# Software Reliability Modelling and Evaluation via Markov and Semi-Markov Process

by

Shakil Akhtar

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

**MASTER OF SCIENCE**

In

**ELECTRICAL ENGINEERING**

September, 1982

# SOFTWARE RELIABILITY MODELLING AND EVALUATION VIA MARKOV AND SEMI-MARKOV PROCESSES

BY

SHAKIL AKHTAR

SEPTEMBER 1982

# SOFTWARE RELIABILITY MODELLING AND EVALUATION VIA MARKOV AND SEMI-MARKOV PROCESSES

BY

SHAKIL AKHTAR

THESIS

PRESENTED TO THE FACULTY OF THE COLLEGE OF GRADUATE STUDIES

UNIVERSITY OF PETROLEUM AND MINERALS

DHAHRAN - SAUDI ARABIA

IN PARTIAL FULFILMENT OF THE REQUIREMENT

FOR THE DEGREE OF

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

SEPTEMBER 1982

UNIVERSITY OF PETROLEUM AND MINERALS
DHAHRAN, SAUDI ARABIA

This thesis, written by

MR. SHAKIL AKHTAR

under the direction of his Thesis Committee, and approved by all its members, has been presented to and accepted by the Dean, College of Graduate Studies, in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

جامعة البترول والمعادن
كلية الدراسات العليا
COLLEGE OF GRADUATE STUDIES
UNIVERSITY OF PETROLEUM & MINERALS

[illegible]
Dean, College of Graduate Studies

Date: [illegible]

[illegible] 8/9/1982
Department Chairman

THESIS COMMITTEE

Mansoor Alam
Chairman, Dr. Mansoor Alam

[illegible]
Member, Dr. Gunter D. Achilles

Ahmed Milyani
Member, Dr. Ahmed M. Milyani

*THIS THESIS IS DEDICATED TO MY PARENTS AND TEACHERS FOR THEIR LOVE & GUIDANCE*

## ACKNOWLEDGEMENT

I express my sincere appreciation to my advisor Dr. Mansoor Alam for his constant support and guidance, and the other members of the thesis committee Dr. Gunter D. Achilles and Dr. Ahmed M. Milyani.

Acknowledgement is also due to Dr. Fareed M. Zedan, Chairman of the Department of Electrical Engineering at the University of Petroleum & Minerals for the support of this research.

Finally I wish to thank Mr. Muhammad Khalid Butt for carefully typing the manuscript and Mr. Bakheet for making the drawings.

TABLE OF CONTENTS

## LIST OF FIGURES

## ABSTRACT

This thesis develops Markov and semi Markov models for quantitative evaluation of software reliability. Two measures of software reliability termed as operational reliability and operational availability have been introduced which provide better figures of merit than reliability. Quantitative evaluation of total expected cost and the cost due to failures in control transfer of a computer program are described and the overall failure process of the program is developed. Program examples are presented to illustrate the procedure presented in this thesis. Some possible future extensions of the work are also discussed.

# 1. INTRODUCTION AND LITERATURE SURVEY

Software reliability has been the subject of considerable investigation since the late sixties [1-11], as it was observed that highly financed missions could fail due to a minor error in a software. Various definitions of software reliability have been proposed in the literature but the most promising definition seems to be the following.

"It 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" [12].

Therefore software reliability is a function of the impact that the errors have on the system users; it is not necessarily a function of the size of error. A small error such as a missing comma can cause a total mission failure while a large bug can have no effect on the user, provided the user does not demand the defective part.

There are two schools of thought (Bayesian and non-Bayesian) currently found in literature. The first group thinks strongly that the concepts of hardware reliability should be applied with a great care to software reliability [13,19] while the other group believes that there is a very little difference of the concepts between hardware and software [14-18].

As the subject of software reliability research is relatively modern, the differences of opinion often concern the modelling aspects but there is a general agreement on the basic definition of software reliability. Littlewood does not agree [13,19] with the earlier models presented by Shooman & Jelinski, Moranda [14,17] which are perhaps one of the first complete models presented in this field which depends upon the classical reliability theory [40]. Littlewood has extensively contributed in this field [13,19-23] but his approach remains theoretical without providing a practical example. His suggestions are based on the following [13]:

1) Do not use MTTF, MTBF for software, unless certain that they exist, there can be better measured than MTTF.

2) Do not stop at a reliability analysis; try to model lifetime utility (or cost) of programs.

3) Try to devote effort to structural models.

4) Structure should be of a kind appropriate to software, e.g. top down, modular.

5) Use a Bayesian approach and do not be afraid to be subjective.

Shooman, Musa & Jelinski, Moranda are the main contributors of models based on classical reliability theory [14-18,24-29]. Shooman introduced a Markov model for software reliability [24] which opened a new dimension for modelling software for the reliability point of view. Most of his models produce reliability indices such as the number of remaining errors, reliability, availability & MTTF or MTBF. Musa & Jelinski, Moranda [11,14] follow a similar approach. In addition to these models there are some models presented by Shick, Wolverton [30]. Sukert has studied and tested these models extensively with the available field data [31-32] and has shown that the results produced by most of these models are very close to the expected behaviour.

Some authors like Robert, James, John & Larry [33-35] have worked over the combined hardware and software reliability models. Although this idea is new but has not attracted much attention.

Recent trend in software reliability modelling is to incorporate the user profile. User oriented reliability models which take into account user profile have been presented by many authors [21-22,36-38]. Cheung has presented such a model [36] and applied this to a real program having independent modules. He has shown how the program reliability is dependent upon the user profile but has ignored the important aspect of fault correction. An attempt has been made in this thesis to incorporate this aspect in software reliability modelling. Moreover, we have been able to handle non-exponential fault correction density function and have used the method of stages [40] to model such a density function. Recently, a semi Markov model has been developed by Littlewood [21] in which he has taken the costs of failure into account and has developed mathematical formulas for many important quantities, e.g. mean cost in the steady state. In this thesis, we have developed semi-Markov reliability models which are directly applicable to modular structures. The model has been used to produce operational reliability and availability as a function of time.

# 2. SOFTWARE CYCLE

In order to understand the operation of the software systems and to provide the necessary backdrop materials useful in the development of software reliability models, a brief description of the software cycle is presented in this chapter. The total software cycle includes the following phases:

(i) Design and coding.

(ii) Testing.

(iii) Maintenance.

The cost of maintenance and testing is approximately 75% of the total software cost, while design and coding takes only about 25% of the total cost.

## 2.1 DESIGN AND CODING

Design means: "to fashion according to the plan". Therefore, this phase covers software activities beginning with the establishment of requirements and objectives and ending with the writing of program statements, with distinct stages of design. These design stages are described by the block diagram of Fig. 2.1.

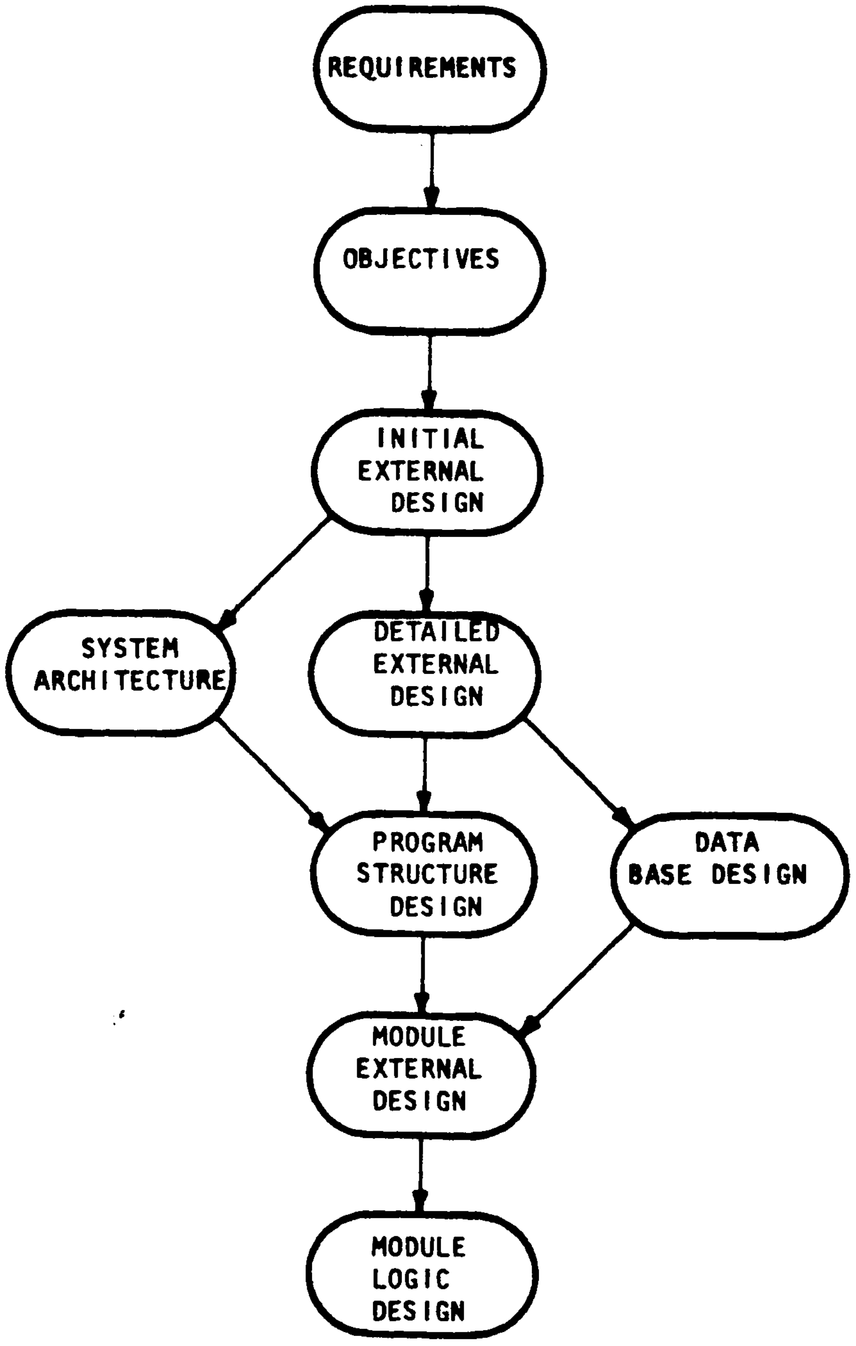

Figure 2.1. Design stages for a large project.

## 2.1.1 Requirements & Objectives

The first step produces a set of requirements, that is, a statement of what the user expects of the product. In fact software errors first creep into a product when requirements and objectives are established. The cause of most of these errors is a misunderstanding of the needs of the user. More errors originate when the requirements or objectives are translated into external specifications.

The purpose of software requirements is to establish the needs of the user for a particular software product.

