	REANO9 ≥ TAGWA34	REANO9 ≥ TAGWA33	REANO9 ≥ TAGWA35	REANO1O ≥ TAGWA34	REANO1O ≥ TAGWA33	REANO1O≥ TAGWA35
1	O			O		
1	1	O		O		
1	O			1	O	
1	1	1	O	O		
1	O			1	1	O
1	1	1	1	O		
1	O			1	1	1
O			O			O
O			1			O
O			O			1

Table 12.1

Building block 13 has been treated after B1.

2.3 Conclusive Remarks

By adding up the lines of our tables, we find that we need 61 test cases to show whether our subroutine contains programming errors or not. The correct runs give us confidence in the tested subroutine but they proof not its correctness in the meaning of formal logic. With some effort we could even reduce this number slightly. The analysis of the previous routines and program parts of the system must show how these cases can be provided.

The result of our analysis is quite encouraging as we easily estimate that the number of conceptually possible paths is more than 48 millions!

Our analysis took approximately 1 to 2 man weeks time. The subroutine was easy to analyze: We had no general WHILE loops and no array addressings depending on other array contents. If such more complicated structures are used, the analysis requires more logic reasoning. Some certainly remain unfeasible by our method. General WHILE-loops e.g. require induction proofs / 6/. Any manual analysis is error prone, as it is a human activity. Therefore it is recommended to have some redundancy in the notation and to go back to the original code quite frequently.

We hope our manual analysis method will provide some practical help for those readers, who have to verify programs whose failing could cause greater damages. The manual program analysis remains important as long as the automatic aids are still unsufficient.

3. AUTOMATIC ANALYSIS OF SOFTWARE

3.1 Preliminary Remarks

Manual analysis of software as a technique for its verification and for improving its correctness is a useful method for comparatively small sized software modules. Sometimes, however, quite large units must be analysed. /7/, for instance, mentions a real-time software package of about 2.000.000 statements. As is well known, the costs for guaranteeing the reliability of the performance of software products are at least in the dimension of the costs for their development. To provide a reliability evaluation of packages of the above mentioned size therefore would be extremely expensive if tried by hand.

In this connexion even simple automatic aids for testing software gain importance, even if their results may be far away from a complete proof of correctness. The development of automatic testing tools started with two approaches: one came from the intention to dispose of a tool that allows to decide whether even large programs are reliabile, the other continued the line of making more comfortable compilers by including code optimisation and debugging routines. Both contributed to the present state of the art of automatic testing tools. In the following, some aspects of the different methods of automatic testing tools will be outlined.

3.2 Test Strategies

For testing entire software packages, different strategies can be used. Before testing, the packages are always decomposed into modules that are investigated in detail.

3.21 Top-Down Strategy

Before the start of testing, the hierarchy of the program modules is evaluated. This hierarchy is represented by the call graph or invoking tree. In this graph the most embracing module is represented by the root node. The invoked modules follow in the hierarchy of their CALLs (fig.3.1).
The top-down strategy starts by testing the most embracing module first. The modules invoked by it must be simulated during this test as 'stubs' to which data can be transferred and that can transfer data to the calling module when the control is given back. In the next step of the top-down strategy the called modules are tested. They possibly invoke submodules by themselves that are tested in the following step and so forth.

3.22 Bottom Up Strategy

In case of bottom up strategy, the sequence of module tests is interchanged; those modules are tested first, that are at the lowest level in the call-hierarchy. The role of the calling module is hereby simulated by the 'driver'. One task of the driver is to provide data for the modules under test /8/.

3.23 Aspects of Both Strategies

The simulation of the environment, i.e. the embedding of a module to be tested is a subtle work in both cases (see fig.3.2). The data passing the interfaces of the module under test and its environment should be carefully selected concerning their significance and the bounds of their specification. A hybrid way of testing using both top-down and bottom-up techniques can be advantageous, as individual peculiarities of the software package can then be used effectively /8/. The test strategies deal with the partitioning of a system into its modules. Besides the verification of modules by applying formal logic, a rough classification into two methods of module testing can be made, namely

- static analysis and
- dynamic analysis.

Static analysis is regarded to be more fundamental; because its carefully interpreted results build the basis for dynamic analysis. On the other hand, many errors that are not found by static analysis can be discovered by dynamic analysis, like certain anomalies of control flow and data flow; moreover, as shown in /9,10/, static analysis has principally also its limits, otherwise the halting problem of the Turing machine would be resolvable.

3.3 Static Analysis

3.31 Lists and Tables

A lot of examinations of the source code are performed by compilers, as syntactical checks or lexical analysis. An important task for automatic static analyzers is to assemble further lexical informations in the form of tables and lists. In future, that could become a compiler's job. The following shows a collection of such lists:

- list of blocks, procedures, tasks
- cross-reference list of objects, (showing additionally whether variables are defined or referenced)
- list of input/output variables
- list of life spans of objects
- list of branching statements
- list of do-loops
- list of real-time statements
- statistics on frequencies of different kinds of statements

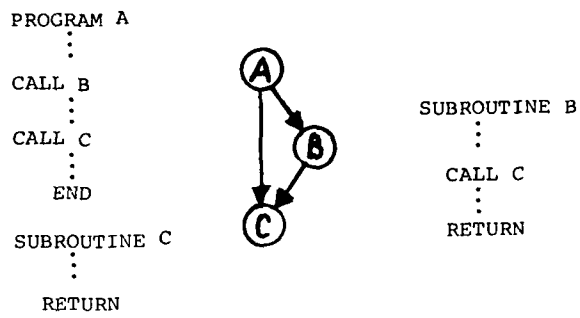
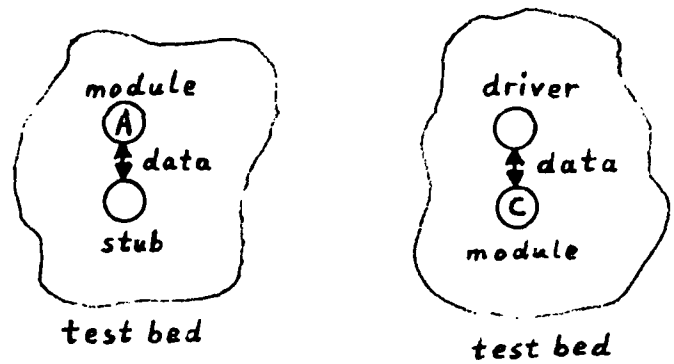


Fig. 3.1: Simple call graph



First step of top down testing corresponding to the call graph in fig.(3.1); test sequence of modules: A,B,C.

First step of bottom up testing corresponding to the call graph in fig.(3.1); test sequence of modules: C,B,A.

Fig. 3.2: Modules under test and their environment

- list of constructions unprocessed by the analyzer.

3.32 Structural Analysis

Much information can be gathered by the structural analysis of a module. In a similar way as the call graph at the system level, the control flow graph (cfg) can be obtained automatically. The cfg is represented by G(N,E) where N is a set of nodes $n_1,n_2,\ldots,n_k$ and E is a set of ordered pairs of nodes, called the edges $(n_{i1},n_{i2}),(n_{i3},n_{i4}),\ldots,(n_{im-1},n_{im})$. The nodes represent the statements of a program, the edges show the transfer of control from node n_{im-1} to node n_{im}. A path in G is a sequence of nodes $n_{i1},n_{i2},\ldots,n_{ik}$ such that every adjacent pair (n_{im-1},n_{im}) is in E. Nodes which have more than one entering edge are called multiple inedge nodes, similarly a multiple outedge node has more than one leaving edge. The cfg of a program is the representation of the total of all paths of a program notwithstanding their possibly very large number.* Similarly to hand analysis certain control flow anomalies like unreachable parts of the code or jumps into do-loops can easily be detected by evaluating the cfg. In addition the cfg is fundamental for finding certain sets of paths which should be traversed during the dynamic test phase. Normally, a test cannot execute all possible paths because their number is too large. Reasonable testing requires convenient sets of paths which - for example -

- cover all edges
- cover all inedge-outedge pairs of all multiple entry and/or multiple exit nodes
- reach at least all nodes.

Various classes of paths for covering a cfg are outlined in /11/. Fig.(3.3) shows a cfg and three corresponding sets of paths as mentioned above.

Finding out input data that lead to the execution of certain paths is one of the most important tasks of analyzing systems. It should be noted that certain kinds of errors such as missing control transfers can not be discovered by structural analysis. A lot of work was spent in investigating the properties of cfg's /12,13/. In general, two classes of cfg's can be distinguished: reducible cfg's and irreducible cfg's. The latter are much more difficult to manipulate. Loop constructions and subroutines with more than one entry point have irreducible cfg's. In this case, an exhaustive analysis of the cfg results in more difficult operations such as node splitting e.a. /12/.

3.33 Path Related Analysis

Other work to be done by static analysis is to find path conditions and to make symbolic executions.

Branchings

A branching in a program appears in its cfg as a multiple outedge node. It is possible to decompose branchings with more than two branches into sequences of binary branchings, even if, therefore, we have either to alter the source code or to resolve it in a lower level language. A binary branching is considered as a conditional program statement of the form IF e THEN b_1 ELSE b_2. The evaluation of e causes either the execution of b_1 if v(e)= true or the execution of b_2 if v(e)= false.

Path conditions

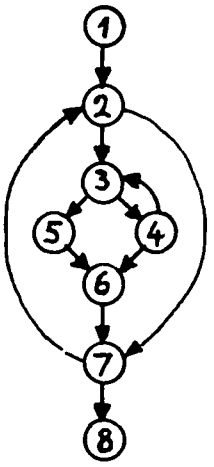
For each path in the cfg, its 'path condition' pc can be found if the pc is first set to 'true' and, following the path from its starting point, each subsequent Boolean branching expression e_i or its denial $\neg e_i$, according to the transfer of control to either statement b_{i1} or b_{i2} on this path, is linked to pc by a conjunction until the termination statement of the path is reached. If a variable that is part of a path condition is assigned during program execution, its symbolic value must be substituted. Each path has its unique pc.

3.34 Symbolic Execution

Each path can be executed symbolically. That means, that the values of input variables of a program are represented by symbolic values instead of numerical or other actual values. Along a selected path, variable values and pc subexpressions are collected and, if necessary, repeatedly substituted by existing symbolic values or expressions until the final point or another interesting point of the path is reached. At this point, the value of the variables and the path condition pc is given by formulae depending on symbolic input variables only. The pc expression shows the conditions for path execution. It describes the possible values of input data that cause the execution of the selected path. By evaluating the formulae of the variables, the correctness of a path execution can be shown in relation to its specification. This is valid for all numerical or other actual values of input variables satisfying the particular pc. The symbolic execution is a powerful method for finding errors in programs /14/. Unfortunately, since the final expressions are so complex they are often unwieldy and, therefore, difficult to understand and to resolve. It is also possible to start the symbolic execution in reverse order, i.e. to begin with an interesting

---

* The cfg of our example in chapter 2.2 corresponds to that part of the structure plan that contains the individual sections and the connection between them. The control flow graph is similar to a structure plan without data movements.



Path sets

1 covering all nodes:

1,2,3,4,6,7,8

1,2,3,5,6,7,8

2 covering all edges:

1,2,7,2,3,4,3,5,6,7,8

1,2,3,4,6,7,8

3 covering all combinations of inedge-outedge pairs:

1,2,7,2,3,4,3,5,6,7,2,7,8

1,2,3,5,6,7,8

1,2,3,4,3,4,6,7,8

Fig. 3.3: Control flow graph (cfg)
(example taken from /12/)

point or the terminal point of a path and to go back to its starting point.*

Infeasible Paths

A further problem is to decide whether a path is infeasible or not. An infeasible path is a path that never can be executed for any possible combination of input data. Unfortunately, the problem of finding all infeasible paths of a program can not be solved in general. We can only determine whether any particular path is feasible or infeasible. At present it is not always practicable to gain all symbolic path expressions of an arbitrary program by automatic tools but it becomes less tedious to find out input data for a special path by using partial results that were found automatically.

3.35 Data Flow Analysis

Static analysis also enables us to have a look at the flow of data. This is helpful for detecting errors and anomalies like a missing initialisation of a variable, a reference of a variable that is undefined, or a definition of a variable that has not been referenced after the last definition. The analysis of data flow has its roots in the design of code optimizing compilers. A data flow analysis routine is, for instance, a part of a FORTRAN-analysing system /15/. Like the symbolic execution, data flow analysis allows to find errors or anomalies in sequences of data assignements. Although it does not produce such powerful results as the symbolic execution, it has the advantage of being easier implemented.

Path Expressions

In general, the use of a variable during the run of a program can be described by three states /16/:
a variable can be - undefined (u)
- defined (d)
- referenced(r)
when a statement is executed. In an assignment statement such as

```
A = B + C
```

the variable A is defined and both B and C are referenced. A variable index is considered as a referenced variable, too:

```
A(I) = A(I) + B.
```

Here A,B, and I are referenced and A is defined. As a simplification, an indexed variable is generally regarded as a simple variable that changes its value if any of its components changes its value. A variable is undefined if its value is undefined or outside the block where it is declared. For instance, a variable that was declared, but not initialised, is undefined. Local variables in blocks are undefined when the termination statement of the block was executed.

During the execution of a path the state of a variable may change; the sequential notation of its states in the temporal succession is a path expression pe; for instance pe=drdrd for the variable A results from the following sequence of statements:

```
A:= B + E
A:= A * 3
B:= A
A:= 1
```

Path expressions containing partial sequences as

```
ur  (undefined and then referenced)
du  (defined and then undefined)
dd  (defined and defined again)
```

are called anomalous; the corresponding part of the source code is suspected to be possibly faulty. The partial sequence ur within an expression of a certain path, for instance, points to a runtime error arising during an execution of a program that runs the path.

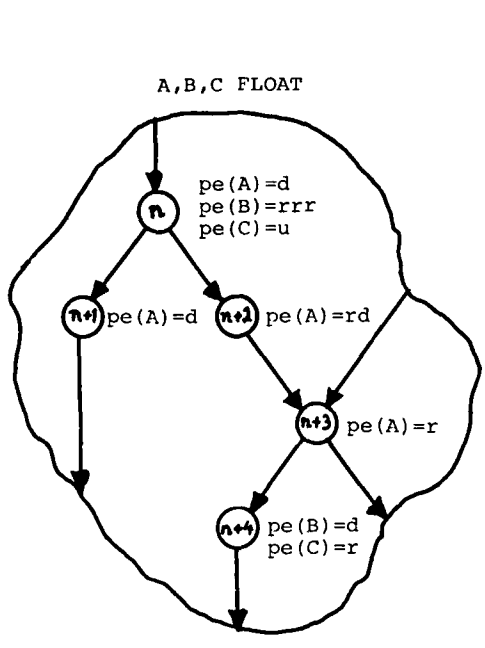
Data flow analysis starts with the determination of path expressions for each variable in each node of the cfg. Therefore, it is convenient to look to a cfg containing only sections** as nodes. An anomaly like ur detected within a section causes a run-time error for all paths containing this section. However, the sequence ur within an arbitrary path expression does not necessary reveal an error; it might be a part of a path expression of an infeasible path. In this case, the two elements of the anomalous partial sequence correspond to different nodes of the cfg. Supposed that all nodes of the cfg are reachable from its entry node and that they are meaningful, all data flow anomalies detected within the nodes belong to feasible paths.

The partial path expressions found for each node are the elements for the expressions of the whole path. It is necessary to detect the anomalies resulting from linking the partial path expressions of nodes, too. That means, the first state of a variable in the partial path expression of a node n is to be combined with the last state of the variable of the path expressions of each path entering node n. This combination then is checked whether it

---

* In our hand analysis we used this symbolic execution in a simple form in connection with building block 3. The problem was not too difficult there, because we need not consider data dependencies from other building blocks.

** What we call "sections" is mostly called "basic block" in the open literature.



source code

```
        :
n1  { FREE C;
      A=B*B;
      IF B>10 THEN DO;

n+2 { A=A+1;
      GOTO LAB;
      END;

n+1 { ELSE DO;
      A=1;
        :
        :
n+3 { LAB : IF A< 0 THEN
n+4 {       B=C;
      ELSE DO;
        :
```

n_o	path	path expressions	kind of path
1	n,n+1	pe(A)=dd,pe(B)=rrr,pe(C)=u	feasible
2	n,n+2,n+3	pe(A)=drdr,pe(B)=rrr,pe(C)=u	"
3	n,n+2,n+3,n+4	pe(A)=drdr,pe(B)=rrrd,pe(C)=ur	infeasible
4	n+3	pe(A)=r,pe(B)=∅,pe(C)=∅	feasible
5	n+3,n+4	pe(A)=r,pe(B)=d,pe(C)=r	"

anomalies: pe(A)=dd on path 1 (feasible)
pe(C)=ur on path 3 (infeasible)

Fig. 3.4: Part of a cfg, corresponding parts of source code (notation PL/1-like), and corresponding path expressions.

is harmful. In an analogous way, the first state of a variable of each directly succeeding node is combined with its last state within node n and checked. Fig.3.4 shows a part of a cfg, the corresponding statements, and the partial path expressions of the nodes for the variables A,B,C.

Automatic Detection of Data Flow Anomalies

From the field of global optimisation, two algorithms can be used for detecting data flow anomalies /16/: the algorithms for solving the

- live problem and the
- availability problem.

The live problem is to decide whether the value of a variable or expression must be saved for a later use during a program's execution. The availability problem rises when the same value of an expression is used several times within a program run. For saving execution time, it is desirable to take the existing value of an expression instead of computing it again: if it is still available. For the solution of both problems, a set of tokens is established in relation to the cfg of a program and its variables and/or expressions /16/.

For each node n of the cfg, three disjunct subsets of tokens can be generated: gen(n), kill(n), and null(n) whose elements are the identifiers of values and whose union is the set of tokens: $gen(n) \cup kill(n) \cup null(n) = tok(n)$. $\alpha \in gen(n)$ means that α is to be computed within node n; $\alpha \in null(n)$ says that α was not affected within node n; $\alpha \in kill(n)$ can be used to show that α is to be overridden and/or is to become unknown within node n. The three sets describe all actions performed on the elements satisfactorily. The solution of the live problems shows whether on one or more of all paths leaving node n the next action concerning a certain value is that it is to be computed or not.

The solution of the availability problem shows whether on each path entering node n, the last action concerning a certain value was that it was to be computed or not.

A more generalized form of the two solutions results from substituting 'object' for 'value' and 'attached by an universal (given) action' for 'computed'. Then, the two solutions can be used for detecting anomalous data flow: therefore, 'values' is substituted by 'identifiers of variables' and 'attached by an universal (given) action' by

- defined or
- referenced or
- undefined.

The repeated application of the modified 'live' and 'available' algorithms according to certain rules /16/ results in the detection of data flow anomalies. Afterwards it is examined whether they belong to feasible paths or not.

3.4 Dynamic Analysis

Software testing while running a program is called dynamic analysis. The simplest way of dynamic analysis is to run a program with certain input data and to check whether the output data coincide with the expected data. For testing large software packages, care must be taken for embedding the modules to be tested in a suitable environment and for preparing meaningful data. This problem was mentioned earlier in connection with system test strategies. More sophisticated systems use the results of the statical analysis to prepare meaningful test runs, especially such results that facilitate the choice of distinct paths and their input data.

Test systems can be established for interactive operation; then, for the operator, it is possible to insert breakpoints, to alter data with regard to the compliance of certain path conditions, and to look at intermediate results.

3.41 Instrumentation

Another way of testing is to instrument programs, i.e. to insert special statements ('probes') for different purposes as for

- producing 'post-mortem' data bases that contain the execution's history
- counting the executions of groups of statements
- detecting data flow anomalies as mentioned in chapter 3.35.

Instrumenting a program for creating 'post-mortem' data bases is standard in available debugging tools, frequently under the name 'trace'. Recording the number of executions of statements or groups is particularly important if the input data for covering sets of paths are unknown. The dynamic detection of data flow anomalies is related to the executed paths and, therefore, easier to achieve than by static analysis. Another way of testing a module is to run it as a 'black box' without knowledge of its internal structure; for reasonable testing, the input data should then be chosen under statistical aspects.

3.42 Providing Input Data

As mentioned previously, the selection of appropriate test data is a very problematic task to be done with different respects as to create data for

- semi automated tests (e.g. to create data for node/edge covering sets of paths as mentioned in chapter 3.32)
- black box tests.

Creating input data for running all paths, mostly has its practical limits in the great number of paths. The most relevant resources to determine test data for path related testing are the results of the static analysis of path conditions.

3.5 PEARL Analyzer

The considerations of this paper on automatic testing tools were made in connection with the development of an analysing system for modules written in PEARL, a project that is supported by BMFT. As pointed out earlier at the level of module testing, many relations exist between the methods of analyzing and those used for special requirements in compilers. From this came the decision to choose an intermediate language as the starting point for the analysis of modules. The intermediate language is generated during the first passes of a particular PEARL-compiler. It is the basis for the analysis to be performed by the PEARL-analyser. The analyser is scheduled for carrying out some of the static analytical methods as mentioned above, starting with the structural analysis of modules. We have in mind to provide a tool that is particularly helpful for safety assessments. In connection with the analyzer a set of rules will be provided whose observation during program development will facilitate the analysis. Particular care is taken to provide a convenient presentation of the analysis results.

4. CONCLUSION

We tried to cover some important aspects of both analyzing programs by hand and by automatic means. Concerning hand analysis we focused on practical aspects, in automatic analysis we put theoretical questions in front. We deliberately omitted such important aspects as using formal logic or induction proofs. The general feeling is, that arbitrary WHILE-loops cannot be successfully treated without induction proofs. In software packages for engineering requirements, however, it is frequently possible to avoid loops whose repetition number depends on intermediate calculation results from the loop. We therefore believe that the methods discussed in this paper cope with a considerable amount of software verification questions arising in industrial computer applications. We are quite optimistic that even high requirements on software reliability can be met by applying these methods or tools; this seems to hold above all if during programming a certain discipline had been followed and tricks or tricky constructs had been avoided.

We also did not cover the important area of program specification. We emphasize that any formal verification is only possible by comparing the results of an analysis or a test to a requirement. The more complex the tasks to be fulfilled by our software are, the more difficult it is to judge intuitively whether a particular result is correct or not. Concerning our hand analysis example we were in the lucky situation of having the program specification in a form similar to the logic diagrams that were derived. In such a case performing a practical test has only marginal importance.

We feel that in the future software testing and the related theory and tools will increase in significance. In the long term automatic aids clearly will win. For the ultimate steps of analysis and for treating some nasty program constructs, hand analysis, however, may remain to be of some interest.

5. ACKNOWLEDGEMENT

The authors express their thanks to their colleague, Dipl.-Math. Josef Märtz for his careful reading of this manuscript and for many important improvements.

6. REFERENCES

/1/ Second International Conference on Software Engineering, San Francisco, Cal., Oct.1976, IEEE Cat.No. 76CH1125-4C

/2/ Third International Conference on Software Engineering, Atlanta, Georgia, may 1978, IEEE Cat.No. CH1317-7C

/3/ W. Ehrenberger:
Systematische und statistische Methoden zur Gewinnung von Zuverlässigkeitskenngrößen für Programme, VDI Bericht No. 307, 1978

/4/ KWU-Berichte RE 23/013/77, RE 023/046/77, RE 23/047/77 zum Förderungsvorhaben P 4.1/6; F-KWU/1, Kennwort: Prozeßrechnersoftware Teile 1,2 und 3, 1977, PDV, KFK Karlsruhe.

/5/ W. Ehrenberger, G. Rauch, K. Okroy:
Program Analysis - A Method for the Verification of Software for the Control of a Nuclear Reactor, in /1/

/6/ C. Reynolds, R.T. Yeh:
Induction as the Basis for Program Verification, in /1/

/7/ C.G. Davis:
Testing Large Scale, Realtime Software Systems, Proceedings of the Infotech State of the Art Conference on Software Testing, Regent Centre Hotel, London, Sept. 20-22,1978

/8/ H.M. Sneed, K. Kirchhoff:
New Automated Tools for Analyzing Program Input/Output Behaviour, Proceedings of the Infotech State of the Art Conference on Software Testing, Regent Centre Hotel, London, Sept. 20-22, 1978

/9/ R. Fairley:
Static and dynamic testing methodologies, Proceedings of the Infotech State of the Art Conference on Software Testing, Regent Centre Hotel, London, Sept. 20-22, 1978

/10/ J.E. Hopcroft, J.O. Ullmann:
Formal Languages and their relations to Automata Addison-Wesley (1969)

/11/ Ed. Miller jr.:
Tutorial on Software, Software Research Ass., 13.-17.2.78, Saig/W.-Germany

/12/ F.E. Allen, J. Cocke:
Graph-Theoretic Constructs for Program Control Analysis, IBM Research Report RC3929, Yorktown Heights, N.Y. 1972

/13/ M.S. Hecht, J.D. Ullmann:
Analysis of a Simple Algorithm for Global Flow Problems, Conf. Record, ACM Symp. on Principles of Programming Languages, Boston, Mass. Oct. 1973, pp 207-217

/14/ J.C. King:
Symbolic Execution and Program Testing, CACM vol.18 no.7 (July 1976)

/15/ L.D. Fosdick, L.J. Osterweil: DAVE - A Fortran Program Analysis System, Dept. of Computer Science, Univ. of Colorado, Boulder, Colorado 80302, 1975

/16/ L.D. Fosdick, L.J. Osterweil:
Data Flow Analysis in Software Reliability, Computing Surveys, Vol.8, No.3, Sept.1976

SOFTWARE QUALITY AND ITS ASSURANCE

Dipl.-Ing. P. Weigel

Head of Section for Quality Assurance of Electronic Systems
Bundesamt für Wehrtechnik und Beschaffung, Koblenz (FRG)

SUMMARY

The ever increasing complexity of software for electronic data processing systems and the growing criticalness of the application areas have resulted in the fact that, in addition to the cost and time factors, the quality of software products is gaining more and more importance. This paper provides an overview of the major quality characteristics of software and their assessment standards.

The causes of failure and the development of software are discussed, and the technical means and measures for eliminating faults and impacting the software quality are described. In addition to the technical measures, the organizational means of software quality assurance are handled.

1. Introduction

In recent years, it has been more and more realized that software (SW), like any other product, must first of all be viewed from an economical aspect. As in the case of other products, therefore, the decisive factors are quality, cost and time.

Under the pressure of increasing cost, growing complexity and progressively greater criticalness of application areas, this has led to a change of attitude toward software development* and maintenance, as a result of two NATO sympoisa. (NAUER, P. ..., 1969 and BUXTON, I.N. ..., 1970). The change in attitude is characterized by the term "software engineering" which was meant to be provocative.

The goal was to introduce, into the field of software, engineering methods characterized by disciplined, scheduled, checkable, transparent, standardized and methodical action.

Although software engineering has not yet developed to be a separate engineering discipline and many wishes are still open, a multitude of principles, methods and aids have been made available in the past few years for the rationalization of the development process and for the enhancement of SW product quality.

The economic significance of software is emphasized by the fact that an overproportional rise of SW costs versus overall costs of technical systems can be obverved.

An investigation performed for the FRG under an order of The Department of Research and Technology (DITTMER, E. ..., 1976) revealed that, as early as in 1975, the annual personnel costs for the generation and maintenance of EDP software amounted to 7 billion DM, whereas the annual expenditures for rent/interest, depreciation and maintenance of hardware amounted to 6 billion DM. Other investigations have shown that the test effort for the development of SW is betwenn 30 % and 50 % of the overall effort, and that the maintenance costs (debugging and modification) amount to as much as 70 % of the development costs.

As "software engineering" regards software development as an integrated process where all phases of the life cycle (from the first approach through implementation) of software are considered, Quality Assurance being part of software engineering must arrange for an integrated system of assurance and control measures. The application of software technological methods and aids must result in an overall minimization of costs for the quality functions (fault prevention costs including test costs) and of the costs due to poor SW quality (fault costs).

2. SW Quality - Quality Characteristics
Assessment - Assessment Standards

In general terms, quality is defined as being the condition which makes a product or an activity suited for the accomplishment of given requirements. As a rule, the requirements result from the application purpose of the product or activity (DIN 55350).

In this connection, it must be noted that the quality is defined relative to given requirements (specifications) only.

For the user, the useful value of software depends on a number of properties whose weights result from the problem and the marginal conditions relevant for operation.

---

* "Development" as used in this paper is meant to cover the entire software generation process (concept, design, coding, implementation).

The major quality components which, in their entirety, constitute the SW quality, are shown in Fig. 1 and are briefly defined hereinafter: (WEIGEL, P. ..., 1977).

"Functional fidelity" is a characteristic describing the correctness, completeness and exactness of the functions required in the specifications.

"Efficiency" defines the performance behaviour of the software in terms of time within a given hardware system, as well as the capacitive load on the given hardware resources.

"Flexibility" with its components "adaptability", "transferability" and "changeability" describes the capability of adaptation to new functional requirements of the user, or to a new hardware environment, respectively.

"Ease of use" defines how well the software is adapted to the user's requirements in terms of handleability, ruggedness (insusceptibility to incorrect operation), learnability, understandability.

"Maintainability" is a quality characteristic defining to what extent software deficiencies can be analyzed and corrected by average personnel without affecting other SW quality factors and without assistance by the software originator.

"SW reliability" eventually is defined as the property indicating with which degree of probability correct results will be provided for any type of possible input cases of a given task in the event of faultless hardware.

While the terms of quality, quality assurance[1], faults[2] and, consequently, all terms of a quality system derived therefrom are transferable to software, this is difficult in the case of the term reliability as usually defined for hardware.

Reliability, as a quality component, in contrast with software reliability, is defined in the case of hardware products as the probability that a given system, for a determined time interval, will operate satisfactorily under given conditions.

Application of this definition to software is problematic due to the failure mechanisms differing from those of hardware (with software, there is no material wear or physical overlead as cause of failure; the time of fault occurrence is covert, in reality depending on the instantaneous load only).

The term of SW reliability in the first definition mentioned above initially has a merely practical meaning due to the fact that SW is not 100 % testable. It enables, by means of probability models with corresponding test and analysis methods, to derive information as to the number of faults still inherent in the product.

Frequently, SW reliability is defined with reference to the "actual application" (not to a requirement or specification). This means that it is used as a standard for the completeness and quality of the SW specification. From the point of view of the program generator, this definition is most unsatisfactory as it only obscures the fact of inadequate quality planning during the definition phase.

In the past, assessment of SW has primarily been performed on the basis of its efficiency and of functional characteristics. In view of the usually high failure and consequential costs for SW, other quality characteristics such as maintainability and flexibility are gaining more and more importance. The efficiency of SW is still playing an important part in the control and monitoring of real-time processes, while in commercial data processing its importance is decreasing in favour of the requirements with respect to program flexibility.

The assessment of software is accomplished in the "functional range" based on attributive criteria (presence or absence of a certain function). Metric assessment standards, e.g., can be used for the assessment of the accuracy with which a defined function has been realized. As to efficiency, the criteria with metric properties are preponderant (reply times, occupation of computer components).

In connection with the quality components "flexibility", "ease of use" and "maintainability", subjective attributive assessment criteria are still applied preponderantly. An objective assessment can be reached here, too, by breaking down these quality characteristics into a multitude of attributes and by generating checklists. In this case, the assessment can be accomplished indirectly by the assessment of the SW generation process or the assessment of secondary characteristics, such as, for instance, documentation level, program structure, standardization, programming style, etc.

It is mainly in the case of SW projects with a high safety risk or risk of damage that a quantitative assessment of reliability is desired nowadays. As previously mentioned, the existing literature describes a number of reliability models with corresponding test methods (SHOOMAN, M.L. ..., 1975).

---

1) Quality Assurance: all measures for achievement of the required quality

2) SW-fault: unpermissible deviation of a characteristic value from the required value

By the definition of quality objectives with the requirement to exactly describe and, if possible, quantify the characteristics during the definition and design phases, the SW quality can be impacted in a manner advantageous to the user.

Quality assurance without planning and definition of the quality requirements is impossible.

3. Software Generation Process - Software Faults

The software generation process can be regarded as a sequence of translation steps: a problem to be solved is translated, via various intermediate steps (intermediate products) into a detailed set of computer instructions. The initial product, on the one hand, consists of information for the solution of a problem, stored on magnetic tapes, punched cards or strips, on the other hand of documentation describing the SW product and how it is handled.

In Fig. 2, individual steps of the SW generation process and the resulting intermediate products are shown, based on the so-called quality cycle.

Software faults will arise whenever an inexact or incomplete translation is made when proceeding from one problem solution step to the next, more detailed step, i.e. whenever specified and defined characterisitcs are misinterpreted, or mitted when passing from one development step to another, or when realizing any characteristics which were not requested.

The basic causes for the generation of SW faults are:

- insufficient communication between SW developer and SW user, or between various SW developers;
- difficulties encountered in handling and minimizing complexity;
- insufficient accuracy of translation of functional and design requirements in the various development stages (due to insufficiently qualified personnel or use of inadequate application of these tools).

The fault, like the quality, is a relative term which loses its sense without the existence of an explicit or implied requirement or standard.

With software, the reference standard originates from the requirements of the user and the quality and acceptance criteria derived therefrom. In classifying the faults that occur by categorizing the faults in accordance with development stages, one may coarsely distinguish between problem analysis faults (concept and development faults) and implemetation faults (programming faults). Various investigations show (BOEHM, B.W. ..., 1975 and ENDRES, A. ..., 1975) that frequently more faults arise due to insufficient comprehension of the problem rather than due to incorrect programming. Moreover, the former type of fault mostly is more difficult to detect and far more costly.

During the utilization phase, faults may arise due to incorrect SW handling or poor maintenance (software changes and debugging).

4 How to influence Software Quality

Basically, there exist two different possibilities for influencing the quality of a product, viz.

- preventive technical measures;
- measures for fault detection and correction.

The additionally important organizational measures are discussed in Section 5.

The preventive measures are also denoted constructive methods, those for fault location and correction are designated analytical methods.

Experience shows that, as the complexity of products rises, the quality improvement measures, in view of the high costs involved, must be shifted to the early phases of product development. This applies in particular for SW products where a major part of the faults can be traced back to the definition and design phases, and a reduction of the maintenance costs (which, for some systems, are as high as 70 % of the development costs compared to the overall life expectancy) is imperative.

4.1 Preventive Measures for Influencing the SW Quality

Today's software technology presents a number of methods and tools by means of which the SWquality may be decisively influenced in a preventive manner, and by which the above failure causes may be eliminated or at least diminished.

The methods and aids, on the one hand, orient themselves to the type of software to be developed (administrative SW, technical/scientific SW, system SW) and the associated,

partly different, development methods, on the other hand to the programming language used and, last but not least, to the development environment.

Systematization of the software-technological methods and aids is rendered difficult by the fact that many activities are repeated in the various development phases.

The methods and tools affect the process and the product in a different manner, and it is difficult, in the majority of cases, to quantify the influence on SW quality.

Often, however, it suffices to know the application of the methods cannot but improve the quality, all the more as the costs involved are but low compared to the development costs, and as the quality improvement is almost always accompanied by an augmentation of productivity.

It is important to know that the constructive methods and tools do not only favourably affect the SW quality as regards functional fidelity (correctness, exactness), but, in particular, also such quality characteristics as maintainability and flexibility.

If, for instance, the quality characteristic "maintainability" is investigated more closely, one will detect that its quality component "learnability" can be favourably impacted by constructive methods which support the uniformity, the modularity, the transparency of algorithms and the documentation level. The testability, also a partial attribute of maintainability, is supported by constructive methods for the recognition of the instantaneous processing condition and the history of the processing conditions.

In Fig. 3, the major software-technological methods and tools are shown. A distinction has been made between methods and tools which directly affect the technical functions and characteristics of the SW product (or partial product), and such methods and tools which support the activities accompanying the project. An exakt separation, however, is not always possible, just as the transition from higher echelon methods to specific methods and tools is flowing.

4.2 Methods for Fault Detection and Correction

Fault detection and correction become all the more difficult and expensive the later during the development process the faults are detected.

Faults in program development affecting any "design quality characteristics" (maintainability, ease of use, flexibility) can be detected and corrected in a meaningful manner during the early phases of program development only. Both the test methods and the methods for fault correction depend heavily on the SW technological methods and tools described above and applied for the respective SW development, as well as on the decisions made during the design and implementation phases.

At this point in time, the fault-tolerating systems must be mentioned, i.e. systems with automatic fault detection and correction. In connection with software, similar thinking as with complex hardware seems to occur, namely that in many cases it is more economical to make redundant (fault-tolerating) systems rather than to increase the effort for quality assurance beyond a defined threshold. It must be noted, however, that by additional automatic mechanisms additional failure sources might be introduced. (MYERS, G.J., 1976 and KOPETZ, H., 1976)

4.2.1 Fault Detection

Tests and checks must be initiated as early as ever possible, and must be performed in a manner related to the individual project steps. Their goal is

- to find out if the required functions and quality characteristics of the previous development step have been realized;
- to point out any inexakt, incomplete or incorrect specifications;
- to detect any possible problem areas.

As already stated, many software faults may be traced back to poor problem analysis, problem processing and problem solution.

The checks starting during the definition and design phases, therefore, concern the intermediate SW products available as documentation. It is expedient to provide for at least three test steps:

- review of user requirements;
- review of "external specifications" (system specifications);
- review of "internal specifications" (program, component design)

For the program generation process proper, the SW implementation, the following test steps must be passed, depending on the size of the project:

- component tests
- integration tests
- system tests/acceptance tests.

The importance which is today attached to SW testing can be recognized from the fact that meanwhile the SW managers are prepared to invest up to 50 % of the overall development costs in SW testing (with particular emphasis today being on the functional characteristics). This area which is not yet methodically

secured poses particular problems to quality assurance as it is impossible to prove that a non-trivial program is free from faults. Even in the case of simple programs, a very small portion only of the possible input cases can be tested during the test phase.

The goal in SW testing, therefore, must be to find faults, not to prove that the software is correct. This principle is new in test planning and performance, and must not be underestimated as to its effect. It must reflect in the attitude of the personnel entrusted with the tests.

The test methodology for SW comprise test strategies and test tactics. The test strategies cover the manner of proceeding in connection with the verification of entire systems. As a rule, one will proceed from the components over higher assemblies to the overall system ("inside-to-outside" procedure).

The test tactics cover the verification of individual components. Here, too, there exist two different manners of proceeding which mostly are applied concurrently:

- the review of the functions of a program by means of the functional test, also known as the "black-box" tactics, with the specifications being used as the basis.
- passing of all program paths and instructions at least once, also known as structural test or "white-box" tactics.

As in non-trivial programs an exhausting test of all functions for all possible input parameters is neither thinkable, nor a checkout of all program paths with all parameters is feasible, special emphasis rests on the problem of selection and generation of suitable representative test cases.

Selection and generation can be performed along three basic principles:

- Orientation along the type of expected input cases. The functions are tested at the rate of their occurrence during actual operation.
- The test cases are selected based on the possible safety or damage risk. The test cases are scheduled so that functions and program portions where a fault might result in a particularly high damage are tested most intensively.
- Selection of test cases orients itself above all by the weak points of a SW product, i.e. functions or program portions particularly susceptible to faults as a result of their level of difficulty are subjected to special tests.

In conclusion, I wish to mention that today there exist various approaches to prove the correctness of programs (LONDON, R.L., 1975).
Concrete applications, however, are scarce, as these methods are most complex, and as it is difficult to deduce from the programs the conditions to be proved.

4.2.2 Debugging

The majority of the defensive methods and aids mentioned in Fig. 3 exert a positive influence on debugging, too. In connection with "debugging", one must distinguish between fault location, fault elimination and determination of the cause for failure. With fault diagnostics, the reverse order is generally followed (fault isolation) compared to the order employed in verification of programs during the SW test (cf. above).

Software technology offers additional aids for the implementation (coding and integration) and utilization phases, enabling to observe and check program performance, thus supporting fault diagnosis (Fig. 3).

For debugging, too, programming aids are known (MYERS, G.J., 1976), supporting the performance of modifications as well as monitoring and checkout of the modification. It should be clearly understood that, whenever a non-trivial fault is concerned, the SW drops back to a corresponding step of the design phase, with the same rules for checkout and monitoring having to be applied as during initial generation of the SW product.

Above all, attention must be paid to avoiding, by debugging, to impair the design quality characteristics. Secondary characteristics such as the modularity of the systems and the transparency of the algorithms are susceptible to being adversely affected by frequent debugging.

The determination of the failure causes is meant to avoid the repeated occurrence of similar faults. The questions to be asked are:

- Why did the fault occur?
- What is the reason for the fault not having been detected during earlier test steps?
- What would have had to be done in order to prevent the fault or to detect it earlier?

5. Quality Assurance as a Management Function

In modern thinking, quality assurance as a management function is understood to be a coordinating and integrating service during the product generation process. This means that it is not a member in the chain of order performance, but the organization of cooperation and communication of all functions in all phases of product generation which impact the quality of the product.

Quality assurance, in this context, means the correct operation in quality matters of all organisms involved in planning, design and production.

Quality assurance as a management function can be subdivided into

- Quality Planning (definition of specifications, rules and measures by which - based on the defined quality level - the disred quality is ensured in the various development phases of the product);
- Quality Audit (continuous checkout of quality, i.e. measurement of deviations of quality characteristics from the defined quality level);
- Quality Control (minimization of difference between actual and required quality by corrective action).

The above manner of thinking can also be applied to the management of SW quality, although this is not yet generally accepted. (WEIGEL, P.,1977)

Fig. 4 shows the development process for a major software system. It permits to recognize the above sub-functions of quality assurance and their impact on the development steps.

Quality Planning generates a quality assurance plan summarizing, in a project-oriented manner, all quality assurance rules, work orders and organization instructions. This plan must, inter alia, cover:

- definition of responsibilities, cognizance and competence in connection with all quality matters;
- information on test concept and test planning;
- procedures for the performance of design reviews;
- procedures for configuration control and monitoring;
- procedures for fault reporting, fault analysis and fault correction;
- information on the use, generation and monitoring of test aids (e.g. test and simulation software, data generators etc.).

Quality Audit, by review and test, intervenes in the various SW generation steps, while Quality Control analyzes the data obtained by quality audit and exerts a correcting influence on the development process.

Major aids for software quality control are the procedures for configuration control and monitoring, rules and procedures for the establishment and monitoring of a software library for central management (reproduction, identification, storage and output) of objects subject to configuration monitoring, as well as formalized procedures for fault reporting, fault analysis and fault correction. (MC GINNIS, F. 1973; FOSTER, R.A., 1977)

It is of essential importance for Quality Assurance as a coordinating and integrating function that the responsibilities and cognizances be unambiguously defined, and that the interfaces with all organisms involved in the project be clearly determined.

Like for hardware, there is no general rule for the organization of software quality assurance; it is always dependent on the structures of the company and the project.

Based on experience gained with hardware projects, however, it is advisable, also in the case of software, beyond a certain size of an enterprise or a project, to set up an independent quality organization for the uncompromising enforcement of the specified quality assurance measures. In any case, it must be ensured, whenever an organizational independence of software quality assurance is impossible or inexpedient, that an objective assessment, supervision and control of quality is always possible.

Instead of a quality assurance organization independent of the project manager, a test team headed by a so-called test manager and responsible to the project management exists for many software projects. Its tasks are test planning and performance.

By analogy with the procedures for the development and procurement of complex hardware, contracts awarded by the GMoD for software are increasingly stipulating quality assurance requirements.