The second process in software development is the development of objectives. Objectives are specific goals for the software product. Establishing software objectives is primarily a process of establishing tradeoffs.

A good set of software objectives defines the goals of a software product without implying any particular implementation and specifies how tradeoffs are to be made in later design processes. Two types of objectives must be established: product & project objectives.

Product objectives define the software goals from the users point of view. Among other things product objective must contain reliability objectives.

Project objectives are goals to be met during the development processes, goals that are not directly manifested in the final product.

### 2.1.2 External Design

External design is the process of describing the intended behavior of a product as it would be perceived by a human observer external to the product. Producing a complete and correct external specification is the most challenging problem in software development today.

For external design a valuable principle to follow is the idea of conceptual integrity [ 12,41]. Conceptual integrity is the harmony (or lack of harmony) among the external interfaces of the system. A system without conceptual integrity is a system with no underlying uniformity; such a system results in an overly complicated user interface and additional complexity in the software structure. The easiest way to achieve conceptual integrity is to produce an external design with lesser people. Depending upon the size of the project, one or two people should have the responsibility for the external design.

### 2.1.3 System Architecture

The system architecture process decomposes the system into a set of programs, subsystems or components and defines the interfaces among them. System architecture process is a necessary step in the design of systems but not in the design of programs. If the external specification describes a program, the system architecture step is unnecessary.

We can say that a software system represents a set of solutions to a set of distinct but related problems, and then rely on intuition to distinguish systems from programs, e.g. an operating system, an airline reservation system and a data base manager are examples of systems. A text formatting command in a time-sharing system & a compiler are examples of single programs and do not require the system architecture process.

### 2.1.4 Program Structure Design

The design of program structure includes the definition of all modules in the program, the hierarchical structure of the modules and the interfaces among the modules.

The traditional method of managing complexity is the idea of "divide and rule", often called "modularization". A design methodology called "composite design" is usually used for program structure design [12,42]. Composite design leads to a program structure of minimal complexity, which has proven to increase reliability, maintainability and adaptability. Composite design can be described by the following software properties.

### 2.1.4.1 Module independence

The primary way to make a program less complex is to decompose it into a large set of small, highly independent modules. This can be achieved by maximizing the relationship within each module and minimizing the relationship among modules.

### 2.1.4.2 Module strength

It is a measure of the relationship within a module. Determining the strength of a module involves analyzing the function or functions performed by the module. Among various kinds of module strength, functional strength is the best form of module strength. A functional strength module is a module that performs a single specific function such as "turn off valve y," or "execute EDIT command", etc.

### 2.1.4.3 Module coupling

Module coupling, a measure of the data relationship among modules, is concerned with both the mechanism used to pass data and the attributes of the data itself. The design goal is to define module interfaces so that all data passed between modules is in the form of explicit simple parameters.

### 2.1.4.4 Module size

Module size has a bearing on a program's independence , readability, and difficulty of testing (e.g. number of paths), one could satisfy the criteria of high strength and minimal module coupling by designing a program as one huge module, but it is unlikely that high independence would be achieved by doing so.

Module external design (Fig. 2.1), is the precise definition of all module interfaces. The next step, module logic design, is the design of the internal logic or procedure of each module in the system, including the expression of this logic as program statements.

Data base design is the process of defining any data structures external to the software, for example the design of file records or data base records.

However, in a real design effort, the picture of the flow of design processes is not quite simple as this. There is a considerable amount of feedback and overlap between processes. For instance, flaws may be discovered in the objectives during one of the external design steps, requiring one to back up momentarily to correct the flaw in the objectives [12].

## 2.2 TESTING & MAINTENANCE

Testing is the process of executing a program with the intention of finding errors. It is not done with a goal to show the absence of errors, rather it is done with a goal to show the presence of errors. The testing process is directly related to the design process. If some error is detected during the testing process, it will actually lead to the design process. The relationship between design and testing processes is shown in Fig. 2.2.

### 2.2.1 Module and Integration Testing

There are various types of module & integration testing described below:

#### 2.2.1.1 Bottom up testing

In this approach the terminal modules (the modules that call no other modules) are first tested in isolation. Then the modules having CALL to one of the already tested terminal modules are tested. The higher level modules are not tested in isolation; they are tested with the previously tested lower modules. This process is repeated until the top is reached. At this point module and integration testing for the program is complete.

#### 2.2.1.2 Top down testing

In this approach the program is merged and tested from the top module in the program structure. After this module is tested, the modules called by this module are merged one by one and the combination is tested.

When some lower module is called by a module under test some fixed outputs are returned. The tested module is thus simulated to the functions of missing modules by the help of "stub modules".

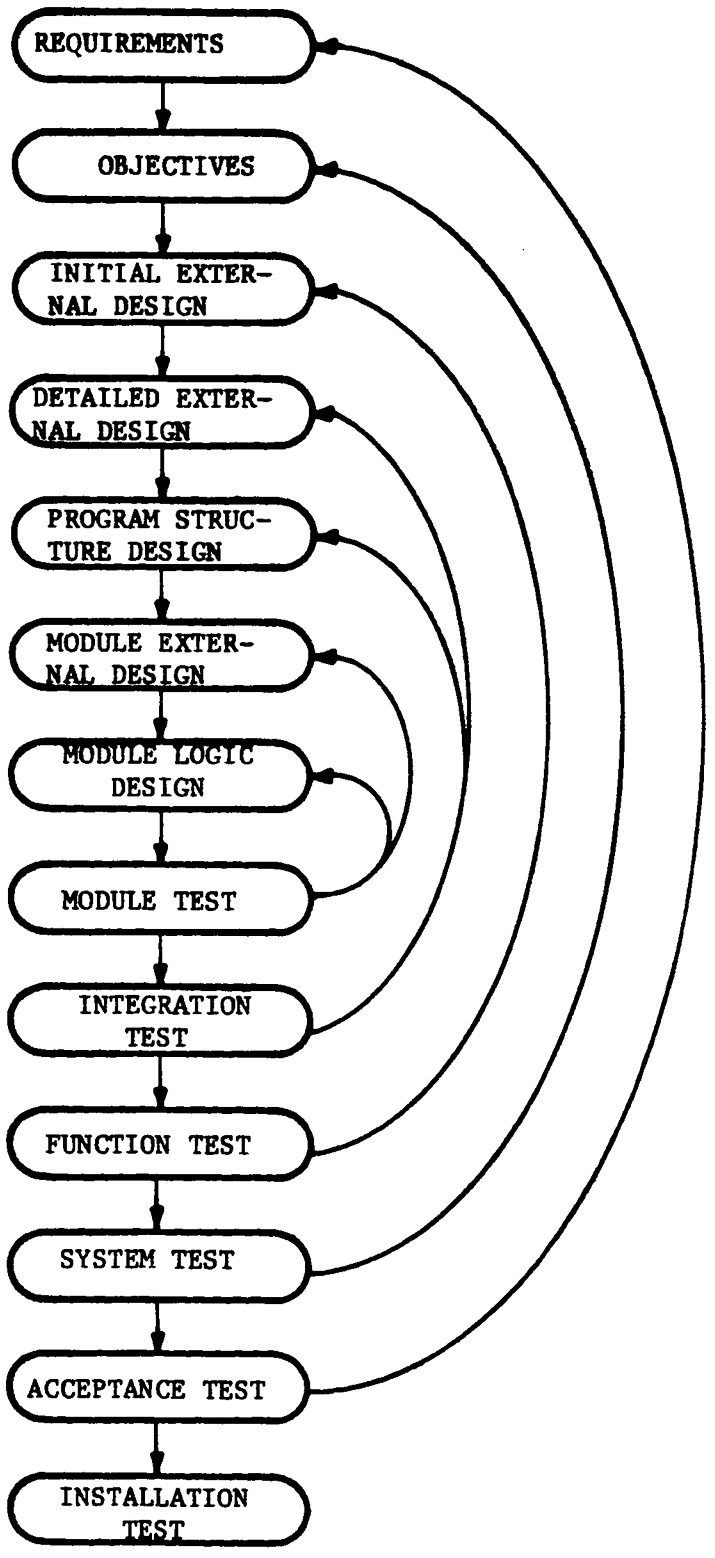

Figure 2.2. The testing processes and their relationships to the design processes.

#### 2.2.1.3 Sandwich testing

This testing is a compromise between bottom up and top down testing.

In this testing one begins using top-down testing and bottom-up testing simultaneously, integrating the program from both the top and the bottom and eventually meeting somewhere in the middle.

### 2.2.2 Function and System Testing

After module testing has been performed and integration or interface testing has been completed, the testing process is just only beginning.

The program or system has yet to be tested against its external specification (function testing) & its objectives (system testing).

The purpose of the function test is to find discrepancies between the program and its external specifications. The prerequisite for a successful function test is a precise and accurate external specification. If the external specification is incomplete or unclear, then the function test cannot end up successfully.

For system integration testing, the method of module integration testing can be used except for some exceptional cases.

### 2.2.3 Acceptance Testing

Acceptance test is a validation process that tests the system to the initial requirements. This testing is performed by the customer or user organization, not by the software development organization.

### 2.2.4 Installation Testing

The purpose of installation testing is to find any mistakes that were made during the installation process.

Installation tests should be designed and written by the software developer and shipped with the product and its documentation.

## 2.3 CONCLUSIONS

Following conclusions can be obtained about the complete software cycle:

i) Software design is a very involved process and a reliable program structure is designed by using modular approach.

ii) Complexity of a program is reduced by maximizing the independence of each component of a program.

iii) Program testing is easier when it is written in modular form and standard techniques exist to test this type of programs.

iv) Structured programming facility exists which is very helpful in module design and coding (Appendix 'A').

v) Debugging process is easier when modular approach is used.

## 3. AVAILABLE MODELS

Some of the models currently available in literature are described below.

### 3.1 SHOOMAN MODEL

The model proposed by Shooman is based on the following assumptions:

i) The total number of machine language instructions in the software program is constant.

ii) The number of errors at the start of integration testing is constant and decreased directly as errors are corrected. No new errors are introduced during the process of testing.

iii) The difference between the errors initially present and the cumulative errors corrected represents the residual errors.

iv) The failure rate is proportional to the number of residual errors.

Based on these assumptions we have:

$$e_r(x) = e(o) - e_c(x)$$

where $x$ = debugging time since the start of system integration.

$e(o)$ = Errors present at $x = 0$, normalized by the total number of machine language instructions $= E_o/I$

$E_o$ = Number of initial errors.

$I$ = Total number of machine language instructions.

$e_c(x)$ = Cumulative number of errors corrected by $x$, normalized by I.

$e_r(x)$ = Residual error at $x$, normalized by I.

According to assumption iv) we have,

$$\lambda_s(t) = K_s\, e_r(x)$$

where

$t$ = operating time of the system.

$K_s$ = constant of proportionality

$\lambda_s(t)$ = Failure rate at time t.

The reliability and MTTF corresponding to the above failure rate are:

$$R(t) = \exp [- \int_{o}^{t} K_s \, e_r(x) \, d_x]$$

and

$$MTTF = \frac{1}{\lambda_s(t)} = \frac{1}{K_s \, e_r(x)}$$

## 3.2 THE MARKOV MODEL

In this model system is assumed to go through a sequence of up & down states (Fig. 3.1).

Up state implies that an error has not yet occurred or is already corrected.

While the down state means that an error has been encountered and has not yet been corrected.

Using the Kolmogrov equations, state probabilities, availability & MTBF can be found out.

There are other models in literature [14-23] to find these quantities but the recent trend is to compute these quantities from users point of view. Software system can provide a reliable

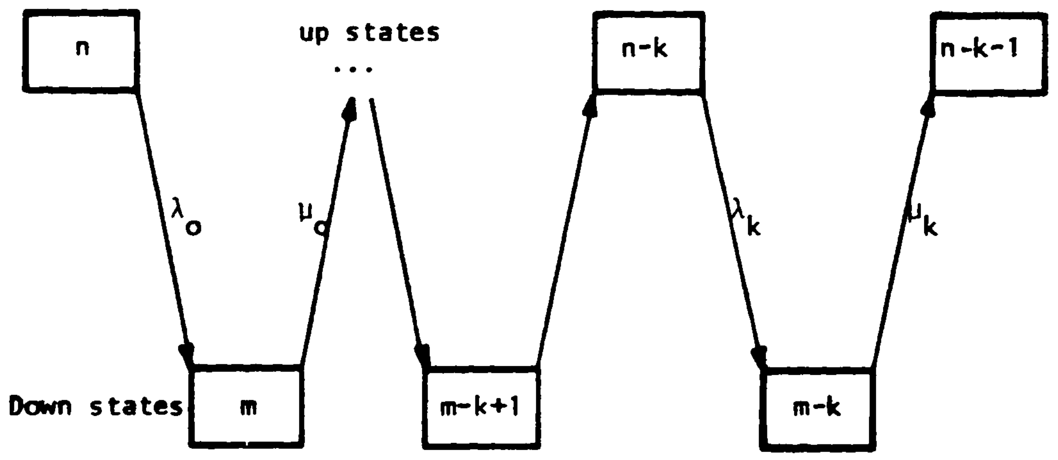

Figure 3.1. The Markov model.

service to a user when it is known that error exists, provided that the service requested does not utilize the defective part. This emphasizes the importance of modular structure of programs and hence, on the module reliabilities. Models which include the user can thus be termed user oriented models.

## 3.3 USER ORIENTED RELIABILITY MODELS

The basic definition of software reliability is rephrased as follows:

"Probability that when a user demands a service from the software system it will perform to the satisfaction of the user".

Further assumptions are as follows:

i) A user profile is the probabilistic distribution of the set of processes that can be generated by the program in an application environment.

ii) Programs are assumed in modular form, where each module is logically independent component of the system.

iii) Reliabilities of the modules are independent i.e. modules are statistically independent.

iv) Transfer of control among program module is Markov process i.e. transfer of control to the next module depends only on the present module and not on the past modules thru which the program has come to the present module.

v) Module reliability functions can be determined.

One such model is presented by Cheung [36]. Further assumptions in addition to one listed above are as follows:

i) Program graph has a single entry node and a single exit node.

ii) Every node in the graph is a state of the Markov process, with the initial state corresponding to the entry node of the program graph.

iii) Two states C & F are added as the state of correct output and failure.

Using these assumptions we take an example of a 10 module program as shown in Fig. 3.2. Now states C & F are added to indicate the states of correct output & failure (Fig. 3.3).

The state transition probability matrix is obtained by using the module reliabilities & probability of transition from one state to another.

Reliability of the program is therefore the probability of being in state 10 multiplied by the reliability of module number 10.

The time dependent state probabilities can be obtained

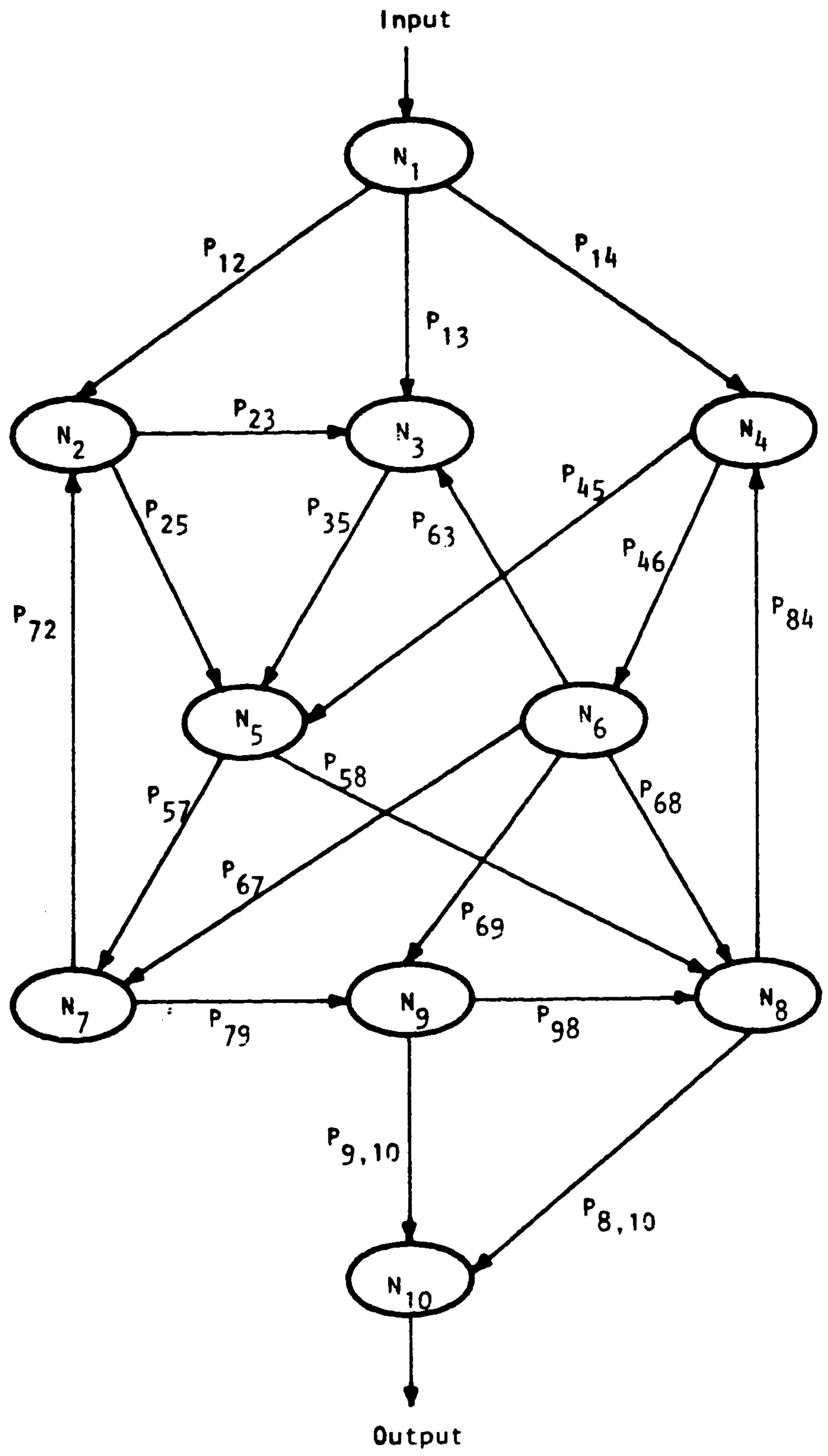

Figure 3.2. A program control flow graph.

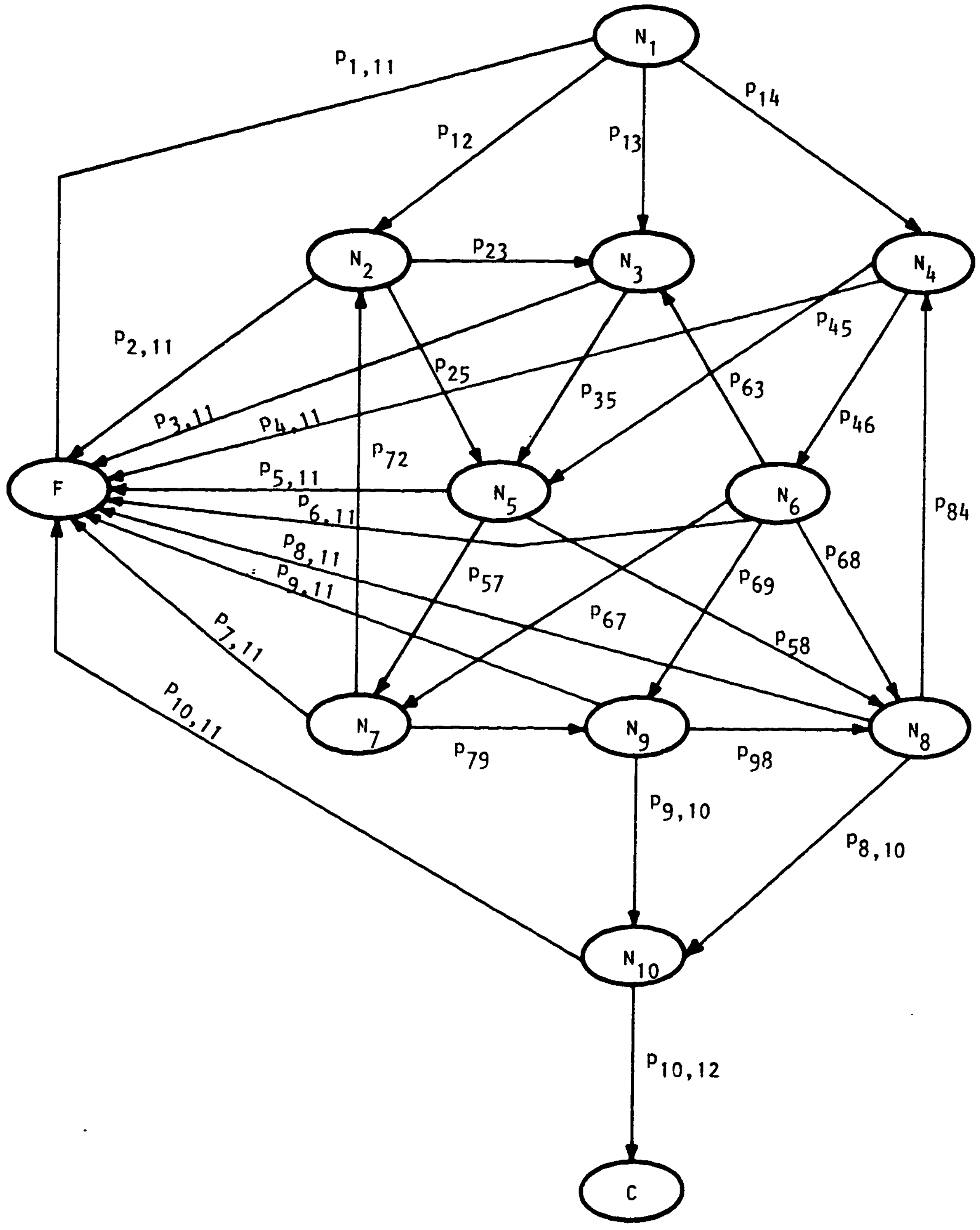

Figure 3.3. The Markov software reliability model of the program.

from the following equation:

$$p(n) = p(o)\, P^n \tag{3.1}$$

where $p(n)$ = The state probability vector at time n.