The requirements for set-up and monitoring of a quality assurance system as well as for the generation of a quality assurance plan are based on the NATO specification AQAP-1 and MIL-S-52779, respectively, the latter one covering software. The NATO-AC-250 Group is at present preparing an appropriate Software Specification.

6. References

BOEHM, B.W.; McClean, R.K. et al., 1975 "Some experience with automated aids to the design of large-scale reliable software", Proc. of the int. Conf. Reliable Software.

BUXTON, J.N., and RANDELL, B. (Ed.) 1970, "Software Engineering Techniques", NATO-Conf. Rome 1969, Birmingham 1970.

DITTMER, E., et al. 1976, "Einsatzmöglichkeiten softwaretechnologischer Methoden mit Normungseffect", Diebold Deutschland GmbH, Frankfurt, sponsored by Bundesministerium für Forschung und Technologie.

ENDRES, A., 1975, "Software als Qualitätsprodukt", IBM Nachrichten 25. Jahrg.Heft 227.

FOSTER, R.A., 1977, "The Role of Quality Assurance in Software Development", Second NATO Symposium on Quality and its Assurance, London.

KOPETZ, H., 1976, "Softwarezuverlässigkeit", Carl Hanser Verlag, München-Wien.

LONDON, R.L., 1975, "A view of Program Verification", Proc. of the Intern. Conference on Reliable Software, Los Angeles.

MYERS, G.J., 1976, "Software Reliability, Principles and Practices", John Wiley & Sons, New York.

McGINNIS, F., 1973, "The Role of Quality Assurance in Advanced Technology", First NATO Symposium on Quality andits Assurance, München.

NAUR, P. and RANDELL, B., (Ed.), 1969 "Software Engineering NATO Conf. Garmisch 1968, Brussels.

SHOOMAN, M.L., 1975, "Software Reliability: Measurement and Models", Proc. Ann. Reliabil. and Maintainabil. Symp.

WEIGEL, P., 1977, "Qualitätssicherung von ADV-Software", Second NATO Symp. on Quality and its Assurance, London.

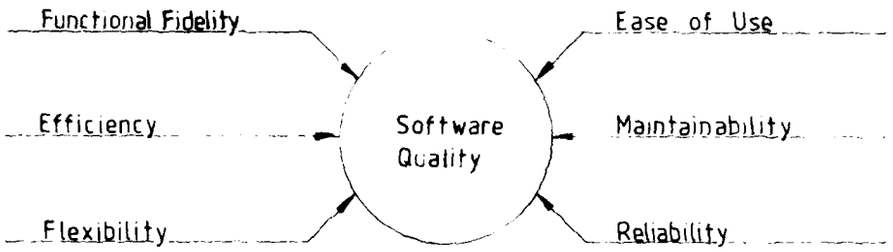


Fig.1 Quality-characteristics of computer software

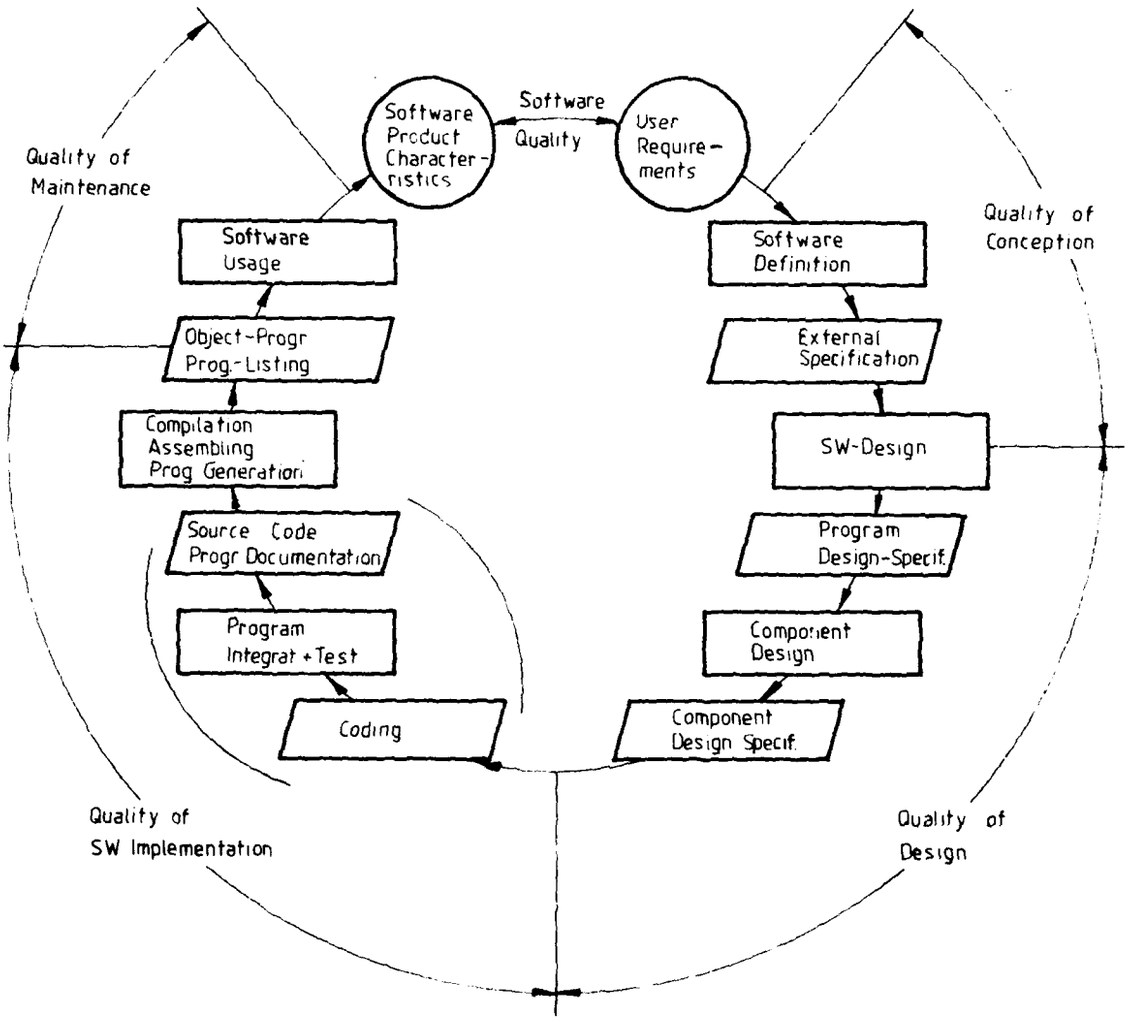


Fig.2 Quality cycle and software-development-process

Direct influence on technical functions and characteristics of the SW-Product (intermediate Product)

Methodology	Techniques/Practices/Tools	Influence on Software-Quality
Structuring Methods (for Functions, Data Processes) - hierarchical structuring - modularization - step-wise refinement	- use of structure-forming languages - Macrosoftwarestructuring - structure graphs - structured coding - HIPO - Michael Jackson Notation	by means of: - reduction of degree of complexity, improving the communication between SW-Developer and SW-User as well as between the SW-Developers; due to this reduction of functional and logic errors, which can propagate pu to the final program; - influencing the SW-structure already at definition and design with the objective of optimization of Flexibility and Maintainability;
Description Methods (for Functions, Data, Processes)	- formal specification languages - problem oriented special languages - Data description languages - HIPO - Decision Tables - Cause-Effect Graphs - Flowcharts	
Technical Constructional Methods - uniform coordination of construction (interfaces, control-flow, data flow) - use of standardized SW-Components - Design of testable systems - Design of machine-independent systems - Design of fault-tolerating systems	- Structured Programming - Defensive Programming - active and passive fault detection measures - dynamic Redundancy (using different algorithms) - Fault-isolation/Reconfiguration - Algorithm-Collections or Catalogues	- by standardization of constructional principles improvement of Maintainability and Flexibility; - by influence on "SW-Reliability" by design of fault-tolerating systems and appropriate independent Program-components;

Fig.3(a) Methods/techniques/practices of software engineering to influence software quality

Influence on SW-Quality by project-accompanying measures

Methodology	Techniques/Practices/Tools	Influence on Software-Quality
<u>Verification/Testing/ Auditing</u> - project accompanying inspection - result inspection	- Formal procedures for Requirement-Review, Design Review, Code Inspection - Structured walkthrough - Check lists - Questionnaires - Guidelines for Test- and Integration-Strategies - Interactive Test-Tools - Diagnostic Aids (Dumps, Traces, Backtraces) - Test-Monitors - Test-Data-Generators - automatic-Test-Analysis	- Prevention of the propagation of errors through various development-phases by executing testing as soon as possible - Prevention of logical faults and improvement of Maintainability and Flexibility by checking transparency, uniformity, modularity, level of documentation
<u>Influence on SW-development environment</u>	- Dialogue-languages - Dialogue-Test-Tools - Compiler, Assembler, linkage Editors - Program-Generators - Automatic Specification- and Documenation aids - Qualification of the SW-Development-Personnel	
<u>Management Methods</u> - SW-structure-oriented programming-organization - Clearly defined and separated development phases (work breakdown structures) - Deadline Management - Independent Quality Assurance Organization - Generation of Documentation during development process - Influencing Programming Style	- SW-Configuration Management - Critical Path Planning and Scheduling - Programming Manuals - Guidelines for programming - Problem-reporting-system - Work Instructions (in writing) - Formal Procedures for Fault-Correction and SW-Modification - Software Development library - Documentation Guidelines - Documentation administration - Guidelines for generation, duplication and release of programs - Guidelines for transportation, storage and identification of data carriers	By use of appropriate Management Tools the overall Quality of a product can be improved, this is especially true for Software, due to: - up to now inadequate standardization - high portion of personnel resources for Software-Development - the degree of complexity - the lack of software-technological Methods and Tools in some areas - special characteristics of SW as an informational product - the fictious ease of change of SW

Fig.3(b) Methods/techniques/practices of software engineering to influence software quality

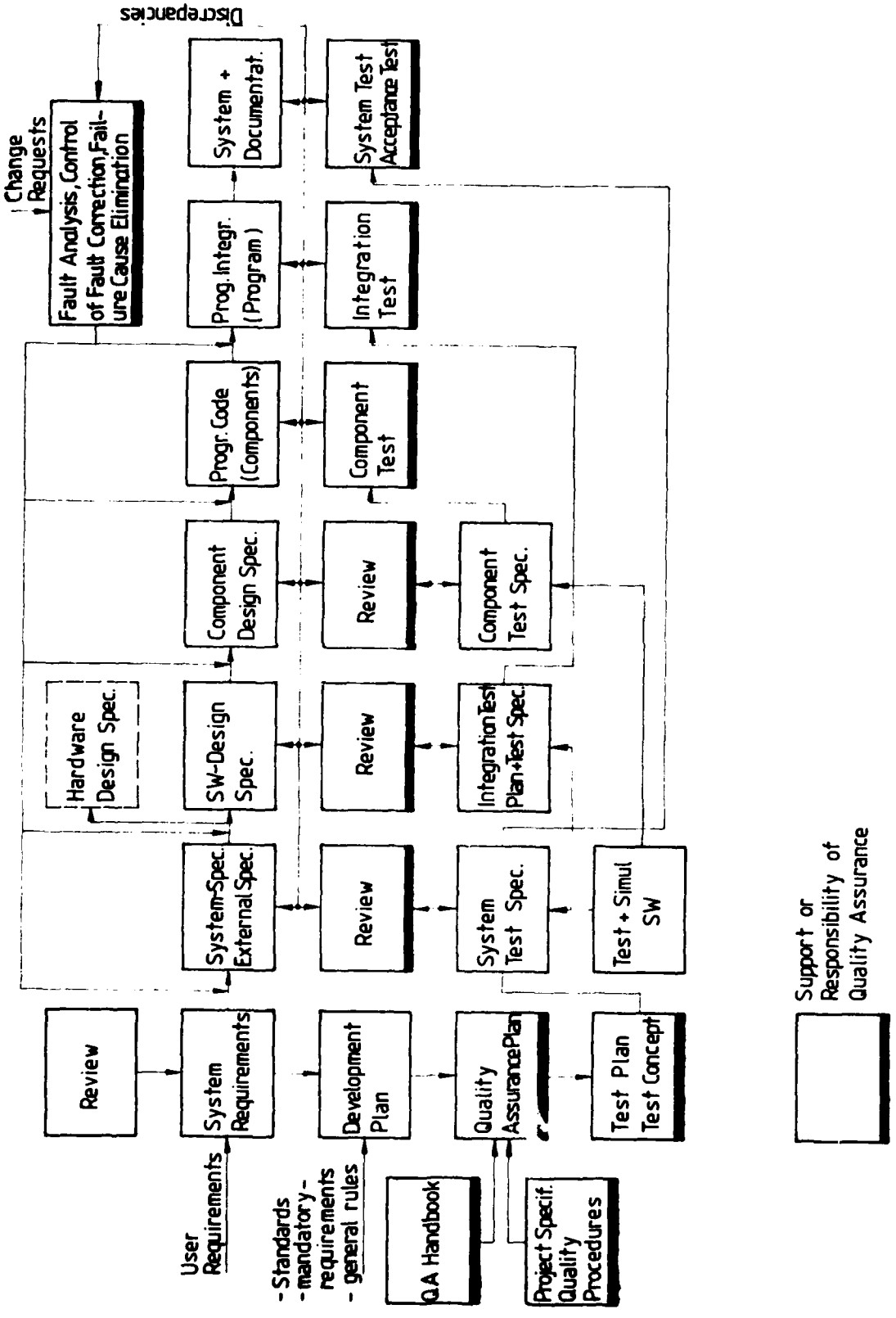


Fig.4 The role of quality assurance in the software development process

SOFTWARE DEVELOPMENT FOR TORNADO - A CASE HISTORY FROM THE RELIABILITY AND MAINTAINABILITY ASPECT.

D. J. Harris

EASAMS Ltd.,
Lyon Way,
Frimley Road,
CAMBERLEY. Surrey. U.K.

SUMMARY

The paper presents the methods and procedures adopted in the development programme, for the PANAVIA Multi-Role Combat Aircraft (MRCA) now named TORNADO, to produce software of the required quality. Software development for TORNADO has been undertaken in four successive, but overlapping phases, namely definition, writing, testing and delivery. The paper identifies the key features, in these four phases, that have contributed to software reliability and maintainability and indicates those that are worthy of repetition in future projects, those to be avoided and those being improved now.

1. INTRODUCTION

This paper is based on EASAMS experience over the past 10 years in the development programme for the PANAVIA TORNADO aircraft.

The anticipated life of both hardware and software in the TORNADO programme requires the investment of effort to ensure the quality of these items. In the case of software the term quality is used to mean the maintainability and reliability of a program which also contribute to the effectiveness of that program. The need for such quality applies equally to the software for the TORNADO Avionic System and to the software used to support main program development including assembler, simulation and utility programs.

Maintainability has been interpreted as the readability and high level of comprehension of programs as documents in their own right. This includes program organisation and structure together with programming conventions, which aid visual comprehension of program flow and purpose, thus speeding error localisation and program modification. Emphasis has therefore been placed on the production of software documentation of adequate depth, together with standards and control procedures.

Reliability has been interpreted as operational correctness and has led to a concept of progressive and detailed program testing associated with comprehensive test documentation and test control procedures.

This paper describes how software maintainability and reliability objectives, based on the above interpretations, have been realised in terms of methods, procedures and quality assurance standards.

In order to establish the background to TORNADO Software Development it is necessary to introduce:

- the project organisation and EASAMS role therein
- the TORNADO Avionics System and the place occupied in that System by the Computing Sub-system, hardware and software, and
- TORNADO software and its relation to the Avionics Development Programme.

1.1 The Project Organisation and EASAMS Role Therein

The International Project Organisation is shown in Fig. 1. The Defence Ministries of the three countries involved formed the NATO MRCA Management Organisation (NAMMO) at policy level and the NATO MRCA Management Agency (NAMMA) to execute that policy. NAMMA placed a contract on PANAVIA for airframe and avionics development, PANAVIA in turn placed a contract on its partner companies, BAe, MBB and AIT for airframe development and a contract on EASAMS for the management, design, development and integration of the avionics system.

EASAMS Limited (formerly Elliott Automation Space and Advanced Military Systems Limited) is a Systems Company of GEC Marconi Electronics Limited. EASAMS was formed in 1962 to conduct forward looking system studies and project studies, to provide an operational analysis facility, to provide a management organisation to undertake the co-ordination, monitoring, technical design and analysis and control of system development projects, and to provide advice and training in project management.

EASAMS role in the project is that of Co-ordinating Design Authority for the avionic system and Technical Authority for avionics equipment development. EASAMS has placed sub-contracts on collaborating systems companies, ESG in Germany and SIA in Italy, in accordance with agreed international worksharing arrangements.

To carry out the work the three systems companies EASAMS, ESG and SIA formed and staffed two tri national engineering teams:

- the Central Design and Management Team (CDMT) which is based at EASAMS in the UK and undertakes the design, development, integration and management of the entire avionics system
- the International Software Team (IST) which is based at ESG in the FRG and provides the software for the avionic system and for development test purposes according to requirements produced by the CDMT.

The three Systems Companies also undertake work "in house". Each acts as technical authority for the equipments which are procured from the avionics industry of their own country - the commercial authority being the corresponding partner company of PANAVIA, with whom close liaison is maintained at all times. In addition, each Systems Company has and is undertaking avionic system integration and development using comprehensive in-house ground rig facilities. The CDMT itself is responsible for planning, conducting and analysing flight trials of the avionic system in a Buccaneer carry aircraft and also co-ordinates support by all the System Company teams, to PANAVIA's own ground and flight trials programmes.

The overall task thus encompasses:

- system, sub-system and equipment design and development
- software design and development
- system, sub-system and equipment integration, ground and flight test
- programme management and control
- documentation and training
- provision of development and in-service test equipment.

It will be appreciated that the development task, which started in January 1969, has been conducted in a challenging environment. The requirements of international collaboration and worksharing, the large number of organisations and companies involved, the geographical separation coupled with the need for good communications and the pressures of timescale and cost have all brought their attendant problems. It has been necessary throughout to invent and introduce management procedures specifically adapted to the project to overcome such problems.

1.2 The TORNADO Avionic System and Computing Sub-system Hardware/Software

1.2.1 The Avionics System

Within the Weapon System a number of major functions are carried out by the Avionics System. These functional areas and the interface with the Weapon System are illustrated in Fig. 2. Three areas, Navigation, Weapon Delivery and Automatic Flight Control, are closely integrated and are particularly influenced by the requirements of the critical mission phases. These critical requirements are:

- low level, highspeed flight
- first pass acquisition of targets
- accurate weapon delivery

The system must, in addition to satisfying these critical requirements, provide communications and other facilities to enable TORNADO to operate within NATO and National frameworks. Its detailed design must be such that there is an acceptable workload for aircrew, in operation, and for groundcrew in supporting and maintaining the aircraft. The range of avionics tasks carried out under the broad headings of Fig. 2 is listed in Fig. 3.

1.2.2 The Computing Sub-system

a) Hardware

Navigation, Weapon Delivery and Display calculations are carried out in the Computing Sub-system. This sub-system provides a service facility to the Navigation, Weapon Aiming, Displays and Controls sub-systems essentially of integration of sensors, automatic and manual moding and control of the system and concise display of data to the pilot and navigator. The sub-system comprises three primary equipments, the Main Computer and two Interface Units and communicates with some 37 other equipments ranging from control panels to tactical sensors largely using a digital data transmission system.

- The Main Computer

  The Main Computer, the LITEF Spirit 3, is a stored program digital computer with two independent programming levels, multiple arithmetic and index registers, large instruction repertoire, single address working, parallel operation and a priority interrupt system

- The Interface Units

  The Interface Units are provided to service the pilot's and navigator's cockpits since digital interfacing at control panels was not cost effective. These units provide conversion of data and switch signals from analogue to digital and vice versa.

- Data Transmission

  The TORNADO digital data transmission system was chosen primarily to offer high noise immunity and stability, flexibility to accommodate varying signal types and significant reduction in aircraft wiring due to multiplexing. A standard format is employed for digital data transfers on TORNADO and each equipment interfacing with the system performs the necessary data conversions internally. A transmission link is provided for each data transmission path between equipments.

b) Software

The program in the Main Computer carries out the following basic tasks:

- processing of Vertical and Horizontal Navigation Data from the basic sensors operating in four integrated modes of Navigation. Incorporation of Position Fixing Data
- calculation of lateral steering demands against a stored automatic flight plan or against a route manually entered during flight
- automatic and manual moding of the navigation and fixing sensor control and associated calculations
- calculation of pointing angles for the tactical sensors and processing of the angle, range and height data from these sensors
- calculation of ballistic trajectory of bombs for accurate Weapon Aiming
- calculation of lateral steering demands for an automatic attack on a target
- generation of release cues for automatic weapon release
- concise display of Navigation, Fixing and Weapon Aiming data to the Pilot and Navigator.

These tasks present considerably varying demands on the flight program from logical decision making for moding to complex trigonometrical calculations for fixing. Additionally, these tasks have to be performed at varying iteration rates from 0,1 Hz for some of the navigation calculations to 50 Hz for sensor stabilisation and ballistic calculations. In addition to the Operational Flight Program an External Ground Test Program is being developed. This program which is loaded in place of the operational program provides the groundcrew with extensive facilities for diagnostic and checkout tests on the installed system.

1.3 TORNADO Software and its Relation to the Avionic Development Programme

The approach in testing is aimed at providing progressive and increasing confidence in the avionics system prior to flight trials in TORNADO prototypes. Avionics system and software development is thus undertaken in a series of successive test stages of increasing scope and complexity as indicated in the outline avionics programme shown in Fig. 4.

- Stage 1

  At Stage 1 a rig facility is utilised which permits individual software items to be checked on computers as soon as they are written and before they are tried out with other parts of the avionics system.

- Stage 2

  At Stage 2 the ground test rigs provide the prime facilities for the avionics equipment and system integration and are the first stage where representative avionics equipments are brought together, and also combined with the software. The Stage 2 rigs are supported by additional computers which are used for data acquisition and replay, for open loop testing (stimulation) and for closed loop work (simulation). The avionics system can thus be exercised in a dynamic manner as far as is possible on the ground.

- Stage 3

  There is a need, however, for more representative dynamic testing of both hardware and software at an early stage. Sensors such as Radar and IN cannot be fully tested on the ground therefore Stage 3 of the development comprises flight testing of the avionics in specially equipped Buccaneer flying test-bed aircraft. The advantage of using these aircraft are that they have well-proven airframe and engines thus permitting concentration on avionics problems; and that they can be fitted with comprehensive flight test instrumentation specially for avionics. This instrumentation includes high-data-rate recording of digital and analogue signals as well as a variety of photographic data.

- Stage 4

  The avionics system itself needs to be integrated with other aircraft systems such as the electrical power generation and distribution system. This is carried out on Stage 4 ground test rigs which are also used to assist in solving problems first found in TORNADO flight trials.

- Stage 5

  The final stage of avionics development testing is on the TORNADO prototypes.

Of the nine prototype TORNADO aircraft three are being used extensively for avionics testing. The avionics system is also being evaluated by the Official Test Centres on the pre-production aircraft. This sequence of the main stages of avionics integration and testing is also shown in Fig. 5. The total programme of avionics system work is extensive and is shared between many test sites. Proper planning, co-ordination and assessment are therefore essential together with 'feedback' from trials into design.

2. SOFTWARE DEVELOPMENT FOR RELIABILITY AND MAINTAINABILITY

Having established the background and environment, the phases in the development of TORNADO software may be examined in more detail to ascertain how attention was given to software reliability and maintainability objectives and quality assurance requirements. The essential phases are:

- software definition
- software writing
- software testing
- software delivery

2.1 Software Definition

The software definition phase involved:

- the preparation of software requirement documentation correlated with the overall avionic system design requirements
- relation of software development to the key events in the development of the total avionic system.

2.1.1 System and Software Specification

Successful development of any avionics system, comprising both hardware and software, requires close liaison between the major contractors and the customer throughout all phases. The Design Specifications are an essential vehicle for this liaison and the standard, definition and degree of attention paid to such documentation bears significantly on the effectiveness of the liaison to agree an adequate system definition. The application of formal design documentation standards, from the early stages of design, will enable a more thorough understanding of the system hardware and software by both the customer and contractor. It will also enable some of the inevitable design deficiencies to be identified early and corrected. The system, hardware and software, is defined by a series of specifications. At the highest level they define the general organisation and performance characteristics from which lower, more detailed levels, are defined. At the lowest level these specifications will define individual components of a specific equipment. The source of the various documents and the timescales over which they are produced varies according to the specific level, however the format and hierarchy of the levels ensure a consistent and compatible set at an early stage.

Experience, in the TORNADO programme, has shown the benefit to be gained from such emphasis on full documentation of the design from the outset. Insistence on detailed and in-depth documentation helped to ensure the application of common engineering standards by all project participants. Detailed attention to interface specifications, particularly that for the data transmission system, certainly reduced the number of problems arising in development. The avionic system design documentation structure is shown in Fig. 6 and will be described further to show the place of software definition documentation therein.

The basic requirements document for the Avionics System, the "Performance and Design Requirement" (PDR), is issued by NAMMA after agreement by the Nations and forms an Annex to the PANAVIA/EASAMS Development Contract. The PDR is the highest level document. The documentation, prepared by the avionic teams, at the lower levels are summarised below.

a) System and Sub-system

The expansion of the PDR into a set of definitive documents at sub-system level is formally contained in a set of "Sub-system Specifications" which have been prepared by the CDMT and are maintained by CDMT throughout the development phase. The sub-system specifications detail for each sub-system, the component parts of the sub-system, both hardware and software. They define the structure and external interface of the sub-system and identify the major characteristics of the equipment comprising the sub-system including function, interface, performance, safety, reliability and maintainability.

The sub-system specifications have been used as the vehicle for defining and agreeing the detail system functions, performance and architecture with NAMMA throughout development. Initially the sub-system specifications defined operationally visible functions and system performance during the initial Feasibility Phase. During Development they have been progressively extended until they have become the baseline for the sub-system operation and performance against which the actual sub-system performance is now being measured.

b) General Applicability Specifications

There are a number of aspects of equipment design which are common for every equipment installed in the system. These aspects include environment, installation build standards and general standards and general quality requirements. Whilst comprehensive sets of Government standards already exist for equipment design it was necessary to choose from the options provided in these standards those which were applicable to the project. These aspects were all referenced in an Equipment Model Specification which was drawn up as one of the initial tasks of Project Definition. The Equipment Model Specification includes aspects of a general nature which are common to all equipments and provides a format within which the specific technical requirements are defined to provide each individual specification.

c) Equipment Specifications

Equipment Specifications took their requirements for facilities, accuracy and interface from the relevant sub-system specification. Initially they defined the technical requirements to a level sufficient for tendering. After the vendor was selected however, the specification became the contractual technical definition of the equipment, to be procured initially for Development and finally for Production. All equipment specifications follow the same format as defined by the Equipment Model Specification.

d) Software Requirements

Software falls into two basic categories; dedicated to a specific equipment process and general system software. Dedicated software (firmware) is part of an equipment process and as such is embedded into the design for that equipment. In general it is not specified separately at a sub-system level but is covered in the characteristics and performance requirements detailed in the equipment specification for that equipment. This software is often proprietary to an equipment manufacturer and it is the responsibility of the equipment manufacturer to specify, design, test and maintain the programs. The system software, within the main computer, is defined in a series of Software Requirement documents. These are equivalent, on the software side, to the equipment specifications.

The Software Requirements identify all the operational requirements of the program by defining the logical and mathematical tasks to be carried out. The breakdown of the total software definition into a series of software requirements is based on identifying manageable system functions which can be assigned a design task. The structure of the final programs, however, may not necessarily reflect one by one the breakdown of the software requirements although they will be very similar. The Software Requirements are prepared in accordance with a defined standard, to ensure a common format and depth of treatment and cover the following items:

- the scope of the requirement
- other related and applicable documents including sub-system and equipment specifications
- a description of the system function covered by the requirement and its relationship to other software requirements
- a functional description of the software task covering all tasks to be undertaken listed in a logical sequence. This will include descriptions of crew actions and switch functions and use of control and status signals defining software modes. All calculations necessary to achieve the desired transfer characteristic will be detailed. The program iteration rate accuracies and priorities will also be defined.

The Software Requirements also contain two annexes where appropriate; one, called the Logic and Equation Development (LED) Annex, containing derivations from first principles of calculations in the requirement, the other, called the Outline Test Requirements Annex, giving recommended methods to be used for testing various functions in the requirement at Stages 1 and 2. Throughout development procedures have been applied to control changes to Software Requirements as well as other documentation defining the system. A Software Requirement Change Request may be formally raised by any group or establishment involved in design, development and rig and flight testing of software. The Request will contain full details of the proposed change for assessment purposes. During processing the Change Request may change status to a fully defined Software Requirement Change Instruction. All Change Requests are forwarded to the CDMT for configuration control and design assessment purposes. The feasibility of the Change is checked and hardware and other software dependencies examined and associated documentation raised if required. The IST will also examine the Change for impact on the associated program if and when it is implemented. The Requests may affect overall system design and operation and are also submitted to PANAVIA for comment. Once the Change Request has been authorised all affected documentation is updated including such items as sub-system and equipment specifications and test procedures. At this stage an authorised Software Requirement Change Request is also treated as an agreed Program Change Request and is implemented accordingly.

e) Supporting Specifications and Documents

During development a number of supporting documents must be produced to cover items at sub-system level and below not specifically covered in the main line documentation. Amongst these the most important, from the viewpoint of software development, is the Interface Control Document which lists all equipment interface data.

2.1.2 Basic Program Definition

In order to simplify the development and testing of the software and to facilitate progressive testing of the system, the software has been divided into a number of separate operational flight programs (called "series"); each series being an extension of the previous one in that additional system functions are added. The functions contained within each software series were as follows:

Software Series 1 (SS1)	Basic Navigation Flight Trials Display and Recording
SS 2	Extended Navigation and associated modes Display of Basic System Data Flight Trials Display and Recording
SS 3/4	Full Navigation En-route Steering Position Fixing Extension of Display Capability Flight Trials Display and Recording
SS 5	Full Navigation En-route Steering Position Fixing Weapon Aiming - Basic Modes Attack Steering Full Display Capability System Check or Flight Trials Display and Recording
SS 6	As for SS 5 but with full Weapon Aiming and final incorporation of modifications from flight trials.

For each of the above software series the CDMT prepared a so-called "Design Data Set" which served as the baseline for the software writing task undertaken by the IST. The Data Set specified the following items:

- the avionic system functions to be covered in the software series
- the applicable Software Requirements and agreed associated amendments and change requests
- the applicable issue of the Interface Control Document
- related Sub-system Specifications
- items to be delivered by the IST for the software test phase
- differences between equipment development standards which may affect software.

2.2 Software Writing

In order to undertake the software writing task it is necessary to:

- define the structure of the programs
- define rules for program preparation in detail
- prepare documentation describing the programs.

The degree of effort applied to these aspects relates directly to software reliability and has an even greater impact on software maintenance.

2.2.1 Program Structure

a) General considerations

Structuring programs for avionics systems requires careful consideration of the trade-offs between store usage, time loading and ease of program modification and system maintenance. In this work the architecture of the computer has a significant effect on the organisation of the program. The major influences are:

- interrupt organisation
- method of addressing store
- word length
- instruction repertoire

Since the program has to run at varying iteration levels and invoke specific functions according to the system moding, it is necessary to have a program routine which acts as a supervisor over the control of the individual program modules. This program is obviously an overhead in both time and store requirements which must be kept to a minimum consistent with reasonable flexibility. This is particularly important to the customer since a supervisor which imposes a large number of interface requirements on the program modules it controls would make program modification extremely difficult to implement with any confidence without a highly detailed knowledge of the complete program.

b) Major Program Components

The program is broken down into four major components:

- task packages
- common sub routines
- program data base
- supervisor

which are organised as shown in Fig. 7. These are briefly described below:

Task Packages - These are the main program modules and their breakdown is determined principally by function and iteration rate. Packages are activated by the supervisor at their required frequency and all communication between packages is achieved via the data base.

Common Sub routines - when a number of packages require a similar calculation the routine for this calculation is common. Access to the common sub routine is achieved via the data base.

Program Data Base - the data base comprises all non-code storage used by the program and covers the following areas:

- input/output areas
- work store
- computer variables
- constants
- link points

Supervisor - the supervisor provides the overall control of the programs and is itself activated by the interrupt from the real time-clock. The most important routine in the supervisor is the Package Scheduler. All task packages are activated in a predefined sequence and the task of the scheduler is to define the sequence at the start of any one program cycle of the packages to be entered during the cycle. When a package has been completed it returns to the scheduler which activates the next package in sequence. In addition it controls the following:

- interruption of lower priority packages by higher ones
- allocation of work store areas which are shared between packages at the same iteration rate
- selective activation and de-activation of packages according to instructions from other packages.

c) Package Organisation

All packages are entered from the scheduler and the order of entry depends on the package iteration rate and the logical dependency within the program. Packages iterated at a high rate are:

- navigation moding and velocity calculation
- fixing and weapon aiming, moding and status
- fixing and weapon aiming calculations
- pilot display drive

Packages iterated at a medium rate are:

- steering

Packages iterated at a low rate are:

- navigator display control handling
- pre-smoothing for optimum filtering for navigation
- navigators display moding, drive calculation and output

Optimum Filtering Package are iterated at even lower rates. Finally, a Main Computer self check is carried out as a background task in the remaining spare time. The organisation of the software packages is shown in Fig. 8.

2.2.2 Rules for Program Preparation

A Programming Guide, prepared by the IST, is provided to all programmers in the Team using the LITEF Spirit 3 Assembler Language for TORNADO Main Computer programs. Specific references are given, throughout the document, to the requirements for the Operational Flight Program but the underlying principles, providing an engineering discipline for software development, are applicable for all main computer programs produced by the IST. The guide details program construction, components and constraints and how hardware interrupts and program level facilities are to be used.

Programming standards are given with special regard to the appearance and usefulness of the source code listings, which will be the prime source of information for programmers later responsible for the maintenance of main computer programs produced by the IST. Interface requirements between program sections are defined and the use of the Assembler is also covered in the Programming Guide. Programmers involved in preparation of operational flight programs will thus use the guide as a basic information standard together with the Design Data Set for the applicable program (software series) defining all the data on which the program is to be based.

The importance of the source code listings cannot be over emphasised when considering future maintenance activities. Every effort has to be made to ensure that the source listings are regarded as the most reliable document produced by the IST. Reliability in this area means that these documents reflect the current state of a particular program more accurately and in more detail than any other documents produced by the IST. They are, by definition, the most up-to-date type of documentation because they are automatically produced by the updating process. In order that all the information necessary for the support and maintenance of a program can be incorporated into the source listings in a logical and standardised manner the IST are required to adhere to requirements contained in the Programming Guide when creating a new source list or updating an existing one. The source line format is fully defined. Great importance is attached to adequate comment in the listings. It is required that all coding is fully commented to convey the global role of an instruction and not simply a literal translation of the instruction into English. In general this will consist of a comment per line of code. However, if a particularly difficult, obscure or elegant instruction sequence is used a paragraph of comment is required to precede the section of code. All sequences of code are broken down into small functional sections which are preceded by a paragraph of comment giving a brief title to the section and, if appropriate giving the name of the responsible programmer, a reference to the applicable SWR, a reference to the applicable software specification and a list of applicable Software Requirement Change Requests. Because of their importance source code listings are subjected to quality assurance audit to ensure they conform to the requirements.

2.2.3 Program Documentation

The purpose of the development program documentation is two-fold. Firstly,to provide information to all programmers to ensure that software packages are compatible, to enable programmers at test sites to make rapid modifications and to provide a better understanding of the avionic system to test engineers and aircrew. Secondly, to provide a foundation on which the in-service user technical publications can be based for later program maintenance purposes. The development documentation is organised on three levels of progressively increasing detail. The documents corresponding to these levels are program specifications, package specification and flow charts and source code listings. The organisation of the documentation is shown in Fig. 9.

The program specification, based on the Software Requirements, describes the program components, packages and sub routines, specifies the program data base and input/output area and defines the functions of the scheduler. The package specification, is also based on the Software Requirements and is task orientated. The package internal structure, routines, sub routines and interfaces are all defined. The requirements for flow charts and source code listings are covered in the programming guide. The aim has been to conduct the program and package software writing task in parallel with the preparation of program documentation. The two tasks should run together.

2.3 Software Testing

The progressive approach to software testing, within the overall avionics development programme, has been outlined in section 1.4 and is examined in more detail here.

In order to define and control the testing of software on the various rigs a documentation system has been evolved which specifies the tests to be carried out and defines the test methods according to the rig facilities on which the tests are to be run. While a program is being written by the IST programmers the design engineers in the CDMT are defining and developing test methods as an independent check. When the program is handed over to Stage 2 for proving (i.e. checking for satisfactory operation with prime equipment) tests are developed in detail and Test Specifications and Procedures brought to a state to enable formal software tests to be made. After the formal tests have been made, prior to formal release of a program for delivery to flight trials sites, a statement of the standard of the program is made, based on the completed test procedure, in the technical description delivered with the program.

2.3.1 Test Stages

2.3.1.1 Stage 1

The major function of Stage 1 is to provide a working environment for the IST to develop and test programs. Testing takes place in three phases.

Phase 1 - Testing of individual program modules

Phase 2 - Initial integration of a Software Series

Phase 3 - Final integration and complete program testing.

The facilities to achieve this are provided by two rigs - Stage 1A and Stage 1B. Stage 1A principally provides facilities for assembly, compiling, editing and debugging of the software. Stage 1A is a commercial facility, a SIEMENS 4004, with disc and magnetic tape backing storage and with line-printer, visual display unit, card input/output and paper tape input/output as shown in Fig. 10.

Two main programs operate in the Stage 1A computer:

- the Assembler and
- the Emulator.

The Assembler accepts the main computer assembly code card decks and generates object code paper tape form suitable for loading into the main computer.

The Emulator program provides a software model of the Spirit 3 instruction repertoire on which programs are initially run. The Emulator accepts either object code card decks output from the Assembler or paper tape and generates trace outputs as specified by the control cards. In addition the emulator generates a paper tape binary version of the loaded program suitable for direct loading into the main computer.

Stage 1B provides an environment in which the real time aspects can be checked in a Spirit 3 computer but without the complications of a full avionic equipment rig. By testing programs on this rig greater confidence can be gained in the software before it reaches Stage 2. Problems arising at Stage 2 which are directly attributable to the software will be fed back to the IST for investigation at Stage 1B. The Stage 1B rig comprises the main computer and necessary peripherals together with the essential Navigators Displays and Controls. Stage 1B is supported by an external computer facility based on a PDP 11/40 which both stimulates and monitors the programs running in the main computer and provides facilities for open loop simulation. When stimulating the Stage 1B rig dynamically the PDP 11/40 will use data recorded on magnetic tape resulting from the processing of avionic system computer models provided by the CDMT. The resulting performance of the operational flight program will be compared with the predicted performance of the program derived from the models.

2.3.1.2 Stage 2

The Stage 2 rigs provide facilities for avionic equipment and software integration and were the first stage where the software and hardware were brought together. Due to the high workload and compressed timescale two Stage 2 rigs, one at EASAMS and one at ESG have been initially operated in parallel with each other taking the lead in a different sub system area. The workload has now decreased to a level requiring only one Stage 2 rig. The EASAMS rig has now been closed and the remaining system and software development task has been transferred to the rig at ESG, now operated by the International Rig Team (IRT) staffed by EASAMS, ESG and SIA.

A "bird's-eye view" of typical rig layout is shown in Fig. 11. Equipments were integrated into the rigs in a planned sequence. Control of the standard of equipment integration was by testing against defined procedures starting with single equipments and continuing through a controlled build up to an integrated group which is used for testing a particular software series.

The following prime equipments are used on Stage 2 for software and system testing:

- Main Computer
- Navigators Displays and Controls
- Navigation Sensors
- Pilots Displays and Controls
- Flight Control Equipment
- Forward Looking Sensors

The equipments are provided with supplier's special-to-type test equipment (STTE). The STTE together with standard laboratory test equipment was used initially for 'on arrival' testing and for control and monitoring of equipment functions during hardware testing. Each Stage 2 rig was also supported by an external computing facility which comprises a PDP 11/40 digital computer, a data interface unit and an analogue computer. At ESG this facility is shared between Stages 1B and 2. The facility is used for data acquisition and replay, for open loop testing and for closed loop work as shown in Fig. 12. For closed loop work simple models of the aircraft control system and aerodynamics are used to enable software and system functioning to be examined in a reasonably representative dynamic environment. Facilities are provided for driving the basic and forward looking sensor equipments (downstream of r.f. and inertial elements) so that operation of Main Computer software can be checked with data which have the characteristics appropriate to the outputs from the data processing elements of sensor equipments.

2.3 1.3 Stage 3

Stage 3 flight trials are essentially concerned with the flight development of forward looking sensors and the Navigation and Weapon Aiming sub-systems. Responsibility for planning and conducting the trials rests with the CDMT. The concept behind the Stage 3 program is to expose the Avionics System to the critical high-speed low-level sub-sonic flight regimes which will be experienced in TORNADO at an early stage in the system development.

Stage 3 provides the first opportunity to conduct airborne trials of the integrated sub-system and thus problems associated with software operation under dynamic condition will be revealed leading to recommendations for both hardware and software modifications as necessary. A Bench Harness at Stage 3 provides a facility for exercising software on the ground in the TORNADO avionics system configuration as flown in the Buccaneer aircraft. Facilities include test equipment for acceptance and maintenance and for limited manual stimulation to establish an integration standard (software and hardware). The Buccaneer aircraft is fully instrumented, as shown in Fig. 13, to record the performance of the hardware and software within the flight envelope of the aircraft. The test flights are also carried out over instrumented ranges and the results from both the range and aircraft instrumentation are then transferred to the ground replay and analysis system for subsequent processing. The Stage 3 ground replay and analysis system provides 'quick look' facilities for the 'on site' decisions by the trials team and collates and formats data for detailed performance analysis by the CDMT.

2.3.1.4 Stage 4

Stage 4 rigs, at the aircraft companies, provide a logical continuation of the hardware and software integration tests performed during Stage 2 activities. The emphasis is on integration with the aircraft itself and includes equipment checks on installation, cooling and aircraft power supplies. Further tests are conducted with both hardware and software the ultimate objective being to achieve clearance for flight.

2.3.1.5 Stage 5

Stage 5 flight trials are aimed at proving the entire avionic system in the TORNADO itself and achieving operational clearance. At Stage 5 software testing is not conducted in isolation but occurs as part of each trial of the avionic system. At both Stage 4 and 5 queries and problems attributed to software are handled through the same procedures developed for Stages 2 and 3.

2.3.2 Control of Testing

Tests must be both visible and repeatable. Tests therefore must be completely documented and recorded. It also means that the status of a software series and the status of the rig equipment must be compatible and completely identified so that problems arising from an integration test can be returned and investigation repeated.

Three series of test documents have been produced during the software development programme. These are:

- Outline Test Requirements (Annex to the Software Requirement)
- Software Test Specifications
- Software Test Procedures

A Software Test Specification is produced for each software series from which software test procedures are produced for each rig and test site. Procedures have also been devised for the management and control of software queries, errors and program changes. Fig. 14 shows the overall flow of software development testing and covers the relationship, described in this section, between the test documentation, the control procedures and the test activities. A hardware/software compatibility matrix is also maintained for the test sites.