$P$ = State transition probability matrix.

$p(o)$ = Initial state probability vector.

$n$ = Discrete time = 0,1,2, ----

The results obtained for the parameter values as given by Cheung [36] are tabulated in Table I [Appendix 'B'] & Table II [Appendix 'B'] They are plotted in Figs. 3.4 & 3.5. Steady state probability of state 12, is observed to be the same as that obtained by Cheung [36].

This method is specially useful for computer analysis and for large number of states.

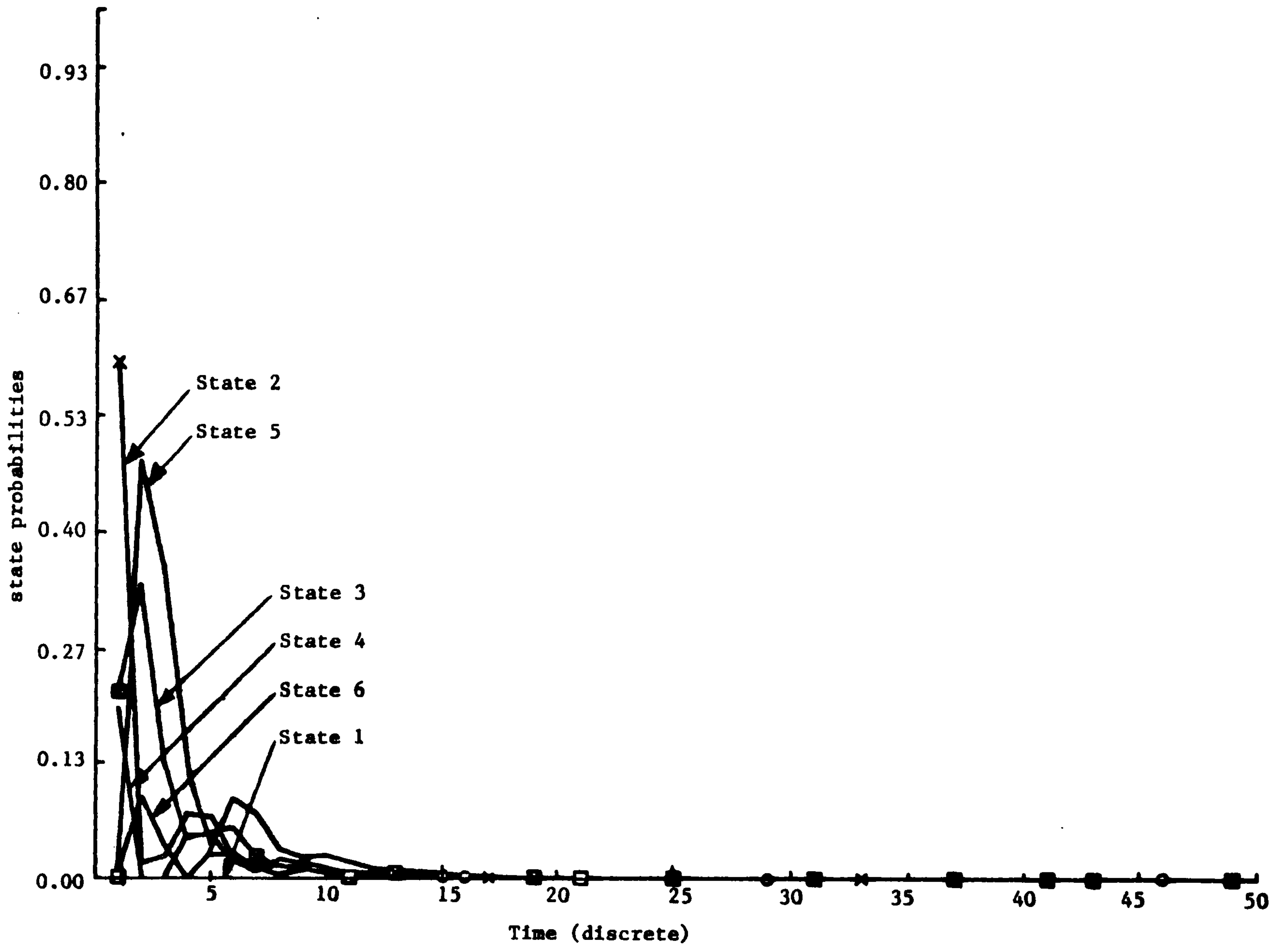

Figure 3.4. Cheung's model plot (states 1-6).

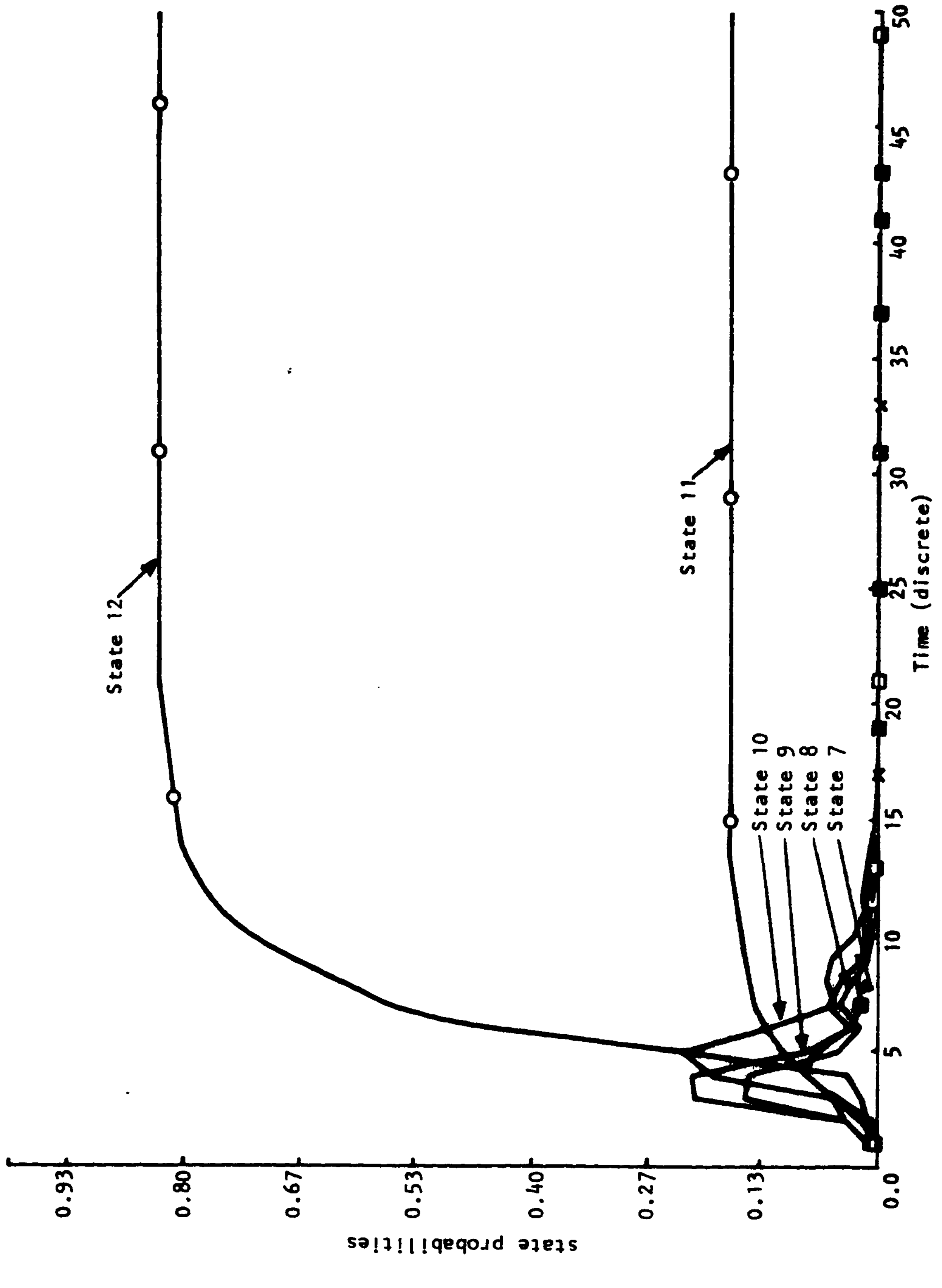

Figure 3.5. State probabilities for Cheung's model.

# 4. PROPOSED MODELS

The models presented in this thesis, are based on the following assumptions:

i) Programs are assumed in modular form.

ii) Reliabilities of the modules are independent.

iii) Module reliability functions can be determined.

iv) Program graph has a single entry node and a single exit node.

v) Every node in the graph is a state of the Markov process with the initial state corresponding the entry node of the program graph.

vi) Two states C & F are added as the state of correct output and failure.

vii) Fault correction in the failure state takes finite amount of time as opposed to Littlewood's assumption [21] of instant fault correction. Therefore 'availability' could provide another reliability index.

viii) Whenever a failure occurs in some module i, fault correction takes the process back to module i, provided that the fault is corrected.

Two new definitions of operational reliability and operational availability are introduced as follows:

Operational reliability:

It is defined as the probability that the program executes successfully for a particular input condition.

Operational Availability:

It is defined as the probability of the program being in the successful state C for various random input conditions, input condition being determined by a factor called input rate which is the transition rate from state C to state 1 in the state transition diagram.

## 4.1 (n+2) STATE MARKOV MODEL WITH FAULT CORRECTION

In this model, it is assumed, that the transfer of control among program modules is governed by a Markov process. If n is the number of modules in a program then this model will have n + 2 states.

State diagram for the 10 module program example with fault correction is shown in Fig. 4.1. Keeping the same values of module reliabilities and path probabilities as for Cheung's model example we have state F (state 11) & state C (state 12) probabilities (operational unreliability & operational reliability respectively)

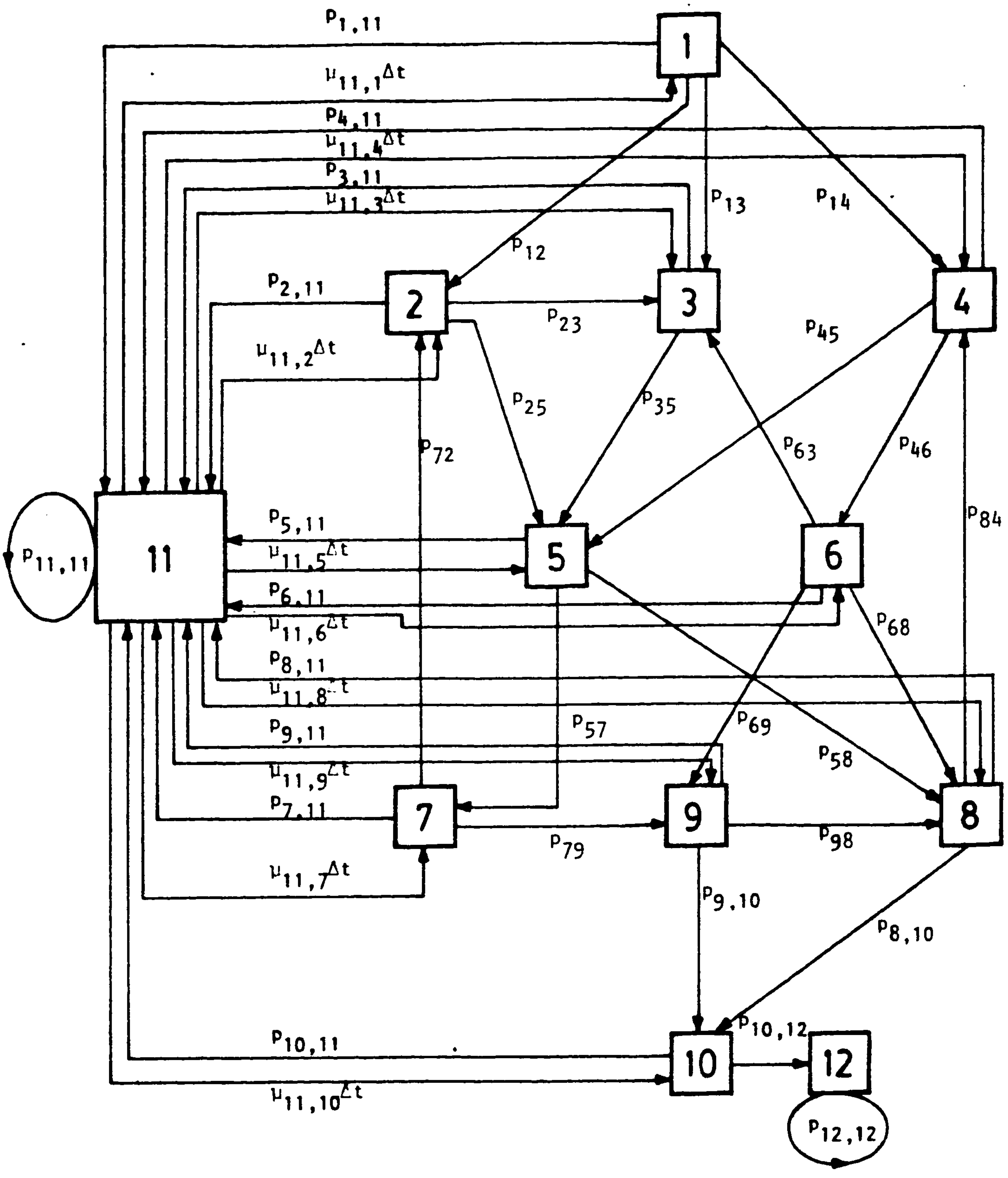

Figure 4.1. 12 states model with fault correction.

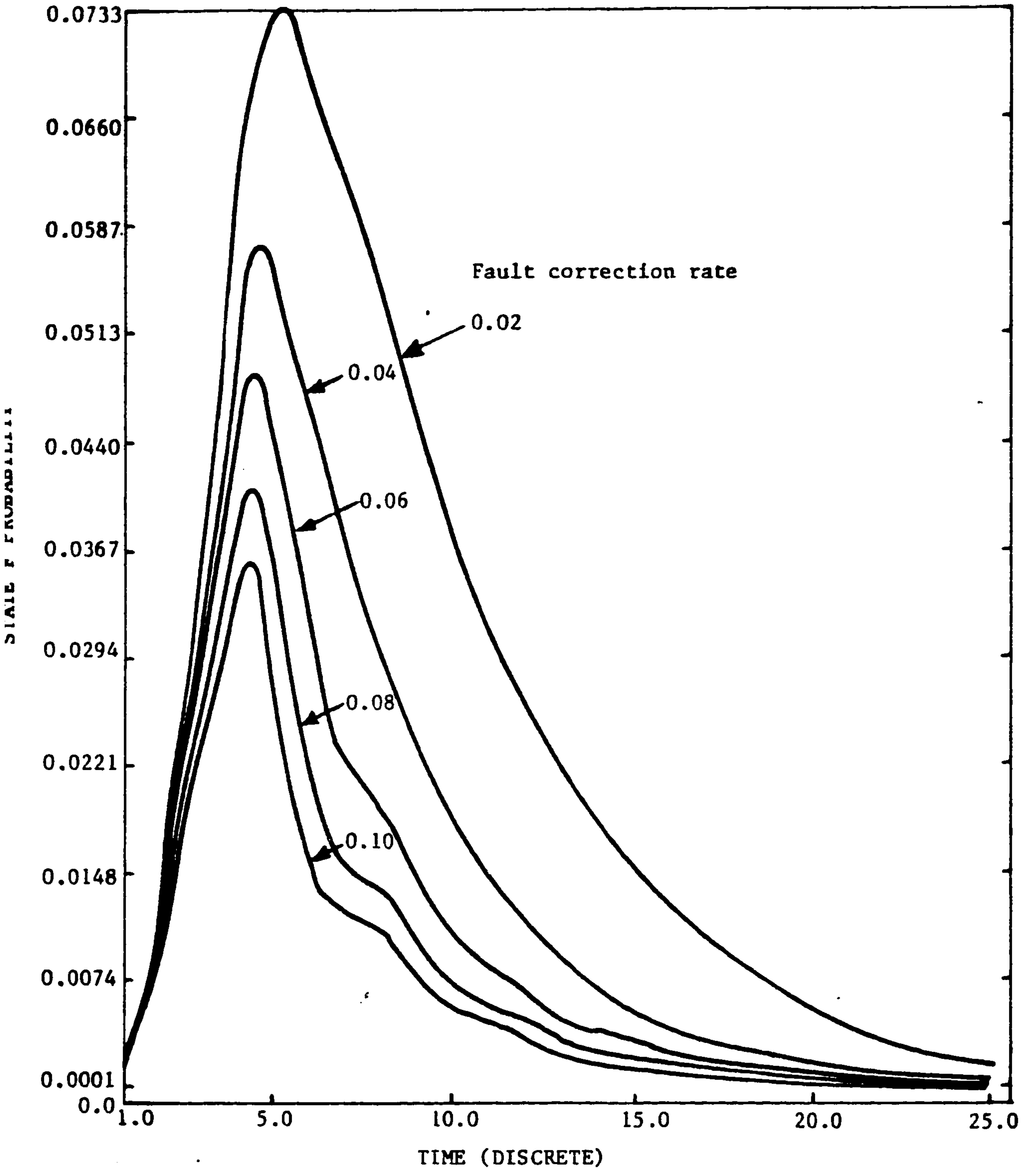

Figure 4.2. State F probabilities as a function of time at various fault correction rates.

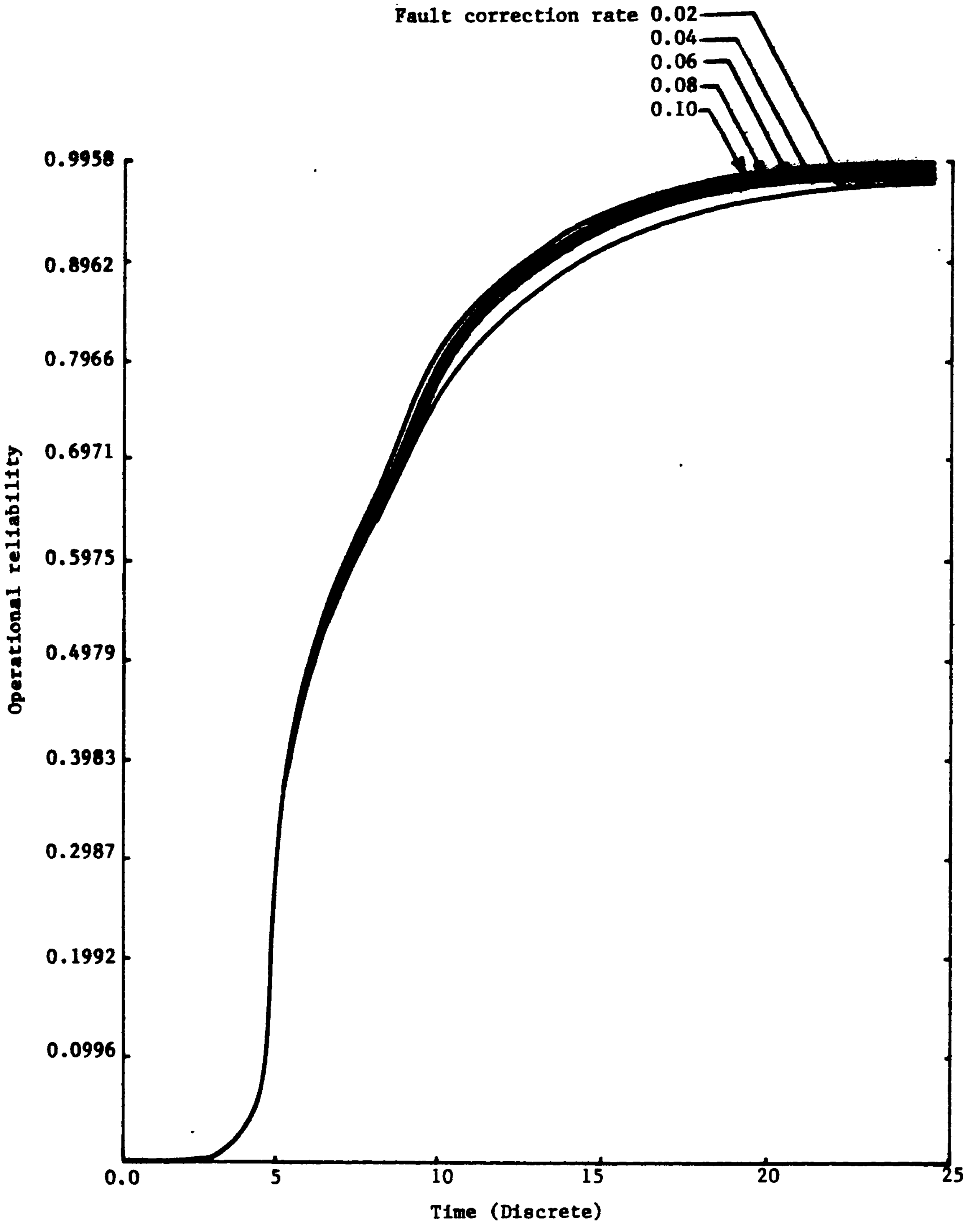

Figure 4.3. Operational reliability as a function of time at various fault correction rates.