2.3.2.1 Outline Test Requirements

The Outline Test Requirements are produced by the CDMT. Generally these are produced by the authors of the Software Requirements and lay down in general terms the tests to be carried out on the software to establish its effectiveness. Only general attention is given to test facilities since specific tests are tailored to the individual rigs by the Test Procedures. These Outline Test Requirements form the basis of the test specifications.

2.3.2.2 Software Test Specification

The Software Test Specifications define the tests to be performed on a particular Software Series at all stages. These specifications list the parameters to be tested, together with their values and ranges and the expected results under the defined conditions are tabulated. The basic requirements of the Software Test Specifications are then included in the Functional Test Procedures for Stage 1B and Stage 2 and in the Flight Test Procedures for Stage 3.

Software Test Specifications are produced by the CDMT. Initial definition of the specification will be taken from the Outline Test Requirements produced by the authors of the Software Requirements Data derived from CDMT studies will be used to point to areas where problems are likely to arise and results from system performance simulations will be used to check main computer program results. The Software Test Specification (and Procedures) will be arranged so that the simple most fundamental tasks are carried out first in order to progressively increase confidence in the program. The following levels of test are defined in the Software Test Specification:

a) Routine Tests

Initial tests on software are designed to establish that the program has entered store correctly and will cycle without fault warnings being indicated.

b) Static Tests

The program logic is checked to establish that all modes specified in the Software Requirements can be entered correctly. For this purpose all acceptable and obviously unacceptable key actions and orders of operation which are effectively different, are checked. Static input values are set to exercise the program in each mode and the relevant results are tabulated.

c) Open-Loop

Input parameters are varied manually or automatically, initially without any particular attempt to be representative of true system values to establish the behaviour of the program. Subsequently, long term tests with static or limited dynamic inputs are carried out to determine equation stability, error build up and system lags. Predictable worst case as well as typical values are used to establish that the program will operate with unfavourable input conditions.

d) Closed-Loop Tests

Limited closed loop dynamic tests are carried out to gain the maximum confidence in the software prior to flight tests. The complexity of the closed-loop testing is determined by the sophistication of the rig facilities and the time available for testing. A practical example of closed-loop testing is the closing of the azimuth loop for steering and limited weapon aiming checks.

e) Flight Tests

The final stages in software/system development are the flight tests, firstly at Stage 3 in the Buccaneer and then in the prototype series TORNADO aircraft at Stage 5. During these stages specific software checks are not defined but the functional performance of the avionic system will be established with due regard to the contribution of software. A number of areas within the software will obviously not be fully checked until flight testing commences since it is only possible to close all loops when the equipment is under dynamic conditions.

2.3.2.3 Software Test Procedures

Software Test Procedures are produced for each rig and are based on the tests defined in the Software Test Specification. These procedures are tailored to the facilities available on the particular rigs and provide step by step instructions to the operator. All tests specified in the Test Procedures are designed to be repeatable. The layout of Test Procedures enables the set-up procedure, expected result and observed result to be tabulated together on one page. The Software Test Procedures are used for reporting the results of software tests and are signed by the appropriate inspection and quality assurance authority.

a) Stage 1B Software Test Procedure

The Software Test Procedures for Stage 1B are produced by the IST in conjunction with the CDMT. Formal acceptance of the tests carried out on Stage 1B is not required since the software is not considered accepted until it has been run on Stage 2 against the Stage 2 Test Procedures. Stage 1B however, is the first rig on which an integrated test procedure will be run and as a consequence some 'debugging' of the test specification and procedure will be necessary. By carrying out this activity on Stage 1B the program, when tested on Stage 2, should have had the more obvious misconceptions in either the software or the test procedure filtered out.

b) Stage 2 Software Test Procedure

The Software Test Procedure for Stage 2 is produced by the CDMT in conjunction with the IST and Stage 2 personnel. Experience gained in using the test procedures for Stage 1B is reflected in Stage 2 Software Test Procedures and the tests are expanded to accommodate the additional facilities available in the Stage 2 rigs. Testing on Stage 2 of the software functions is divided into two sections - Software Proving and System Function Testing.

- Software Proving

  Software Proving commences when a particular software series is delivered to the Stage 2 rigs. A period of 3 to 5 months is allocated on the rigs for this exercise, the length of time being dependent on the particular software series under test. Tests at this stage are against the Stage 2 Software Test Procedures, during which clearange of queries and program will continue, culminating in a formal test against the Procedures. Completion of a formal test marks the milestone of formal delivery of that particular software series to Stage 3 and 4.

- System Function Testing

  This phase of testing is not aimed primarily at software but at the overall system operation combining software and hardware as an integrated system. Thus, with the increasing complexity of tests, software testing will merge with the system function tests. A number of tests are run prior to certain flight trials using test conditions representative, as far as possible, of a planned flight profile. The data for setting these test conditions is provided by the CDMT from outputs derived from the avionic system computer models. The primary object is to avoid abortive flight trials due to a system anomaly which could have been established on the ground.

c) Stage 3 Flight Test Procedure

Flight trials are undertaken against a Flight Test Procedure. This procedure will again embody test requirements and profiles which have been specified by the CDMT. The detailed versions of these procedures will be prepared by the CDMT Stage 3 team taking account of the restriction of the ranges and aircraft available. The contribution of software to flight performance is checked by testing integrated functions within the avionics system. Particular manoeuvres in trials may be designed to exercise the particular elements of the system including software function.

Prior to equipment being installed on the aircraft an initial integration of the hardware is carried out on Stage 3 bench harness. Similarly software is also exercised on this facility since it is the first time it will have been tested with flight standard equipment. Test Procedures will be laid down for this exercise and will be tailored according to the facilities for stimulation available on the bench harness. These tests will be based on the earlier tests carried out during the software proving exercise on Stage 2.

d) Stage 4 Test Procedures

Procedures, produced by the aircraft companies, are designed and used to test the software for flight clearance purposes. The objective will be to ensure the software is safe to fly and that meaningful results will be obtained from flight trials.

e) Stage 5 Test Procedures

Procedures for TORNADO flight trials, produced by the aircraft companies, as for Stage 3 are not designed to test software in isolation from hardware. The objective is to progressively test the Weapon System as a whole and achieve operational clearance.

2.3.2.4 Procedures for Handling Software Queries and Program Changes Arising during Testing

In any extensive software development programme, as in the TORNADO project, early establishment of procedures for the management and control of software queries, errors and change is essential. Specific procedures have been defined for:

- the control of software queries
- the control of program changes
- the control of changes to software requirements

The first of these procedures, that for software queries, may require one of the other two procedures to be invoked as dictated by the answer to the software query. The purpose of the software query procedure is to identify and record software problems, together with their solutions, encountered at the IST, CDMT or any other flight test site or rig establishment. The initiator of a query is required to complete a form in accordance with the rules of the procedure and forward it to the IST representative at the site; or directly to the IST who are responsible for recording and co-ordinating all software queries as they are received from the test sites. The CDMT and IST will jointly investigate queries. Solutions generally fall into the following categories:

- operator error due to misunderstanding of system procedures or status. In such cases a full explanation is entered on the query form and the query is cancelled.
- further investigation required on-site. The originator is required to provide more information
- hardware/software incompatibility which cannot be rectified by software change
- program change required
- software requirement change required
- procedural change required.

In all cases once a solution to a software query has been agreed and validated it is copied to all test sites. The purpose of the program change procedure is to enable control of software changes and corrections to programs.

All test sites, the CDMT and the IST may raise change requests for programs that have been officially issued by the IST. A program change, once agreed and validated, may also necessitate a software requirement change. As in the case of queries the initiator of a program change completes a form in accordance with the rules of the procedure and sends it to the IST, with copies to all test sites

and to the CDMT, for assessment. The CDMT and IST will assess the request for system design and software aspects respectively and if necessary may return the change request to the initiating site for further appraisal. The next step is for the IST to program the change and test it at the initiating site and also to issue the change, on a provisional basis, to all other sites. Once the program change has been checked as satisfactory, clearance of the program change tape is given by the IST to all test sites.

2.4 Software Delivery

Once a satisfactory level of testing has been achieved at Stage 2 "advanced" delivery of programs is made to Stage 3 and to the aircraft companies for further testing at Stages 4 and 5. Advanced delivery from Stage 2, prior to the programme milestone of formal delivery, enables an earlier start to be made in program testing at the later stages and permits the development programme impetus to be maintained.

Formal delivery of software involves the following steps, all of which contribute to the quality of the programs and hence their ultimate reliability and maintainability. The steps are:

- formal testing of the program against a validated test procedure under quality assurance inspection conditions
- preparation and identification of program material for delivery in accordance with defined procedures
- preparation of associated program documentation
- certified release of the program
- storage of master program material in protected conditions

2.4.1 Formal Testing under Quality Assurance Conditions

The object of formal testing of programs and program changes under inspection conditions is to ensure that the required comprehensive tests have been properly carried out and that statements of program standard can be given. Such testing contributes directly to certified release of programs to PANAVIA in accordance with quality requirements. A formal test of this kind is conducted at the end of the test phase at Stage 2 when the technical specialists are satisfied that the program is as free as possible of errors and that the test procedures are sufficiently comprehensive. The timing of formal testing is also constrained by the requirements of the overall avionics development programme.

On notification that formal testing can commence quality assurance representatives will:

- ensure that the program test procedure has been agreed by the appropriate technical authority
- check the program status, test software status and rig equipment and configuration status
- witness the conduct of the test formally recording all deviations and ensure that a test report is prepared.

2.4.2 Preparation and Identification of Deliverable Program Tapes

On completion of formal testing a master program tape is produced together with a master source code listing. An official identification code is also allocated for deliverable versions of the program tape. Prior to preparation of a deliverable tape the master tape is fully or partially analysed and compared with the master listing to confirm that the correct master tape has been received. The identification code is also punched on a separate tape. The identification tape and master code tapes are then copied in a continuous sequence on Mylar tape to form a deliverable program tape. This tape is then verified against the master tape to ensure the copy process is correct. Finally, a quality assurance certificate is prepared to record and certify that the procedure has been followed.

2.4.3 Preparation of Associated Program Documentation

The deliverable program tape is accompanied by a Technical Description and a source code listing. The Technical Description provides information such as the purpose and characteristics of the program, references to applicable documents, differences from previous issues, technical details and descriptions, comments and restrictions for users, tests undertaken, deviations, concessions and requirements not programmed, self check and fault diagnosis, preparation of input data tapes, tape loading order and all outstanding software requirement change requests and software queries.

2.4.4 Certified Release of the Program

All deliverable program tapes are certified by quality assurance as being true copies of the master, this being stamped on the tape heading. A release note accompanies all software deliveries and is used to directly relate the deliverable program tape, the procedures used for the formal quality assurance test, the technical description and the source listing.

2.4.5 Library Facilities

All master program tapes are stored under conditions designed to record and control their movement, to ensure that all tapes are correctly identified and to protect all tapes from damage.

3. SUMMARY AND CONCLUSIONS

The development of software for TORNADO started in depth in January 1972 after the initial feasibility and definition studies had been completed. Stage 2 testing of software started in June 1973 with Software Series 1 and is still underway with testing of Software Series 6 (the initial production standard) at this time. Stage 3 trials commenced in January 1974 followed by Stage 5 in April 1975. Production software is now under test at all stages and is being evaluated by the users at their national air test centres.

Over the development period the methods, standards and procedures described in Section 2 have been evolved and applied. At this time, looking back over the programme, it is possible to identify key features which contribute to software reliability and maintainability and to draw conclusions and make recommendations under the following headings:

3.1 Management and Organisation

Key features:

- Early definition of organisation, responsibilities and work plans
- Early definition of operational procedures
- Involvement of the software user from the start
- Application of constructive quality assurance
- Good communications and adequate personal contact.

Close liaison between the software design team in the CDMT and the software programming team in the IST would have been improved if both had been at one site but international worksharing requirements prevented it. Geographical separation has been a problem.

Early establishment of procedures for definition, writing, modification, testing and issuing of software is obviously necessary but management action is vital to ensure the procedures are available before the results of the work. Of particular importance in this area is the development of associated audit and quality assurance controls to ensure that standards are being followed and met and that the product is subject to inspection.

Close liaison with the customer and user has been both essential and beneficial. Communication with the customer, using the program definition documentation as a means, has helped to remove misunderstandings and to get program specification correct. User representatives have been resident with the CDMT and IST to participate in the software development programme and have formulated the policy for in-service maintenance of software.

3.2 Software Definition

Key features:

- Formal Software Definition Documentation keyed into the Avionic System Design Data Base
- Early definition of formal procedures to control software definition
- Breakdown of the software into "prototype" programs of increasing completeness and complexity tied to overall avionics development objectives.

The Software Requirement documentation has generally proved satisfactory but the need for careful validation and integration of the documents before programming started soon became evident. Specifications of this kind must be detailed and in-depth. Special note should be made of the fact that an author of a Software Requirement has to attach an annex to it giving test recommendations. The object is to ensure the production of testable requirements.

3.3 Software Writing

Key features:

- Programs structured to enable precise specification of modules for coding purposes and subsequent maintenance
- Early production of programming standards
- Production of program documentation in parallel with code
- Language for program implementation

Early production of programming standards contributes directly to reduction of errors, and hence reliability but experience showed the value of continual review and the need to tighten standards in the light of experience.

The Assembler program itself has been used to impose programming standards by placing restrictions on allowable code. Further enhancements to the Assembler, for this purpose, will continue to be developed.

Automation of program documentation, prepared whilst programs were being written, would have brought forward its availability and hence increased its accuracy and is recommended for implementation. The Host computer system used at Stage 1A must therefore be large enough to enable the documentation to be held in the system together with the program.

From the program writing and maintainability viewpoint there would have been benefit in writing programs in a high level language instead of Assembler but available computer store and time loading did not permit this. As always the task expanded to fill and overflow the available store in spite of storage increases, in the main computer, during the development programme.

Development delays have been reduced by ensuring that the programming team checks all programs and documentation, prior to delivery, against defined test procedures and quality assurance requirements. It is apparent, however, that errors can be detected early prior to testing by visual inspection of programs by persons, other than the author, with good system and software knowledge. Implementation of formal visual inspection procedures is recommended.

3.4 Software Testing

Key features:

- Software testing by the programming team, on a Host Computer (Stage 1A)
- Early testing on the operational main computer (Stage 1B)
- Early software integration with representative equipments (Stage 2)
- Early airborne testing (Stage 3)
- Further stages of test (Stages 4 and 5)
- Formulation of test procedures, at all stages, to check every testable path
- Production of test procedures by the system design team, rather than the programming team, as an independent check
- Creation of a system to record and respond to all software queries arising at any test site.

The objective of the test philosophy followed in both the hardware and software development programme has been the early detection of problems and errors. This has required substantial investment in both test hardware and test software as seen in Stages 1 and 3. The aim was to reduce avionics testing on TORNADO, at Stage 5, to a proving exercise the essential development having been completed in the earlier stages. This aim has been largely but not totally achieved. Testing at each stage, with extension of facilities and with checks under dynamic conditions, revealed particular groups of problems and errors. Basic program errors were detected at Stage 1. Errors associated with hardware/software integration were detected at Stage 2 while most dynamic software system problems were not detected until Stage 3 and Stage 5. The number of errors arising showed a downward trend as a software series passed from one stage to the next but the error frequency generally rose to a peak at the commencement of testing at each stage.

Consideration could be given to further extension of the capability of the Stage 1A and 1B test rig used by the IST. Due to improvements in commercial Operating Systems more operator interactive program testing may be performed on the commercial host computer at Stage 1A. The programming team would thus be able to undertake more extensive program testing than previously was the case. At Stage 1B the external computer, linked to the main computer, could be enlarged to enable the simulation of avionic equipments by software. This is recommended because it enables more thorough testing of programs before delivery to next stage users, makes the software testing more independent of hardware availability and makes possible the simulation of different equipment modification states. The latter is particularly valuable when software maintenance is performed centrally for sites with differing configurations of equipment. Such improvements in Stage 1 facilities would give the greatest benefit for software implementation in a high level language.

Whilst initially program testing was based on manual operation of test procedures increased emphasis is being given to automation of program testing. When a modification is introduced into a program it is necessary to check that modified program remains totally correct by exercising all possible test paths through it. Automation of such repetitive testing saves time and ensures its completeness. In addition software has been developed to enable automation of test case construction to contribute towards more rapid and reliable testing.

In a development programme such as TORNADO where software is directly under test, or under test as part of the avionic system, at some seven sites in the participating countries the need for a well defined software query and programme change control system cannot be stressed enough. Flexibility had to be built in, to meet requirements as program testing spread out from Stage 1 through to Stage 5 in all countries, and rapid response to keep pace with the development programme. The latter required programming team representatives at some sites empowered to make changes to software immediately on-site.

3.5 Software Delivery

Key features:

- Formal test prior to delivery under quality assurance conditions
- Preparation and identification of program material for delivery under controlled conditions
- Certified release relating program material, test procedures, technical documentation and source code listings.

The features listed above all directly contribute to definition of tests in the field and the relationship of hardware and software. Careful control of these features is essential in order to relay software queries back from the test sites to the central programming team in an identifiable manner.

3.6 Conclusions

We believe that the attention given to the key features, described in this paper, have contributed to software reliability and maintainability in the TORNADO programme and we hope to continue to improve and extend such features in the production and in-service phases.

On the basis of experience gained in TORNADO emphasis, in any future project, should be placed on:

- Use of formal system description and high level languages
- Production of documentation covering all aspects of software definition, writing and testing
- Comprehensive testing as outlined in this paper
- Management and maintenance of a skilled programming team during the life of the project.

4. ACKNOWLEDGEMENT

The author wishes to thank his colleagues in the CDMT and IST for their help in the preparation of this paper.

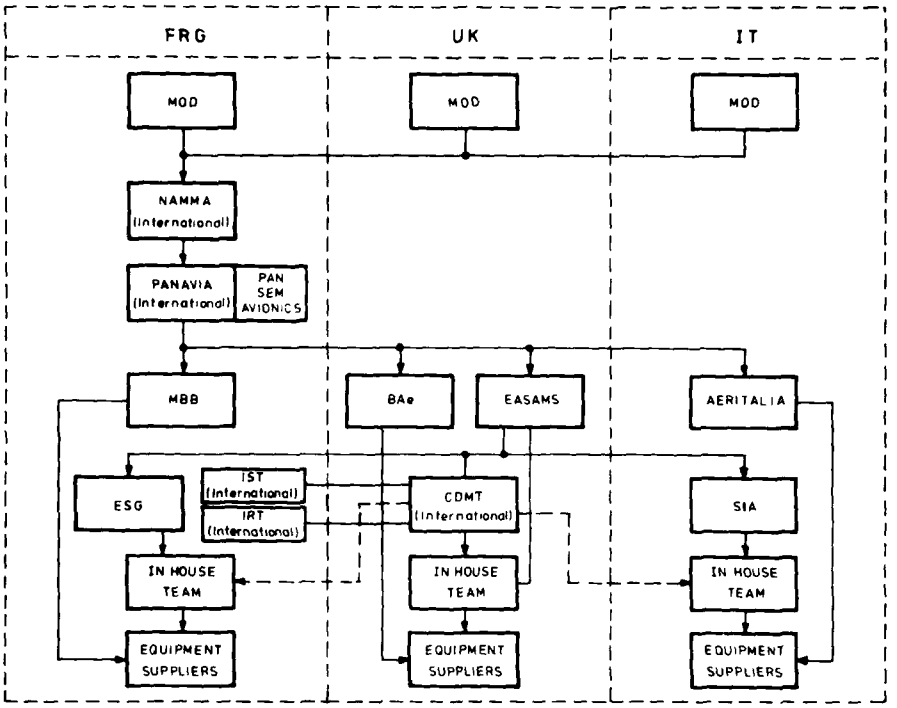


Fig. 1 TORNADO PROJECT ORGANISATION FOR AVIONICS DEVELOPMENT

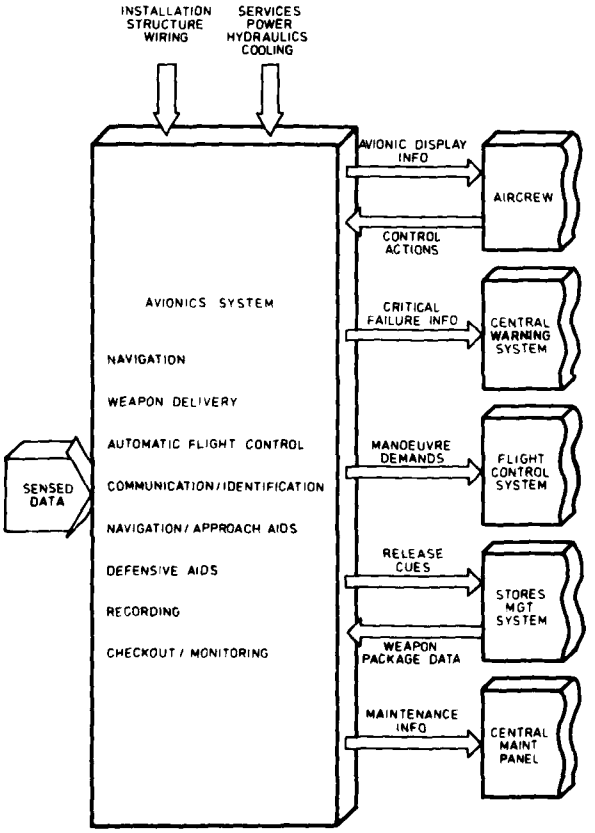


Fig. 2 AVIONICS SYSTEM - MAJOR FUNCTIONS AND INTERFACES

NAVIGATION
- Basic Sensors
- Horizontal and Vertical Navigation Calculations
- En-Route Steering
- Display of Basic and System Data
- Mode Control

WEAPON DELIVERY
- Acquisition of Targets (and Fixpoints)
- Measurement of Target Relative Position
- Weapon Aiming Calculations
- Attack Steering
- Release Cues
- Mode Control

AUTOMATIC FLIGHT CONTROL
- Vertical and Lateral Guidance Demands
- Manoeuvre Demands in Aircraft Axes
- Mode Selection and Engagement Logic
- Safety Aspects

COMMUNICATIONS, IDENTIFICATION, NAV/APPROACH AIDS
- Communications Control
- V/UHF
- HF
- IFF
- TACAN
- ADF/Homer
- Approach Aids

RECORDING
- Voice Recorder
- Displays Recording
- Accident Data Recorder

DEFENSIVE AIDS
- Radar Warning

CHECKOUT AND MONITORING
- Built-in Test Features

Fig 3 MAJOR AVIONICS TASKS

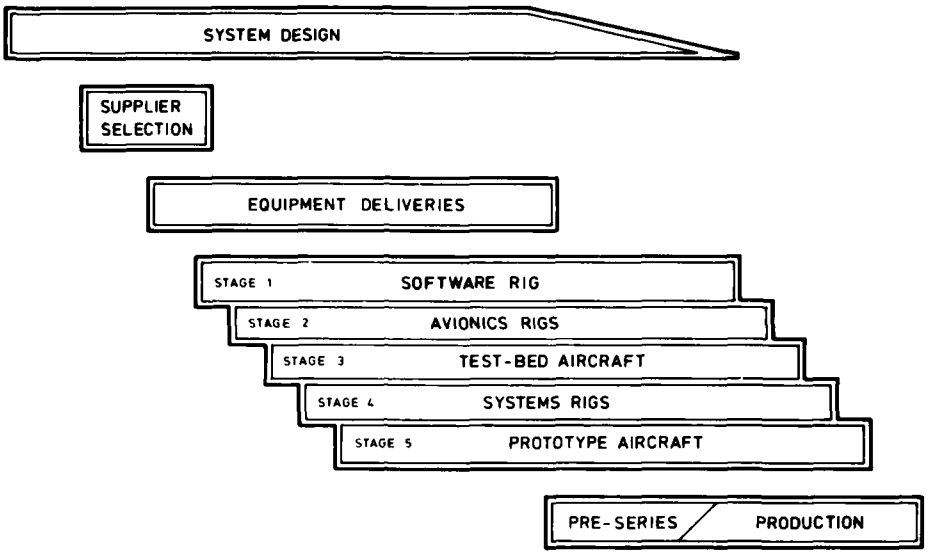


Fig 4 OUTLINE AVIONICS PROGRAMME

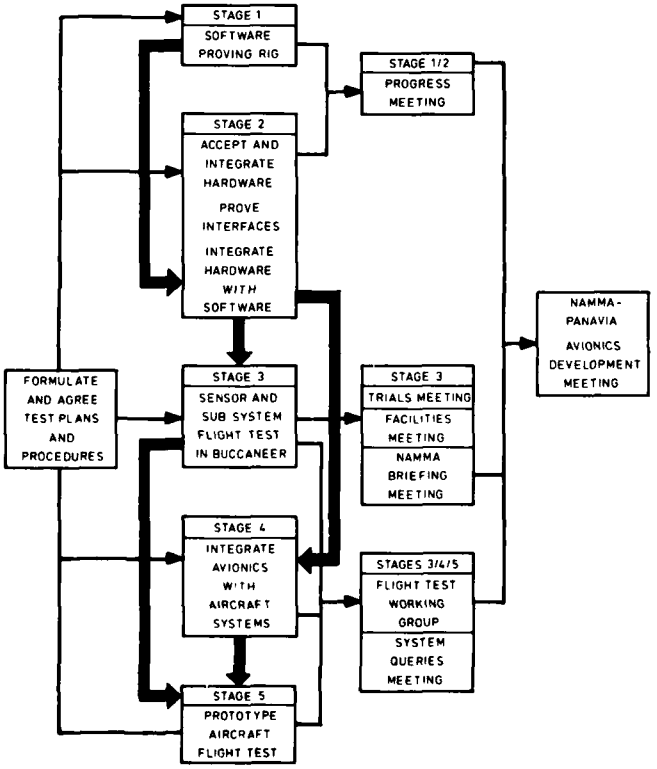


Fig 5 AVIONICS TEST AND INTEGRATION CONTROL

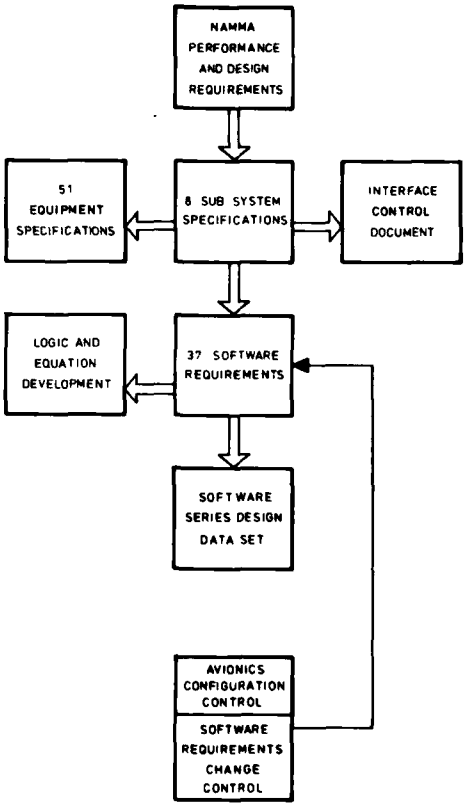


Fig 6 AVIONIC SYSTEM AND SOFTWARE DESIGN DEFINITION DOCUMENTATION

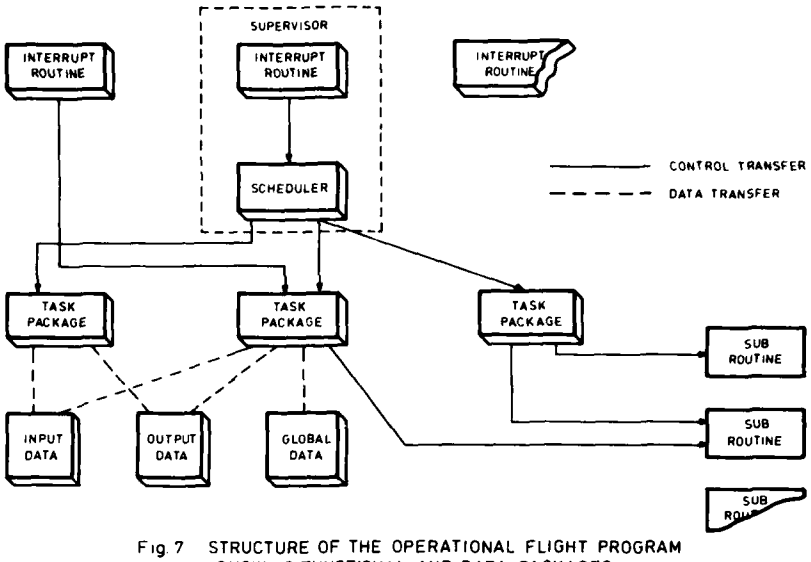


Fig 7 STRUCTURE OF THE OPERATIONAL FLIGHT PROGRAM SHOWING FUNCTIONAL AND DATA PACKAGES

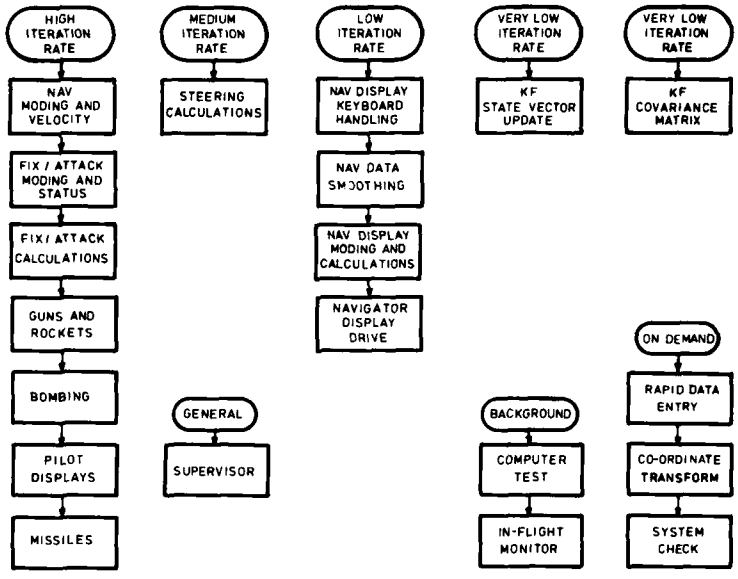


Fig 8 OPERATIONAL FLIGHT PROGRAM PACKAGE STRUCTURE

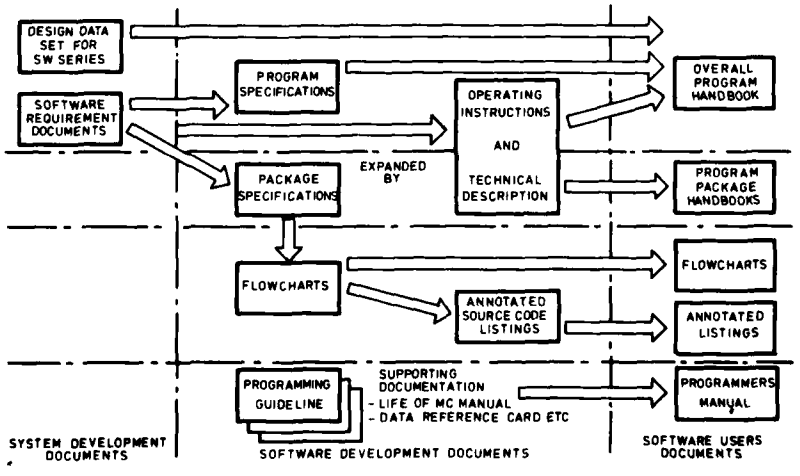


Fig 9 IST PROGRAM DOCUMENTATION STRUCTURE

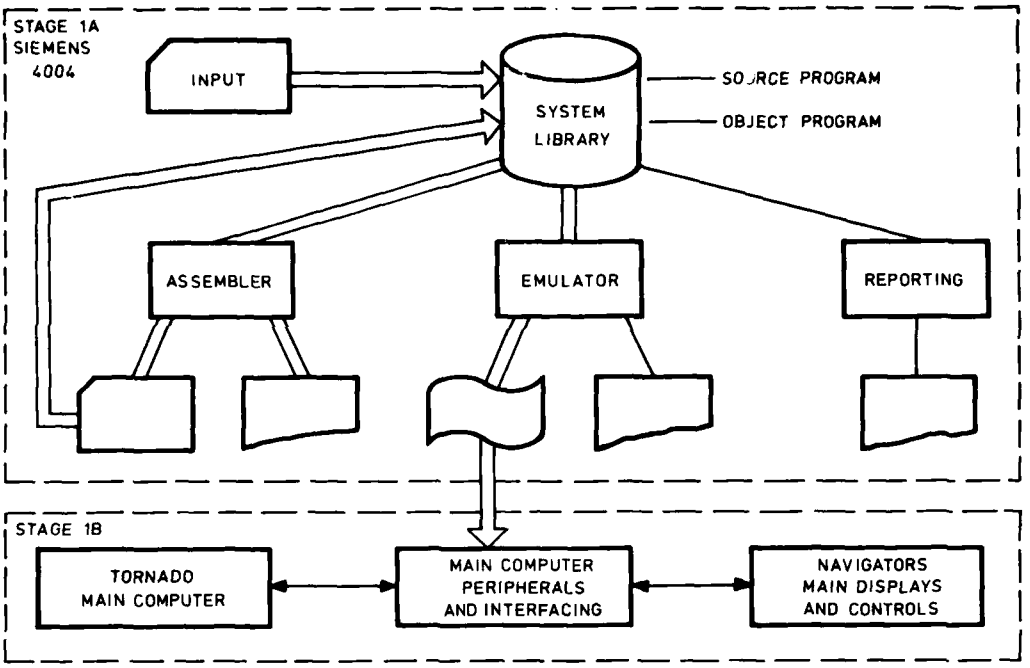


Fig. 10 STAGE 1A / STAGE 1B ORGANISATION

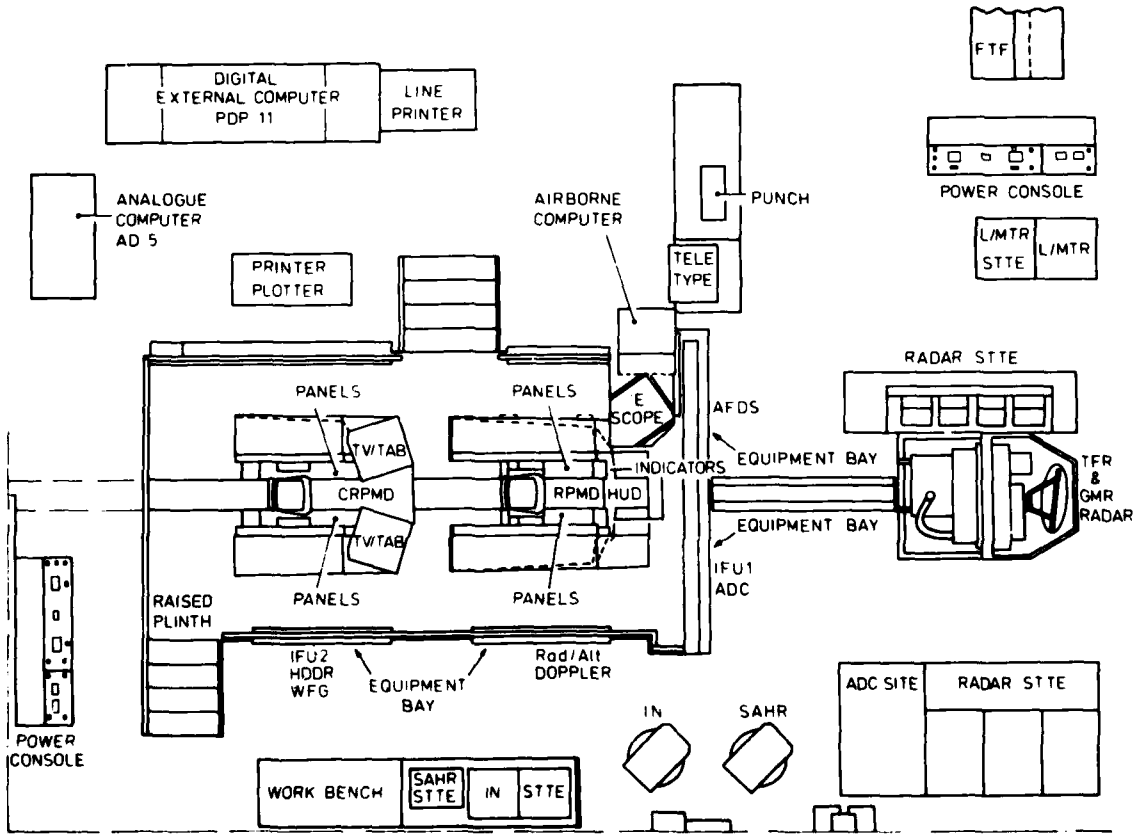


Fig. 11 STAGE 2 RIG CONFIGURATION

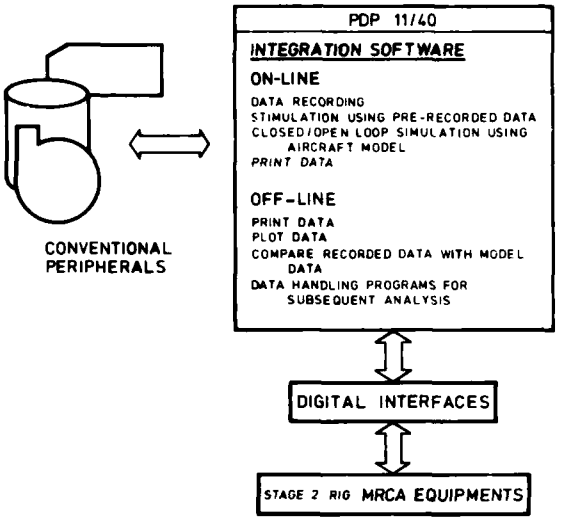


Fig.12 INTEGRATION SOFTWARE UTILISATION

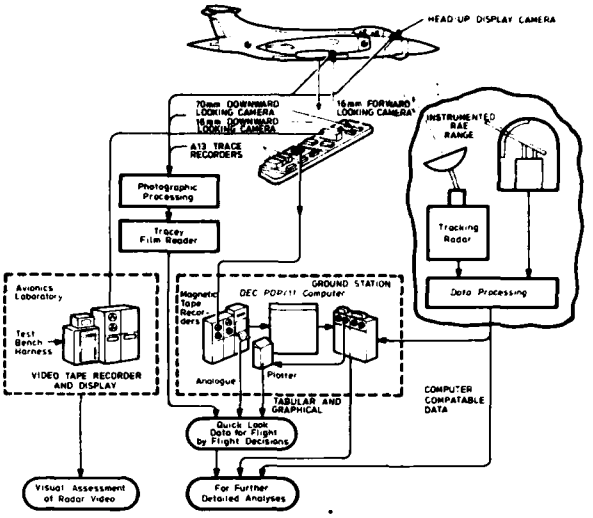


Fig.13 DATA ACQUISITION AND PROCESSING IN BUCANEER AVIONIC HACK AIRCRAFT

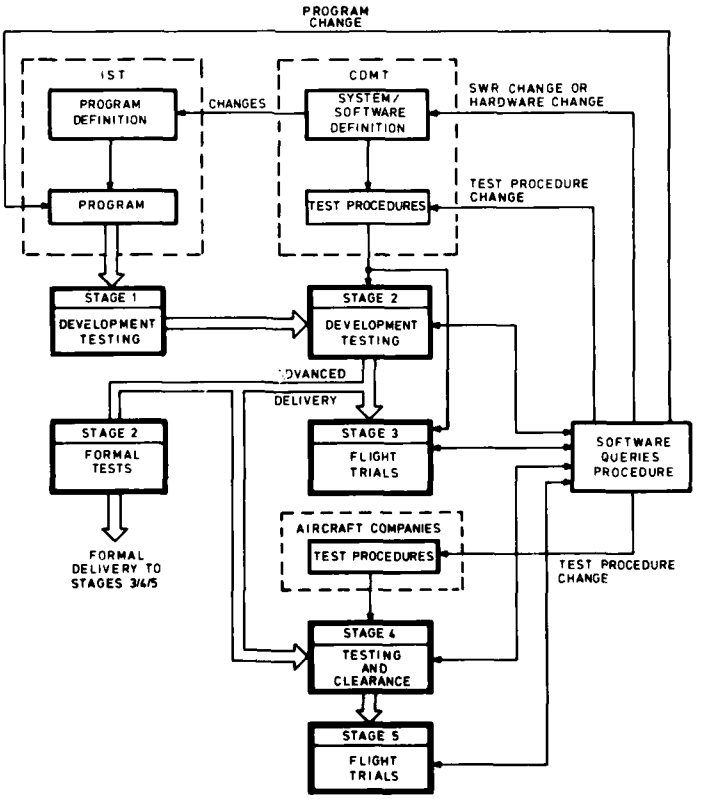


Fig.14 FLOW OF SOFTWARE DEVELOPMENT

DISCUSSION

A.Sukert, US

Of all of the different areas and features that you described, which area or areas do you feel had the biggest impact on improving the reliability and maintainability of the TORNADO avionics software.

Author's Reply

Two features are regarded as having the most impact.

The first of these is *early* detection of errors. The test philosophy followed through stages 1 to 4 has enabled the detection and correction of many software static and dynamic errors prior to development*. Flight trials involving software development have therefore been less extensive.

The second feature has been the concentration on detailed documentation including software requirements, program specifications, source code lists, program technical descriptions and extensive software test schedules and test result reports.

* and flying in TORNADO.

W.Ehrenberger, Ge

(1) Obviously simulation of the software environment provided your major test method. Did you use other test methods too?

(2) There are two possibilities of testing software from a completely statistical point of view, either (a) to test the software completely, or (b) to provide all possible states of the environment with sufficient probability.

You obviously tried the latter. Do you have any quantitive feelings about the completeness of your test and are you giving a probability figure for something arising during the operation which had not been previously tested?

Author's Reply

(1) Yes, in the early stages many of the software requirements were themselves programed in a high-level language to check the logic and computation capability and to provide test data for imput to the testing at stage 2.

(2) It's very difficult to state quantitative figures. We have not taken any predictions or made measurements of the kind that has been described. We ensure as far as possible that the test procedures used are extensive. We make sure that all imput that can be made by the crew are checked.

I can't specify as regards a figure.

F.S.Stringer, UK

How has the visibility of software been arranged?

Can modifications be made in a way which will allow continued visibility despite complexity of the system?

Will the rig remain operational during the service life of the aircraft?

Author's Reply

Visibility of the effects of software has been ensured through the test procedure philosophy followed in the project, especially at stage 2 where hardware and software are integrated together and the effect of software on hardware can be seen.

All software modifications are fully tested at stage 2 to ensure that they themselves do not introduce further errors. Repetitive testing is thus required and automation is being introduced to save time.

A stage 2 rig, or similar facility will be used to provide industrial software support during the production phase and will be eventually replaced by software maintenance facilities, based on stage 2 hardware and software test tools, operated by the service users during the in-service phase.

A.Andrews, UK

In view of the fact that TORNADO software impinges on flight safety (cf. Jaguar which does not) have any new requirements been imposed for integrity?

Do you see such standards being laid down in the future?

Author's Reply

In TORNADO the main computer, and hence the software, are safety involved but not safety critical. While safety requirements, as such, have not been imposed on the software, the safety requirements of critical systems (e.g. terrain following) have required fail-safe design of the software and main computer.