$$\begin{bmatrix}
0.0 & 0.59940 & 0.19980 & 0.19980 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.00100 & 0.0 \\
0.0 & 0.0 & 0.68600 & 0.0 & 0.29400 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.02000 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.99000 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.01000 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.38800 & 0.58200 & 0.0 & 0.0 & 0.0 & 0.0 & 0.03000 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.38000 & 0.57000 & 0.0 & 0.0 & 0.05000 & 0.0 \\
0.0 & 0.0 & 0.29850 & 0.0 & 0.0 & 0.0 & 0.29850 & 0.09950 & 0.29850 & 0.0 & 0.00500 & 0.0 \\
0.0 & 0.49250 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.49250 & 0.0 & 0.01500 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.23750 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.71250 & 0.05000 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.09750 & 0.0 & 0.87750 & 0.02500 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.01500 & 0.985 \\
0.02000 & 0.02000 & 0.02000 & 0.02000 & 0.02000 & 0.02000 & 0.02000 & 0.02000 & 0.02000 & 0.02000 & 0.80000 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.00
\end{bmatrix}$$

Figure 4.4. The transition probability matrix for 12 states model with fault correction.

at various fault correction rates as shown in Tables iii & iv, (Appendix 'B') respectively. Their plots are shown in Figs. 4.2 & 4.3. Figure 4.4 shows the transition probability matrix for the fault correction of 0.02.

The plot of state C (state 12) probabilities (operational availability) at various input rates is shown in Fig. 4.6. The state transition diagram is shown in Fig. 4.5. The values are tabulated in Table v, (Appendix 'B'). The transition rate matrix for input rate of .02 is shown in Fig. 4.7.

## 4.2 (2n+2) STATE MARKOV MODEL WITH FAULT CORRECTION

This is a modified version of the model presented in last section. In this model we can consider a nonconstant fault correction rate from state F which is the failure state. In the previous model we add an intermediate state between failure state and other states so that fault correction takes place through an intermediate hypothetical state. This is known as method of stages.

The new state transition diagram for our program example is shown in Fig. 4.8.

State C operational probabilities is listed in Table vi (Appendix 'B') and the transition rate matrix is shown in Fig. 4.9. The plot of state C probabilities is shown in Fig. 4.10.

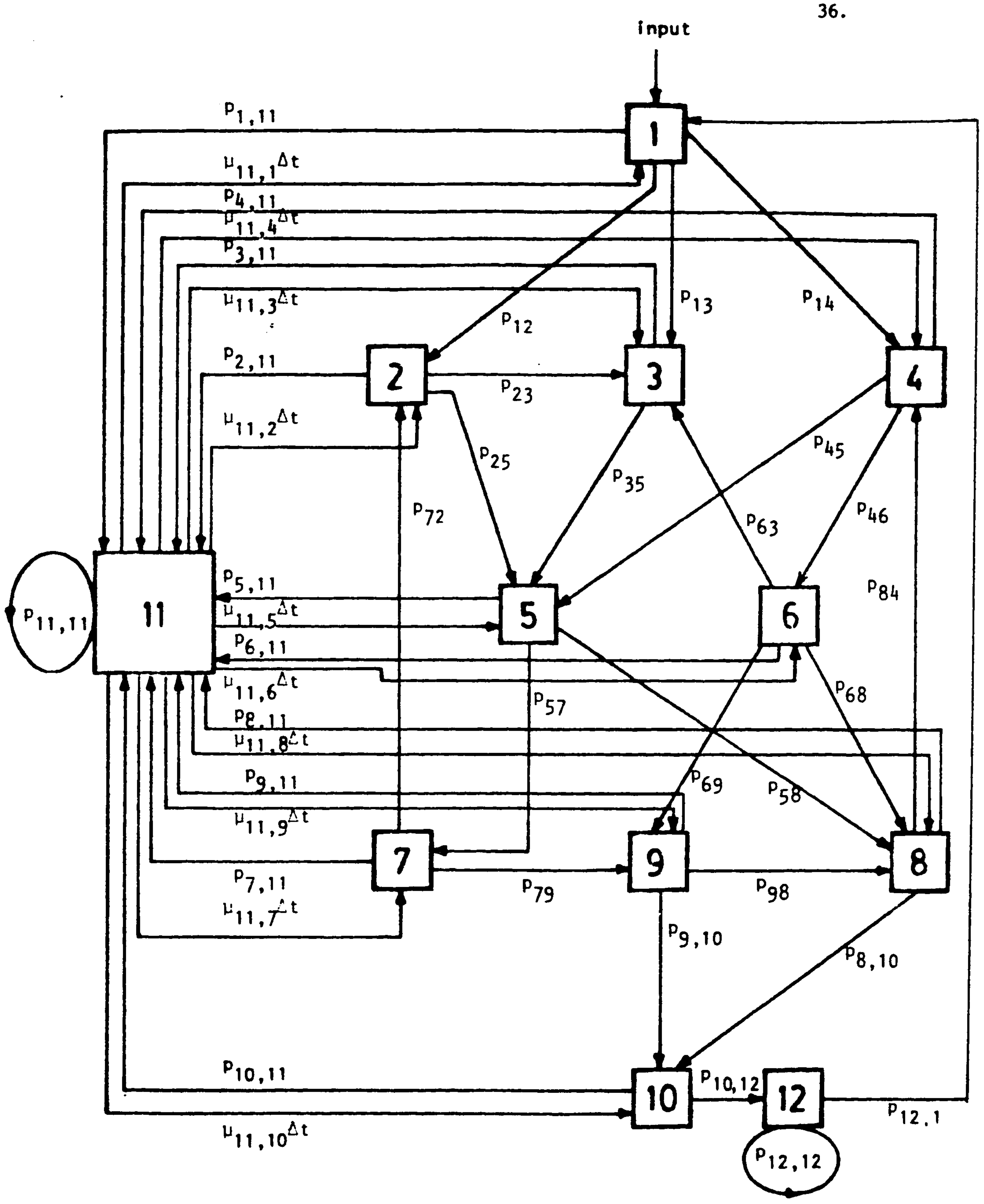

Figure 4.5. 12 states model with fault correction and variable input.

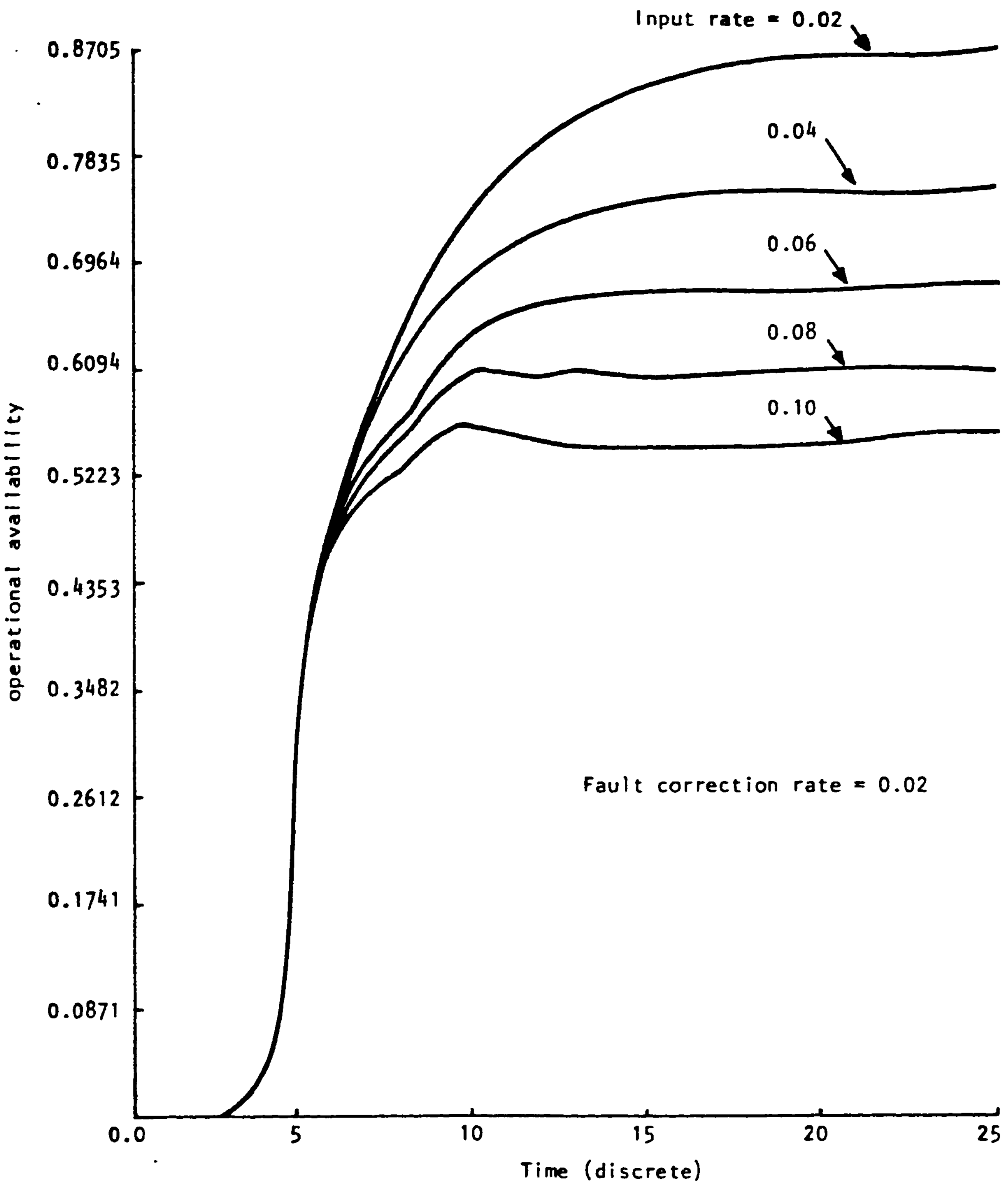

Figure 4.6. Operational availabilities at different input rates.

$$\begin{bmatrix}
0.0 & 0.59940 & 0.19980 & 0.19980 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.00100 & 0.0 \\
0.0 & 0.0 & 0.68600 & 0.0 & 0.29400 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.02000 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.99000 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.01000 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.38800 & 0.58200 & 0.0 & 0.0 & 0.0 & 0.0 & 0.03000 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.38000 & 0.57000 & 0.0 & 0.0 & 0.05000 & 0.0 \\
0.0 & 0.0 & 0.29850 & 0.0 & 0.0 & 0.0 & 0.29850 & 0.09950 & 0.29850 & 0.0 & 0.00500 & 0.0 \\
0.0 & 0.49250 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.49250 & 0.0 & 0.01500 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.23750 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.71250 & 0.05000 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.09750 & 0.0 & 0.87750 & 0.02500 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.01500 & 0.985 \\
0.02000 & 0.02000 & 0.02000 & 0.02000 & 0.02000 & 0.02000 & 0.02000 & 0.02000 & 0.02000 & 0.02000 & 0.80000 & 0.0 \\
0.02 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.98
\end{bmatrix}$$

Figure 4.7. The transition probability matrix for 12 states model with input rate 0.02.

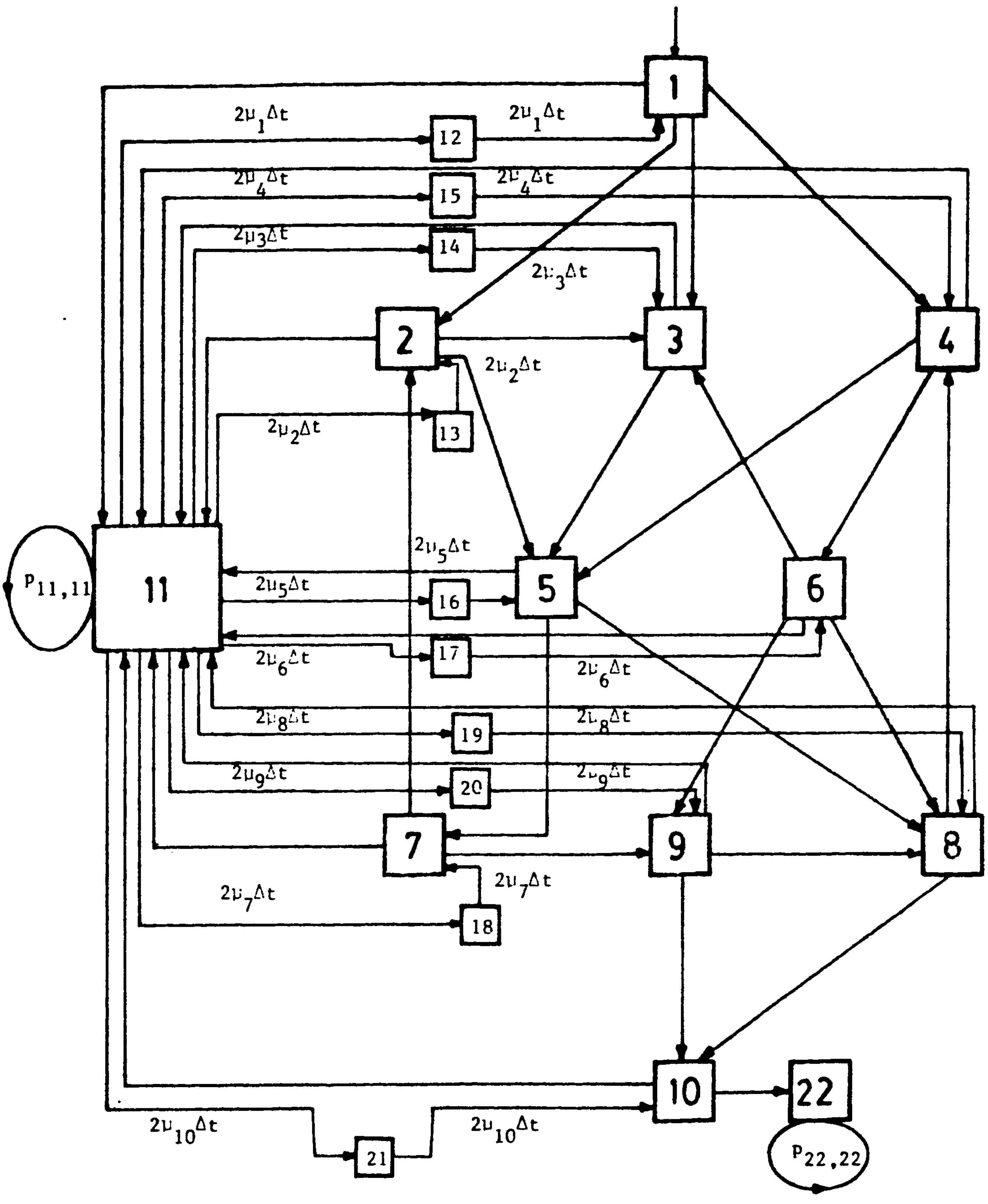

Figure 4.8. 22 states model with fault correction.

$$\begin{bmatrix}
0.0000 & 0.5994 & 0.1998 & 0.1998 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0010 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0000 & 0.6860 & 0.0000 & 0.2940 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0200 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.9900 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0200 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.3880 & 0.5820 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0400 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.3800 & 0.5700 & 0.0000 & 0.0000 & 0.0500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0000 & 0.2985 & 0.0000 & 0.0000 & 0.0000 & 0.2985 & 0.0995 & 0.2985 & 0.0000 & 0.0050 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.4925 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.4925 & 0.0000 & 0.0150 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0000 & 0.0000 & 0.2375 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.7125 & 0.0500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0975 & 0.0000 & 0.8775 & 0.0250 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0150 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.985 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.5000 & 0.0500 & 0.0500 & 0.0500 & 0.0500 & 0.0500 & 0.0500 & 0.0500 & 0.0500 & 0.0500 & 0.05 & 0.0 \\
0.0500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.9500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.9500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0000 & 0.0500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.9500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0000 & 0.0000 & 0.0500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.9500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.9500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.9500 & 0.0000 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.9500 & 0.0000 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.9500 & 0.0000 & 0.0 & 0.0 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0500 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.9500 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.05 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.95 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0
\end{bmatrix}$$

Figure 4.9. Transition probability matrix for 22 states model with fault correction.

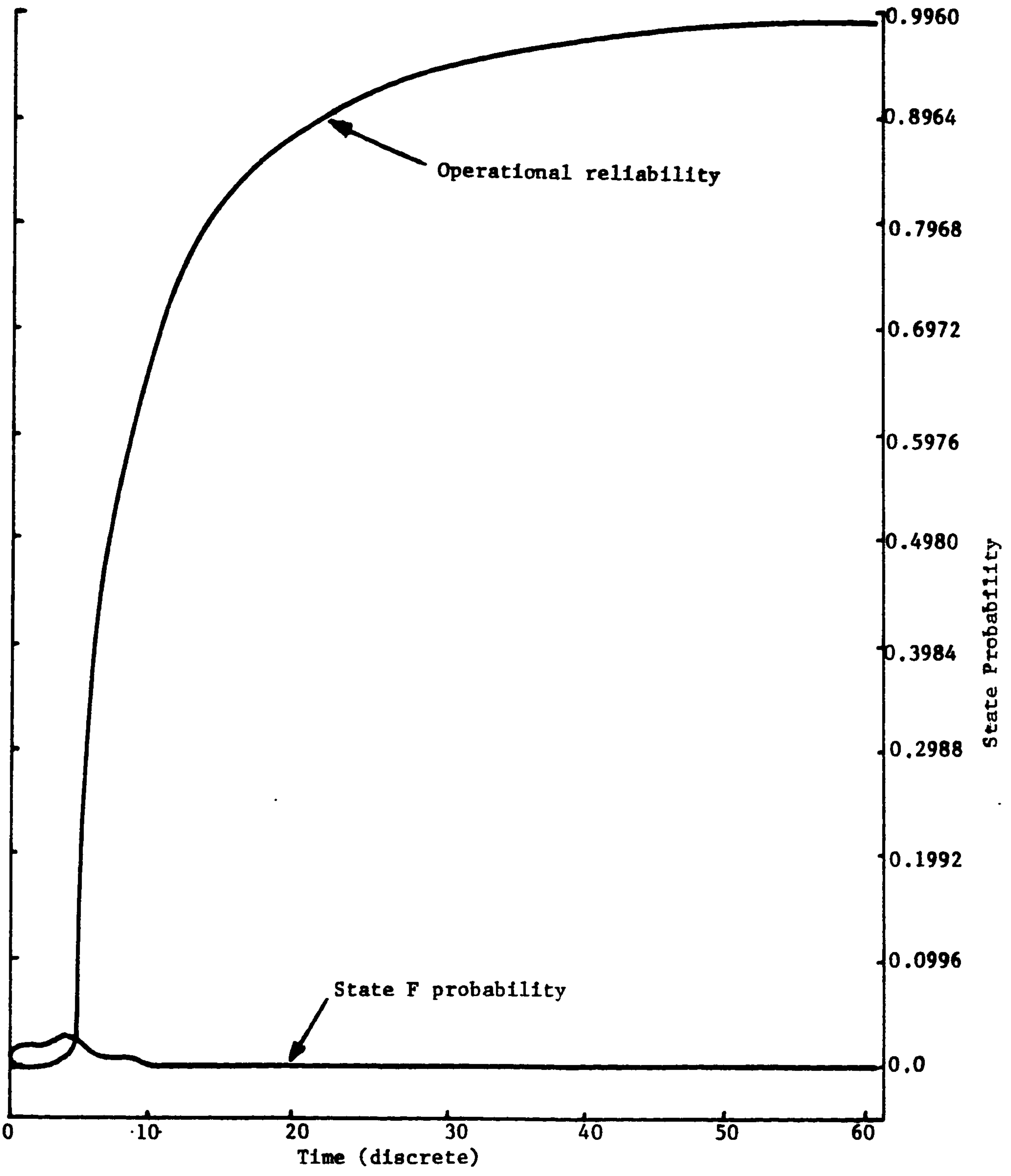

Figure 4.10 Operational reliabilities as a function of time for 22 states model

## 4.3 (2n+2) STATE SEMI MARKOV MODEL WITH REPAIR

In this model, the transfer of control among program modules is described by an imbedded Markov chain but the time spent in each state (normally called the holding time) is now taken as a random variable with any general probability density function. Since this model is not exactly Markov, we call it a semi-Markov model. The holding time $\tau_{ij}$ are positive, integer valued random variables each governed by a probability mass function $h_{ij}$ called the holding time mass function for a transition from state i to state j.