Safety standards for software must feature prominently in future avionic systems which will involve even more distributed computing facilities and associated highway data transmission.

H.A.T.Timmers, Ne

(1) Are redundancy techniques applied to the software, of the kind described by Mr Heiner in his paper NR 29 i.e. to get a solution via different approaches?

(2) Can you quote a figure which a reference to the cost of S/W development e.g. to the total A/C development program?

Author's Reply

(1) Redundancy techniques are applied within the system but not within the software itself. Computer storage and time loading has not permitted the approach suggested in the earlier paper.

(2) The results of cost analyses, that have been undertaken are not yet available and so a reliable estimate for the cost of software development as a proportion of the overall avionic or aircraft development program cannot be given. However, an impression can be gained from consideration of the resources employed in development which included a software writing team of some 40 people, support by the central system design team involving up to 20 people, provision of personnel and hardware at stage 2 and conducting of tests, provision of staff, ground equipment, airborne instrumentation and Buccaneer aircraft at stage 3 and conducting of trials and support to software tests at stages 4 and 5.

INTEGRATED LOGISTICS SUPPORT ADDS ANOTHER DIMENSION TO MATRIX MANAGEMENT

Richard M. Drake
ILS Programs Manager
Westinghouse Electric Corporation
Defense and Electronic Systems Center
Integrated Logistics Support Division
1111 Schilling Road, Hunt Valley
Maryland 21030, U.S.A.

SUMMARY

Matrix Management has been applied to the management of complex Department of Defense Systems. Another dimension is added by applying Matrix Management to Integrated Logistics Support. In other words, the role of the Integrated Logistics Support Program Manager is the Matrix Management integration of the various elements of Logistics Services with the Administrative, Engineering, and Operations functions responsive to Programs (Projects) requirements and Industrial Business or Financial Objectives.

Specially covered will be -

- The advantages and disadvantages
- Typical Organizational Diagrams
- The Program Management Process

Particular emphasis will be placed on Integrated Quantitative Planning and ILS Products. (For example: Communicating and Direction, Life Cycle Cost, etc.).

1. INTRODUCTION

To understand the application of Matrix Management to Integrated Logistics Support (ILS), Westinghouse interprets ILS as a management and functional process for unified, coordinated acquisition of logistic resources required to support systems and equipments at all echelons of maintenance throughout their planned period of usefulness. The concept involves the scientific management of all necessary logistic products and services over the system life cycle with particular emphasis upon coherence, timeliness, execution, and reliability.

All major defense programs are being impacted by a vigorous DoD effort to reduce the cost of ownership over the useful lifetimes of new systems. As each program proceeds through sequential phases of development, contractual documents reflect an increasing emphasis on credible design-to-cost goals, on cost effectiveness as a requirement in engineering and operational tradeoffs, and on reduced total life cycle costs. Mounting historical evidence reveals that logistics support costs represent a high significant portion of total system life cycle costs and often far surpass development and production costs (Figure 1). It should be no surprise, then, that new contracts require a systematic, concerted effort to consider logistics support implications in engineering, operational, and management tradeoffs in each segment of the contract work breakdown structure. DoD expects that a thorough definition of support requirements for each alternative in tradeoffs will have a far-reaching and favorable influence on the ultimate costs of ownership. Recent Requests for Proposals (RFPs) for full scale development programs issued by the military services contain these requirements, and the corollary requirement for an Integrated Logistics Support Program.

1.1 Typical ILS Program

To explain the various elements of logistics services and the integration with matrix management, a typical Integrated Logistics Support (ILS) Program as depicted in Figure 1 is designed to identify all of the planning factors and resources that the customer and user will need to sustain the timely, efficient, and economical support of the system throughout its projected useful life. The typical ILS Program is required to produce the following essential elements of system support:

- An overall concept for the maintenance of subsystems, equipment and software that is compatible with the customer's organizations and procedures for resupply, training, and operational command support.
- Trained cadre of operations and maintenance personnel for the initial system.
- Technical orders, manuals, and other procedural data.
- Packaging, storage, and handling provisions compatible with the intended use environments and modes of transportation.
- Support equipment required for test, operations, maintenance, training, and deployment.
- Spares and repair parts for all levels of maintenance.
- Facilities required to support unique test, operations, and maintenance requirements.

Within the context of the typical system, these are deliverable products. Qualitative and quantitative requirements for each of these elements are determined by a comprehensive Logistics Support Analysis (LSA). (Figure 2). In our company, logistics Engineering is the group or discipline that participates directly in the total system engineering process, interacts with the system effectiveness technologies of reliability and maintainability and develops the logistics data base for the system. It provides the primary support and functional integration of the system engineering technologies and program end items for the ILS Program Manager, as depicted in Figure 3.

1.2 Life Cycle Cost

The combination of current economic trends, rising inflation, the on-going reduction in "buying power", budget limitations, etc. has created an increasing awareness and interest in system/product cost. Through this awareness and interest, we have come to the realization that in numerous instances we do not actually know the total cost to date of many of our systems and products currently in the inventory and being utilized by the consumer. In other cases where systems and products are being evaluated, the measured costs far exceed initial expectations, particularly with respect to those elements of cost associated with sustaining system operation and logistics support. Also, it has been recognized that the greatest impact on total cost results from decisions made at the early stages of the system/product life cycle.

In essence, experience has indicated that we must orient our thinking in terms of total life cycle cost, and not just a segment of cost such as the development cost of a system, the purchase price of a product, or the production cost of an item. Further, we can accomplish much in the area of resource conservation by minimizing overall life cycle cost in the process of designing, producing, and utilizing new systems and products of the future. Thus, life cycle cost becomes paramount in the decision-making process from the beginning, and total cost must be considered as a major evaluation criterion factor along with other parameters such as system/product performance, effectiveness, size and weight, capacity, producibility, supportability, and so on. (Figure 4.)

Matrix Management provides further emphasis in life cycle costing--both from the standpoint of introducing cost as a major parameter in the design and development of a new system or product; and as a management technique employed to aid in the decision-making process. (Figure 5).

1.3 Post Production Support Continuity

Post Production Support Continuity is that ILS management discipline that requires Support Planning and Implementation which provides a cost effective solution to spares and repair needs for the duration of the operational period following the cessation of production utilizing contractor facilities and know how. Both cost and lead time savings will result from such a program.

The proposed approach is one that enlist the contractor financially, contractually, and physically. It will provide a vehicle for lowering operating and support costs by promoting DoD and Contractor efforts to improve reliability and maintainability, to investigate logistic support alternatives and to apply warranties and other contract innovations.

This is one more very important area to consider in discussing matrix management for ILS, since the current defense posture and economy has added emphasis to the necessity for complete life cycle cost (LCC) analysis thru the Post Production Support period. In the 1980's and 1990's, DoD logisticians must cope with logistic support of existing inventories dating back, in some cases, to the 1950's. (Figure 6)

1.4 Reliability Improvement Warranty

An additional contracting technique for encouraging contractors to design equipment with the optimal life cycle cost is the Reliability Improvement Warranty (RIW). An RIW is a provision in a fixed price acquisition or fixed price equipment overhaul contract in which the contractor:

- Is provided with a monetary incentive throughout the period of the warranty to improve the production design and engineering of the equipment so as to enhance the field/operational reliability and maintainability of the system/equipment.
- Agrees that, during a specified or measured period of use, he will repair or replace, within a specified turn-around time, all equipment that fails.

A fixed price for the RIW coverage should be agreed upon during negotiation of the acquisition contract, preferably in competition. The objective of an RIW is to motivate and provide an incentive to contractors to design and produce equipment which will have a low failure rate as well as low repair costs. The F-16 program has a contract which includes RIW on the radar. The RIW for the F-16 will be discussed later in this session.

2. GENERAL INFORMATION

Where are we to find the leaders who can bring order out of chaos? Whose philosophy is that there is a place for everything and everything can be put in its place?

As the turn of the century, the American Educator and Philosopher John Dewey said, "The power to command belongs to those who can master the resources available and carry through the actions under-taken".

If this "sounds like a job for superman", don't dispair . . . it's also a job for a Westinghouse Program Manager using MATRIX MANAGEMENT.

The term, "Matrix Management" is a part of our Westinghouse business vocabulary but not everyone uses the term to mean the same thing. In this discussion, it will be treated as an application of Program Management.

"Webster's Dictionary says one meaning Manufacturing, Finance, and Contracts, for example. And, on the horizontal scale would be the numerous programs Westinghouse is conducting to meet customer needs. Each of these programs cut across the customary departmental boundaries".

Complementing this, another dimension is added by the elements of ILS, such as Supply Support, Technical Logistics Data, Training, Logistics Engineering, and Field Engineering and Support.

The Department of Defense has been using this technique for years, and in order to serve that market, we find it necessary to adopt their form of organization. In fact, because of the nature of the products and services Westinghouse sells to DOD, it would be virtually impossible to work in any other framework. (Figure 7).

As an example of top management confidence here at Westinghouse, a Logistics Technology Seminar was recently presented at our Hunt Valley installation. It gave our commercial divisions a chance to evaluate how matrixing might benefit their operations.

2.1 Actual Practice

In actual practice, being a Program Manager under the Matrix System is similar to running an independent business. The Program Manager takes on the role of the owner. He deals with such things as profit and loss statements, investment control statements, and cash flow analysis.

The concept's origins and the system evolved from the traditional Project engineering approach, the Program Manager has a greater responsibility than just solving the technical aspects of the problem.

He works with the Business Management or Contracts people and the Finance Manager in much the same way a private businessman would use his lawyer and CPA.

There are several guiding principles involved in the management of a program. The overriding one is to satisfy the customer. I would caution, this has to be done while ensuring the company of a reasonable profit on the program... profitability is not a dirty word.

Getting things done in this kind of an environment requires a special kind of person. He has to be able to work in a "people-sensitive environment". He has to be able to motivate people to help him satisfy his end needs. (Figure 8).

But most of all, he has to be able to accomplish his tasks by working within the system, without the disruption of the Division's Operations.

He has to be a planner. Based upon sound planning decisions made at the start of a program, he has to be able to look into the future. He is responsible from the start to the finish of a program. Only excellent planning can achieve successful results.

As an anology, I suggest the success of Matrix Management also depends on how well the Program Manager accepts his staff as team members. If the Program Manager is a good leader, each of the individuals on the staff has been made a team member. Then, Matrix Management will work like a charm.

I would like to discuss the average Program Manager. It's a high visibility job. While the individual might be anywhere from a medium-code professional to a high-level manager, he has to be an entrepreneur. He's a risk-tasker. He can't be afraid to stick his neck out. But there's more to it than that, he is responsible for making it happen.

The Program Manager functions as an agent for the General Manager and frequently reports -- reporting on the sensitivity of his program -- to the Division Manager. (Figure 9).

[illegible] been a number of comments that we may have become over-matricized... that you have to [illegible] several layers of management to find a single part. We have to admit, this has [illegible] at times, but we have taken steps to eliminate that kind of structuring.

[illegible] organizations have been changed, eliminating some of the layers in sub-Program [illegible] were present (for example, putting the Manufacturing Departments with their [illegible] Managers).

[illegible] better management approach, I would prefer to regard Matrix Management in a [illegible] advantages are many. First, it is the least costly form of organization.

[illegible] Manager to devote his time to the complex issues of the program and to [illegible] functions, tasks and priorities. He's not constantly distracted by

It is probably the most efficient and productive organization form.

The functional representatives such as Technical Logistics Data, Supply and Support, or Logistics Engineering, also gain experience working in this type of environment.

They gain expertise from one program to another. Lessons learned can be applied to another program, either by assignment or through the representatives functional department home.

2.2 Typical ILS Program Manager's Team

Provision is made so that each program within the ILS Division (ILSD) becomes the prime responsibility of a distinct program management team. An ILS Manager, approved by the ILS Division General Manager, is designated for each program. He is given full responsibility for all operational and financial aspects of the ILS program. The ILS Manager recruits program team members from the ILS Division product areas and management service groups. Team personnel receive program direction from the ILS Manager. They have clear-cut authority, delegated by their functional management, to draw upon all resources of their functional organizations to meet program requirements. (Figure 10).

It involves the sharing of critical skills. Sometimes there is only one individual in an organization who has a unique and required skill. This way, that individual is not tied to one project. His or her expertise can be spread around as needed. Or, at other times there may not be enough work to justify assigning a person fulltime.

Another consideration is that Matrix Program organization is an attractive arrangement for highly skilled professional people who want to work on various new and challenging programs. These are the individuals who come from the functional departments of the product line who assist the Program Manager in the Matrix Program organization.

Staffing problems are minimized. It's easier to accommodate changes in program manpower requirements. The program starts off with a small number of people, and as it grows it calls on the people it needs. Then, as the program declines, it frees them again.

In the Matrix Management approach, I believe the entire management team works toward successfully achieving program objectives with strong feelings of responsibility, interest, concern and pride.

The Program Manager must be able to prevent and deal with excessive overhead, decision strangulation, and uncontrolled layering (Matrixes which lie within Matrixes which lie within matrixes).

Some critics have pointed out that in a fast-moving and fast-changing program, the Matrix organizational approach may not be able to achieve reaction times that are fast enough to meet program requirements. I do not dispute this.

But the two other frequently mentioned shortcomings of the system are: First, that communication is more difficult in the Matrix system: and second, bias of functional heads may subtly work against the priorities desired by general management.

In response to these areas, I believe matrixing puts a heavy demand on planning and communication. Good planning can minimize the shortcomings.

Planning is, in fact, an excellent way to communicate. It's a great management tool. If you don't own the resources, then planning is the only way to communicate the need for those resources to the managers of those departments that do own them.

This way there is no unilateral determination of the schedule. It requires participation.

2.3 The Integrated Logistic Support Plan

The ILS Manager uses a task-oriented system of program planning and control to administer his program. The basic system is adapted, when required, to meet sensibly the requirements of DoD Instruction 7000.2 (Performance measurement for Selected Acquisition), MIL-STD-499 (System Engineering Management), and MIL-STD-881 (Work Breakdown Structure for Defense Material Items). Because the system provides the elements for effective program control and visibility, the ILS Manager can enforce his business decisions and keep both his customer and ILSD Management informed of program status. The term customer is used here to mean any agency, either external or internal to Westinghouse, which orders services or materials directly from the ILS Division.

The ILS Manager uses an Integrated Logistic Support Plan (ILSP) as a primary management tool. A separate ILSP is developed for each program and is based on the results of a Logistic Support Analysis (LSA). The LSA is a systematic, comprehensive analysis, conducted on an iterative basis throughout the acquisition cycle. It is the single analytical logistic effort within the system engineering process which identifies, defines, analyzes, and quantifies the logistic support requirements.

Initially, the LSA develops qualitative and quantitative logistic support objectives. As the program progresses, these objectives are refined into system/equipment design parameters for use in design/cost/operational availability/capability trade-offs, risk analysis and development of logistic support capabilities. The initial LSA effort evaluates effects of alternative hardware designs on support costs and operational readiness. Known scarcities, constraints, or logistic risks are identified, and methods for overcoming or minimizing these problems are developed.

During system design, the LSA is oriented toward assisting the designer in incorporating logistic requirements into hardware design. The goal is to create an optimum system/equipment that meets the specification and is most cost effective over its planned life cycle. Logistic models, such as the USAF Optimum Repair Level Analysis (ORLA), may be used (if appropriate) to predict and evaluate complex support requirements. These models are compatible with, and do not duplicate, other system engineering models.

The ILSP is formulated, based on the results of the LSA. It provides a comprehensive description of support and test equipment, facilities requirements, personnel required by skill, type, and number, spares and repair parts, and quantification of maintenance and operational support needs. Because the LSA is iterative in nature, the ILSP is a dynamic document which continually grows with the availability of information, and provides for integration of logistic elements into program planning, development, test and evaluation, production, and operational processes.

The preferred method for conducting the LSA is defined in MIL-STD-1388. The U.S. Army has implemented MIL-STD-1388 through the medium of DARCOMP 750-16, and has developed a COBOL computer program to manipulate the mass of data. This program is in use by all of the services. The LSA process normally depends on inputs from Design Engineering, Reliability Engineering, Maintainability Engineering, Human Factors Engineering, and System Safety Engineering, which all participate in the system engineering process to evolve the operational elements of the system.

The support elements, however, are defined under the leadership and cognizance of the ILS organization. It is important to note that most of the essential elements of system support are predicated upon the maintenance plan and its resource requirements, so there is a vital interface between Maintainability Engineering and its products - analysis, prediction and design support, and Logistics Engineering and its products - quantitative design requirements for deliverable support equipment, trained personnel, T.O.s/Manuals, spares and repair parts, and other support resources.

2.4 Some Problem Areas

The Matrix organization because it cuts across so many lines does cause at times conflict. But in regarding the conflict that occurs, I would like to make a vital observation. Conflict is healthy if properly managed. (I hope someone comes up with a word other than conflict which better describes this process).

If there were no conflict, none of your big problems would surface in time to do anything about them. There's no solving of sticky problems without sitting down and openly discussing them.

I feel this is healthy if these discussions can take place in an atmosphere of mutual respect.

One of the main challenges facing the Matrix Management programs is finding the right people and placing them in the proper jobs.

The Program Manager has been called a coordinator, a teacher, delegator, leader, father confessor, and perhaps a few other names we can't mention.

The requirements for the job boil down to three basics:

1. Education -- Engineering or Technical Education, Finance or Business.
2. Experience -- A broad array of experiences: Administrative, Design, contact with legal, accounting and sales, shop operations, and ability in public relations and public speaking.
3. Training -- Must include actual OJT with experienced Program Managers who can provide the kinds of background and guidance to introduce the trainee to the various facets of work involved in a project. And, formal training.

There just aren't any people coming out of our school system with all the talents and training needed, this is why we continue our excellent in-house training programs, the ILS Program Management seminars, and the career management of our personnel.

The need for logistics courses at the undergraduate and graduate levels in colleges and universities is evident. Beyond college, we are supporting the pursuit of the CPL (Certified Professional Logistician) which is given by the Society of Logistics Engineers.

3. CONCLUSION

In summary, I would like to say, the people we look for must have a unique blend of the basic logistics disciplines and business knowledge. Everything is based upon experience, training, and the desire and ability of the Program Manager to make a program successful.

The Matrix Management approach assists the Program Manager in running such a successful program. Industries, businesses, government agencies, institutions and individuals alike have been dealing with various facets of Matrix Management, logistics support, life cycle costing for years. This approach ties integrated logistics support into a managable system and adds another dimension to Matrix Management.

Examples are shown in Figure 11 ILS Innovations.

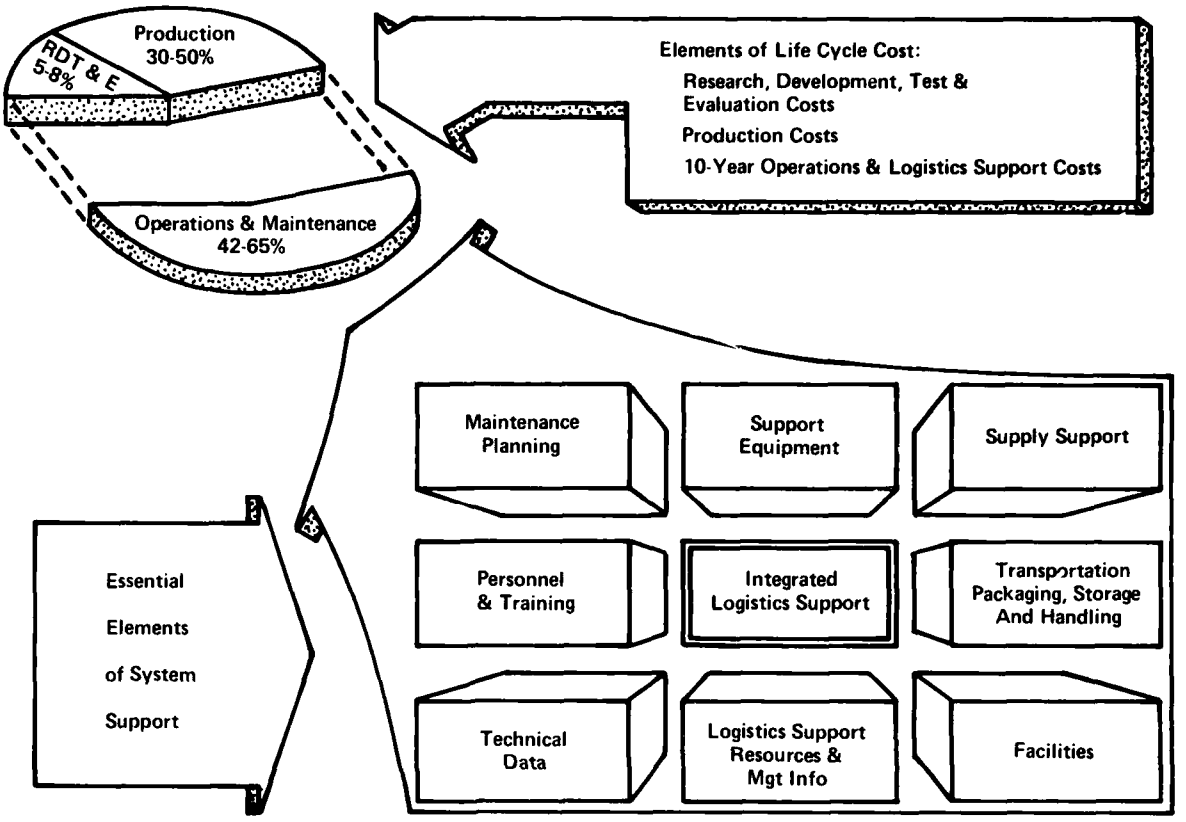


FIGURE 1 - Integrated Logistics Support

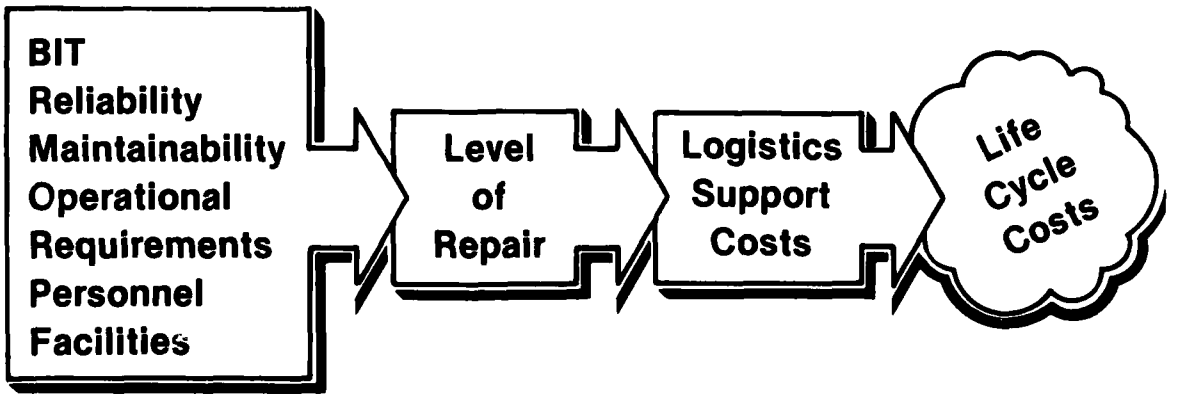


FIGURE 2 - LSA Procedure

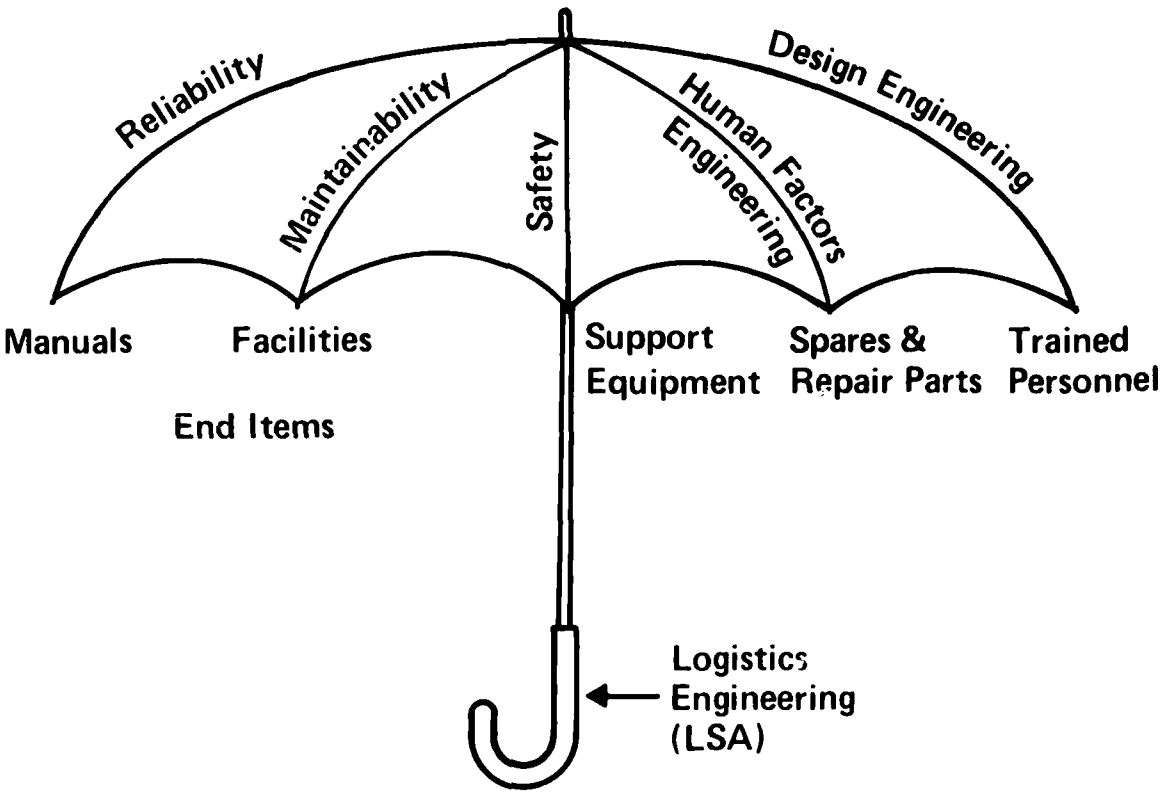


FIGURE 3 - The Systems Engineering Umbrella

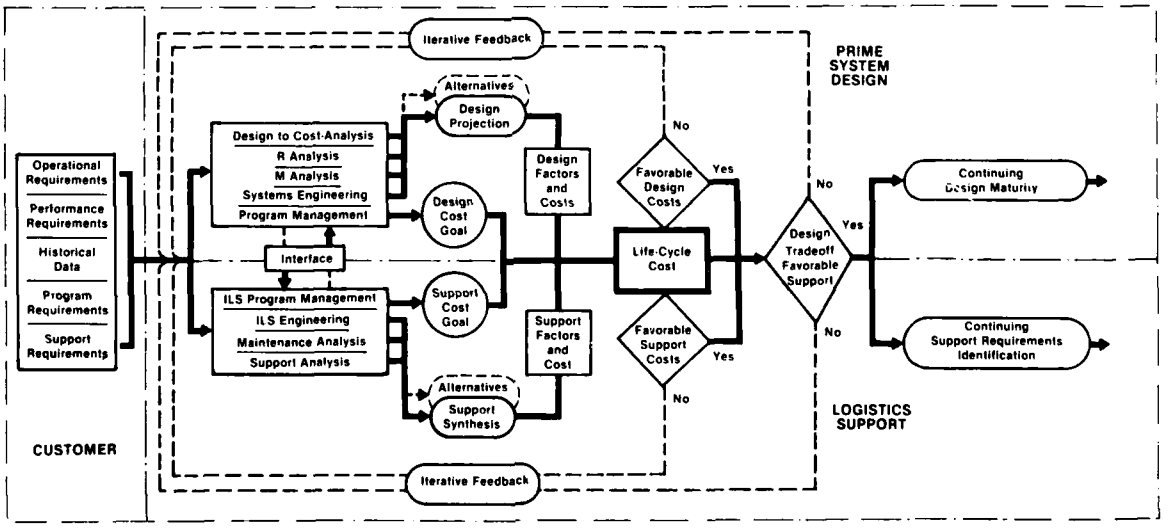


FIGURE 4 - Life Cycle Process

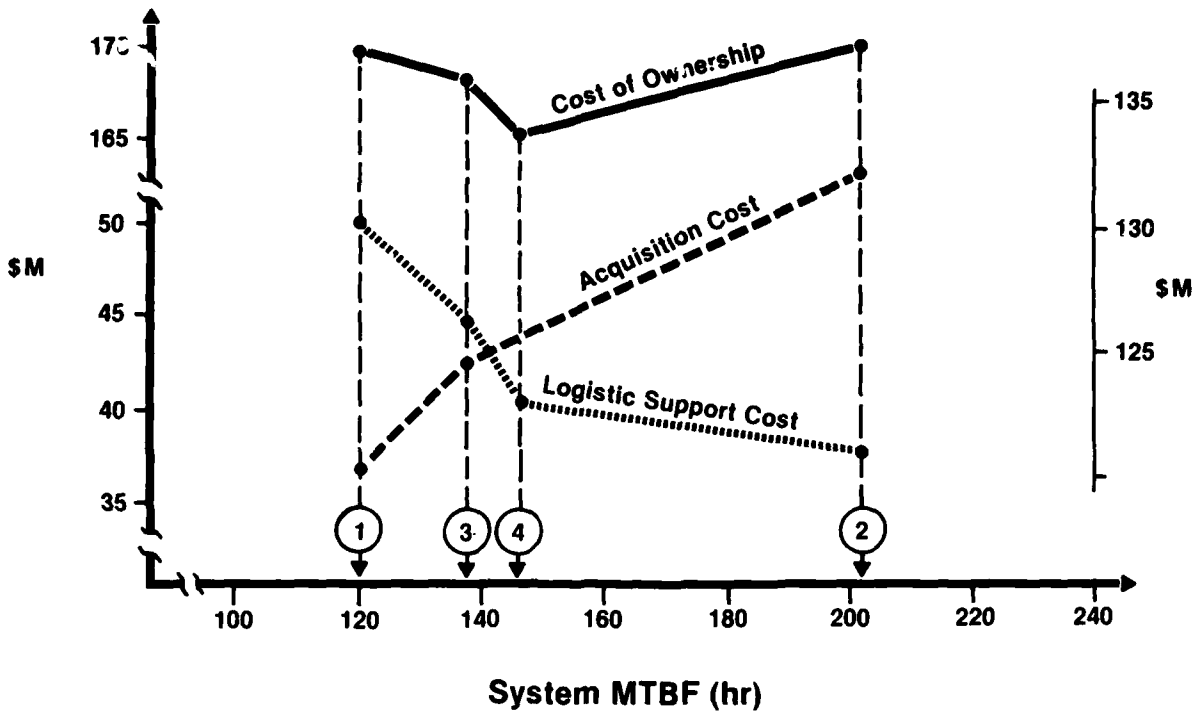


FIGURE 5 - Reliability Cost Trade-off

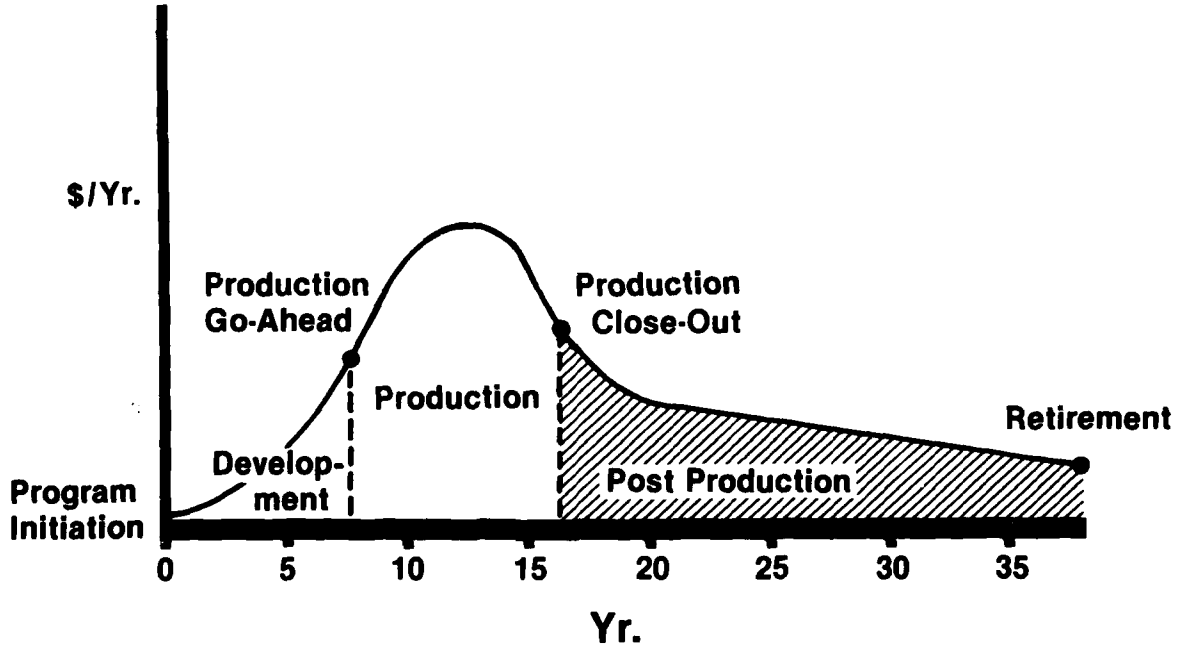


FIGURE 6 - Typical Program Expenditure Projection

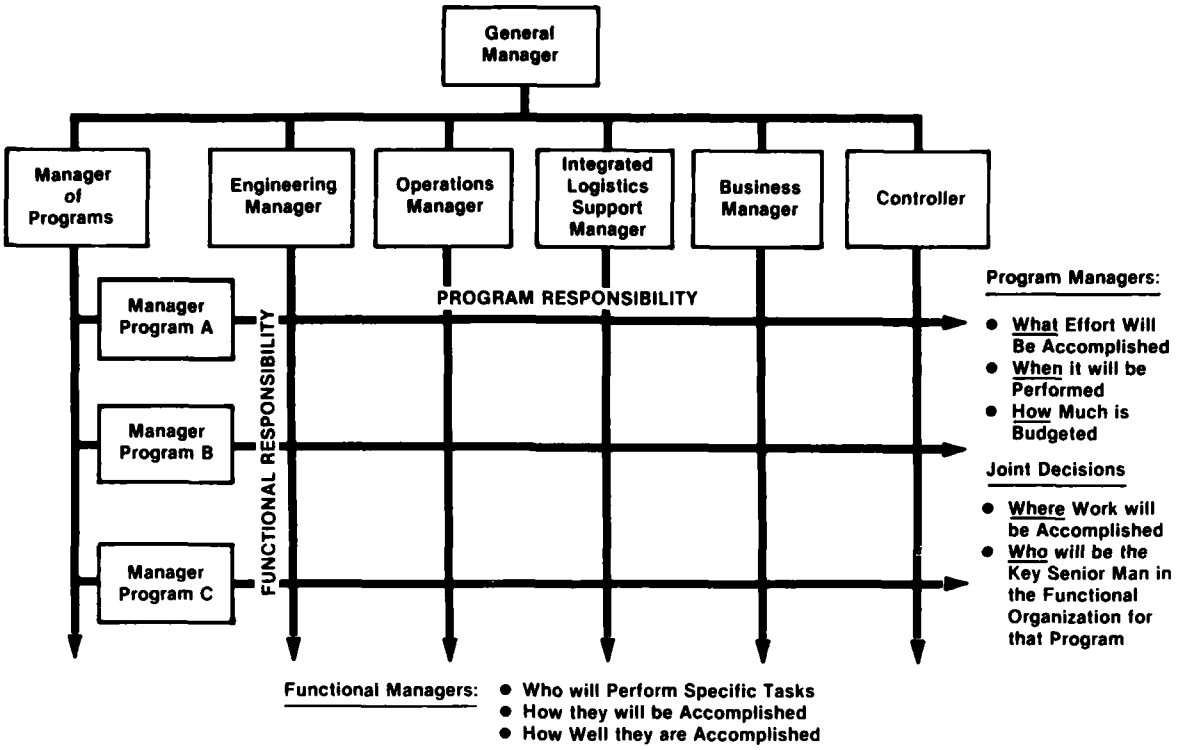


FIGURE 7 - Matrix Organization Relationship of Program Management to Functional Management

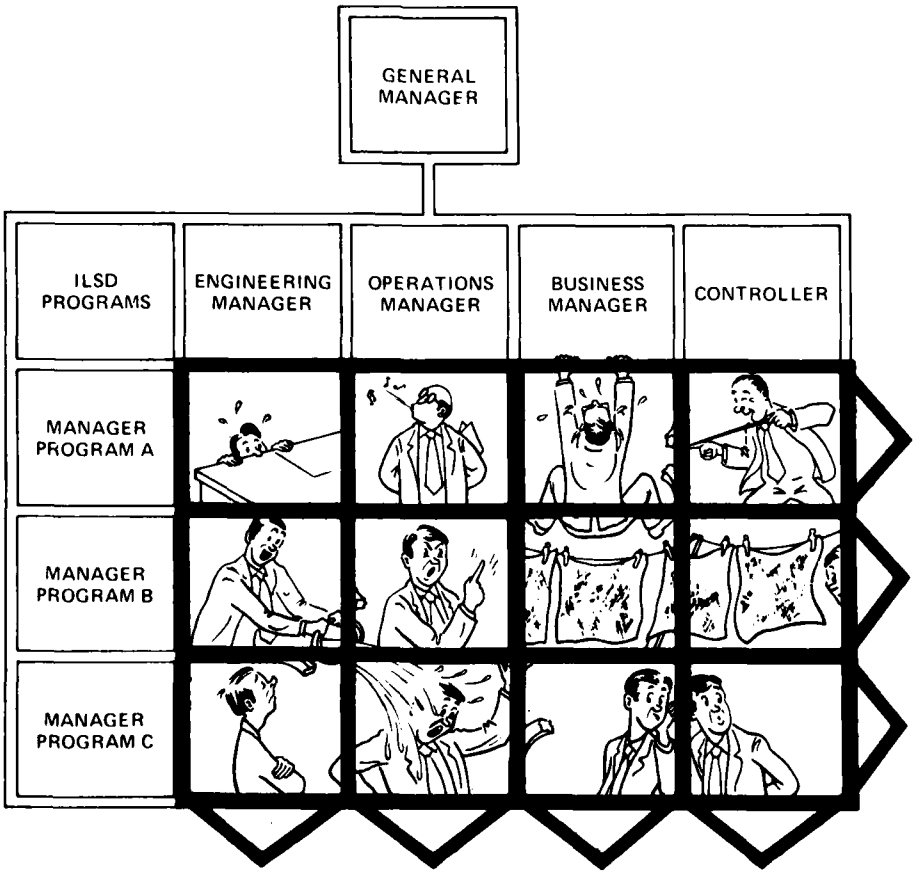


FIGURE 8 - Incorrect Matrix Organization

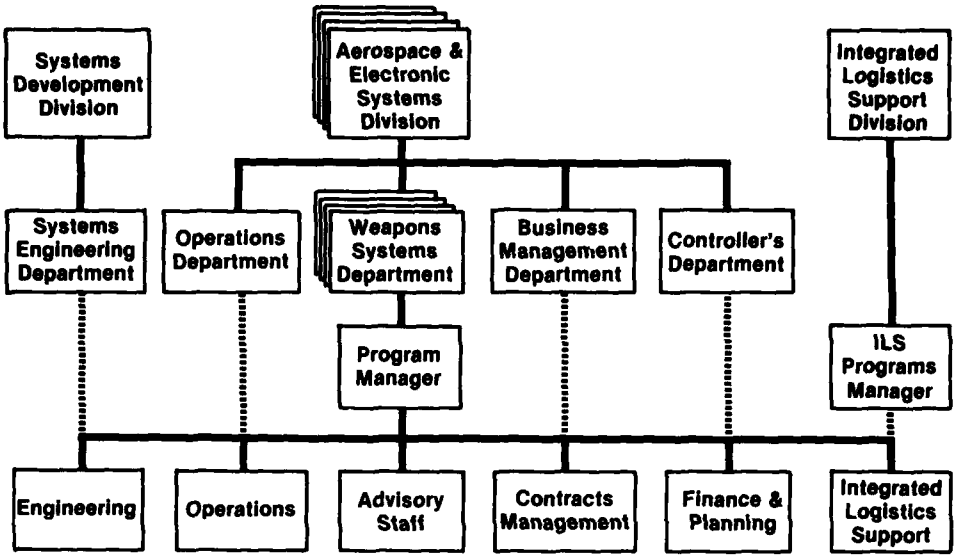


FIGURE 9 - Typical Program Management Organization

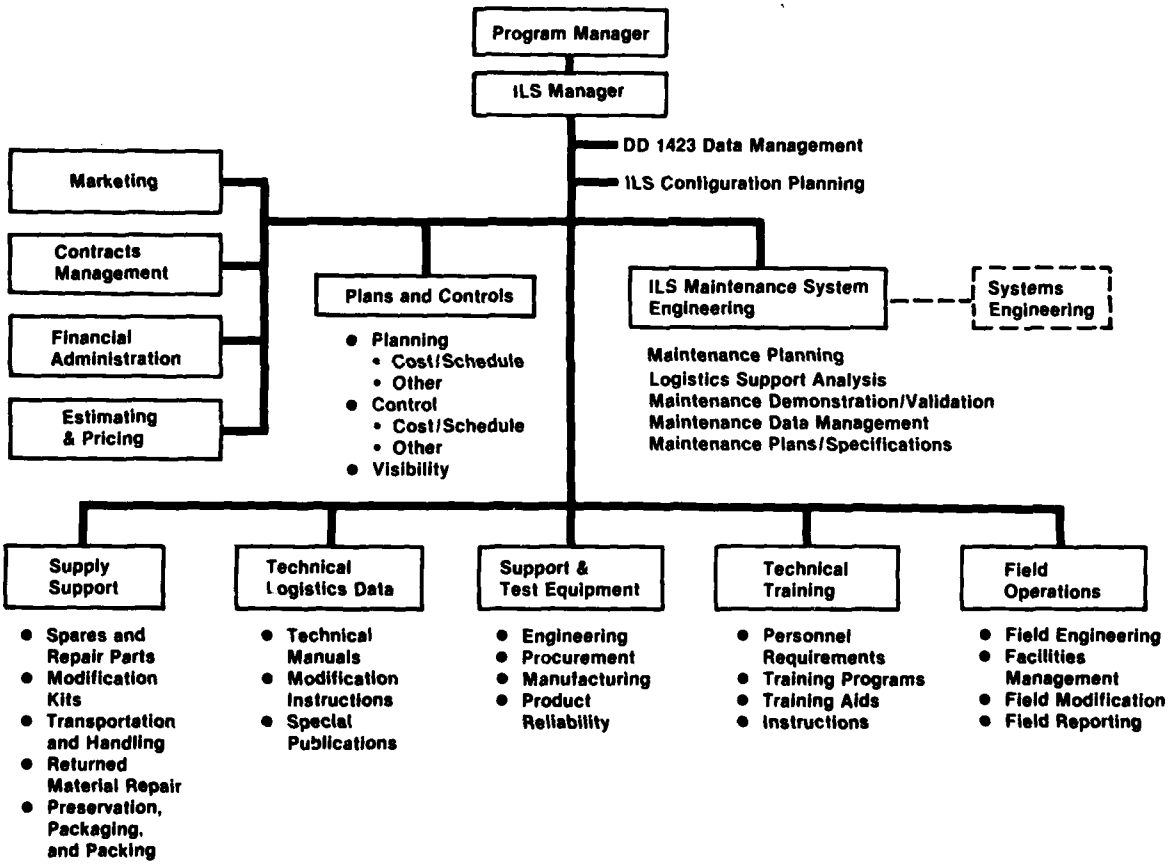


FIGURE 10 - Typical ILS Team for Large Program

LOGISTICS PLANNING

LCC Optimization Programs
Extended Warranty Programs
Availability Analysis Techniques
Wire Data Transmission
Out-of-Production Support Programs

TECHNICAL DATA

Job Performance Aids (JPA)
Mechanized Data Production
Tone/Line Conversion
Graphics Production
Task Analysis Techniques
Automated Readability

TRAINING

Training Requirements Analysis
Screening/Testing Services
Audio/Visual Techniques
Productivity Improvement Techniques
Interactive Training Systems
Skills Training Institutes

SUPPLY SUPPORT

Mechanized Provisioning
Spares Optimization Programs
Supply Continuity Programs
Spares Integrity Processes
Direct Expedite Services

TEST EQUIPMENT

Test Analysis Techniques
Remote/Contact Less Test
Built-In Test
Electronics Functionilization
Human Factors Studies

FIELD SERVICES

Maintenance Audit Systems
Direct Support Systems
Maintenance Data Feedback Systems
Electronics Depot Systems

FIGURE 11 - ILS Innovations

"MEK" - A NEW PROCEDURE FOR DEVELOPMENT OF MAINTENANCE POLICIES

Klaus Lewandowski
Product Support Manager
Messerschmitt-Bölkow-Blohm GmbH
Postfach 80 11 60
8000 München 80

SUMMARY

With the following presentation a survey of a procedure for the development of maintenance policies will be given. The procedure has been developed originally for the Tornado-programme but shall be standardized in the future for general application in German-Air-force weapon system projects.

The procedure is based on a detailed collection and evaluation of the maintenance expenditure expected for the new weapon system. The step-by-step evaluation process includes participation of the military authorities at an early point in time. Because of the relatively high amount of data and information to be processed, the procedure uses EDP. As the result of the procedure validated and detailed information are provided to all logistic disciplines.

Gentlemen!

It is well known to each logistic engineer that the coordination of the logistic activities required to make military equipment/weapon systems operable and serviceable is one of the most important logistic problems. Coordination means to ensure that the provision of

- technical publications
- test and ground equipment
- test programmes and procedures
- spares management data
- spare parts for all maintenance levels
- training and training aids
- facility requirements
- personnel requirements
- repair and overhaul resources

and other support services are well harmonized in view of their interdependences and means to insure that they are available in time. Obviously this type of harmonization and coordination requires a common basis; a basis which provides sufficient information to initiate and to perform the tasks.

Comprehensive regulations and procedures have been issued by the military authorities around the world on how to perform the logistic work, whereby often relatively independent specifications and procedures are given for the different disciplines, whereby the required coordination shall be achieved and secured via sufficient programme management systems and procedures. "ILS" - Integrated Logistic Support - is the magic formula for this, whereby "ILS" stands for a strictly cooperation requirement of the different disciplines within the companies, continuously monitored and supervised and also sometimes assisted by the military authorities.

This proceeding leads generally to the fact, that the majority of the individual decissions required for the daily work on the logistic subjects are based more on the individual knowledge and experience of both - the industrial and the military specialists - rather then on harmonized, reproducible and proved technical data sources.

Admittedly, there are - at least within "American orientated" programmes - such type of data sources, as the already known results of the maintainability analysis in form of the "Maintenance Engineering Analysis Sheets" -MEAS-. These data sheets, distributed to all peoples concerned within the logistic work, provide information on the maintenance expenditure expected and give a glimpse into the complexity of the equipments and systems. But the MEA has a considerable disadvantage: In cases of multi national programmes, such as the Tornado programme for instance, where the weapon system is developed and produced for different user forces with different logistic systems, the MEA must be related to abstracted, standardized definitions. The manhour expenditures can be given only as "spanner-in-hand-times", which neglect all influences caused by the customers itself, e.g. additional manhour/manpower-requirements for preparational work, way- and waiting times, administrative work etc. Also the trade- and skill level requirements can only be given in a standardized format because nearly every nation has other personnel structures, and the same applies for quality assurance requirements, working areas etc. The MEA results therefore are primarily used to give the customer confidence that the maintainability requirements can be met and not as the basic information source for the whole logistic work.

To overcome this, we have had in Germany lengthly discussions in 1973-74 between industry and military authorities on how to develop a solid and sufficient data source for the imminent logistic decisions - first for the Tornado programme but possibly later also for all further weapon system projects. The result has been an EDP - based procedure which has proven its effectiveness, which I'll now present by the following:

The basic idea, the philosophy, of the procedure is, as shown with fig. 1, to investigate and to estimate the maintenance expenditure expected for the whole system. This maintenance expenditure is caused and influenced by:

- the system technology; the number and complexity of sub-systems, equipments and components;
- the reliability and maintainability attributes including the given life-time limitations;
- the operational requirements and logistic concepts.

The best possible investigation of the expected maintenance expenditure is of crucial importance if you remember that all logistic support is provided only to maintain the weapon systems operability, assuming that "maintenance" includes operational support.

In contrast to the maintenance expenditure the maintenance capacity has to be seen, whereby the overall capacity is composed by military (depot/level) and industrial capacities. Harmonization of expenditures and capacities, by strict observation of operational requirements and given logistic concepts, then leads to the overall maintenance policy of the weapon system. The maintenance policy must provide information on:

- the manhour requirements expected for all maintenance branches and levels to allow manpower calculations and establishment of maintenance organization;
- the qualitative and quantitative training requirements (trades and skill-levels, and depth of maintenance for each skill-level) to allow definition of required training programmes;
- the spare part requirements, at least for the reparable items, to provide realistic quantity forecasts for the initial provisioning process;
- the test and ground equipment requirements for all branches and maintenance levels, together with the related utilization rates to allow preparation of inventory lists for each maintenance organization;
- the technical publication requirements to initiate the preparation process and to support the validation/verification tasks;
- the equipment and structural component repair and overhaul requirements to establish the required R + O programmes including the industrial R + O support contracts;
- the scheduled maintenance requirements which after grouping and harmonization allow the preparation of inspection manuals and work card decks;
- the infrastructure requirements based on the knowledge of the maintenance tasks of each branch and working area.

To investigate the overall expenditure as precisely as possible, the expenditure for each individual or independent component of the system must be estimated separately. "Expenditure" means all possible maintenance tasks and "independent components" means components which require any type of maintenance and which can be maintained (repaired-inspected-exchanged) separately. Naturally, the term "component" has to be seen on different levels, for example:

- Equipments (black boxes, line replaceable units, etc.)
- Modules (plug-in units, cards)
- Accessories (tubes, connectors, cables)

The breakdown of the entire system into independent maintainable components is therefore the first step in our procedure. (fig. 2) For this breakdown the known system codifications according to MIL-M-38769, MIL-STD-780D or GAF-TO 00-5-23783 are suitable. Nevertheless, we have found that the usual five-digit code is not sufficient, seven digits are the minimum for a satisfactory breakdown.

The next main step in our procedure then is the "Maintenance Data Collection". All maintenance tasks for each individual component are collected on a EDP-input data sheet, whereby all available data sources are used. Besides the known data sources, such as MEA-results, vendor brochures, DMEA-results and others, we use a special "Module Analysis" to investigate the specific attributes of electronic equipment modules. The following data and information are listed:

1. Item Identification Informations:

System Code (WUC)	Quantity per System
Specification Number	Testability
Nomenclature	Aircraft Panel No.
Part Number	Location (Zone)
NATO Stock Number	Record Card Requirement
Manufacturer Code	

2. Maintenance Informations:

Task Description	Repair Level Code
Arising Code	AGE
Malfunction Code	Special Facilities
Cause of Malfunction Code	AGE Code
Working Class	AGE Using Time
Maintenance Level	Location Code
Frequency	
Down Time	
Skill Level	
Trade Code	
Number of Personnel	
Manhours	
Spare Parts	
Spare Parts Quantity	

The "Module Analysis" in addition provide information on:

Module Fastening	Signal Characteristics
Parts Fastening	Test/Diagnosis Dummy Rqmt
Technology	Cover Types
Function	Screen Types
Frequency Range	Fillings
Connections	Mean Component Exchange Time
Test Connections	Mean Adjustment Time
Estimated Unit Price	Mean Test Time

Naturally, this data collection process appears very extensive, but remember the before mentioned data sources provide most of the required information. Therefore the workload is more on the administrative data collection than on investigation. Care has to be taken only on the specific military data, e.g. maintenance level, trade, skill level and work locations. Our experience has shown that it is very useful to task maintenance specialists with the data collection who have formerly been involved in military maintenance activities. The results are then very realistic.

After data collection and EDP storage, the next step in our procedure is the "Validation Process". For this working lists, containing all stored assorted information in one format, are printed out and distributed to the military authorities responsible for logistic decisions. The leading authority then calls a "Technical Evaluation and Assessment Meeting" (TEAM). Participants on these meetings are

- aircraft manufacturer
- equipment supplier (if required)
- customer officials including procurement officials
- NDQAA

The meeting, chaired by the leading military authority, reviews and discusses mainly the following informations, whereby at the beginning of the meeting a presentation of the item under discussion is provided:

a) Task Frequencies (Arising Rates)
The task frequencies are derived during the data collection process mostly from the specified values of the equipment or system specifications. Experience has shown that there is often a considerable difference between specified values and those proved in-service. The reason for this is very complex and shall not be discussed here. For a realistic maintenance expenditure forecast, however, a realistic frequency forecast is of the greatest importance. The TEAM therefore discusses the probability of the given frequencies and com-

pares them, if possible, with experience gained from similar equipment/components already in-service. If there is a remarkable difference without acceptable reasons, a new frequency will be established. This is a pragmatic attempt, however it must be noted that these established frequencies have in no way any influence on specified reliability figures which are contractually binding. Figure 3 shows two examples.

Equipment	Specified Defect Rate	Experience		TEAM Assumption	Remarks
		F-104	F-4		
Magnetic SBY Compass	0.03	4.12	2.79	0.15	Reliability Improvement credibility
Transmitting Rate Gyro	0.48	3.21	0.35	0.66	Equipment more complicated than the F4-equipment

Fig. 3

Special attention will be given to the frequencies for scheduled maintenance tasks, e.g. inspections, lubrications, life time limitations, etc.

b) Scheduled Maintenance Requirements
As already known, all modern maintenance concepts try to avoid scheduled maintenance. Because of this, for each scheduled requirement a detailed engineering justification has to be provided, which needs acceptance by the customer. Several special procedures are known to roentgenrize scheduled maintenance requirements. The most common one is the civil airline industry procedure, called "MSG" or "EMSG" ("Maintenance System Guide respectively" European...). The questionnaire schemes of these procedures can be applied, either prior to or during the TEAM, discussed and the required decision made. For assistance on structure component, the results of a "SSI" ("Structural Significant Item") Analysis must be available.

The TEAM also has to discuss and decide on the required harmonization phasing of scheduled maintenance intervals.

The next item under discussion concernes the proposed maintenance levels for the individual tasks:

c) Maintenance Levels:
The proposed maintenance levels are discussed and validated by the responsible authorities. These decisions can possibly be later on corrected if the "Optimum Repair Level Analysis" (ORLA) of the Analysis Phase requires this, but for the majority of the maintenance tasks the final level can be determined.

The next following items are then the decisions connected with personnel-requirements:

d) Personnel Requirements:
Personnel requirements are influenced by the trade - and skill level definitions for each maintenance task and by the commitment of the individual work areas for the tasks (e.g. flight line, shelter, work shop, etc.) and - last but not least - by the quality assurance requirements. Of greatest importance for the manpower - calculations to be performed later on is the precise forecast of the manhour-requirements. Therefore, for each maintenance task, the additional time requirements for preparational work, administrative work, etc. are estimated and added to the data set. Fig. 4 shows the table with the extra charges for additional time requirements used in our programme:

Maintenance Task	Location		
	extra charges for maintenance tasks to be performed in the flight-line/ shelter area	extra charges for maintenance tasks to be performed in the maintenance hangar and in work shops	extra charges for maintenance tasks to be performed on other locations
	MMH	MMH	MMH
Re-Fuelling	0,2	-	0,3
De-Fuelling	0,4	-	0,4
A/C towing	0,5	0,5*	0,5
AGE provision	0,5	0,2*	0,3
A/C securing (safety precautions)	0,5	-	0,5
obtaining spare parts	0,8	0,5	0,5

cont. ./.

cont. ./.

Way-times for maintenance personnel and QA-inspectors	0,4	0,2*	0,2
preparation of forms	0,2	0,2	0,2

Fig. 4

(*no extra charges for work to be performed in work shops)

e) Technical Publication Requirement:
Another point of discussion for the TEAM is the definition of the technical publication requirements for the related item. The detailed knowledge of the maintenance requirements for the item allows us to decide on the type and depth of manuals and catalogues required. This decision is also added to the data set.

The 4th major step in our procedure now is the evaluation and analysis part:
At the beginning the scheduled and unscheduled maintenance tasks are separated. The scheduled maintenance tasks are sorted according to their periodicities (frequencies), grouped into work packages and then the position of each individual task within the package marked. This allows the preparation of work-flow charts and draft work card decks. These drafts are presented to a special TEAM for final acceptance.

The unscheduled maintenance requirements are investigated in view of candidates for an "Optimum Repair Level Analysis" (ORLA) which then can be performed separately. Computer models for those evaluations are already known and available. Results are also discussed at the special TEAM, and possibly the data store corrected. Additional evaluations are performed for the workload of the individual branches and working areas, what can also lead to possible corrections of the data store.

After this evaluation and analysis step, or in parallel to it, a cross-reference is made to the initial provisioning data store. During the whole procedure, the parallel running IP-process is supported by provision of information for quantity and maintenance level requirements. As a feedback, the IP-data bank delivers to the maintenance data bank the "figure-line-index" numbers for all registrated spare parts, and also the NATO stock numbers for the reparable items itself. This proceeding facilitates all further work within the logistic disciplines.

The last step in our procedure now is the summarization and presentation of the results. For higher management level, a summarized report is prepared, which is used to publish the guidelines for the further logistic activities. For the working level, detailed lists are provided, for example:

- Manhour and related manpower requirements for each branch on wing- and depot level, together with the trades and skill levels
- Spare part requirements for each maintenance level and working area
- IPC-cross reference list
- AGE-requirements including utilization rates for each branch at wing and depot level
- Technical Publication requirements for all maintenance levels
- Maintenance expenditure information for preparation of the maintenance manuals
- Repair and Overhaul concepts for each individual component

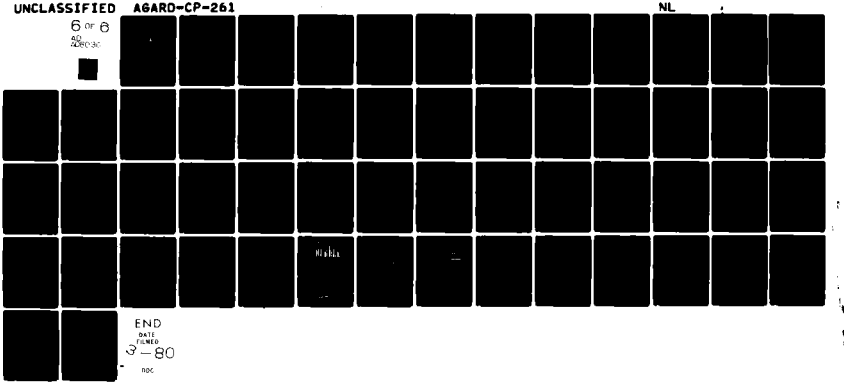
In addition lists for special purposes, e.g. spare part lists assorted to price classes or equipment test requirements, are provided. Very important also is the provision of the facility requirements derived from the maintenance requirements of each individual branch and location.
The data bank itself can either be updated and used for in service repair and overhaul management, and other maintenance management purposes, or filed and used for comparisons in future development programmes.

Fig. 5 gives a summary of the amount of data processed and stored for a modern weapon system. At this point in time, activities are initiated by the military authorities to standardize the procedure to make it applicable for all future weapon system projects. The related GAF-TO guide line will possibly be made available by the end of this year.

AD-A080 301 ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT--ETC F/G 9/5
AVIONICS RELIABILITY, ITS TECHNIQUES AND RELATED DISCIPLINES.(U)
OCT 79 M C JACOBSEN

UNCLASSIFIED AGARD-CP-261 NL

6 OF 6
AD
A080301



END
DATE FILMED
3-80
DDC

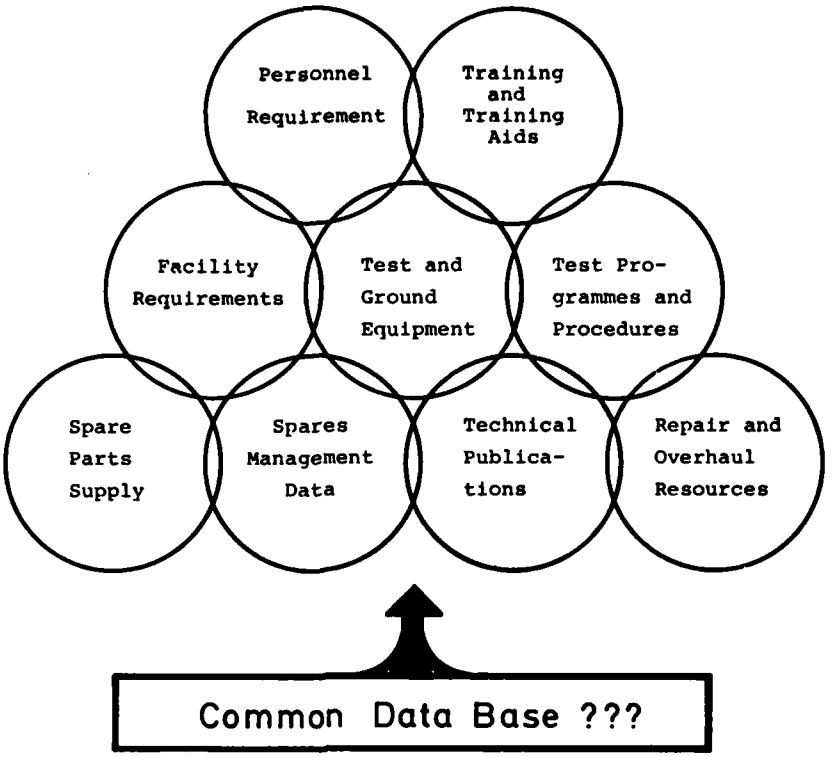


Fig.1 Co-ordination of logistic activities

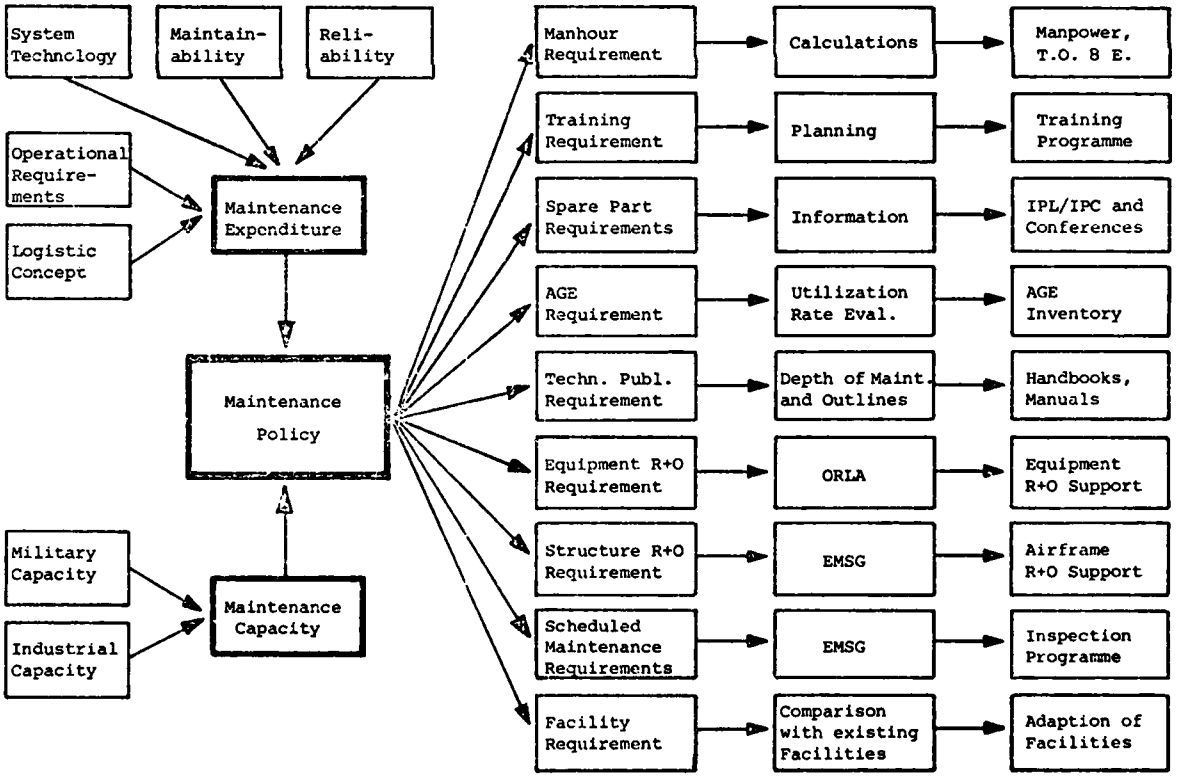


Fig.2 "MEK"-procedure philosophy

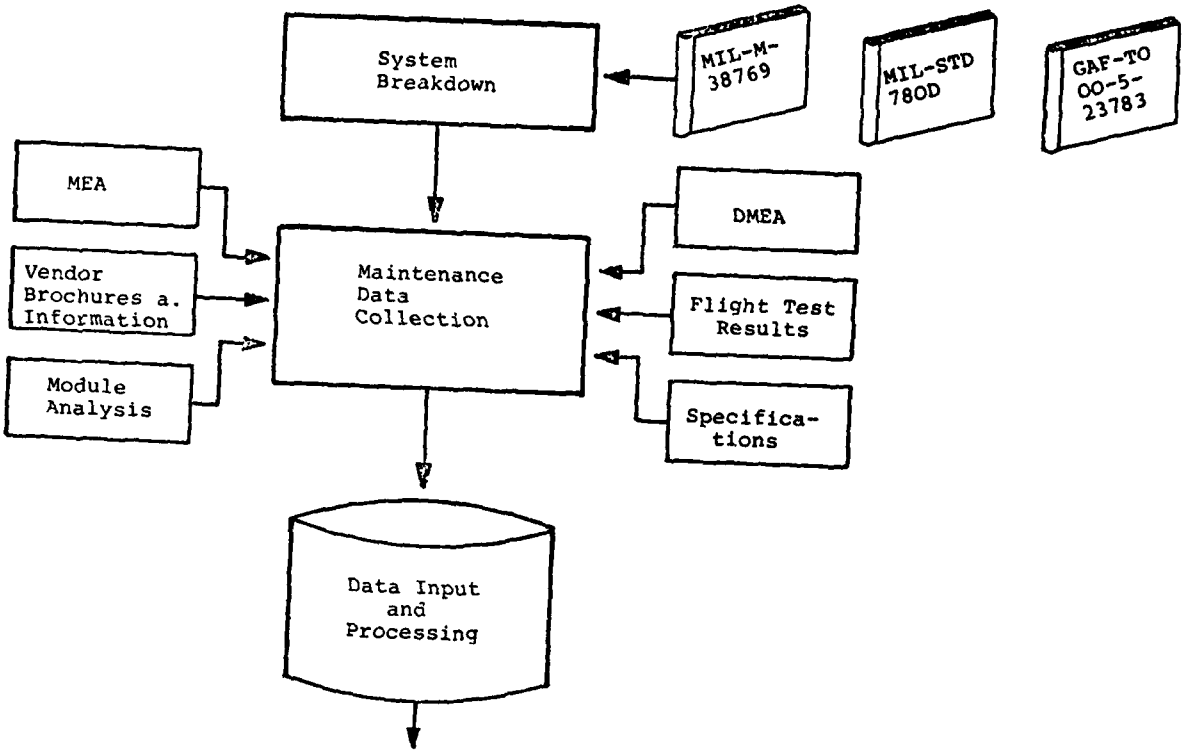


Fig.3 "MEK"-procedure, step 1 and 2

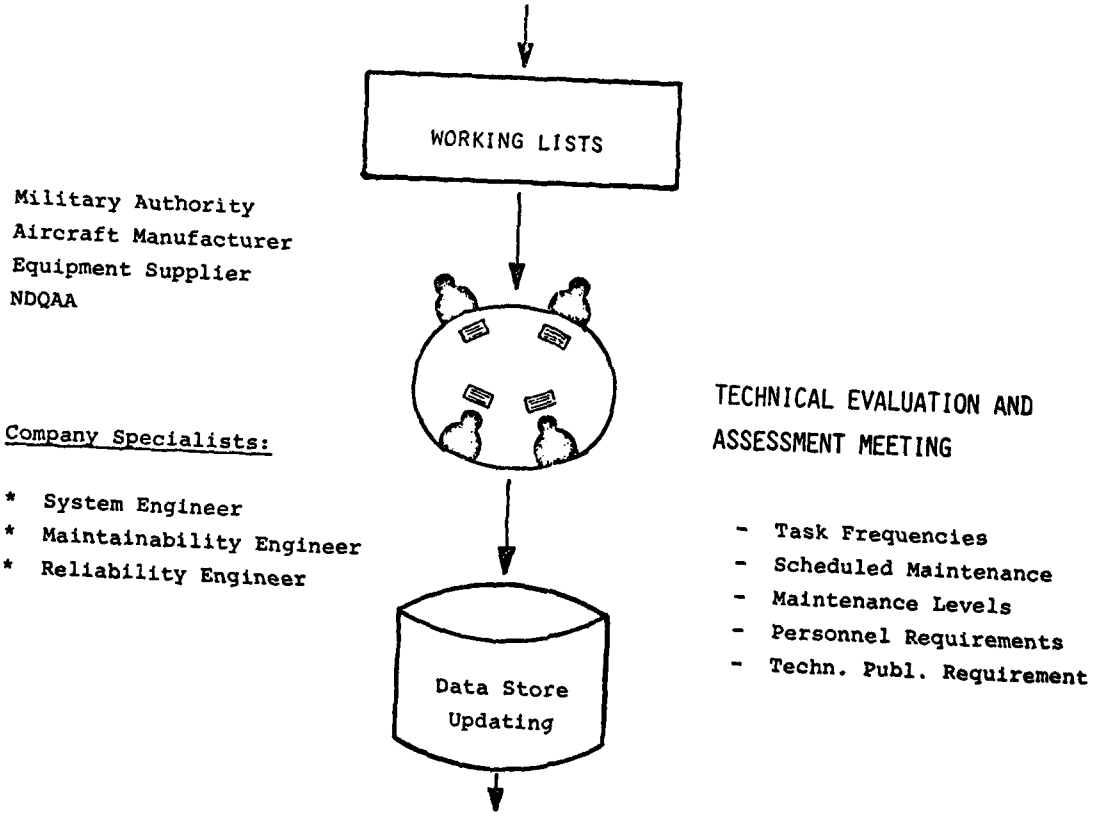


Fig.4 "MEK"-procedure, step 3

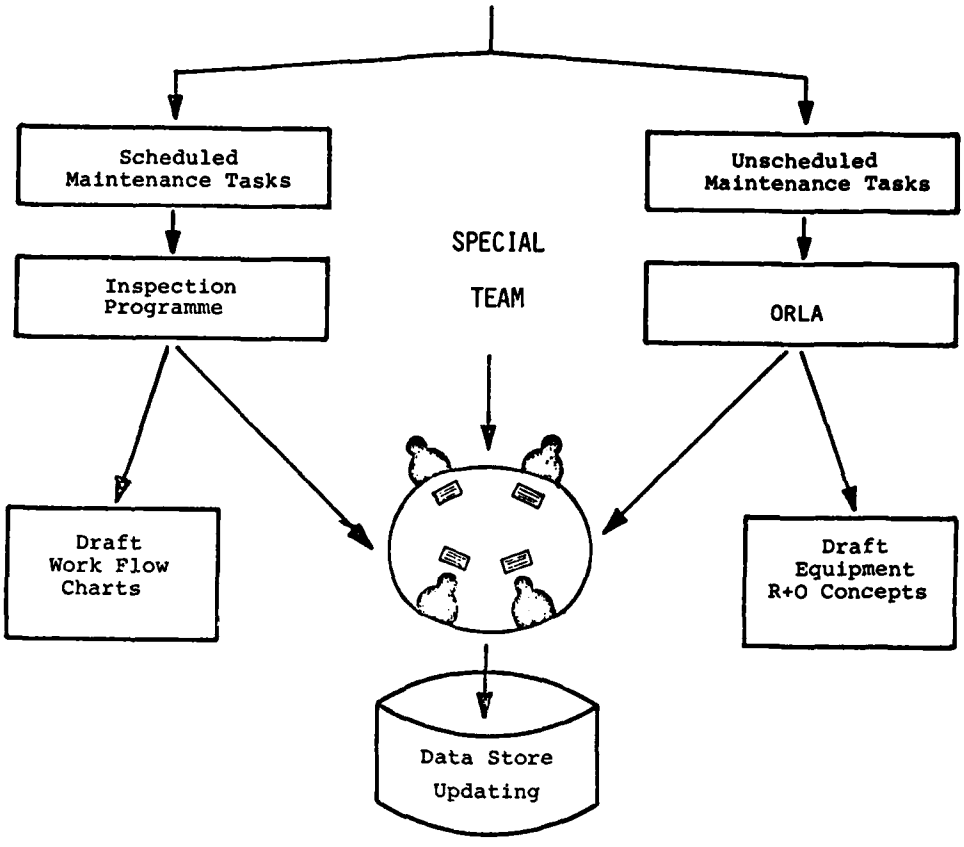


Fig.5 "MEK"-procedure, step 4

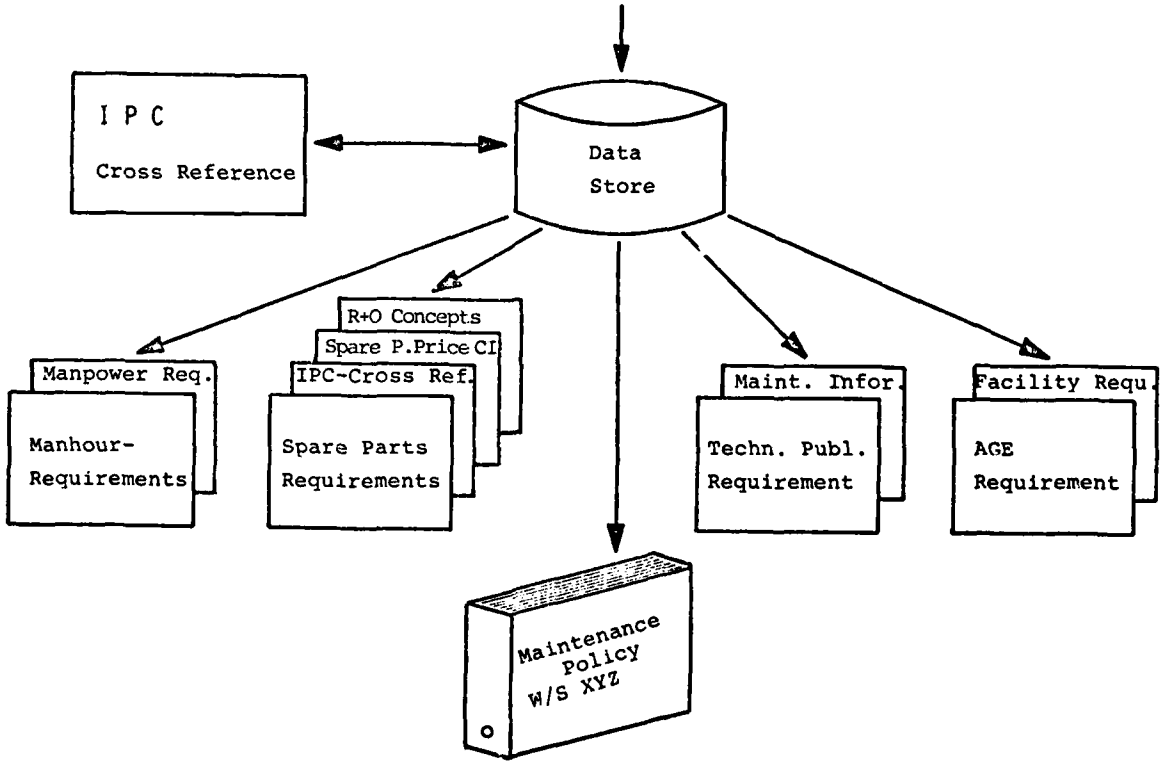


Fig.6 "MEK"-procedure, step 5

Number of Components	3500
Number of Maintenance Tasks	15000
Number of TEAM's	50
Number of Modules to be analyzed	730
Number of Evaluation Lists	20

Fig.7 Summary of the Tornado-MEK-procedure

DISCUSSION

J.N.Basmaison, Fr

Quelle est la fréquence de remise à jour des données "taux de remplacement prévus" pour ajuster les stocks de rechanges prévus.

Réponse d'auteur

Nous avons commencé le recueil des données il y a 4 ans et nous avons mis à jour nos bases de prévision de rechanges depuis, deux fois.

A.Andrews, UK

You have pointed out the need for realistic assumptions on reliability which are not unduly influenced by the reliability targets.

This statement appears to be contradicted by the examples shown e.g. Magnetic SBY Compass, F4 experience 2.79, Team Assumption 0.15, Specified Rate 0.03.

On the face of it such a large improvement seems difficult to justify except by reference to the specification.

Author's Reply

The specified defect rate was 0.03. Comparison with similar equipment has shown that their defect rates have been much higher (4.12 and 2.79). The TEAM therefore concluded to raise the specified value 5 times to 0.15. The state of the art of the new compass seems to be better than the older ones, so that the TEAM came to the result that 0.15 would be realistic.

THE IMPORTANCE OF INTEGRATED LOGISTICS SUPPORT CONSIDERATIONS DURING DESIGN

ROBERT C. RASSA
ILS Manager

Westinghouse Electric Corporation
Integrated Logistics Support Division, Hunt Valley, MD

Abstract
Many problems arise when proper consideration is not given to the Integrated Logistics Support elements during the earliest phase of a program. The relationship and interdependency of these Integrated Logistics Support elements and their relationship to the design of the "prime mission equipment" is explored. The roles of the key personnel involved in the initial considerations are also examined. Some important rules for the successful implementation of an Integrated Logistics Support plan are presented and developed.

Integrated Logistics Support, or ILS, includes many disciplines which are familiar to most of us in the aerospace and weapons system community. These disciplines include Maintenance Systems Engineering; Test Equipment (both factory and field); Training; Technical Logistics Data; Spares; and Field Support. Without all of these disciplines, our military weaponry cannot be maintained in the field. A deficiency in just one area is sufficient to render a system unable to perform its mission.

This simple fact is well known by anyone who even has a hint of how the military system operates. What may not be quite so obvious is the interdependency of the Logistics Support disciplines, and the domino effect that a deficiency in one area has on another. For example, let us look at a hypothetical radar system whose maintenance concept calls for replacement of printed circuit boards, as opposed to "black boxes," at Organizational level. Functional radar faults are indicated by Built-In-Test.

The radar has a fault indicated in a particular functional area. There are ten p.c. boards involved in that function. What does the maintenance technician do? Well, if he is relatively new (a high probability) he'll pull out the Tech Orders for the radar and go to the section concerning the functional area where the fault lies. The Tech Order will probably outline some additional testing that can be performed, using "standard instrumentation" and/or a piece of Special Support Equipment. So the technician gets the piece of SSE out of the hangar and dollies it up to the radar, and runs a self-test as prescribed in the Tech Order.

Now he begins to hook up the test equipment. Again the Tech Order tells him how. Plug P1 into J1, P2 into J2, P3 into A1A8J16, etc. When all the hookup is complete, we begin the testing. Our technician follows the book, more or less. He has had some training, of course, and remembers something about this test set, but this is only the second time he has used it an he doesn't remember quite all the details. But things start to happen and he starts to get some readings.

Now our technician has to interpret these readings in order to make a diagnosis of which of the 10 p.c. boards is actually at fault. Here the Tech Orders really come into play, for they will lead the technician by the hand through a diagnostic routine, a fault analysis chart, as signal observations are made and additional tests are run.

Pretty soon the fault has been localized to only 5 of the orginal 10 boards. Do we replace all five and call it quits? No, because we only happen to have 3 of the 5 spares on hand. You see, last week the same fault occurred on another aircraft, and the technician couldn't get the test set working right, so he replaced all 10 boards. It fixed the fault, but depleted the stores.

So we have to continue our troubleshooting the "right" way—by the book. The book, of course, being our Tech Orders books. As our technician plugs along, he finds some reference designations that don't seem to match the hardware, but the problem is not serious and he figures out what should have been printed.

Then we find that many of the procedures are cumbersome and repetitive and take a lot of time to perform. Next our Chief Maintenance Officer is on the scene, wondering what the deuce is taking so darned long.

To make a long story (and maintenance action) short, we finally get Brand X's Tech Rep out of the hangar—where he was working on another radar, of course—and he recognizes the fault. He pulls the one board he knows to have failed and replaces it with a spare board—which fortunately was in stock. Naturally the spare board also didn't work, but in order to have a Cinderella ending, the Tech Rep just happened to know where there was a good spare, and we're back on the air.

What has this story demonstrated? Well, to start with, a maintenance concept that allows the Built-In-Test/Fault Isolation Test (BIT/FIT) to signal an ambiguity level of 10 is sorely lacking. If you are replacing cards as the prime maintenance action you require a far better degree of fault isolation. The test eqipment that was designed for this hypothetical system for beyond BIT/FIT required a lot of manual setting and interpretation. The training provided for the test equipment was really limited to the basic theory of operation and use and did not include a lot of hands-on training because a full

radar system wasn't available at the time. The Tech Orders had not yet been fully verified. And, of course, the base had not ordered enough spares, and many of what they did have would not work in the system.

Our example has drawn attention to the basic interdependency of the various Integrated Logistics Support elements. Now we'll explore each element in turn and establish some criteria.

The Systems Maintenance Concept is where it all begins, and the establishment of this Maintenance Concept is not solely the function of the contractor ILS organization. It is established by the military Project Manager working with the contractor, and it appears in the overall Statement of Work or similar document. The decisions on the various factors compirsing the Maintenance Concept are based on many things, including contractor recommendations, performance, deployment plans, system concept, cost, schedules, military policy or preferences, trade study results, math modeling, etc., etc., etc. To go in depth into any of these is beyond the scope of this paper, and we will begin at the point where the maintenance concept has been established and now we must implement it. Our relationship must be made clear, however, and that is this: the maintenance concept is dependent to a great degree on the validity of the concept and the factors upon which it was based.

Now the contractor Integrated Logistics Support organization comes into play, and here is where the first and most critical steps must be taken. These steps require that a very close working relationship be established and maintained between the Prime Mission design engineers, systems engineers, and with the ILS maintenance systems engineers. The importance of this relationship cannot be emphasized enough. Even small disturbances in their communications link can have profound effects on the availability of the Prime Mission equipment further downstream.

We can examine this delicate relationship in greater depth to gain a better understanding of how it should function. We'll start with a breif definition of responsibilities.

Systems Engineer: Overall systems design cognizance. Responsible for coordinating the overall design activities, ascertaining that system performance parameters are met and applicable specifications are adhered to, etc.

Design Engineer: Individual hardware design activity. Responsible for hardware design of certain designated parts of the system.

Software Engineer: Has responsibility for implementing the software required by the system design approach.

Reliability Engineer: Responsible for ascertaining that the specified reliability goals are met.

Maintainability Engineer: Responsible for formulation of an acceptable combination of design features, repair philosophies, Built-In-Test, and maintenance resources to achieve the specified level of maintainability at the optimum life cycle cost.

Maintenance Engineer: Responsible for developing maintenance procedures, maintenance task analysis, maintenance instruction, and the logistics resource requirements in terms of personnel and test equipment needed to satisfy the maintainability requirement.

Logistics Engineer: Responsible for providing the analytical inputs to the Maintainability and Maintenance Engineers concerning the various logistics support elements.

Note that the Prime Mission Engineers are the Systems, Hardware, Software, Reliability, and Maintainability types, while the Logistics and Maintenance Engineers are the Logistics Support experts.

From these definitions it can be seen that there is really no one individual that is totally responsible for implementing the "maintenance concept." All of the engineers thus identified share a part of this effort, and the responsibility. For example, the hardware design engineer, the systems engineer, the software engineer, and the maintainability engineer must all work together to provide the Built-In Test capability that has been specified. The hardware designers must provide enough circuit access in each board, each box, in each device, so that the software man can write the appropriate BIT routines. The systems engineer must oversee this effort and ensure compatibility between the various major components of the system, while the maintainability engineer assumes that the overall maintenance concept is being implemented. The reliability engineer monitors all design activity and assigns reliability to each major part of the overall system in order to make sure that system reliability goals are met. However, no matter how reliable a system is, it will eventually need maintenance, and this is the area we must address.

One major responsibility of the hardware design engineer has not yet been mentioned, and it is one of the most important of all. What I am referring to is Design for Testability, and the responsibility of the hardware design engineer to provide, from the very beginning, a design that can be tested. This applies to the major assembly level, to the subassembly level, to the printed circuit board level, and even down to the chip level, where it is probably the most important of all. For the very best support equipment the very best factory test equipment, can only do as good a job as the design of the item is testable! And it must start right at the very first levels of design and be carefully monitored.

There is a distinct relationship between Built-In-Test and Design for Testability. A good BIT system requires that the design follow Design for Testability guidelines. A design that is unsatisfactory from a design for testability standpoint will generally not allow a good BIT/FIT system to be implemented. Of course, a good Design for Testability grade does not guarantee good BIT, but it certainly does facilitate it.

The success of your Design for Testability criteria also directly affects your implementation of factory test equipment. Most companies in the aerospace business generally have a facility base of company-owned test equipment (capital equipment), and the mainstay of their capital equipment is generally automatic test equipment (ATE) for the p.c. boards, and wire wrap boards and backplanes, etc. The existence of such capital ATE generally results in company "design guidelines" (which should be, and in many cases are, design rules) which will make the designs from the engineers conform to the factory ATE capability, in terms of logic speeds, capability, maximum numbers of pins, test connectors, etc. Some more subtle factors are also addressed in these rules, such as circuit partitioning, the ability to break loops, and similar items. These rules are excellent in most cases and can form the basis for good "Design for Testability" procedures.

Continuing on with our BIT discussions, we find that BIT is extremely important from another aspect because, except for the Tech Orders and Training that are not associated with Special Support Equipment, it is really the stepping-off point for the remainder of the Logistics Support functions. BIT is really the first outpost of the logistics army that is mobilized when trouble is suspected.

Furthermore, support equipment designs are generally based upon a "beyond BIT/FIT" posture, and are geared to giving the maintenance technician the additional test capability he needs to make an accurate repair decision. There is definitely an overlap with the BIT capability itself, but the overall effectiveness of the support equipment comes from its ability to access signals and make meaningful measurements. The requirements for the support equipment, or Part I of the Support Equipment Requirement Document in Air Force terminology, are usually written by the Maintenance engineer. It is this man or group of men who must fully understand the prime mission equipment, how it operates, what its various signal paths are, what its failure modes are likely to be, what its predicted reliability is, and, most important, what its test access is when installed.

It is this man who must begin formulating his support equipment concepts while the prime mission equipment is still being conceived. It is this man who must be given accurate information on the prime design at all times, right from the very start. He must be informed of all changes once designs have been set, and he should really be informed prior to implementation, especially if field testability is potentially affected. And it is this man, in conjunction with the Support Equipment designer (who generates Part II of the Support Equipment Requirement Document), who should establish requirements for test connectors to the prime mission equipment designers.

"A system without Test Connectors is like a day without love" should be our motto. There should always be an abundance of test connectors planned right from the start, consistent with the packaging philosophy and spare/volume limitations. Experience has shown that we generally don't have enough test connectors or test access at system level, so we should plan on as many as possible. We are back at Design for Testability again, in case you didn't notice. Keep in mind that good test access on a p.c. board level does not guarantee good access on a system level. System test access should be what our maintenance engineers are looking for.

The maintenance engineer has a very difficult role indeed. His task of anticipating all of the logistics requirements based upon his understanding of the prime mission design means that he must keep his crystal ball polished, for he must in turn translate the anticipated logistics requirements into reverse requirements, to be implemented by the PME engineers, such as the item just noted concerning what test access is required.

The engineering personnel we've been discussing, being highly skilled and experienced, can readily assess this: "reverse impact" that the logistics elements have on prime design. Our perennial problem is how to do it in a timely manner, and how do we implement the necessary design actions implied by the logistics elements.

Two things are crucial:

1) The Maintenance/Logistics Engineers must be involved from the very beginning.
2) The Maintenance/Logistics Engineers must have a mechanism to influence the design.

These two items, which I'll call "Robert's Rules of Logistics," are related and we need to establish another rule in order to demonstrate how. We'll refer to our Maintenance Engineer and our Logistics Engineer collectively as ILS engineers.

Rule No. 3:

ILS Engineering will have impact on Prime Equipment Design.

What rule 3 says is that as the ILS engineers do their jobs, they will undoubtedly find areas in which the logistics implementation is simplified and costs reduced if changes are made to the PME. This has been proven time and time again. But often the "executive decision" is to not make a change because it will affect cost and/or schedule. This brings us back to the relationship between rules 1 and 2—if the ILS engineers must influence PME design, they should do it as early as possible, in order to minimize, or even avoid, any cost or schedule impact.

There is a serious problem here which is usually recognized too late on any given program. The problem is related to how much influence the Logistics engineers have in the PME design, or how much "clout" they have.

Anyone with an eye on the political system of today realizes, I am certain, that there are many ways to influence someone. Some of these ways, from what I have observed, are far more effective than others. Unfortunately, the most effective ones probably are a bit out of place in our particular environment, so we'll deal with the less dramatic approaches.

The Program Manager is the key man to successful implementing of the maintenance concept, because he must recognize the need for his ILS engineers at the very start, and set basic ground rules that the Logistics engineers are an integral part of the total engineering team, should have an equal voice in all discussions, especially those involving decisions. They should be consulted at all times concerning test access, test methods and requirements, all those items that we discussed earlier. The general test and support requirements for the type of PME being designed should be outlined fully at the very start of the planning and design phases. And this leads to number 4 in Robert's Rules of Logistics:

PME designers should design to logistics requirements as well as performance requirements.

Once rule 4 has been implemented and understood, we will begin to see better and more effective logistics elements. How often have we hard something similar to ". . .This is the most sophisticated processor ever put in a radar, it was all I could do to get it working properly, and you want me to louse it all up just to add some silly test connector?. . ." The answer to this question is Yes and No. Yes, we do want you to add the test connector and No, we don't want you to louse up your desing. The problem with the engineer who poses this kind of question is that he was only designing to a performance specification, whereas what we needed was for him to be designing from both a functional and a logistics standpoint. This gives the logistics engineer the "clout" they need.

The two just cannot be separated. You can't do your basic design and go back and add logistics considerations like you would add shutters on a house. The logistics considerations are a part of the overall design considerations and are a part of the foundation for the house, not just decoration or trim. But too many industry Program Managers treat Logistics that way—trim, decoration, afterthought. They simply must place high emphasis on the logistic design implications.

Another side of the coin is the Program Manager who has given Logistics the proper consideration from the very beginning. The Logistics engineers participate in the design decisions all the way. The maintenance concept implementation is well integrated into the system design. And then a glitch develops somewhere. The program manager is faced with an over-run in cost and a schedule slippage because of an unforseen problem. I believe the acceptable terms are "cost growth with attendant schedule modification!"

Where do you suppose the effort is cut back to reduce the projected over-run? Certainly not in the performance area where the reduced effort will show the first time you fly the thing, and our industry program manager begins to update his resume. So where? Ah, yes, in the Logistics area. We'll just take out some of those extra isolation circuits for test connectors, and we'll reduce . . . etc. You've seen it all before.

The thing to avoid is compromising the logistics implementation to save a few bucks for a few days at the end, because someone will pay a hundred times over in the end, with increased maintenance costs over the life of the system. And this "someone" is, of course, the military customer, the end user. So it behooves the end user to use some incentive to cause the industry Program Manager to keep his early logistics planning and implemention intact. Unfortunately, when we have development problems, which are the rule rather than the exception, what choices do we have as the sacrificial lamb? The military Project Manager is going to be unwilling to relax any performance parameters, or to buy a reduced number of systems, as this man, like our industry Program Manager, is also looking to move ahead in his job and not be forced into early retirement.

What I am implying is that the Project Manager, the military spokesman for the system, is often a part to decision on cutting back Logistics Support considerations and impacting future costs to the military, for the sake of saving some front-end costs or schedules. It is the age-old issue of "Acquisition Costs vs Life Cycle Costs," which has been addressed numerous times and for us to discuss it any further here is again beyond the scope of this paper.

It is also not necessary to delve any deeper into the Integrated Logistics Support disciplines that we discussed earlier and highlighted in our opening example, for their inter-relationship is fairly obvious by now. Equally obvious is the dependency of all of them on the success of the maintenance engineer and the logistics engineer—our key Logistics engineers—at the front end of the program. For whatever deficiencies exists in the prime mission equipment, be it a lack of test pionts, or a lack of good circuit partitioning, or deficiencies in Design for Testability criteria, or whatever, the remaining Logistics elements must try to make up the lost ground. Lack of adequate test access makes our Support Equipment more complex, which makes it harder to use, which may impact the skill level requirements of the technician and make the Tech Orders more elaborate, which may make the training program more extensive, etc., etc., etc. All of which increases our overall costs and makes maintenance much more difficult. The bottom line here is really less availability at greater cost. And that is the direct opposite of what we are trying desperately to achieve.

So far our attention has been focused on the military procurement cycle, from beginning to end, for a typical weapon system dedicated to satisfying a typical military need—where only one country or military agency is involved. How does all this apply to an existing, developed weapon system being bought by a second country? The answer to this question is not very simple, and depends upon many factors.

First, if the weapon system is being procurred as is, with no new or modified electronics being added or specially developed, then little opportunity exists to implement the "front end" Logistics Support considerations described herein. The BIT/FIT, for example, is generally difficult to improve once it has reached its full development potential. However, some opportunities do exist to implement improvements or refinements, or other changes necessitated by the peculiar military theater in which the weapon system will be deployed and maintained.

These changes will most likely be the area of Intermediate and Depot level test equipment, although certainly the Technical Orders, Spares provisioning philosophies, and Training methods and material can readily be altered to suit the second country requirements. Since detail discussions are dependent upon specific weapon systems and applications, we will select one hypothetical case to demonstrate.

Let us assume that country X has developed a lightweight attack fighter, and plans over 750 copies to be built. There will normally be a specific suite of both intermediate and/or depot level test equipment designed and built for the avionics. The cost of one set of this equipment, which includes, perhaps, eight automatic test stations, is roughly four times the cost of one plane.

Let us say that country Y desires to procure this plane. They will buy 75 aircraft. This country has also recently procured 50 of another fighter, along with its set of unique depot test equipment. Here it makes sense to look at alternatives to the procurement of the full set of depot test equipment for each aircraft, since one set can handle the needs of many hundreds of aircraft, and expensive equipment sits idle much of the time, when small numbers of aircraft to be maintained. One possible alternative is to see if the avionics from one aircraft can be tested on the test from the other. Even though this will involve new documentation and Technical Orders, plus new test programs and adaptors, it may still be far more cost effective than procuring an entire new suite of test equipment.

The main point to be made here is that even though a weapons system is fully developed, opportunities do exist to lower logistics support costs. The best way to identify such opportunities is to work closely with the manufacturer of not only the aircraft, but also the avionics systems. Their maintenance engineering personnel can identify numerous such opportunities, and careful applications of logistics principles, then after development of the weapon system, can lead to savings and improved maintenance.

So remember Robert's Rules of Logistics, make your ILS engineers a part of the team—even after weapon system development—and jump on the train to success.

THE INTEGRATED MANAGEMENT OF RELIABILITY AND MAINTAINABILITY IN PROCUREMENT

by
S.F.Shapcott and K.A.P.Brown
Procurement Executive
Ministry of Defence
Castlewood House
77 New Oxford Street, London WC1A 1DT

In response to the concern expressed by the Service User with the increasing cost of the maintenance of avionic equipments and of the need to improve reliability, a joint MOD(PE)/Industry Working Party was set up to determine an effective procurement strategy for reliability and maintainability (R&M). The strategy that emerged is now documented in "DCAD Technical Publication 1/77: Achievement of Avionic Reliability and Maintainability through Integrated Management". This policy has been ratified and adopted by the Air Systems Controllerate of MOD(PE), the Service User and the UK Avionics Industry. This paper reviews the evolution of these policies, outlines the requirements of this strategy and indicates how this strategy will be implemented.

INTRODUCTION

From the earliest days of engineering endeavour reliability was approached from the viewpoint of safety margins. When failure occurred extra material was added to the component or assembly to increase its strength and thus prevent further failure. Experience gained from the Second World War showed that operational effectiveness and mission success were severly jeopardised if equipment failed to work reliably and when required. This was also realised in the commercial arena in the early years after the war by the numerous airline operators who saw that the key to financial success was reliable operation and easy maintenance. However, with the advent of electronics, the mechanical approach to reliability, using safety margins, was no longer applicable and an approach based on statistical definitions and calculation emerged.

The electronic age also heralded a period of rapid technological advance and with these advances has come ever increasing demands for greater performance, more facilities, smaller size and less weight. There has also been heightened awareness that with increasing complexity and need for equipment to give reliable service when required can only be achieved by a combination of good design, manufacture and maintenance. These factors, together with the ever present problem of costs, have dictated the need for an approach to reliability and maintainability, (R&M), that binds together the various reliability techniques in a coherent management approach.

The methods of defining R&M in statistical terms have not produced a satisfactory way of defining these characteristics. Over the last few years there has been a growing realisation that a new approach based on reliability expressed in terms of performance and cost would have to be evolved. As a result the emphasis on the statistical aspects of definition has been reduced and much more importance has been placed upon improving management and engineering practices. It is now generally accepted that planning and management are vital factors in determining R&M. Thus by treating these as integral parts of the total project activity, increases in reliability can be achieved and the cost of ownership reduced.

In the Procurement Executive we have been aware of the particular roles the User, Procurer and Contractor have to play with regard to the attainment of R&M. There has been considerable progress in laying down sound foundations for developing and manufacturing equipment the Services need, with the required performance, reliability and quality. This progress has resulted in, amongst others, the publication of Defence Standard 00-10 - General Design and Manufacturing Requirements for Service Electronic Equipment - which include aspects of reliability; Defence Standard 05-21, Quality Control Requirements for Industry; and the BS 9000 system for Components of Assessed Quality. There have also been many studies conducted by the Procurement Executive into every aspect of R&M and work is still going on to provide the right framework in which realistic R&M requirements can be stated and fully realised, together with the equipments' other performance requirements.

The purpose of this paper is to describe the advances that have been made in the management of R&M on avionic and guided weapon projects, and the work that still has to be done. These advances have taken shape in the form of the recently published document known as "DCAD 1/77 Technical Publication: Achievement of Avionic Reliability and Maintainability Through Integrated Management"; it is now Controller Aircraft's policy to apply these principles.

This document, unlike previous publications, attacks the management problems by bringing together a great many R&M techniques and placing them into a framework which provides a uniform criteria for management activity throughout the whole life of each project. This document is a big step forward in ensuring that R&M is managed in exactly the same way as other main performance characteristics. It is also significant that the document is accepted by the User, the Procurement Executive and the major companies in the UK Avionics Industry; but although it has been conceived in the field of avionic systems, the message and general principles contained therein have a very much wider application.

EVOLUTION OF DCAD TECHNICAL PUBLICATION 1/77

The document had its birth in 1976 when, after a presentation made to various senior members of the MOD by GEC-Marconi Electronics Ltd, Controller Aircraft set up a joint MOD/GEC-Marconi Working Party to study the proposals made by the Company. The Working Party in due course had its industrial membership expanded to include staff from a number of major electronic and aerospace contractors representing their industries as a whole. From the MOD there was an equally wide representation from the Procurement Executive and the Air Force Department, with the Working Party being chaired by the Director General Air Electronic Systems. The first issue of the document came out early in 1977 but did not include the part covering maintainability aspects. After further work on maintainability, the document was

re-issued in March 1978 and now forms the basis of Controller Aircraft's R&M policy on avionics and guided weapon projects. His policy also recommends that the principles should be considered by Project Directors on other types of projects. Supporting the Technical Publication is a set of supplementary documents which have been prepared to provide managers with references to the R&M disciplines available to them. These supporting documents were published in December 1978.

THE DCAD TECHNICAL PUBLICATION 1/77

As has been indicated, the publication sets out the method of integrating the R&M interests and requirements of the User, Procurer and Contractor, in a continuous and evolutionary fashion. It also explains how these aspects of equipment procurement must be considered in the course of the decision-making process from the outset of the project right through to its in-service stage. It clearly states that it is the responsibility of the MOD(PE) Project Director to ensure that contracts embody the appropriate R&M requirements, and that he must also ensure that these are compatible with the overall project requirements and constraints. Furthermore he must co-ordinate all the activities directed towards achieving these requirements.

The main aims of the Technical Publication are to focus the Project Director's attention on:

a. Those R&M activities which must be considered by the User, the PE and the Contractor when defining the requirements for the equipment.

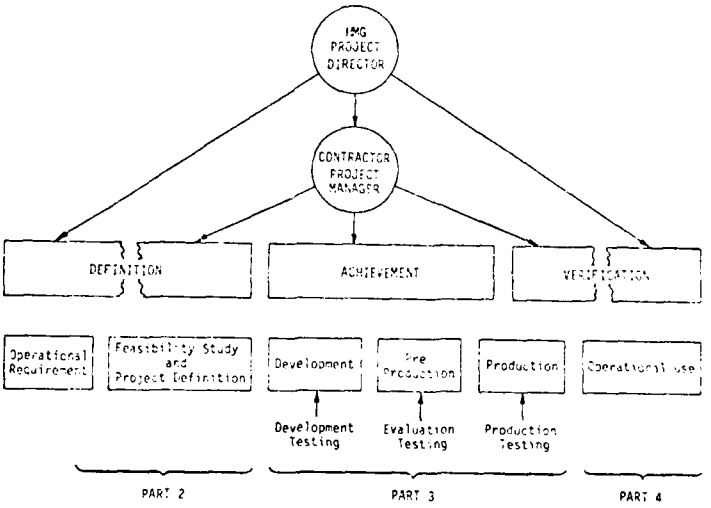
b. The maximum R&M likely to be achieved given the constraints of other factors.

c. What has to be done to achieve R&M requirements and writing this into the Development and Production Specifications and Contracts.

d. What tests have to be conducted, and at what stage they should be conducted, to verify and demonstrate the achievement of the R&M requirements.

e. What has to be done to assess, and where necessary improve, the in-Service achievement, particularly in those areas where the in-Service environment cannot be accurately predicted or simulated during development.

These aims can be stated simply as Definition, Achievement and Verification, and these, together with the management involvement of both the MOD(PE) Project Director, and the Contractor's Project Manager, can be illustrated as shown below.



The contents of the Technical Publication are divided into four parts:

Part 1 - A general introduction and definition of terms.

Part 2 - Deals with the R&M requirements to be considered by the MOD(PE) Project Director during feasibility and project definition, and the response required from the Contractor. This is an important stage because it is here that the R&M aims are turned into requirements by a process of trade-off studies in reliability, maintainability, performance and cost. It allows the designer to become aware of what is required, and for management to state how these requirements will be achieved. The Contractor must, at this stage, motivate his design and development engineers with a positive awareness of the R&M requirements, and provide them with the resources and support needed to achieve them. Equally the User must agree the balance that is to be achieved between R&M in the light of his in-Service resources and the operational environment.

Part 3 - Defines the planning requirements for R&M, the preparation and control of these plans and the assessment and verification of the achievement during the development and production phases. The main emphasis is that R&M must be designed for, that the design must be tested and any weaknesses overcome. Finally, there must be agreed objective evidence, through demonstration, that the R&M requirements have been attained.

Part 4 - Deals with the assessment of the achieved R&M in service. When equipments go into service inevitably there will be residual design problems which could not have been fully identified during development. The User's engineering advisers have a responsibility for providing the R&M data needed by the Contractor and the MOD(PE) to remedy these problems. Because of the Contractors' special knowledge of the equipment they must state what data they need to ensure that any shortcomings can be overcome quickly and effectively. The feedback of data from the Service User of the equipment is a fundamental feature of this R&M strategy. It is essential for the User to be fully committed to providing the data required by the Contractor if equipments are to attain the reliability required: this is a responsibility that the User cannot abdicate.

THE DCAD 1/77 SUPPORTING DOCUMENTS

These supporting documents are a source of information to Project Directors, Project Managers and their staffs, whose projects invoke the requirements of the DCAD 1/77 Technical Publication. This supporting documentation takes the form of a series of short documents dealing with the R&M elements referred to in the Technical Publication. Each supporting document has a standard form and gives guidance on what, why, when, how and who. These documents are not intended to be definitive statements on the subjects they address, rather they identify other source documents which can be used for the technical achievement of the R&M elements referred to in the Technical Publication. The contents of the supporting documentation are:

Introduction
Design Review and Appraisal
Review of Requirements
Reliability Predictions
Trade-off Studies
Failure Modes and Effects Analysis
Parameter Change Analysis
Development Reliability Testing
Selection of Equipment
Reliability Assurance Testing
Stress Level Verification Testing
Engineering for Maintainability
Maintainability Prediction
Maintainability Demonstration
Vendor Control
Burn-In
Defect Reporting
In-Service Data
Design Codes of Practice
Training

IMPLEMENTING AND MONITORING THE DCAD 1/77

One of the important features of the Technical Publication is that it allows Project Directors and Project Managers to have the flexibility to achieve the required balance between reliability, maintainability, performance and cost appropriate to their projects. A number of projects have been nominated by Controller Aircraft on which these procedures will be applied and these cover the main types of avionic and guided weapon systems.

There is a clear and unambiguous commitement by the Air Systems Controllerate to the principles contained in the Technical Publication and in their application to avionic and guided weapon projects, and every encouragement is being given to Project Directors of other types of projects to use the document. Clearly implementation of the principles will vary from project to project and within the Controllerate provision has been made for expert help and assistance to be made available to Project Directors and their staffs to obtain an effective approach to R&M. If the full benefits of this policy are to be gained it will be necessary to monitor the effectiveness of the implementation of these procedures and to identify any area where potential problems in applying this Document exist.

For this monitoring to be done effectively requires close contact with the projects where the principles are being applied to be able to obtain objective evidence in the light of problems which may arise on a particular project. Equally, industry has a vital part to play since they have to executive the strategy of this R&M policy and it is hoped that they will monitor the effectiveness of this policy on their activities. Finally the User, as has been stressed, must be as committed to this strategy as the PE or Contractor, and he must ensure that he fulfills his commitments and make a contribution to the monitoring of the implementation of these procedures.

To provide a forum for the discussion of problems of mutual interest concerning the overall policy and its implementation a small liaison group has been proposed. It will be chaired by the PE and bring together those who have a direct contribution to the implementation of the DCAD 1/77 policy with a membership drawn from the PE, Industry and User.

It is also intended that the Liaison Group will report periodically to an Air Systems Co-ordinating Committee. In addition the Liaison Group, through the Co-ordinating Committee, will report progress being made with this policy to the Committee for Defence Equipment Reliability and Maintainability (CODERM). This committee has the responsibility to survey Reliability and Maintainability practices over the whole of the MOD and thus these reports will be a valuable contribution to the policies being evolved with MOD as a whole.

CONCLUSIONS

For far too long R&M have been given insufficient attention or left to the care of off-line bodies. It has been very easy for equipment sponsors to set unrealistic R&M targets without being required to justify these requirements in performance or cost terms. Likewise contractors have been able, using statistical techniques, to show by prediction that these targets are attainable without any consideration of the engineering or technological contraints which might limit the possible achievement of these targets or make such targets not cost effective. The consequence of this has been, in general, a failure to achieve specified reliability and maintainability requirements either in terms of system availability, mission success, maintainability or logistic support.

The approach advocated in the DCAD Technical Publication comprises a set of management principles based on the engineering concept that a good design taken properly into production is the only way to achieve reliable and maintainable equipment at an acceptable cost of ownership. It is recognised that reliability and maintainability cannot be divorced from performance or cost, and it is therefore necessary to treat them in the same way as other performance characteristics, integrating the management of them with that of the project as a whole: in other words reliability and maintainability by DESIGN in every sense of the word.

The DCAD Technical Publication has already had wide acceptance in MOD(PE) not only in those areas associated with avionics, but in areas concerned with aircraft equipment in general and electronic procurement as a whole. For avionics and guided weapons the Air System Controllerate is firmly committed to establishing the use of these principles and assessing their effectiveness, and giving every encouragement and support to others who wish to apply them.

Equally strong is the commitment to see that R&M achievements are demonstrated prior to equipment being offered to the User.

It is recognised that a consequence of this R&M strategy is a possible increase in cost and timescales. The acceptance of such increases is a matter for the PE, but at least with the policy that has now been adopted the potential increases in reliability and ease of maintenance can be assessed against any consequential increases in procurement costs or timescale.

The reliability and maintainability strategy that has been adopted rests on a number of facts. The integration of the management of reliability and maintainability with that of the project as a whole; the attention to R&M from the earliest stages of design right through into Service life; the commitments of the Procurer, Contractor and User to the attainment of R&M and the demonstration of these achievements. On this foundation it is believed that equipments can be procured which gives the User what he needs; Equipment that works reliably, when required, and at a reasonable cost of ownership.

RELIABILITY AND SUPPORT DATA FOR STATISTICAL EVALUATION

by

Group Captain A.Andrews
Head of RAF Maintenance Data Centre
RAF Swanton Morley
Dereham, Norfolk
UK

SUMMARY

The ultimate test of reliability is experience in the field. However equipment reliability is not an absolute property, it interacts strongly with the environment. The relatively uncontrolled environment in the field poses particular problems in capturing and interpreting reliability and support data. The organisation and procedures for aircraft data collection in the UK Royal Air Force and Royal Navy are described, together with the methods of storage, retrieval and analysis which have been evolved in order to give a responsive service to military and industrial agencies. Some principle applications of the data as part of an integrated reliability management programme are outlined. The problems of relating in-Service reliability to inherent reliability as measured during development are discussed.

1. INTRODUCTION

1.1 The Maintenance Data Centre at RAF Swanton Morley in Norfolk, England, is responsible for a management information system called the Maintenance Data System (MDS). One of the aims of the MDS is to provide reliability information for engineering managers. Unfortunately reliability is not an absolute property of an equipment; it is a characteristic which interacts strongly with an environment which in Service it is impractical to control and the results can be difficult to interpret accurately.

1.2 This paper describes the procedures for collecting and analysing reliability data in the Royal Navy and Royal Air Force and some principal applications during an integrated reliability management programme. The problems in collecting and interpreting field data are discussed and difficulties in relating it to reliability measured in the design and development stages of new equipment are described.

1.3 It is unnecessary to emphasise the penalty of aircraft unreliability for this symposium, but to summarise our problem in the RAF, unreliability is the cause of a significant proportion of our aircraft being unavailable for operations, the source of

over half our maintenance manhours per flying hour and it costs us over £200 million a year.

2. Defect Data Collection

2.1 So what do we at MDC do to help reduce the burden of unreliability? The first thing we can do is to quantify the problem. To measure a complex phenomenon like reliability, we have found that it is not sufficient to take a sample, we need to measure every arising. Defects reported to us occur at the rate of 1 a minute or about 500,000 a year. This is a formidable input workload but even so the sample size on any particular defect from which one aims to draw valid statistical conclusions can be quite small. 100% defect data capture is therefore our starting point.

2.2 Fortunately from our point of view tradesmen have to record work done for safety reasons and we capture the information simply by taking a copy of the job card. In order to tell us what we need to know about each defect, each job card will have 25-30 entries to cover the information in Figure 1.

2.3 In order to complete a history of each defect, the key elements of information are:

Symptom of Defect

Cause of Defect

Action taken to rectify Defect.

2.4 This is a sound recording philosophy, but there is a snag; that not all this information will become known at the same time or place. The symptom will be found on the aircraft but the cause of a defect may not be uncovered until the defective part has been removed and repaired perhaps in the Station workshops or perhaps at the depot or manufacturer, much later. In fact an average of 2 job cards are raised for each defect and we link them by an unique identifier so that makes over a million job cards a year to be collected (Figure 2).

2.5 In order to cope with this large volume of input, data compression is essential and the information on each job card is coded and retained on magnetic tape. This presentation is not about computers so I will simply say at this point that at MDC we have an ICL 1904S computer with 128K of store which operates in batch processing mode. It has served us well but batch processing limits the speed of response to requests for information to about 24 hours and we are in the process of upgrading the system to achieve on-line access to the computer files.

2.6 Discretionary Narrative Reports

The defect job card reporting system is supplemented by narrative reports which are raised if the originator believes a defect requires a specific investigation. There are also defect-related incident and accident reports which are raised for flight safety reasons,

reports of foreign damage and so on.

2.7 These reports are microfilmed for storage, but to ease retrieval they are indexed and abstracted on the computer files which link job cards and related narrative reports. This enables us to retrieve the complete reported history of a defect arising (Figure 3).

2.8 The narrative report has an important function in highlighting serious defects, but still too many engineering managers and equipment manufacturers act as if this report represented the sum total of defect experience, ie, no reports equals no problems, but only 1 to 2% of defects are covered by narrative reports. For accurate statistical interpretation of Service experience therefore we must look to the analysis of defect reports on job cards.

2.9 Defect Data Bank

To sum up the input stage, our Defects Data Bank consists of some 2 years worth of data ie, over 2,000,000 records each with 25 to 30 pieces of information, which is readily accessible. Older data is transferred to archival files which go back to 1972, so data is never lost. This is our data base from which we attempt to measure the in-Service reliability problem.

OUTPUT

3.1 Now we come to the real pay-off, the output. We have two main types of output:

a. Routine Outputs

b. An Interrogation Service

3.2 Routine Outputs

Like all similar data systems we have a family of routine printouts to meet the specified needs of customers. This includes periodic listing of defects and reliability summaries of aircraft and major equipments, presented in various ways. We also have "standard" printouts produced on demand but to a standard layout such as the "Case History" of a defect.

3.3 These outputs are intended for particular applications but they lack flexibility, indeed they can be positively miselading if used for a purpose for which they were not designed.

3.4 The Interrogation Service

For this reason, we much prefer the customer to explain his problem to us so that we can produce an answer to meet his specific requirements. To help us to do this we have a system which enables us to control our data retrieval to suit the task and provide an individually tailored answer to every customer. We call this the Defect Interrogation Process (DIP), which is the basis of the Interrogation Service available to any

entitled individual or formation, military, Ministry or manufacturer. The Interrogation Service is a valuable customer service which we have developed to a greater extent than any other information system we know of. Its popularity is such that 45% of computer processing is for interrogations, compared to 2% for routine outputs.

3.5 It works like this. The customer is put into contact with a member of staff whose job it is firstly to help him ask the right question, and secondly to frame a suitable interrogation to meet the enquirer's needs. The Interrogation is set out in three stages:

Stage 1 specifies the Job Cards to be called up.

Stage 2 specifies the Fields to be copied from them.

Stage 3 specifies the order in which they are printed.

3.6 Thus we can if we wish interrogate on aircraft operations only on a particular Station between specified dates. We can limit our Job Cards to those raised on defects found during Flight Servicing or when a Mission failure resulted; we can provide lists of defects where the faulty component had to be replaced; the number of permutations at our finger tips is virtually endless. And having interrogated on these selected Job Cards we can have our data printed out as we wish in chronological order of occurrence, or in order of increasing Item usage,or with all similar symptoms grouped together; the choice is ours.

3.7 Consider again the list of Job Card contents at Figure 1. We can Interrogate on Aircraft type. Print out the defects under the various headings of When/How Found and we have an operational picture we can use in our Scheduling. We can specify one Aircraft Tail Number, print out all the defects in order of ascending airframe hours and we have a ready record of that Aircraft's defect history - this has been used to assist in an Accident Board of Enquiry. We can specify an Engine type, and printing out all its defects in ascending order of running hours gives us the basis for a lifing exercise. Printing them out again, this time in Section/Reference Number order of the defective component and we can see which parts give the most trouble. So in preparing a maintenance policy for a new aircraft we can answer questions like "does failure mode have a direct effect on operating safety?" and "Is there an adverse relationship between Age and Reliability?" with some confidence. So we can now measure reliability in numerous ways.

3.8 Incidentally there is no need at all for a request for data from MDC to be made in a formal manner; a telephone call is perfectly adequate, any authorised user in government or industry has free access to our Data Bank. We get some 15 to 20 tasks per working day.

3.9 Who wants data from MDC?

The latest breakdown of tasks by customer grouping is:

Defence Ministries and Air Commands	28%
RAF Stations and Units	25%
Project teams	22%
Industry	14%
Royal Navy	11%
	100%

The task loading from the Stations and Units is particularly welcome because these are the people who put the Data in and it is heartening to be able to show that they get something back for their efforts.

3.10 What do they want data for?

This is how our latest Analysis appears:

Individual Defects; Fault Diagnosis	30%
Servicing and Lifing of in-Service aircraft	15%
Reprovisioning for current aircraft	15%
Families of defects	11%
Statistics, modelling data	10%
Modifications	8%
Scaling future aircraft	4%
Internal tasks	7%

4. APPLICATIONS OF RELIABILITY DATA

4.1 The importance of avionic reliability has recently been emphasised in the UK by the issue of a publication DCAD 1/77 by our MOD Procurement Executive. This sets out the principles for achieving reliability through integrated management on the part of the Service user, the Ministry of Defence Procurement Executive and Contractors involved. What do we mean by integrated management? Well DCAD 1/77 does not state new requirements; it introduces the concept of a continuous and evolutionary approach to reliability as an integral part of any project with significant development content, from Staff Target through to acceptance into Service. In fact in each development, the application of MDC data has a vital role. This paper allows scope only for a description of the application; the statistical treatment would in each case take a separate paper.

4.2 Pre Staff Target.

Operational analysis studies are usually carried out with the aid of mathematical models for which we provide input data. This helps to shape the staff requirement and determines what

type of aircraft and also the number which would more effectively meet the operational task.

4.3 Staff Targets and Requirements.

In the formulation of staff targets and requirements MDC data is used to set realistic R and M levels. This is particularly important if some kind of contractual agreement is sought as the user has to be confident that the improvement he seeks will provide an adequate return on the reward he offers, while the manufacturer has to be convinced that the targets are achieveable. However there are demonstration problems which will be enlarged on later.

4.4 Equipment Selection.

A major contribution by MDC is in the selection of equipment; historic experience often gives a good guide as to which components or design concepts are to be avoided or preferred.

4.5 Maintenance Policy.

In planning stages for acceptance into Service, field data provides an essential input to the decision process for determining the maintenance policy, numbers of equipment to be purchased, scales and location of test equipment. Again mathematical models are used to simulate the engineering and operational environment, and generate cost options which indicate the most economic maintenance policy.

4.6 Entry Into Service.

Field data is used to compare the actual performance of the equipment against the design requirements. Early in-Service it is used to highlight problem areas for intensive development in order to achieve predicted levels as quickly as possible.

4.7 In Service Monitoring.

Later in-Service, performance is monitored so that engineering effort may be concentrated on the most rewarding areas, bearing in mind operational and flight safety factors. The validity of the servicing schedule content is reviewed. The cost effectiveness of proposed modifications is assessed. Component life can be extended with confidence. Using pre-set alert levels which, if exceeded trigger an automatic output from the computer, we can detect incipient problems before they reach crisis proportions. The effectiveness of the supply support can be monitored by recording the incidence of robbing. The data can also be used to monitor the quality of repair of equipment received from manufacturers and maintenance Units.

5. LIMITATIONS OF DATA RECORDING IN SERVICE

5.1 We find that our data is being used more and more extensively by manufacturers but sometimes there is disagreement on the validity of the data we collect. In particular, in

comparison with the manufacturers predictions and development experience, reliability data collected in the field is invariably more pessimistic.

5.2 Now there are undoubted problems in capturing accurate data in the field; you have to depend on tradesmen who are more interested in completing work on the aircraft than filling in forms.

5.3 We invest a very considerable effort to obtaining good input quality and in fact half our running costs are devoted to this objective. Each job card, containing some 110 alpha-numeric characters after coding, is subject to a rigorous series of computer controlled checks at MDC which produces about a 5% rejection rate, which is not bad particularly as most of the errors can be corrected locally. Nevertheless there is much to be done to improve input quality. Our efforts must start with the tradesmen who provide the data, and as already indicated we try hard to provide useful feedback to Operational Stations to convince them of the value of accurate input. In the long term we aim to introduce computers for engineering management at Station level and we believe that the involvement of tradesmen in local defect data processing will be a powerful stimulus to improve quality. We find that the main sources of error are:

(a) Corruption of the 'linking' process between associated job cards. This breaks the defect loop creating spurious arisings and an incomplete defect history.

(b) Lack of Feedback of defect data from Manufacturers who repair our equipment. Again this results in incomplete defect history and sometimes there is a significant proportion of arisings whose cause is "not known".

(c) Recording and Coding errors. Despite the validation checks at MDC there is a proportion of incorrect or ambiguous data input to the system.

(d) Data Lag; an adherent penalty of a batch processing system. Most job cards are received in a few weeks but it takes up to 3 months to acquire 95%+ of the data which we consider to be a workable data base (Figure 4).

6. COMPARISON OF DIFFERENT TYPES OF RELIABILITY

6.1 We recognise that there are some errors in field recording and we are progressively reducing them. However it should be accepted that they are not the main cause of the discrepancy between the manufacturers predictions and Service experience. The problem arises from the fact that the Services measure a different form of reliability. Various definitions are in use, but essentially the distinctions are:

(a) Operational or Logistic Reliability is measured in the Service, and this is the type of reliability which recognises the effect of all occurrences which place demands on the logistic support systems. So

its the reliability you require when assessing (say) spares and manpower requirements.

(b) Inherent reliability is predicted and measured by the manufacturers during development testing, in general this includes only those failures which are confirmed to be attributable to design and manufacturing quality errors.

6.2 The differences between logistic and inherent reliability are as follows:

(a) Definition of terms - there is seldom an exact agreement in what constitutes and attributable failure.

(b) Rating - field recording systems tend to rate globally by flying hours rather than individually in running time except where an elapsed time indicator is fitted.

(c) Arisings in downtime - significant in Service but difficult to reproduce in development.

(d) Equipment Environment - in development it is difficult to reproduce accurately the effect of temperature, vibration, humidity, spiking power supplies to which in-Service equipment is subjected.

(e) Servicing Environment - the manufacturer prefers not to recognise failures arising from servicing problems such as mishandling, faulty diagnosis but often he is partly to blame (inadequate test equipment, schedules, access, carrying handles, poor location etc).

(f) Secondary Damage and Interface Failures - genuine logistic failures but difficult to predict in development.

6.3 Taking all these factors into account, it is evident that the relatively poor showing of equipment in the field is inevitable and should not be the cause of recrimination by the User Service with the manufacturer, for falling down on his predictions on the one hand, or by the manufacturer with the Service field recording systems for inaccuracy on the other.

6.4 As Logistic Reliability includes Inherent Reliability plus other elements, it ought to be possible to establish a relationship between the two. Indeed it is important that we do so, as only then will manufacturers be content to accept contractual reliability requirements based on in-Service experience; and a worthwhile penalty or incentive contract on reliability is needed to reverse the present anomalous situation under which manufacturers are actually rewarded for unreliability by larger repair contracts.

6.5 In practice of course it is extremely difficult to establish such a relationship, and contractual reliability arrangements tend to be associated with the artificial but controlled environment of development testing. Field data recording has been used as the basis of an incentive contract for the RAF's Hawk trainer aircraft,

but the manufacturer reasonably wishes to exclude those defects for which he does not consider himself responsible eg, secondary damage. In practice this means that the attribution of responsibility for each defect will have to be made individually as the defects arise, by a committee on the reporting Station; a costly process which could be avoided if the relationship between the different types of reliability was better established.

7. CONCLUSION

7.1 It is said that good decisions are born of 10% inspiration and 90% information. MDC's task is to provide the information. For this purpose we aim for 100% defect data capture and provide a powerful and flexible data retrieval system, used by experienced engineering staff, which enables us to process the data in many different ways and produce outputs tailored to the needs of the engineering managers who use us.

7.2 Field data has an important role in any integrated reliability management programme, from preliminary studies through to the in-Service phase of a project; indeed a commitment on the part of the user to collect such data is an essential part of the programme.

7.3 However there is a discontinuity between measured reliability at the development and in-Service phases, which arises from the fact that there is no quantified relationship between inherent reliability and logistic reliability, although there is obviously a dependence between the two.

7.4 If we are to introduce contractual reliability incentives based on the "acid test" of Service experience we have to do more work to establish a correlation between development and in-Service reliability data.

7.5 Equally we need to improve the completeness and accuracy of field data collection, primarily by motivating the tradesman who originates the job cards, by providing useful feedback and involving him in local ADP data processing. These are all subjects of joint interest to member Nations, which I am sure would benefit from a continuing exchange of experience.

JOB CARD DEFECT RECORD

WHAT AIRCRAFT: TYPE; TAIL No: SHIP; STATION; AIRFRAME HOURS; TIME DATE PUT U/s

HOW FOUND: SYMPTOM OF DEFECT

WHEN FOUND: IN FLIGHT FLIGHT PHASE; OPERATIONAL EFFECT; MISSION; FLIGHT SAFETY EFFECT.

ON GROUND SCHEDULED SERVICING; FLIGHT SERVICING; INSPECTION IDENTITY.

WHAT FOUND: CAUSE OF DEFECT; IDENTITY OF ITEM; LOCATION OF ITEM (SYSTEM/ASSEMBLY); AGE; SERIAL No.

WHAT DONE: ACTION TAKEN (REPAIR/REPLACE); MANHOURS; TRADE; TIME DATE COMPLETE.

FIGURE 1

DEFECT LOOP

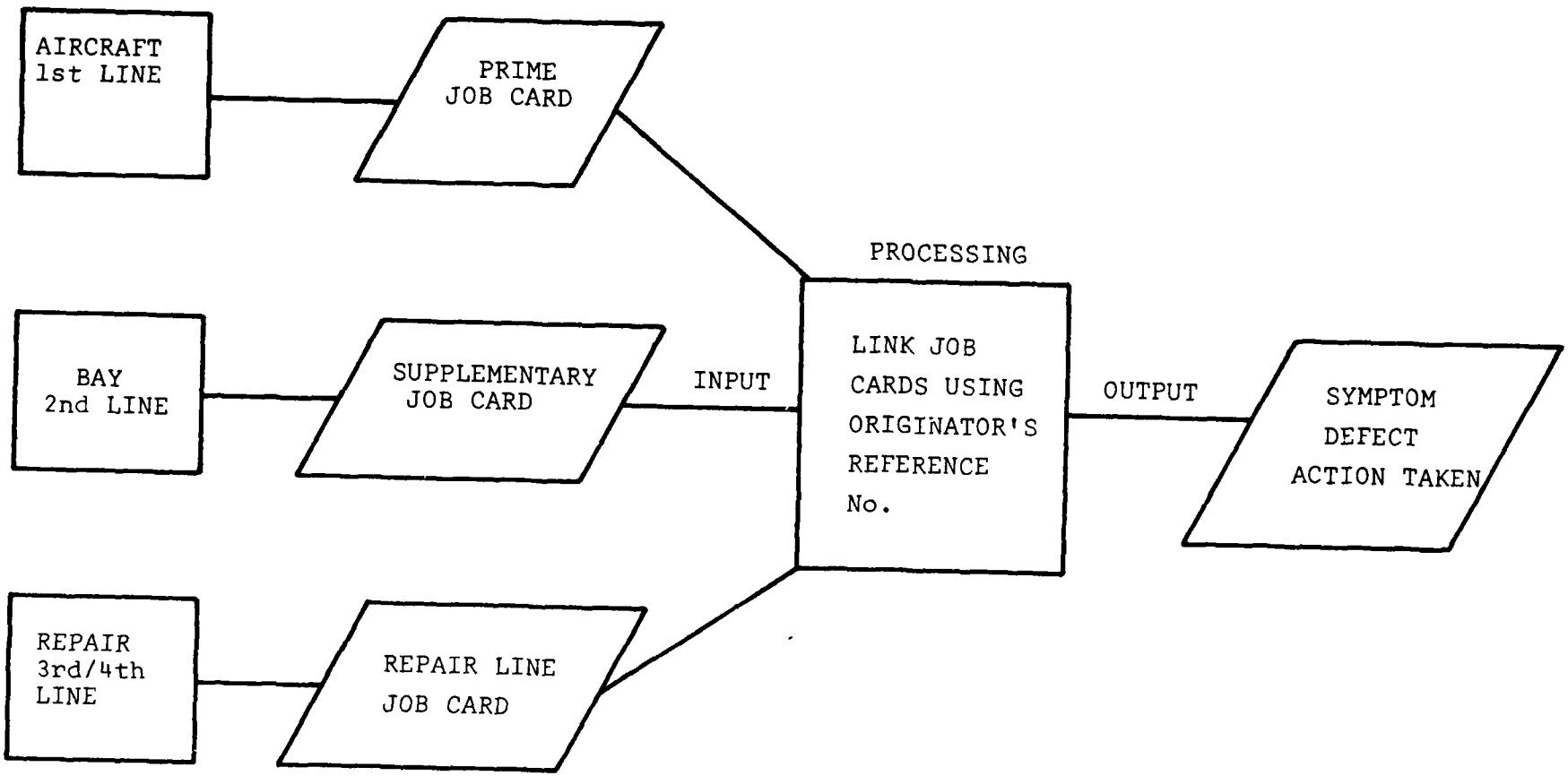


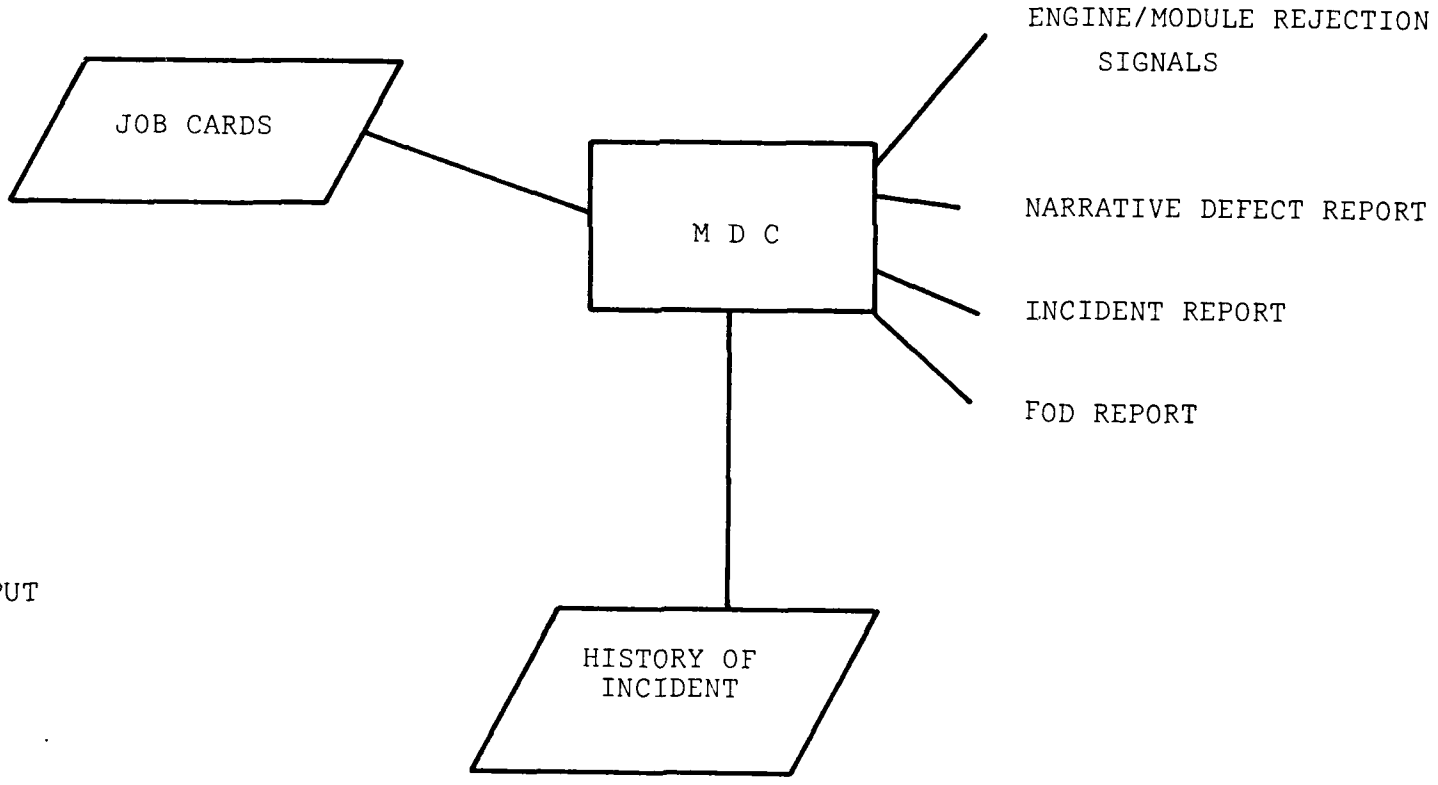
FIGURE 2

INTEGRATION OF MDC OUTPUT

DEFECTS FILES

INFORMATION CENTRE

INPUT



OUTPUT

FIGURE 3

INPUT DATA GROWTH

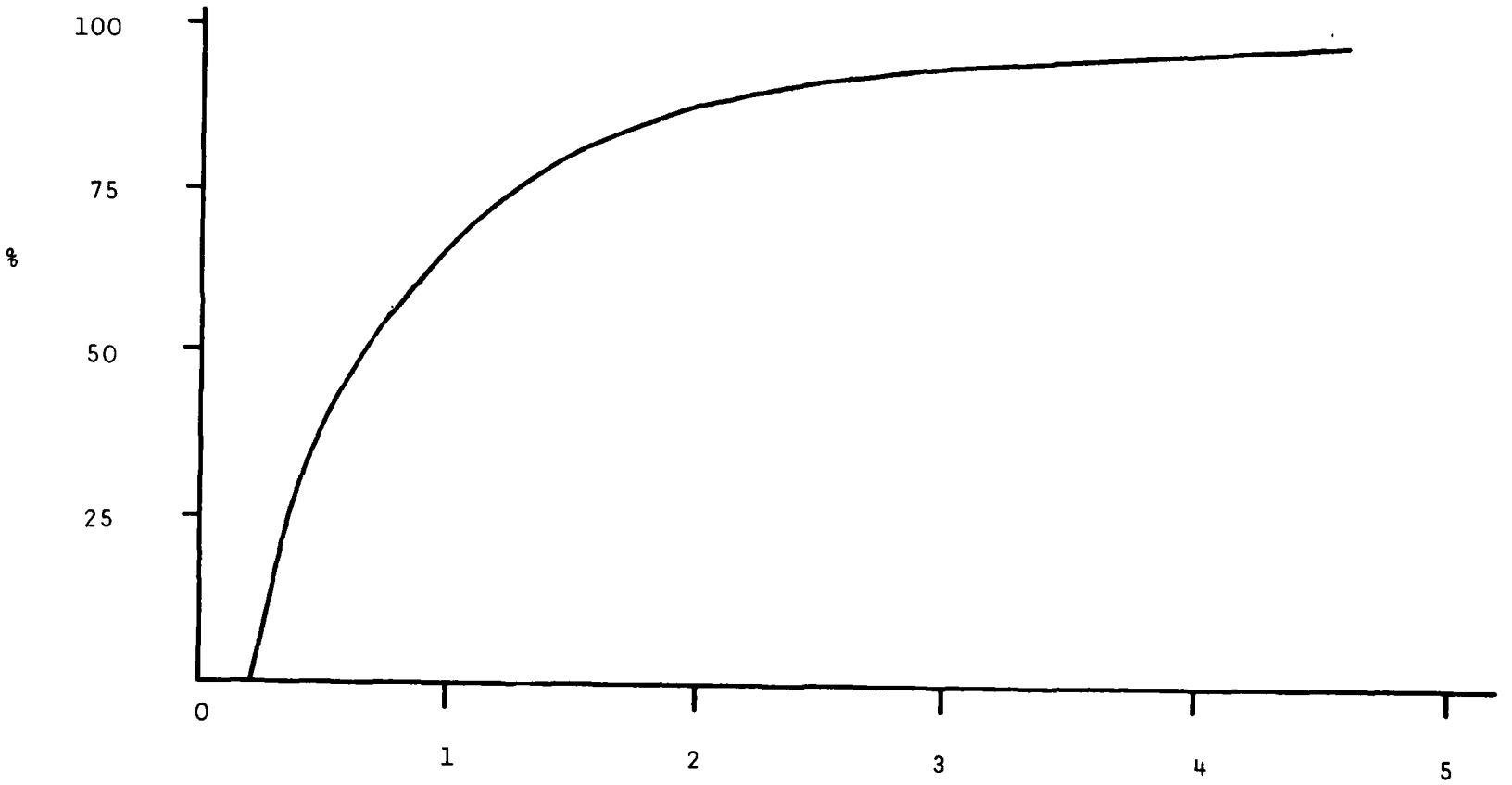


FIGURE 4

DISCUSSION

H.S.Balaban, US

A universal problem for US military data collection systems is accurate measurement of equipment operating time, especially for the equipment which has not failed.

Accurate measurement is necessary for contractual guarantees such as on operational MTBF.

Author's Reply

The RAF/RN has the same problem. The Maintenance Data System has provision for recording operating time where an elapsed time indicator is fitted. Whether this is fitted or not is a matter for the materiel office. Equipments which have not failed would of course not come to the notice of the Maintenance Data System, although the system could identify, by exception, which they were. Given a reasonably long measurement period, I would have thought that the number of equipments not subject to defect would have been small, and that suitable adjustments could be made to allow for the exceptions.

H.Gross, Ge

When evaluating the operational data collected in the German Air Force, it can be observed that the influence of non-technical factors such as season (summer/winter), wing base etc. tends to gain a higher significance than the technical factors themselves and thus may cause misleading evaluation results.

Do the evaluations performed in the RAF Maintenance Data Centre encounter similar problems?

Author's Reply

Non-technical factors do strongly influence defect rates although I would not agree that they have a higher significance than technical factors. The climate affects both the environment humidity, water in cockpit etc. and sometimes the flying rates so it does have an effect, but from a logistics point of view this is a genuine change which the defect recording system should recognise. Human factors have an influence e.g. inexperienced aircrew or servicing personnel but for reliability assessment we use only confirmed defects i.e. those for which the existence has been confirmed during rectification, in order to minimise some of the human factors.

COMPUTER SIMULATION MODEL OF THE LOGISTIC SUPPORT SYSTEM FOR ELECTRICAL ENGINEERING TEST EQUIPMENT

by

Squadron Leader C J P Haynes
Chief Scientist (Royal Air Force) Department
Ministry of Defence, London, UK

SUMMARY

Electrical engineering test equipment (EETE) is a unique subset of the broad spectrum of avionic repairable equipments. In general avionic repairable equipments require a support organisation for the repair of random defect arisings but EETE is unique in that it also requires periodic recalibration and thus an additional facility in the support organisation. In the past, the Royal Air Force provisioning formulae for spares requirements in support of the maintenance system for EETE have been extremely simple. This very simplicity has led to doubts as to their adequacy for provisioning spares for the whole range of EETE. A simulation model of the existing logistic support system for test equipment has been developed as a research tool for evaluating the adequacy of analytical provisioning models. Two further simulations have been developed to model alternative logistic support systems and to compare these with the current maintenance organisation. The present logistic support system, the development of the three simulation models and their potential uses are discussed in this paper.

Abbreviations

EETE Electrical engineering test equipment

TESU Test equipment support unit

INTRODUCTION

For repairable avionic equipment it is necessary to have an associated logistic support system. To set up such a system requires decisions to be made as to the depth of repair to be undertaken at different echelons of the organisation, what proportion of repair, if any, should be done in industry and what level of spares holdings should be procured to support the proposed system. For the broad spectrum of avionic equipments in service in the Royal Air Force these decisions are made with the assistance of a series of computer based mathematical models. These models have not, however, been used for one particular family of equipment, namely electrical engineering test equipment (EETE). In one respect at least, the logistic support system necessary for EETE is more complex than is the case for other avionic repairable equipment. Both suffer from random defects which need to be repaired but EETE also requires periodic recalibration. This adds an extra facet to the maintenance organisation.

Despite the more complex nature of the support organisation for EETE, the methods used by the Royal Air Force to calculate the requirement for spares have, in the past, been much simpler than the mathematical models used for other repairable equipment. This is, perhaps, not surprising in that the effect on the operational task of having too few spare test equipments is not usually as immediate or serious as the effect of having too few spare aircraft components. Nevertheless, the investment in EETE is considerable. In the Royal Air Force there are more than 4000 different types of test equipment varying in number from single figures to thousands of each type. The total inventory is valued at more than £60M with a correspondingly large annual maintenance budget.

It was decided therefore that the simple methods used to calculate the requirements for test equipment spares needed investigating to determine whether or not they were adequate. At the same time a means of assessing the merits of alternative logistic support systems was needed. It was decided that computer simulation models of the EETE logistics support system would meet both of these needs.

The aim of this paper is to describe the development of the models and the uses to which they can be put, illustrating the discussion with a simple example. Our intentions for the future use of the model will be outlined.

THE EETE LOGISTIC SUPPORT SYSTEM

Firstly it is necessary to explain the existing logistic support system for a typical item of test equipment. A simple flow chart of the system is shown in Fig 1.

Direct Exchange Recalibration

Numbers of each type of test equipment are in use in various user units. The numbers authorised will be referred to as the "scale". For any particular piece of test equipment there is one calibration agency which is scaled with a pool of spare test equipment in order that it may operate a direct exchange scheme. This scheme works as follows. When an in-use item of test equipment nears the date for recalibration the user unit informs the calibration agency. They take an item from the calibration pool and if it needs calibration, which is usually the case, calibrate it. It is then dispatched on the next run of the dedicated transport to the user unit. On arrival at the unit the calibrated piece of test equipment is exchanged for the "out-of-calibration-life" item. The latter is then returned to the calibration agency. Once there, it is put into the calibration pool in an uncalibrated state, where it will remain until needed to satisfy a similar requirement in the future.

Loan Recalibration

The first complicating factor in the calibration system occurs when the calibration agency is unable to provide a direct exchange service. This can happen either because there are temporarily no items available in the calibration pool or because the chosen replacement item is found to be defective during calibration. In either event loan recalibration then takes place. This involves the "out-of-calibration-life" item being collected from the user unit by the next routine transport run. It is then returned to the agency, calibrated and sent back to the user unit on the subsequent routine run. For the intervening period the user unit is without the item, hence the term "loan recalibration."

Difficulties in Analytically Modelling Recalibration

This differentiation between direct exchange and loan recalibration is difficult to represent analytically. There is a known calibration arising rate and at first sight it appears possible to calculate a fill rate corresponding to a given pool size. Unfortunately, non-fills from the direct exchange pool result in loan recalibration which makes no demands on the pool thus modifying the effective arising rate as observed by the pool. But the state of the pool at any particular point in time is what determines the arising rate of loan recalibrations. This is one of a number of complicating factors which make an analytical model of the process difficult to determine and to validate.

Repair

In common with any other form of avionic equipment, EETE suffer from random defects. Some of the minor defects may be repaired at the user unit provided repair does not require subsequent recalibration. The remainder of defect arisings are sent to a repair agency, usually in industry, for repair and recalibration. At the same time the user unit demands a replacement from the supply depot which has a repair pool provided for this purpose. Once the defective item has completed its repair it is sent from the repair agency to the repair pool at the supply depot.

Some items are also found to be unserviceable during recalibration at the calibration agency. Minor defects can be corrected at the calibration agency but the remainder have to be returned to industry for repair. In this situation the calibration agency demands a replacement from the supply depot to replenish the calibration pool. This interaction between the calibration and repair aspects of the maintenance organisation further complicates the development of truly representative analytical models.

Present Method for Calculation of Pool Sizes

The present method for calculating the number of items to be procured for the calibration pool is tied to the periodicity of recalibration. If the time between calibrations is 6 months then the calibration pool is set at 20% of the inuse scale. With an annual recalibration periodicity the pool is set at 10% of the inuse scale and with a 2 year cycle the figure is 5%. The repair pool is set at 10% of the inuse scale. These formulae are summarised in Fig 2. These fixed percentages are not rigidly adhered to where the total number of articles in use is small.

It might be reasonable to use a fixed percentage for the calibration pool size since although actual bench times for calibration vary from one type of equipment to another this represents only a small fraction of the total out-of-service time. This includes transit time, administrative time, queueing, etc and may be substantially the same for different types of equipment. However, the incidence of calibration failures varies from equipment to equipment and must in turn vary the incidence of loan recalibration. Thus, setting the calibration pool numbers as a fixed percentage of the inuse scale seems unlikely to give similar levels of system performance, as observed by the user unit, for different types of equipment.

For repair pool calculations it is clear that arising rates and repair times will differ from one type of equipment to another. Therefore it seems unlikely that setting the repair pool size as a fixed percentage of the inuse scale would be satisfactory for the whole range of EETE.

Clearly, more sophisticated mathematical formulae for calculating pool sizes could be developed but because of the complexity of the system such models would inevitably make a number of simplifying assumptions. We needed a means of judging how representative any analytical models would be. Furthermore we were interested in investigating the effects of some fundamental changes in the logistic support systems. It was decided that the best means of satisfying both of these needs was to develop computer simulation models of the EETE logistic support systems.

SIMULATION

A simulation is a computer model of the logical interactions between the separate functional elements of the system being modelled; random sampling techniques are used to simulate the variabilities of the real world. The variables within the model do not have to conform to well known parametric frequency distributions and the assumptions made about the relationships between the variables can be much less restrictive than in analytical models.

Simulation of the Present ("Normal") Logistic Support System

In that a simulation mirrors the logical interactions of the real world, the description of the logistic support organisation (Fig 1) can serve equally well as a description of the simulation logic. The basic simulation (which in future will be referred to as the 'Normal' version) therefore includes all those aspects of the EETE maintenance organisation which have already been described: direct exchange recalibration, loan recalibration, industrial repair, supply depot and appropriate transportation systems. Furthermore, all the significant real world constraints, such as maximum floor loadings, are also incorporated in the model. Defect arisings are generated within the model as a Poisson process.

Additional Versions of the Simulation

In addition to the Normal version of the simulation, the model has been extensively modified to create two further simulations. These have the function of modelling hypothetical changes in the EETE maintenance organisation.

Calibration on Receipt Version.

Under the present system the bulk of the items in the calibration pool are in an 'uncalibrated' state. In this way equipment is only calibrated shortly before dispatch, thus giving a maximum useful calibration life at the user unit. However, when items fail recalibration it is necessary to undertake loan recalibration which is less satisfactory from the user's point of view. An alternative approach would be to maintain a pool of calibrated equipment by recalibrating items as soon as they were received from the user-unit (see Fig 3). It was thought that this should lead to a lower incidence of loan recalibrations but, because of the time spent in the calibration pool, provide a shorter useful calibration life. To compare the merits of this "Calibration-on-Receipt" system with the Normal system a second version of the simulation was developed incorporating all the changes implied by this alternative procedure.

Test Equipment Support Unit Version.

At the time of writing the first two simulation models, a radical change in logistic support organisation was being considered. The idea was that the functions of recalibration and repair should be combined at a single location, a test equipment support unit (TESU). This unit would carry out all the maintenance functions necessary to support a particular type of EETE except for those previously performed by the user unit. The calibration and repair pools would be amalgamated at the TESU which would offer a direct exchange service for both recalibration and repair arisings. The attractions of such a scheme are reduced transportation, reduced turn-round times and reduced holdings of spare EETEs. The most obvious disadvantages are increased capital investment in in-service facilities and a necessary increase in both the range and quantity of piece part spares needed to support in-house repair. A third version of the simulation (see Fig 4) was developed to model this TESU concept.

Validation

Validation of any simulation is always a difficult problem. In this case it involved three stages.

a. Painstakingly going through the simulation logic to ensure that its main features accorded with the real life situations it was intended to model.

b. Satisfying ourselves that the simplifying assumptions implicit in the model would not seriously affect the models representation of real life (most simulations include simplifying assumptions but they are usually much less restrictive in nature than those made in analytical models).

c. Running the models using representative input data and checking that the output was consistent with our expectations. In the case of the Normal version of the model this process was aided by comparing the outputs with observations from the real system.

These three stages of validation were carried out on all three models and no reason to doubt their validity emerged. However, it must be said that so far the simulations have only been run using the input data for one particular type of EETE, albeit a fairly typical example.

Simulation Outputs

The data used for validation purposes was also used to generate the typical outputs described below. It was also the data on which the example of use of the models, discussed later in the paper, was based. The data was that for one type of equipment in service for which the inuse scale was 251 and the value, £350 each. The recalibration period for the item was 6 months and there was a mean of 55 industrial repair arisings per annum. In line with present scaling policy the number of spares allocated to the calibration pool was 20% of the inuse scale ie 50. Similarly the repair pool scaling was 10% of 251 ie 25, making a total inventory of 326 items.

A wide variety of outputs could be taken from the simulations, ranging from an extremely detailed record tracing the progress of every individual item of EETE to a brief summary of the overall system performance.

There are two main outputs currently generated by the simulations. The first is a record of the number of equipments in each possible location sampled once every 20 working days. This information is then manipulated into histogram form showing the frequency of observing a particular number of equipments in each location. An example of this is given at Fig 5, where the output is the histogram of numbers of equipments in use at user units. As shown in Fig 5, this can also serve as a histogram of the 'numbers out of service' since the numbers out of service can be equated to the inuse scale minus the number actually inuse.

The second form of output is a summary of the number of transactions of each type during the period of the simulation under review. This output is then changed into monetary units by ascribing a cost to each different type of transaction (see Fig 6). For example, in the figure for the cost per repair would be included manpower costs, piece part spares cost, overheads etc. The cost of all transactions are then totalled over a ten year period (a reasonable timescale for amortizing the purchase of the items) and added to the capital cost of the equipment to give a life cycle cost. Finally in this same summary output two further measures of performance are recorded; the number of direct exchange recalibrations versus the number of loan recalibrations and the number of defect replacement demands met immediately from the repair pool versus the number that could not immediately be satisfied (the supply depot fill rate). Both of these outputs are measures of how adequate are the numbers of spares in the calibration and repair pools.

Measures of System Performance

From the outputs mentioned above three major measures of system performance emerge:

a. The life cycle cost of the equipment for a given logistic support system.

b. The frequency distribution of the numbers of test equipments in use. The user units are scaled to hold a set number of test equipments and are not entitled to hold any spares. Because of the occasional need for loan recalibration and the temporary loss of equipment due to inuse defect arisings there are usually less than the 'scaled' number of test equipments actually inuse. The mean number deficient (the number out of service) is a measure of the effectiveness of the logistic support system.

c. The direct exchange/loan recalibration ratio and the supply depot fill rate. These measures are secondary in the sense that changing the number of spare items is reflected in both the life cycle cost and the mean number out of service. However, both ratios are useful in that they indicate whether spares provisioning is of the right order and whether the apportionment of spares between the calibration and repair pools is in balance.

EXAMPLE OF USE OF THE SIMULATIONS

At this stage, development and validation of the simulations has only recently been completed. Few results are available to illustrate the use of the models. However, one simple example of their use in choosing between alternative logistic support policies can be given. For the purpose of this example data on the single item of test equipment, described in the sample outputs, was used.

The simulations were run using the input data appropriate to this equipment. Each version of the simulation thus produced a figure for the life cycle cost of the item and the mean number out of service for that particular logistic support system. These results are tabulated in Fig 7. Since the 3 different options, Normal, Calibration-on-Receipt and TESU all result in different life cycle costs and different mean numbers out of service, choosing between them on this basis can be difficult. The TESU system appears to be the best since it is both cheaper and more effective than the other two. But choosing between these two is impractical because Normal is more effective but with a higher life cycle cost.

The next step was to vary the number of spare items in the system thus varying both the cost and performance for each alternative. The range of cases considered is tabulated in Fig 8. The ratio of calibration pool size to repair pool size was maintained at 2:1 for all cases. It was recognised that this would lead to sub-optimal solutions for a given total number of spares available and that for true comparison the ratio would need to be varied in each simulation till a near optimal solution was obtained. If executive decisions were to be made on the basis of these results then this additional refinement would have been necessary. Since the purpose of the results was merely to demonstrate use of the models the 2:1 ratio between pool sizes was kept constant.

Running all three versions of the simulation with the 4 levels of spares scaling resulted in 4 data points for each version to plot on the graph, Fig 9. Using this sort of graph one can choose between logistic support systems either by setting a maximum life cycle cost and selecting the system which gives minimum mean number out of service or conversely by setting a minimum level of performance and choosing the system which achieves that level at least cost. Again the TESU concept appears best under either set of conditions but choosing between Normal and Calibration on Receipt would depend on what constraints on cost and/or performance were imposed.

Assessment of Analytical Models

The other use for which the test equipment logistic simulation was developed was to provide a means for assessing the adequacy of analytical models for scaling spares holdings. Work is progressing in this area but for the present it is sufficient to say that modelling the repair aspects analytically seems to be relatively straightforward and good agreement between analytical model predicitions and simulation outputs has already been observed. Analytical models of the recalibration process are much less easily constructed because of the more complex nature of the interactions involved, particularly those between the direct exchange and loan recalibration processes. It is thought that this aspect may be regarded as analagous to the "lost order" situation in inventory control where orders which are not immediately satisfied are lost rather than held awaiting future satisfaction; it has yet to be demonstrated that such an approach is appropriate to the test equipment recalibration situation.

Use of the Simulation Models for Provisioning

An alternative to using the simulations to assess analytical provisioning models would be to use the simulations as provisioning models in their own right. With over 4000 different types of EETE, use of simulation models, which take a long time to set up, run and interpret, would be both expensive and impractical. Analytical models, once developed satisfactorily, are almost invariably preferable for this type of work.

CONCLUSIONS

EETE is a unique subset of the broad spectrum of avionic repairable equipment. It is unique in that as well as requiring a repair organisation it also demands recalibration facilities. Both of these maintenance functions have to be supported by an appropriate buy of spare items if the user is not to be unduly inconvenienced by the loss of equipment requiring maintenance. Formulating analytical provisioning models is made difficult by the complex nature of the maintenance operation. Simulation models of the EETE logistic support system have been developed as research tools for assessing how representative an analytical model is. Use of the simulation models directly for provisioning purposes would be uneconomic.

The simulation model has been developed in three versions representing different support policy options: Normal, Calibration on Receipt and TESU. It has been demonstrated that these models can be used to choose between alternative support policy options. In this respect the models could either be used to assist maintenance policy decisions when the procurement of a major new type of test equipment is envisaged or by evaluating a range of typical equipments assist in framing general policy for logistic support of EETE.

In the field of analytical provisioning models, development will proceed until adequately representative models are achieved.

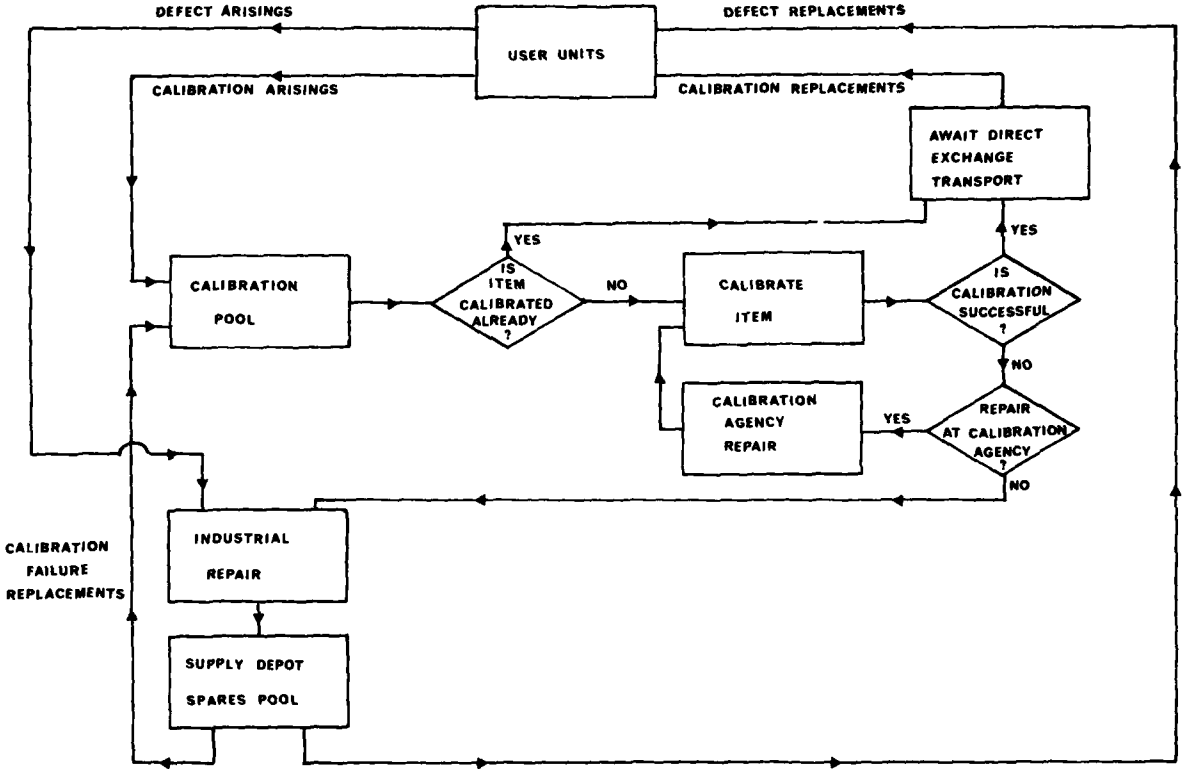


FIG.1 Flow Diagram of the Present ("Normal") Logistic Support System for Electrical Engineering Test Equipment

1. Calibration Pool Scale

Calibration Periodicity Months	Calibration Pool Scale as a % of In-use Scale
6	20%
12	10%
24	5%

Repair Pool Scale

2. The repair pool scale is set at 10% of the inuse scale.

FIG.2 Current Basis for Calculation of Spares Pool Scalings

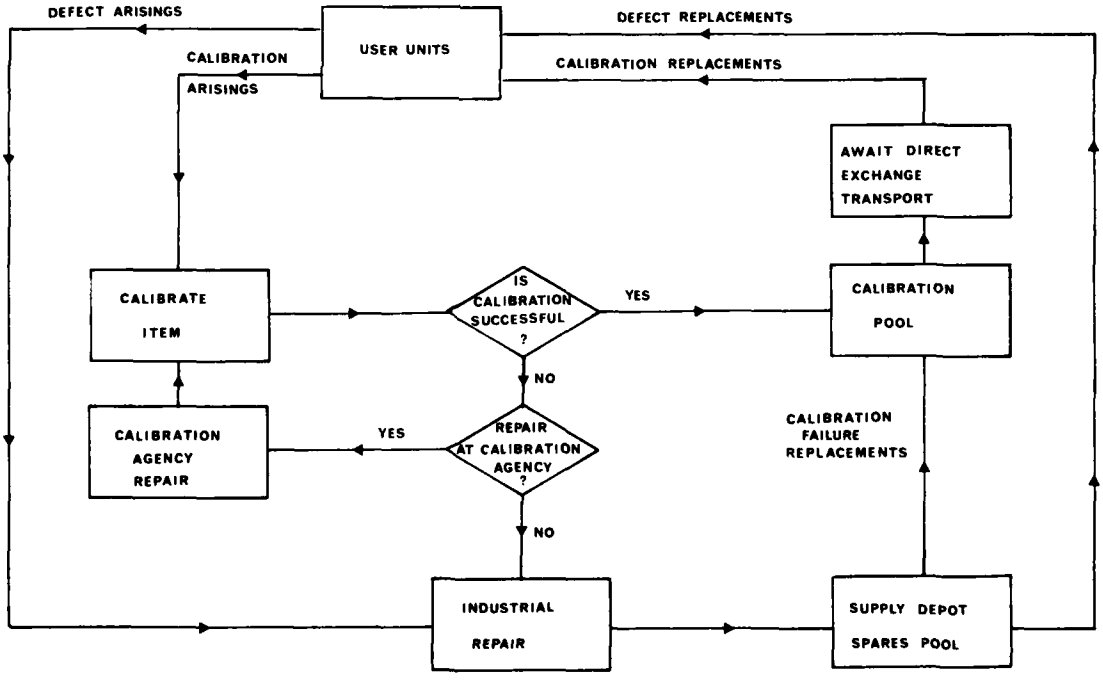


FIG.3 Flow Diagram of the 'Calibration on Receipt' Logistic Support System for Electrical Engineering Test Equipment

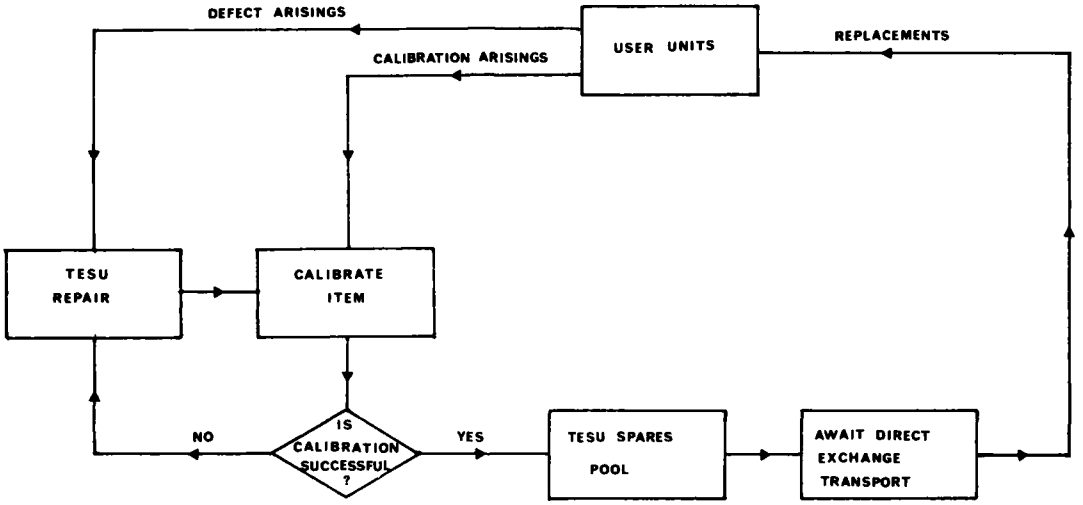


FIG.4 Flow Diagram for the 'Test Equipment Support Unit (TESU)' Logistic Support System for Electrical Engineering Test Equipment

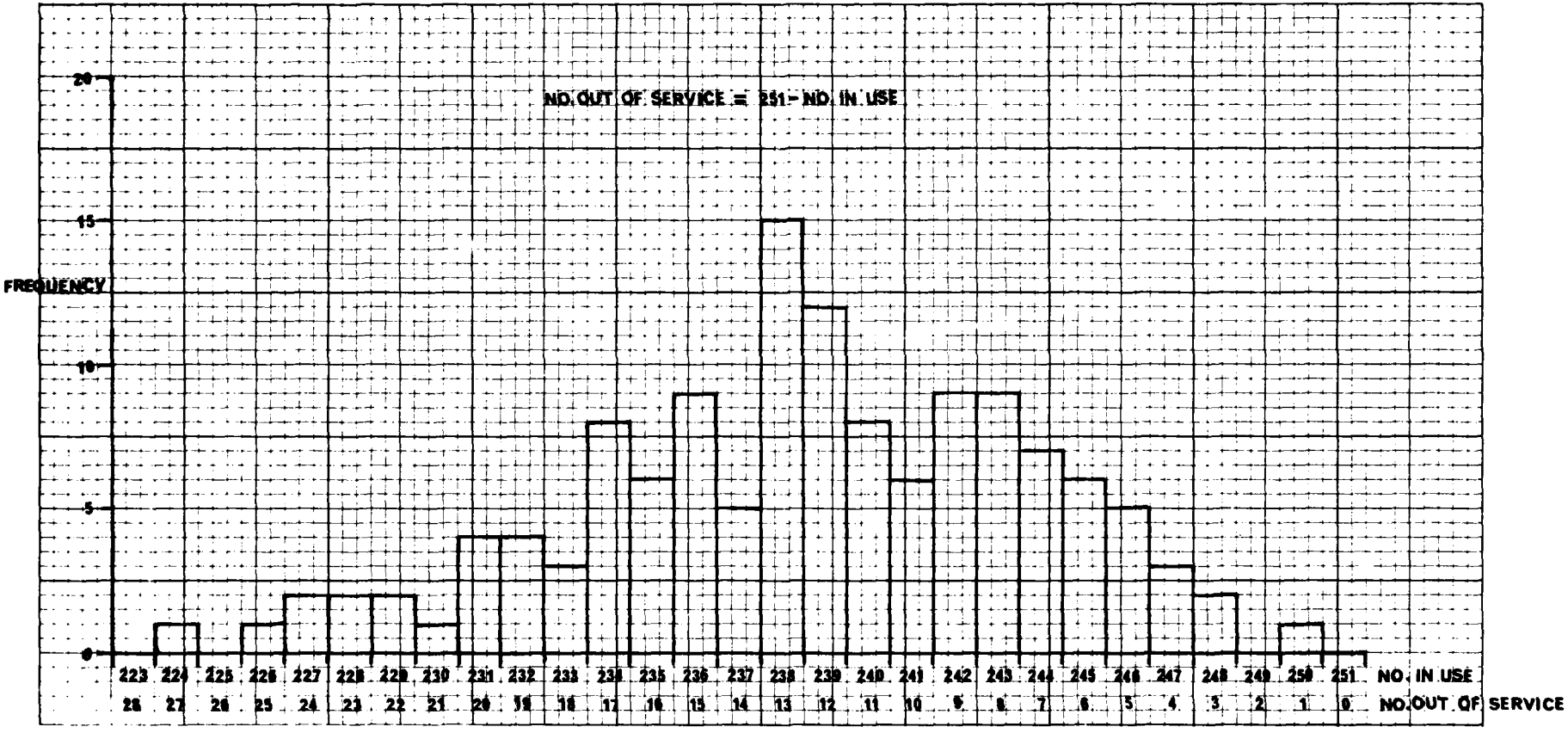


FIG.5 Histogram of Number of Test Equipments In Use and Out of Service

TEST EQUIPMENT SIMULATION

ARISINGS AND TEST EQUIPMENT COSTS

CALIBRATIONS	5602 @ £ 6	.....	£ 33612
STATION REPAIRS	0 @ £ 0	.....	£ 0
CAL AGENCY REPAIRS	4006 @ £ 4	.....	£ 16024
INDUSTRIAL REPAIRS	679 @ £ 150	.....	£101850
COST OF TEQS IN SYSTEM	326 @ £ 350	.....	£114100
	SUBTOTAL	...	£265586
MOVEMENTS BETWEEN:			
BASES AND ESD	115 @ £ 2	.....	£ 230
BASES AND CAL AGENCY	12149 @ £ 2	.....	£ 24298
BASES AND INDUSTRY	177 @ £ 2	.....	£ 354
ESD AND CAL AGENCY	419 @ £ 2	.....	£ 838
ESD AND INDUSTRY	556 @ £ 2	.....	£ 1112
CAL AGENCY AND	626 @ £ 2	.....	£ 1252
	SUBTOTAL	...	£ 28084
TOTAL COST		.....	£293670

Percentage of Recalibrations which are Direct Exchange	81%
Percentage of Defect Replacement Demands met Immediately from Stock (Supply Depot Fill Rate)	78%

Abbreviations

CAL AGENCY	Calibration Agency
TEQS	Test Equipments
ESD	Equipment Supply Depot

FIG.6 Typical Simulation Output - Life Cycle Cost

SYSTEM / RESULT	NORMAL	Calibration on Receipt	TESU
Mean Number Out of Service	12.3	16.5	8.4
Life Cycle Cost	£292K	£273K	£228

FIG.7 Table of Comparative Results for the Three Test Equipment Logistic Support Simulations

	SCALES			
CASE 1	INUSE	CALIBRATION POOL	REPAIR POOL	TESU POOL
Normal	251	20	10	
Calibration on Receipt	251	20	10	
TESU	251			30
CASE 2				
Normal	251	36	18	
Calibration on Receipt	251	36	18	
TESU	251			54
CASE 3				
Normal	251	50	25	
Calibration on Receipt	251	50	25	
TESU				75
CASE 4				
Normal	251	70	35	
Calibration on Receipt	251	70	35	
TESU	251			105

FIG.8 Input Data on Spares Scales Used to Achieve Cost Versus Performance Data

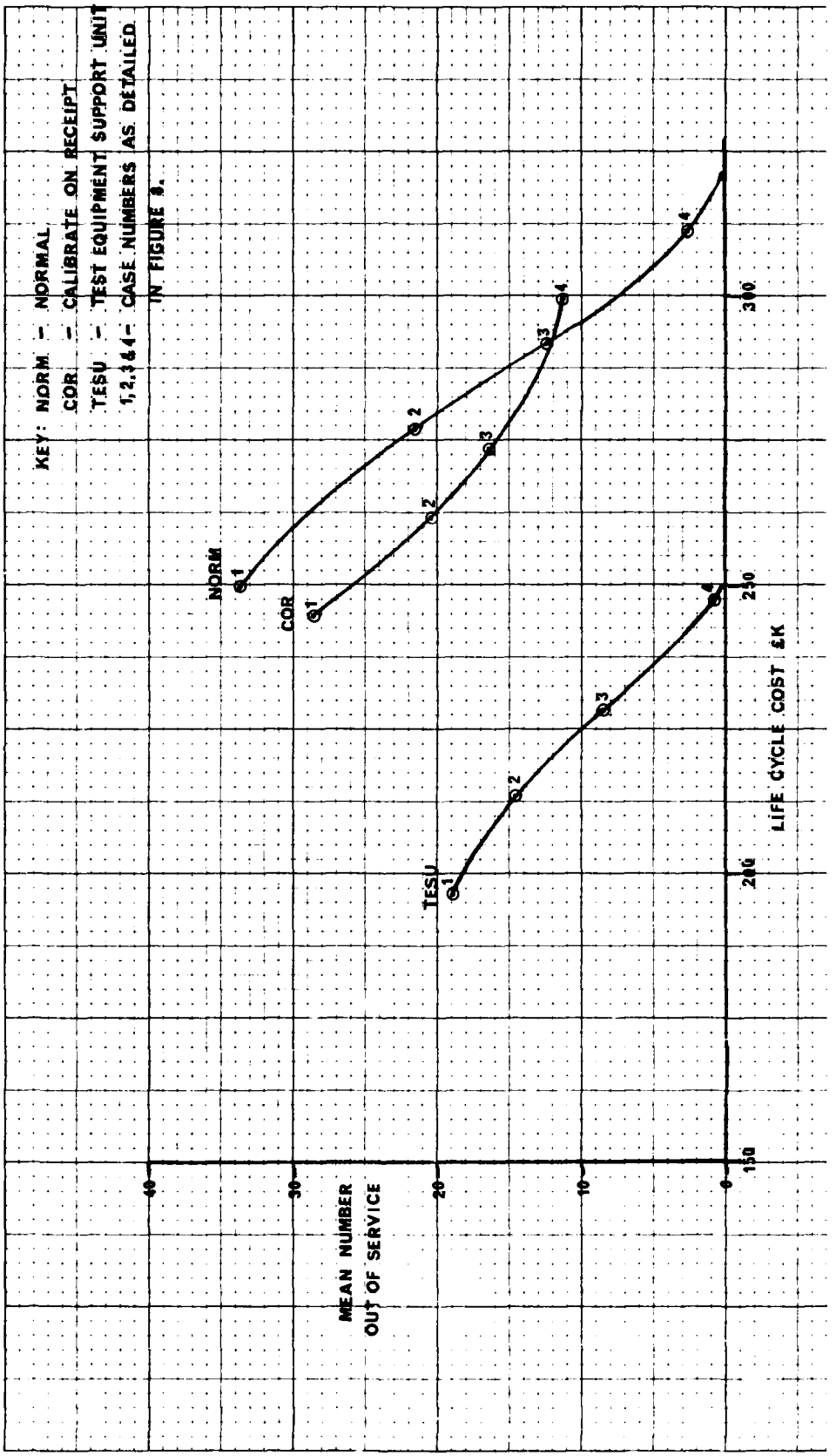


FIG.9 Graph of Performance Versus Cost

DISCUSSION

M.B.Kline, US

In your initial flow chart you stated that when a test equipment was returned for calibration it was only calibrated later when there was a demand for it. You later stated that you examined the situation where it was calibrated immediately upon return. By not calibrating it immediately, one runs the risk of having a defective equipment later on when it is calibrated. Can you comment on this?

Author's Reply

We developed a second simulation to avoid keeping potentially defective items in a pool without them being looked at. As it turned out on the particular item that we chose, the difference between the two models didn't appear to be significant. I suspect that the reason is that when calibration is as short as it is, 6 months, and the scalings are relatively low, that the items spend little time in the pool and the difference between the 2 systems is not all that marked. We have only run this with one set of data at the moment because it is a new development, but I believe that this is something which should always be watched.

R.Voles, UK

In practice, the calibration agency will have a limit to its throughput rate. This will generate queues if the first strategy is used and will increase the mean unavailability time. Does it follow that an appropriate mixed strategy would be better?

Author's Reply

Certainly there are limits on the throughput of the calibration agency and limits on the floor loading were a feature of all three simulations. It is possible that a mixed strategy of a compromise between "calibration on request" and "calibration on receipt" might be better. This would require modification of the existing models.

P.D.T.O'Connor, UK

Will the model be used for decision-making for test equipments which are part of larger ATE systems?

Author's Reply

Where an item of test equipment is a module within a larger ATE, and where that module is one of a reasonably large number of items of the same type, it is possible that the logistic support for the module will be as characterised in the simulation. Given all these conditions, then the simulation model could be used.

THE RELIABILITY IMPROVEMENT WARRANTY AND ITS APPLICATION TO THE F-16 MULTINATIONAL FIGHTER PROGRAM

George Harrison

ARINC Research Corporation
2551 Riva Road
Annapolis, Maryland 21401

SUMMARY

The role a military customer plays in assuring acceptable reliability for his avionics has traditionally consisted of stating the requirement, specifying a test, and having great trust in the intelligence and integrity of his contractor. However, this formula has not always been successful. New techniques being explored on a trial basis by the U.S. Department of Defense are directed toward placing carefully structured reliability assurance incentives and opportunities in the procurement contract. One of these techniques is generally known as the Reliability Improvement Warranty (RIW).

The successful application of the principles of RIW to the F-16 multinational aircraft could mark the beginning of new procurement techniques for many NATO countries. The results of the F-16 RIW program will no doubt be followed with intense interest not only be the U.S. Air Force but also by the four co-producing governments. Other NATO nations could learn from this program and perhaps adapt similar procurement techniques to work within their own particular military objectives, economics, and national laws.

The fundamentals of an RIW procurement, together with some of the variations that have been used to suit particular applications, are described in this paper. Guidelines for RIW application are also presented.

The F-16 multinational fighter program is described, with particular emphasis on two aspects: the development of the F-16 RIW program, and the co-production agreements between the U.S. and the governments of Belgium, Denmark, Norway, and The Netherlands.

INTRODUCTION

The technology for designing and producing highly reliable electronic components and systems has been available for approximately 25 years. We have standards that deal very effectively with reliability allocation, prediction, and testing. For example, U.S. missile programs and space programs in a number of nations attest to the fact that high reliability is possible. However, these types of programs that achieve high reliability are quite expensive. We have learned that we cannot simply specify how much reliability we want and be assured that we will obtain it, as we can for other technical parameters of an electronic system. One reason is that to buy highly reliable parts, and then design and manufacture a highly reliable system is costly and places the contractor's competitive position in jeopardy. Further, it is difficult to measure the reliability of individual items -- something that is easily done for other performance parameters. In the United States, efforts historically have been directed toward comprehensive reliability programs that include parts screening, predictions, more stringent specifications, and rigorous demonstration and acceptance testing. While some improvements have been made, these efforts have produced less than the desired overall result.

As illustrated in Figure 1, the specified reliability requirements for several U.S. military avionic systems were not met in the field environment. The figure compares the specified MTBF with test and operational values and shows that the operational values fall short of the desired objectives. Although not all equipments exhibit such a trend, this type of behavior can increase O&S costs and reduce asset availability.

Unfortunately, achieving a field or operational reliability that is consistent with our expectations still frequently eludes us. We need a new approach. Many equipment manufacturers see the frustrations of their military customers and are quick to advance one of the following causes for poor field reliability:

- There is very little in the economics of a competitive procurement that provides us with the incentive to produce reliable equipment.
- Our equipment is "intrinsically" reliable. Your operational and maintenance environments are inducing the failures.

The Reliability Improvement Warranty (RIW) is a procurement tool that addresses both of these causes.

THE RIW CONCEPT

An RIW is basically a fixed-price contractual commitment by an equipment manufacturer to perform depot-type repairs for a stated number of years. The following key guidelines are being used by the Department of Defense to select candidate systems for RIW:

- Fixed-price procurement

- Competitive procurement
- Equipment amenable to reliability improvements
- Proven technology
- Field testable
- Readily transportable

Consider the economic situation for both the contractor and the government in normal procurement practices and compare it with what occurs under RIW.

Figure 2 shows reliability on the horizontal axis and cost to produce a piece of equipment on the vertical axis. At the beginning of a contract under normal conditions, some minimum acceptable reliability is included in the equipment specification, and the successful bidder is awarded a fixed-price contract.

Figure 3 shows the contractor's initial cost to produce, which, for a particular equipment and a particular contractor, is assumed to be an increasing function of reliability. This is, all other things being equal, the contractor can reduce his cost to produce by reducing component quality, testing, and inspection -- or the quality of a number of other aspects of his program -- thereby reducing equipment reliability. The shaded area between the horizontal "Fixed-Price Contract" line and the "Initial Cost" curve in Figure 3 represents the contractor's profit for a unit. Thus, at any point on the cost curve, the contractor is inherently, although not always knowingly, motivated to move to a lower-cost, lower-reliability, higher-profit point of the curve. On the other hand, the government, which will be responsible for operation and support costs for this equipment, would, at any point on the curve, prefer to move to a region of higher reliability. Therefore, at any point on the "Initial Cost" curve, the goals of the contractor and the government are different and in conflict.

For the RIW procurement, the axes in Figure 4 are again reliability and cost, and it is assumed that the equipment has the same minimum acceptable reliability specification. In this case, however, the contractor is bidding not only on the cost to produce the equipment but also on the cost to repair the equipment for some fixed period of time. For this reason, the value of the fixed price contract is shown at a higher level than without RIW.

Figure 5 has the same "Initial Cost" curve previously considered. However, under RIW, the contractor now incurs the additional support cost, as shown by the dashed line. The "Total Contractor Cost" is depicted by the U-shaped curve in the chart. Again, the contractor's profit is represented by the distance between the horizontal contract-price line and the U-shaped curve. The contractor's maximum profit occurs at the bottom of the U-shaped curve. Thus when he is either to the left or to the right of the bottom, he will attempt to move toward the optimal point. However, when circumstances indicate being to the left of this point, the government will also try to increase the reliability of the equipment. Therefore, in this situation, both the government and the contractor goals are the same. This situation is quite different from that observed earlier.

Under RIW, the contractor no longer seeks the lowest acceptable reliability (MTBF). His interest in reliability improvement is maintained after production since he can still make changes that can reduce his support costs and improve his profits while increasing reliability. The contractor thus achieves maximum profit by controlling and making appropriate trade-offs between acquisition costs and operation and support costs, and the government can realize improved reliability and reduced life-cycle costs.

THE F-16 RIW PROGRAM

One of the most recent DoD programs selected for RIW is the F-16 multinational fighter program. The RIW activities associated with this U.S. Air Force aircraft represent the most complex and ambitious undertaking to date, with nine major avionics black boxes covered under the RIW provisions. These provisions were initially established in late 1974 when two prime contractors, General Dynamics Corporation and Northrop Corporation, were competing for the full-scale development (FSD) and production contract. The market for this development aircraft included not only the U.S. Air Force, but a consortium of the governments of Belgium, The Netherlands, Denmark, and Norway. These four countries were interested in sharing in the production of the aircraft to replace their aging fleet and were also interested in several European aircraft that were competing with two U.S. entries -- the YF-16 and the YF-17.

The competition between the U.S. prime contractors and the competition from the European aircraft were two important factors that led to the decision to include RIW provisions in the FSD and production contract. These provisions committed the winning development contractor to meet long-term obligations for production equipment. When such provisions are contracted for in production, the contractor risks are greater than in "non-warranty" procurements. When they are contracted for prior to full-scale development, the risks are naturally magnified.

The U.S. Government's decision to include RIW in the original contract was a difficult one since no direct experience base existed at the time. The following were two major considerations in the decision process:

- RIW controls would be positive factors in the Europeans' decision to select the U.S. Air Force aircraft.
- The prices for RIW would be obtained in a competitive environment.

In January 1975 the contract for full-scale development and production was awarded to General Dynamics (GD). The European consortium selected the same aircraft, and the United States and its four European partners embarked on a highly complex aircraft production program.

The initial RIW coverage included only U.S. Air Force aircraft (301 planes covered for four years or 300,000 flying hours), with an obligation on the part of the prime contractor to "negotiate with the consortium to extend warranty coverage" consistent with the U.S. Air Force RIW. The European Participating Governments (EPG) decided to undertake the RIW and, in the multinational partnership spirit of the F-16 program, it was decided to renegotiate the RIW during the development phase as a separate contract to include both U.S. and EPG aircraft. Contract F33657-77-0062 was formalized and signed in February 1977, about two years after the initial contract, and is currently being definitized.

HIGHLIGHTS OF THE F-16 RIW PROVISIONS

Nine of the F-16's avionics black boxes (LRUs) are covered under the RIW, as shown in Table 1. The radar antenna is warranted at the "module" level for the entire period of warranty. This module-level warranty permits returning the plug-in modules of the antennas for repair rather than the entire antenna. Six other LRUs begin their warranty at the LRU level and then transition to the module level later in the program. Two LRUs remain at the LRU level of warranty throughout the entire period. These two LRUs have, in addition to the RIW, a guaranteed MTBF.

The warranty applies to all sets of these LRUs installed in the first 250 U.S. Air Force production aircraft and the first 192 EPG production aircraft delivered. The warranty also applies to spares procured to support the first 250 U.S. Air Force and 192 EPG aircraft. Other characteristics of the F-16 contract are presented in Table 2.

MULTINATIONAL LOGISTICS

The RIW program for the F-16 calls for returning a suspected LRU failure to the contractor for repair. Following repair, the contractor places the LRU in a secure storage area that serves as a central supply facility for resupplying the base stock of the using activities. If the secure storage serves as a central supply for the combined rather than the individual needs of all five partner nations, significant saving in spares can be realized. The consortium has agreed to this concept. Spares of each unit type for stocking the bonded storeroom are to be purchased on the basis of the combined demand of the five user countries. There is to be no specific identification of hardware to owner country, since a true sharing of assets is envisioned.

To illustrate the savings that can be realized by commingling spares, we shall consider the warranted radar transmitter unit of the F-16. This unit has an MTBF guarantee, so that maintenance will remain at the LRU level throughout the four-year warranty period. Tables 3 and 4 present the illustrative data used in a simplified analysis to compare separate and commingled sparing at the secure storeroom. As shown in Table 4, a reduction from 67 to 49 in total bonded store spares can be achieved by commingling. A spare radar transmitter is estimated to cost approximately $90,000, yielding the saving shown for each country in the last column, for a total of $1.62 million. Although the relative savings for the units that would convert to a module-level warranty is not expected to be as large, the saving in the cost of bonded store spares attributable to module commingling may still be quite substantial.

Commingling also offers the advantage of providing a sharing atmosphere. If a participating country has an unusual problem, it may receive technical or material assistance from one of the other partner nations, so that all will eventually benefit. Similarly, if a country develops an improved procedure, e.g., in the base fault-verification process, it is to that country's advantage under the commingling concept to disseminate the information to the other participants as quickly as possible. While it is difficult to place a numerical value on these mutual-aid possibilities, such a concept is in keeping with the motivation behind the EPG participation in the F-16 program.

Notwithstanding these advantages, complexities exist in managing commingled spares. For example, one country may use more than its "fair share" because of (1) a higher-than-planned flying-hour schedule, (2) a tendency to stockpile, and (3) excessive transportation times.

The management and distribution of F-16 spares are the responsibility of the Air Logistic Center at Hill Air Force Base in Ogden, Utah. As long as all the user nations remain within their planned values of sparing parameters, the Item Manager will issue spares from

the secure storeroom on a first-come, first-served basis. If a partner consistently exceeds its "fair share" use of spares to the detriment of the other partners, the Item Manager may have to delay the release of spares to that partner until the other requisitions can be satisfied.

In general, the potential disadvantages of commingling are not unlike those of any other partnership venture. With NATO having a nearly thirty-year history of successful partnership, we do not believe that commingling will represent a major problem. We also do not believe that the form of this commingling effort is particularly new for NATO countries. F-4 aircraft system assets, for example, are being shared today among NATO nations. The RIW aspects of the F-16 program and the more comprehensive logistics structure it requires are the new features introduced into the sharing equation.

MULTINATIONAL COMMUNICATIONS

Two types of data must be communicated among the various F-16 participants: (1) supply and accounting data and (2) maintenance and utilization data. Supply and accounting data include the transactions necessary to report failures, to requisition replacements, and to maintain stock-balance records at various sites. Maintenance and utilization data include records of maintenance actions, parts usage, flying hours, etc. The primary means of communicating these two classes of data for the F-16 RIW program will be the established USAF AUTODIN system. The EPG interface with this system, and thus with the warranty communications process, is a switching center in Camp Newamsterdam, The Netherlands.

Figure 6 shows the principals in the communications network for RIW. The following characteristics are illustrated in the figure:

- All four U.S. manufacturers of warranted equipment have AUTODIN terminals that tie into the AUTODIN network.
- The Item Manager at the Ogden Air Logistics Center, all U.S. Air Force bases with F-16s, and General Dynamics also have terminals.
- The four European partners have commercial Telex or TWX facilities that tie into the AUTODIN network through a B-3500 switching center in Camp Newamsterdam.
- Although the Marconi-Elliott facility will repair warranted items from the European aircraft and will serve as a central supply for the Europeans, the Item Manager will issue all release orders to the Marconi-Elliott Atlanta, Georgia division. The internal company "bookkeeping" traffic will be their responsibility.

Since the four major contractor plants will be the depot repair sites for the warranted LRUs, placement of the AUTODIN terminals at these plants will permit the U.S. Air Force to handle requisitions and supply-accounting data in the same way as for a military repair depot. For the most part, all asset-balance accounting and bookkeeping information is generated and promulgated through an automatic address system. Since the European countries do not have AUTODIN terminals, they cannot conveniently requisition an asset and automatically satisfy all the necessary addressees for accounting and information purposes. The procedure to be used by the European air bases is to requisition an asset by sending a TWX or Telex to the Item Manager. The Item Manager then manually generates a requisition for the European air base, and the automated process takes over, notifying all appropriate addressees of the transaction.

Under the F-16 RIW contract, all five governments must furnish certain maintenance and utilization data to the prime contractor, General Dynamics. These data, which include aircraft flying hours, are generated for the U.S. Air Force through the use of standard maintenance and flying-hour reporting forms. Data from these forms are keypunched in a prescribed format and are then automatically used to develop certain statistical reports and trends.

Each European partner has its own form to record similar maintenance and utilization information. To permit each country to continue to use its own standard forms and formats, a data processor has been developed that ties each country's Telex terminals to the B-3500 computer in Camp Newamsterdam. This front-end processor converts each country's format into a standard format. The reformatted data are then transmitted, via AUTODIN, into the U.S. Air Force data system. All of the European data are identified by country, base, aircraft, etc., so that separate maintenance and usage analyses may be performed for any European or United States base.

CONCLUSION

The F-16 aircraft has and will continue for some time to test our abilities in international cooperation, financing, management, procurement, and logistics. Of significant interest to all NATO countries will be the success with which the five enterprising F-16 partner nations meet in the formidable task of achieving higher reliabilities through RIW.

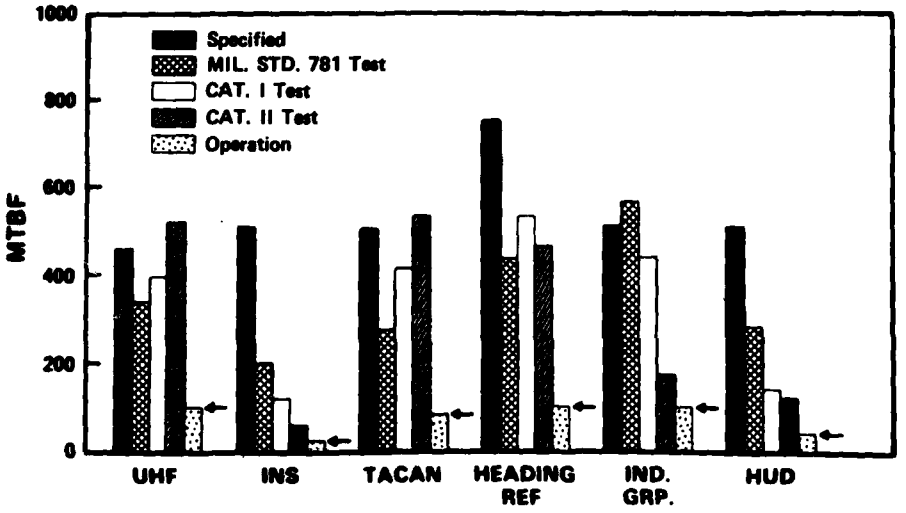


Figure 1. FIELD RELIABILITY PROBLEMS

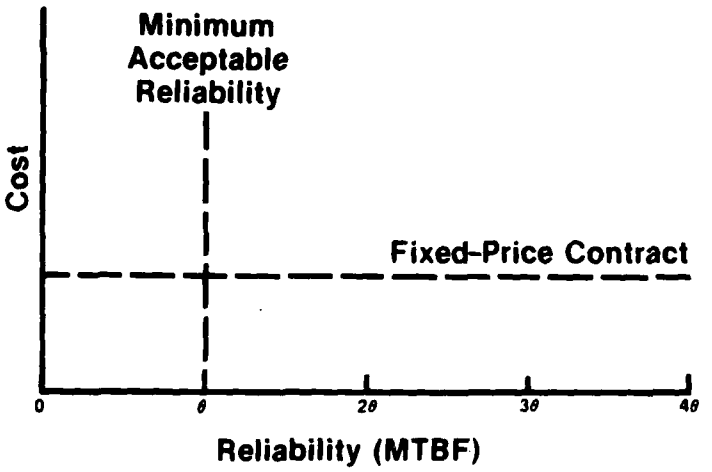


Figure 2. CONDITIONS AT CONTRACT INITIATION

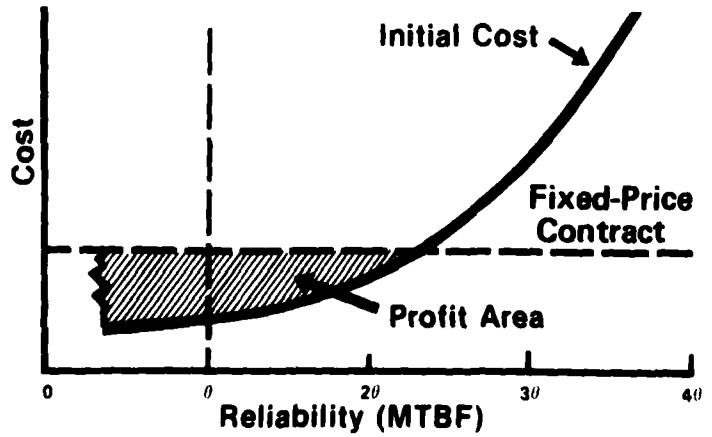


Figure 3. EFFECT OF RELIABILITY ON INITIAL COST OF EQUIPMENT

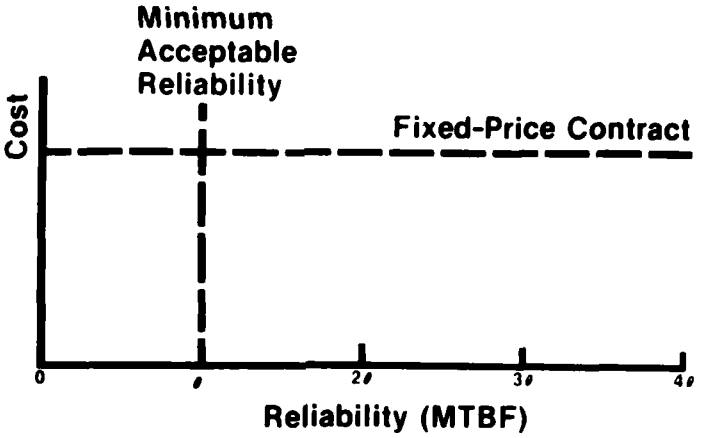


Figure 4. CONDITIONS AT INITIATION OF A CONTRACT WITH RIW

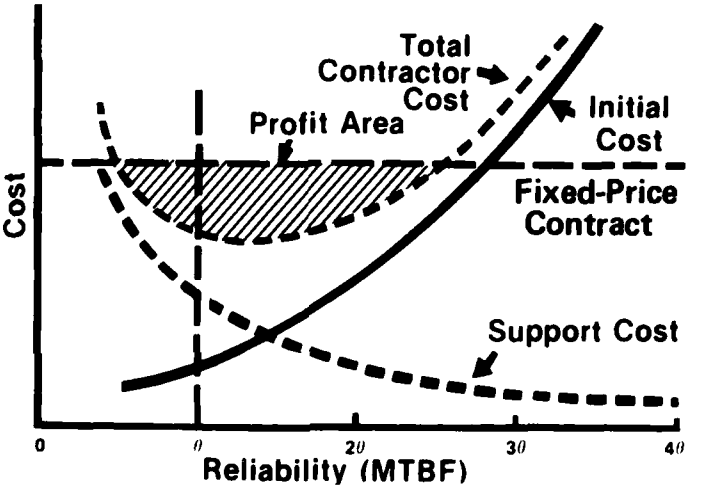


Figure 5. EFFECT OF RELIABILITY ON INITIAL AND SUPPORT COSTS

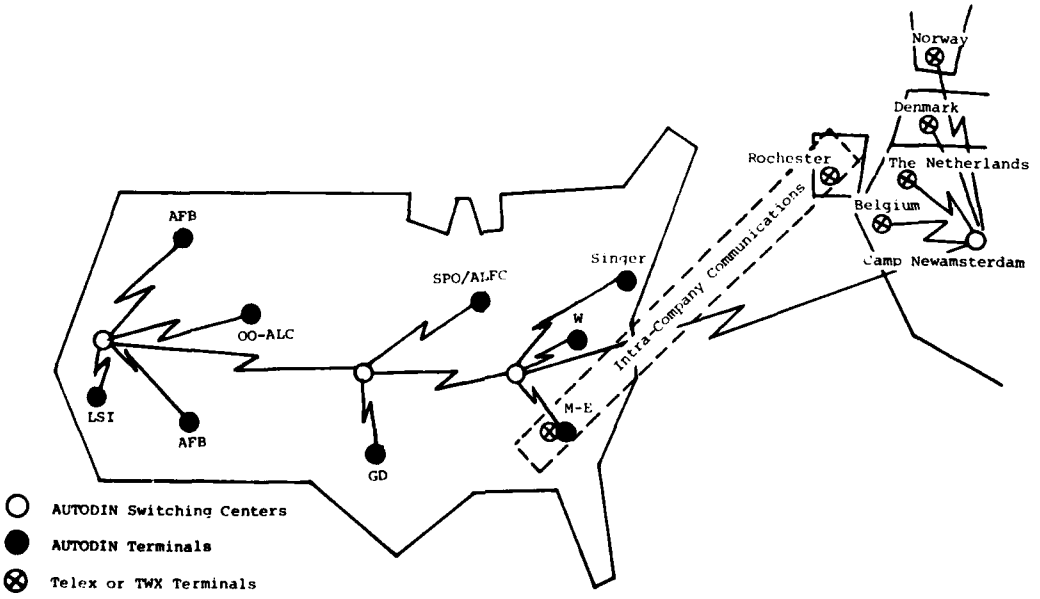


Figure 6. COMMUNICATIONS NETWORK AND FACILITIES

Table 1. F-16 LRU-WARRANTED EQUIPMENT

LRU-Warranted Equipment			Warranty Level/Period			
WUC	Nomenclature	Manufacturer	1979	1980	1981	1982
14AAO	Flight Control Computer	Lear-Siegler Industries	LRU-Level Warranty			
74AAO	Radar Antenna	Westinghouse	Module-Level Warranty			
74ABO	Radar Low Power RF	Westinghouse	LRU-Level Warranty			
74ACO	Radar Transmitter	Westinghouse	LRU-Level Warranty plus MTBF Guarantee			
74ADO	Radar Digital Signal Processor	Westinghouse	LRU-Level Warranty			
74AFO	Radar Computer	Westinghouse	LRU-Level Warranty			
74BAO	HUD (Head Up Display)	Marconi-Elliott	LRU-Level Warranty			
74BCO	HUD Electronics	Marconi-Elliott	LRU-Level Warranty plus MTBF Guarantee			
74DAO	INU (Inertial Navigation Unit)	Singer-Kearfott	LRU-Level Warranty			

Table 2. CHARACTERISTICS OF THE F-16 RIW CONTRACT

Characteristic	Description
Units Covered	9 different LRUs
Aircraft	442 F-16 aircraft
Coverage Period	4 years or 300,000 flying hours
Contract Time	Prior to full-scale production
Air Force Logistics Manager	Ogden ALC
Countries	United States, Belgium, Denmark Norway, and The Netherlands; other countries possible on separate Foreign Military Sales (FMS) contracts
Contractor	General Dynamics (prime) with 4 subcontractors
Price	Range from 2 percent to 6 percent per year.
Coverage Level	LRU (2 units), SRU (1 unit), LRU transitioning to SRU (6 units)
MTBF Guarantee	Yes, on 2 units
"Retest OK" Provision	Under negotiation
Low-Usage Adjustment	Yes
Turnaround Time Requirement	22-day average
User Capability for SRU Fault Location at Base	Planned

Table 3. RADAR TRANSMITTER SPARES DATA - BONDED STOREROOM	
Mean Time Between Demands (Hours)	250
Pipeline Days	
U.S. Air Force	25.0
EPG	30.0
Average	27.2
Operating Hours per Day	0.72
Spares-Sufficiency Level	0.99
Radar Transmitter Cost	$90,000

Table 4. SPARES COST ADVANTAGES OF COMMINGLING THE RADAR TRANSMITTER

Country	Quantity of Aircraft	Quantity of Spares		Dollar Savings Attributable to Commingling
		Separate	Commingled*	
USA	250	29	21	$ 720,000
Belgium	58	11	8	270,000
Denmark	38	8	6	180,000
Netherlands	60	11	8	270,000
Norway	36	8	6	180,000
Total		67	49	$1,620,000

*The 49 total spares required under the commingling concept may be allocated among the participants in more ways than one. The allocation shown in the table reflects an equal percentage of savings over the separate purchase of spares.

DISCUSSION

A.Sukert, US

Does the RIW for the computers, such as the Fire Control Computer, include the corresponding Operational Flight Programs? Or is the RIW just for the hardware only?

Author's Reply

For the F-16 Flight Control Computer, the RIW applies to the hardware only. I would categorically recommend against the application of an RIW concept to "software reliability" until the definition of that term has been thoroughly scrubbed down and is generally accepted by the industry.

F.Wishart, UK

Would you please give some indication of the percentage of contract value which is accountable to the RIW with and without guaranteed MTBF?

Author's Reply

The RIW prices quoted by General Dynamics were for 301 USAF aircraft. Those prices for RIW *and* MTBF guarantee ranged from about 2% to 7% per year of the total contract value of the LRUs. (Not the total value of the aircraft production contract!) After the competition between GD and Northrop was over, the European countries were negotiated into the RIW.

The negotiations resulted in: (1) an increase in RIW price, (2) the inclusion of 192 European F-16s and (3) a reduction from 301 to 250 USAF F-16s.

C.J.P.Haynes, UK

You said that if one of the 5 nations consumes more than its fair share of the combined spares pool then the item manager would delay despatch of spares to that nation. In the case where differences in consumption are caused by differences in defect arising rates, which are quite likely to occur, would delayed despatch still be invoked?

Author's Reply

No. There is no reported intent on the part of the item managers to delay release of a serviceable asset for any reason other than those listed. If differences in MTBF are seen between user nations, however, it would surely be a matter of great interest and investigation.

APPENDIX A

LIST OF PARTICIPANTS

ANDREWS, A. Grp Capt.	RAF, Swanton Murley, Dereham, Norfolk, UK
ATTULY, R. Mr	Aerospatiale, Subdivision Systemes, B.P.96, 78130, Les Mureaux, France
BALABAN, H.S. Dr	ARINC Research Corporation, 2551 Riva Road, Annapolis, MD 21401, USA
BARRE, Mr l'Ing. en Chef Panel Member	STTA, 129, rue de la Convention, 75996 Paris, France
BASMAISON, J.N. Dr	Thomson CSF, 52, rue Guynemer, 92131, Issy-les-Moulineaux, France
BERTIN, C. Mr	Aerospatiale, Subdivision Systemes, B.P.96, 78130, Les Mureaux, France
BINKHORST, P.F. Mr	Fokker VFW, Dept. CB, E Lab, P.O.Box 7600, Schiphol-East, Holland
BOARDMAN, K.W. Mr	Marconi Avionics Ltd., Control Quality Department, Airport Works, Rochester, Kent, ME1 2XX, UK
BRANITSKI, K. Dipl.Ing.	Standard Elektrik Lorenz AG, Dept. CNS/KSO, P.O.Box 400749, 7000 Stuttgart 40, Germany
BRAULT, Y. Mr Panel Member	Thomson CSF, Division Equipements Avioniques et Spatiaux, 178 Bld Gabriel Peri, 92240 Malakoff, France
BRUN, J. Mr	Thomson CSF, D.T.E. 38, rue Vauthier, 92100 Boulogne, France
CHABIN, J.C. Mr	S.A.Crouzet, B.P. 1014, Valence, France
CHARLOT, Mr	Thomson CSF, 178 Bld Gabriel Peri, 92240 Malakoff, France
CIERI, G. Mr	Contraves Italiana, Spa. Via Tiburtina, 965-00156, Roma, Italy
CIVIDINO, B.C. Mr	LMT, 46, Quai Alphonse Le Gallo, 92103 Boulogne, France
COPAGE, C.M. Mr	Smith Industries Ltd., Bishops Cleeve, Cheltenham, Glos, GL52 4SF, UK
DOVE, B.L. Mr Panel Member	NASA Langley Research Center, Mail Stop 477, Hampton, VA 23665, USA
EHRENBERGER, W. Dr	Gesellschaft fur Reaktorsicherheit, Forschungsgelande D8046 Garching, Germany
ELSBERGEN, L. Mr	Messerschmitt-Bolkow Blohm GmbH, Unternehmensbereich, Flugzeuge FF38, Postfach 801160, 8000 Munchen 80, Germany
ERDIM, F. Miss	Testas, Ataturk Blv. 227, Kavaklidere, Ankara, Turkey
FAURY, Mr	LMT, 46, Quai Alphonse Le Gallo, 92103, Boulogne Billancourt, France
GARNIER, M. Mr	STTA, rue de la Convention, 75015, Paris, France
GIRAUD, M. Mr	EMD, Quai Carnot, 92214 St Cloud, France
GOLDBERG, J. Mr	Director, Computer Sciences Laboratory, Stanford Research Institute, International, 333 Ravenswood Ave., Menlo Park, CAL.95025, USA
GREGORY, D.M. Mr	Avionics Department, BAAG, Manchester Division, Woodford, Cheshire, SK7 1QR, UK
GROSS, H. Dr	Messerschmitt Bolkow Blohm, GmbH, UF Systemunterstutzung FE 07, Postfach 80 11 60, D 8000 Munchen 80, Germany
HARRIS, D.J. Mr	EASAMS Ltd., 35/41 Park Street, Camberley, Surrey, UK
HARRISON, G.T. Jr. Mr	ARINC Research Corp., 2551 Riva Road, Annapolis, Maryland, 21401, USA
HAYNES, C.J.P. Squadron Leader	Science 1 (RAF) MOD Main Building, Whitehall, London SW1A 2HB, UK
HEINER, G. Dipl. Ing.	AEG Telefunken, Abt N 14/V6, Postfach 1730, D7900 Ulm, Germany
HILTON, R.G. Mr	Dowty Boulton Paul Limited, Pendeford Lane, Wolverhampton, WV9 5EW, UK
INCE, F. Dr	Marmara Research Institute, Electronics, PK 21, GEBZE, Turkey
IZGI, A. Mr	Selanik Cd. 46/21, P.O.Box 61, Ankara, Turkey
JACOBSEN, M. Mr Panel Member	AEG Telefunken, N14 V3, Postfach 1730, D 7900 Ulm, Germany

KANSU, Y. Mr FMP Panel Member	Director of the Production and Investment, TUSAS, Ataturk Bulvari, 227, Ankara, Turkey
KAYA, D. Col. National Coordinator	Ministry of National Defence, Dept. of Research and Development, (ARGE), Ankara, Turkey
KAZOKOGLU, A. Mr Host Coordinator	TUSAS, Ataturk Bulvari, 227, Ankara, Turkey
KLINE, M.B. Prof.	Code 54 Kx, Naval Post Graduate School, Monterey, CAL.93940, USA
KRAUSE, C. Mr	ESG-FEG, Vogelweideplatz 9, 8000 Munchen, Germany
LINDEN, K. Dipl. Ing.	BWB-NL5, Landshuter Allee 162a, D 8000 Munchen 19, Germany
LORIE, P. Mr	KLM Engineering and Maintenance Div. (SPL/CO) P.O.Box 7700, Schiphol Airport, Holland
MARIANI, S, Dr	SNIA Viscosa GSR, C.so G. Garibaldi, 00034 Cobleferro, Rome, Italy
MAUROMATI, L. Lt.	Ankara Depot, TAF, Ankara, Turkey
MIGNEAULT, G.E. Mr	NASA Langley Research Center, Mail Stop 477, Hampton, VA 23665, USA
MILNER, J.G. Mr	RSRE, Malvern, Worcs, UK
MOLTER, H.H. Dr	Messerschmitt Bolkow Blohm, GmbH, (Abt AQ 11) Postfach 80 11 60, D 8000 Munchen, 80, Germany
NARESKY, J.J. Mr	RADC/RB GRIFFISS, AFB, N.Y.13441, USA
NAUDTS, B. Lt.	GDAL4 Quartier Reine Elisabeth, rue d'Evere, B 1140, Brussels, Belgium
NERI, A. Mr	Contraves Italiana, Spa, Via Tiburtina, 965, 00156, Roma, Italy
O'CONNOR, P.D.T. Mr	British Aerospace Dynamics (PB192) Six Hills Way, Stevenage SG 12 DA, UK
PERBET, Mr	SFENA, B.P.59, 78140 Velizy, France
PEYRET, B.G. Prof.	SFIM, 13, Avenue Ramolfo Garnier 91301, Massy, France
PFAFFENBERGER, O. Mr	Messerschmitt Bolkow Blohm GmbH (FE38), Postfach 80 11 60, D 8000 Munchen 80, Germany
PHALLER, L.J. Mr	Westinghouse Electric Corporation, Defense Electronic Systems Center, Box 746, Baltimore, MD 21203, USA
PLANTARD, J.P. Mr	Thomson CSF, SCTF, B.P.N.10, 91401, Orsay, France
RAMAZAN, O. Capt.	TAF, Ankara, Turkey
RAZON, A. Lt.	Ankara Depot, TAF, Ankara, Turkey
REGULINSKY, T.L. Dr	Air Force Institute of Technology, Dept. of Electrical Engineering, Wright-Patterson AFB, Ohio, 45433, USA
RIESENER, Mr	Erprobungstelle 61, AVWG, Flugplatz, D 8072, Manching, Germany
ROBERTSON, J.C. Mr	British Aerospace Aircraft Group, Warton Division, Warton Aerodrome, Preston, Lancs, PR4 1AX, UK
RUHI, D. Mr	ASELSAN, Ankara, Turkey
SAG, H. Mr	Testas, Ataturk Bulvari 227, Kavaklidere, Ankara, Turkey
SCHRECK, K.J. Mr	ESG Elektronik System GmbH, Postfach 800569, 8000 Munchen, Germany
STEINEBRUNNER, R. Mr	Fa, LITEF, Lorracher Str 1 8, 7800 Freiburg i. Br, Germany
SUKERT, A. Mr	RADC/ISIS, Griffiss AFB, N.Y. 13441, USA
STRINGER, F.S. Mr Panel Member	R177 Building, Flight Systems Dept. Royal Aircraft Establishment, Farnborough, UK
TIMMERS, H.A.T. ir. Panel Chairman	NLR, Anthony Fokkerweg 2, 1059 CM, Amsterdam, Holland
TOKMAK, First Lt.	Turkish Navy, Ankara, Turkey
TUZUNALP, O. Mr	METU, Dept. of Physics, Ankara, Turkey
VAGNARELLI, F. Lt.Col. Panel Member	Aeronautica Militare, P.Adenauer 3, 00144, ROMA/EUR, Italy
VOGEL, M. Dr Panel Deputy Chairman	DFVLR, NE-HF, D 8031 Oberpfaffenhofen, Germany
VOLES, R. Mr Panel Member	EMI Electronics Ltd., 135 Blyth Road, Hayes, Middx, UK
WATTS, M.W. Mr	Air Eng. 33A MOD Main Building, Room 2249, Whitehall, London, SW1, UK
WEIGEL, P. Mr	Bundesamt fur Wehrtechnik und Beschaffung, GPIII 4 Postfach 7360, D 5400 Koblenz, Germany

WEIHE, A. Mr	Messerschmitt Bolkow Blohm, GmbH, Postfach 80 11 60, D 8000 Munchen 80, Germany
WHITE, A.P. Mr	EASAMS Ltd., Camberley, Surrey, UK
WISHART, F.	British Aerospace Group, Warton Div. Warton Aerodrome, Preston, Lancs, PR4 1AX, UK
WOODS, W.M. Dean	Naval Post Graduate School, Office of Continuing Education, Code 500, Monterey, CAL 93940, USA
WUST, P. Mr	Bodenseewerk – Geratetechnik, Postfach 1120, 7770, Ueberlingen, Germany

Interpreters

H.AKBELEN, F.LAMON, M.VRYDAGH

AGARD

Lt. Colonel J.B. CATILLER, Executive

M.Tessier, Secretary

REPORT DOCUMENTATION PAGE			
1. Recipient's Reference	2. Originator's Reference	3. Further Reference	4. Security Classification of Document
	AGARD-CP-261	ISBN 92-835-0254-X	UNCLASSIFIED

5. Originator	Advisory Group for Aerospace Research and Development North Atlantic Treaty Organization 7 rue Ancelle, 92200 Neuilly sur Seine, France
6. Title	AVIONICS RELIABILITY, ITS TECHNIQUES AND RELATED DISCIPLINES
7. Presented at	a Meeting of the Avionics Panel held in Ankara, Turkey, 9–13 April 1979.

8. Author(s)/Editor(s)		9. Date
M.C.Jacobsen		October 1979
10. Author's/Editor's Address	AEG-Telefunken Elisabethenstrasse 3, D-7900 Ulm Fed. Rep. of Germany	11. Pages 560
12. Distribution Statement	This document is distributed in accordance with AGARD policies and regulations, which are outlined on the Outside Back Covers of all AGARD publications.	

13. Keywords/Descriptors

Avionics	Design	Airborne computers
Reliability (electronics)	Computer programs	
Maintenance	Logistics support	

14. Abstract

These Proceedings include the papers and discussion presented at the AGARD Symposium on Avionics Reliability, Its Techniques and Related Disciplines sponsored by the Avionics Panel in Ankara, Turkey, in April of 1979. The meeting provided a state of the art review of topics related to reliability and logistics in avionics systems. The 44 papers were distributed as follows. Seven on general concepts, ten on reliability/availability requirements and demonstration, eleven on reliability and maintainability practices and effects in avionics design, development and production, eight on software reliability, and eight on logistics support aspects.

AGARD Conference Proceedings No.261 Advisory Group for Aerospace Research and Development, NATO AVIONICS RELIABILITY, ITS TECHNIQUES AND RELATED DISCIPLINES Edited by M.C.Jacobsen Published October 1979 560 pages These Proceedings include the papers and discussion presented at the AGARD Symposium on Avionics Reliability, Its Techniques and Related Disciplines sponsored by the Avionics Panel in Ankara, Turkey, in April of 1979. The meeting provided a state of the art review of topics related to reliability and logistics in avionics systems. The 44 papers were distributed as P.T.O.	AGARD-CP-261 Avionics Reliability (electronics) Maintenance Design Computer programs Logistics support Airborne computers	AGARD Conference Proceedings No.261 Advisory Group for Aerospace Research and Development, NATO AVIONICS RELIABILITY, ITS TECHNIQUES AND RELATED DISCIPLINES Edited by M.C.Jacobsen Published October 1979 560 pages These Proceedings include the papers and discussion presented at the AGARD Symposium on Avionics Reliability, Its Techniques and Related Disciplines sponsored by the Avionics Panel in Ankara, Turkey, in April of 1979. The meeting provided a state of the art review of topics related to reliability and logistics in avionics systems. The 44 papers were distributed as P.T.O.	AGARD-CP-261 Avionics Reliability (electronics) Maintenance Design Computer programs Logistics support Airborne computers
AGARD Conference Proceedings No.261 Advisory Group for Aerospace Research and Development, NATO AVIONICS RELIABILITY, ITS TECHNIQUES AND RELATED DISCIPLINES Edited by M.C.Jacobsen Published October 1979 560 pages These Proceedings include the papers and discussion presented at the AGARD Symposium on Avionics Reliability, Its Techniques and Related Disciplines sponsored by the Avionics Panel in Ankara, Turkey, in April of 1979. The meeting provided a state of the art review of topics related to reliability and logistics in avionics systems. The 44 papers were distributed as P.T.O.	AGARD-CP-261 Avionics Reliability (electronics) Maintenance Design Computer programs Logistics support Airborne computers	AGARD Conference Proceedings No.261 Advisory Group for Aerospace Research and Development, NATO AVIONICS RELIABILITY, ITS TECHNIQUES AND RELATED DISCIPLINES Edited by M.C.Jacobsen Published October 1979 560 pages These Proceedings include the papers and discussion presented at the AGARD Symposium on Avionics Reliability, Its Techniques and Related Disciplines sponsored by the Avionics Panel in Ankara, Turkey, in April of 1979. The meeting provided a state of the art review of topics related to reliability and logistics in avionics systems. The 44 papers were distributed as P.T.O.	AGARD-CP-261 Avionics Reliability (electronics) Maintenance Design Computer programs Logistics support Airborne computers

follows. Seven on general concepts, ten on reliability/availability requirements and demonstration, eleven on reliability and maintainability practices and effects in avionics design, development and production, eight on software reliability, and eight on logistics support aspects.

ISBN 92-835-0254-X

follows. Seven on general concepts, ten on reliability/availability requirements and demonstration, eleven on reliability and maintainability practices and effects in avionics design, development and production, eight on software reliability, and eight on logistics support aspects.

ISBN 92-835-0254-X

follows. Seven on general concepts, ten on reliability/availability requirements and demonstration, eleven on reliability and maintainability practices and effects in avionics design, development and production, eight on software reliability, and eight on logistics support aspects.

ISBN 92-835-0254-X

follows. Seven on general concepts, ten on reliability/availability requirements and demonstration, eleven on reliability and maintainability practices and effects in avionics design, development and production, eight on software reliability, and eight on logistics support aspects.

ISBN 92-835-0254-X