The mean waiting time in a state is denoted by $\bar{\tau}_i$ and is given as:

$$\bar{\tau}_i = \sum_{j=1}^{n} p_{ij}\, \bar{\tau}_{ij} \tag{4.1}$$

The cumulative and complementary cumulative probability distributions for the waiting times are:

$$\leq \omega_i(t) = \sum_{m=0}^{t} \omega_i(m) = \sum_{m=0}^{t} \sum_{j=1}^{n} p_{ij}\, h_{ij}(m) \tag{4.2}$$

and

$$> \omega_i(t) = \sum_{m=t+1}^{\infty} \omega_i(m) = \sum_{m=t+1}^{\infty} \sum_{j=1}^{n} p_{ij}\, h_{ij}(m) \tag{4.3}$$

The internal transition probability is given by $\phi_{ij}(t)$ as follows:

$$\phi_{ij}(t) = > \delta_{ij} > \omega_i(t) + \sum_{k=1}^{n} p_{ik} \sum_{m=0}^{t} h_{ik}(m)\, \phi_{kj}(t-m) \quad (4.4)$$

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \qquad \begin{array}{l} i = 1,2, \text{-----}\ n \\ j = 1,2, \text{-----}\ n \\ t = 0,1,2, \text{----} \end{array}$$

In matrix form we have

$$\phi(t) = > W(t) + \sum_{m=0}^{t} [P \,\square\, H(m)]\ \phi(t-m)$$

$$t=0,1,2, \text{---}$$

Here H(m) is the holding time mass function matrix.

and

$$P \,\square\, H(m) = p_{ij}\, h_{ij}(m)$$

where $p_{ij}$ are the elements of matrix P, the transition probability matrix.

and

$h_{ij}(m)$ are the elements of holding time mass function matrix.

If a process is in state i at time zero, the transition interval probability for each state will be 1 because no transition is supposed to have occurred at time zero. As the time t will increase the value of $\emptyset_{ij}$ will change. Plot of three values of $\emptyset_{ij}$ is shown in Fig. 4.11.

The computed waiting time in each state, destination unknown is tabulated in Table vii (Appendix 'B').

The word destination unknown here implies that the waiting times for all possible destinations j are summed up when the process is in state i. The transition probability matrix used here is shown in Fig. 4.9.

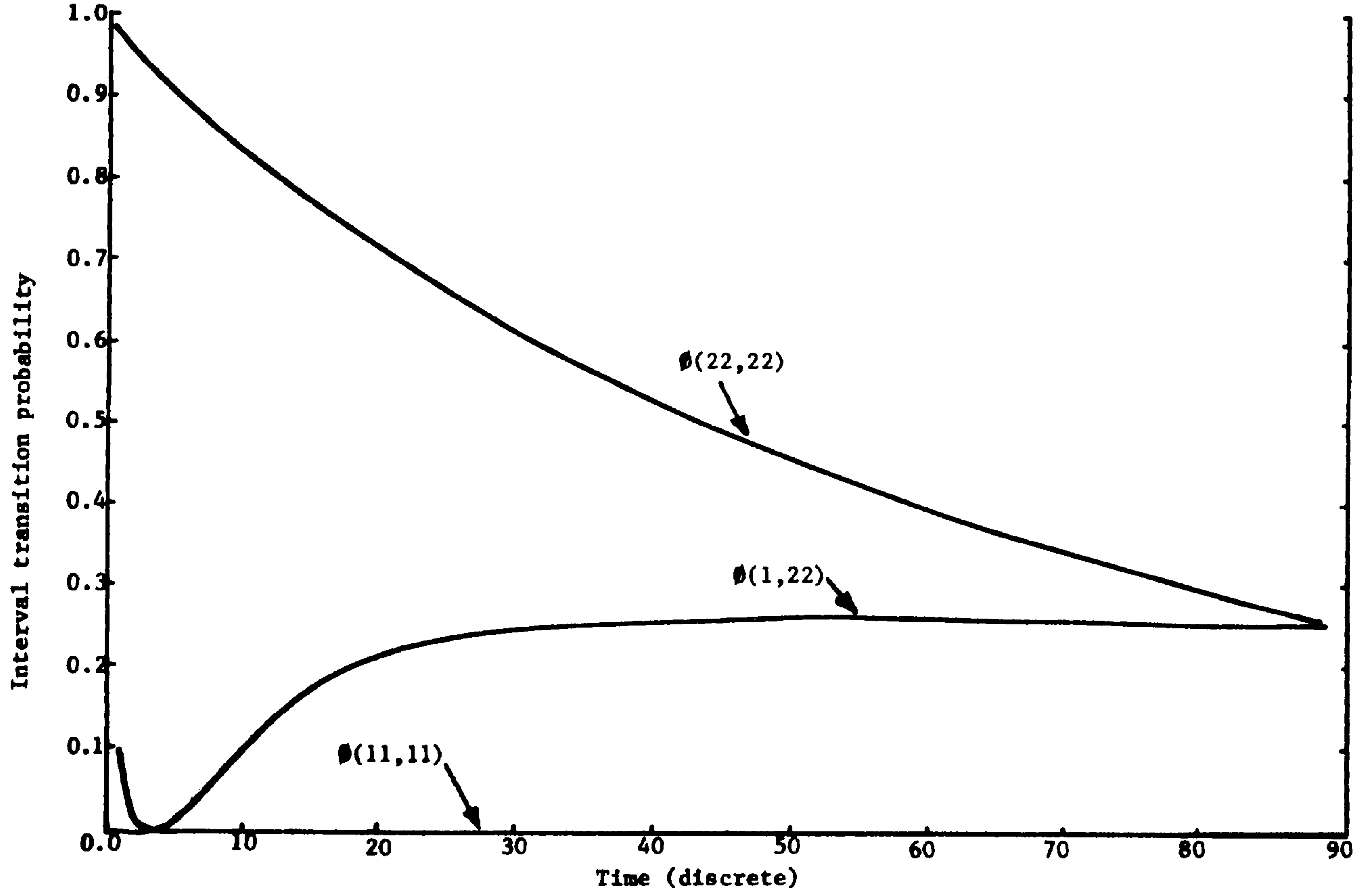

Figure 4.11 Interval transition probability as a function of time for semi Markov model.

# 5. CONTINUOUS TIME MODELS

## 5.1 INTRODUCTION

Models presented in the previous chapter were discrete time models as time was represented by discrete steps rather than continuous variable. The probability at each step is distinct in discrete time process and the probability values can only be obtained at step points and not at any other points. In continuous time process the probability value is defined for each and every point of function variable time.

In this chapter, we present continuous time models by modifying the models discussed in the previous chapter. The continuous time Markov process is just like a discrete time Markov process except that the time between transitions must be distributed exponentially rather than geometrically. This result is expected because the exponential distribution is just the continuous analog of the geometric distribution.

Thus a discrete function f(.) that takes on the value f(n) at the point n where n is a non negative integer could be as well represented at the point n by a continuous type impulse of area f(n). If we used this representation at all points n, the continuous function f(t) can be written as:

$$f(t) = \sum_{n_o=0}^{\infty} f(n_o)\ \delta(t - n_o)$$

The exponential transform $f^e(s)$ of $f(t)$ would then be

$$f^e(s) = \int_o^{\infty} f(t)\ e^{-st}\ dt = \int_o^{\infty} dt\ e^{-st} \sum_{n_o=0}^{\infty} f(n_o)\ \delta(t - n_o)$$

$$= \sum_{n_o=0}^{\infty} f(n_o) \int_o^{\infty} e^{-st}\ \delta(t - n_o)\ dt$$

$$= \sum_{n_o=0}^{\infty} f(n_o)\ e^{-sn_o} \qquad (5.1)$$

putting $z = e^{-s}$ , the exponential transform becomes

$$f^e(s) = \sum_{n_o=0}^{\infty} f(n_o)\ z^n = f^g(z) \qquad (5.2)$$

This shows that the geometric transform of a discrete function is represented by a continuous type impulse string and then for writing its exponential transform simply $e^{-s}$ is substituted for $z$ in the geometric transform.

Assumptions of the previous chapter are valid for the modified models presented in this chapter. The reliability indices, operational reliability and operational availability are used to determine the software performance as before.

A matrix $\Lambda$ is introduced as below.

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & \text{———} & 0 \\ 0 & \lambda_2 & 0 & \text{———} & 0 \\ \vdots & & & & \\ 0 & 0 & \text{———} & & \lambda_N \end{bmatrix} \qquad (5.3)$$

Where each value $\lambda_i$ represents the rate parameter of the waiting time density function at the origin for state i. Thus the equations for the exponential waiting time density function and its complementary cumulative probability distribution have the appearance,

$$\omega_i(\tau) = \lambda_i e^{-\lambda_i \tau}, \ \tau \geq 0 \ ; \quad {>}\omega_i(t) = e^{-\lambda_i t}, \ t \geq 0 \qquad (5.4)$$

Since the mean of a variable that is exponentially distributed with rate parameter $\lambda$ is $1/\lambda$, the mean waiting time matrix is just the inverse of $\Lambda$.

The transition rate matrix A of the continuous time Markov process is defined by:

$$A = -\Lambda(I - P) = \Lambda(P - I) \qquad (5.5)$$

Thus A has the form:

$$A = \begin{bmatrix} a_{11} & a_{12} & \text{---------} & a_{1N} \\ a_{21} & a_{22} & \text{---------} & a_{2N} \\ \vdots & & & \\ a_{N1} & a_{N2} & \text{---------} & a_{NN} \end{bmatrix}$$

$$= \begin{bmatrix} -\lambda_1 \sum\limits_{j\neq 1} p_{1j} & \lambda_1 p_{12} & \lambda_1 p_{13} & \text{-----} & \lambda_1 p_{1N} \\ \lambda_2 p_{21} & -\lambda_2 \sum\limits_{j\neq 2} p_{2j} & \lambda_2 p_{23} & \text{------} & \lambda_2 p_{2N} \\ \vdots & & & & \\ \lambda_N p_{N1} & \lambda_N p_{N2} & \lambda_3 p_{N3} & \text{------} & -\lambda_N \sum\limits_{j\neq N} p_{NJ} \end{bmatrix}$$

(5.6)

The transition rate matrix plays the same central role for continuous time Markov processes as the transition probability matrix P does for discrete time Markov processes.

## 5.2 MODIFIED (n+2) STATE MARKOV MODEL WITH FAULT CORRECTION

The control structure of the program graph is represented in the same way as in non modified model. If n is the number of modules in a program then the model will contain n+2 states, two additional states being C & F. Thus the state transition probability matrix P will remain unchanged which is a (n+2) x (n+2) matrix.

Matrix $\Lambda$ and P can be used to determine the transition rate matrix A as shown in (5.5) which is used in obtaining the state probability differential equations. Once the differential equations are obtained, it can be solved for obtaining the state probabilities as a function of time. The state C probability will be the operational reliability at zero input rate and operational availability at non-zero input rate.

The program flow graph of Fig. 4.1 is taken as an example. The matrix $\Lambda$ for this program flow graph is shown in Fig. 5.1 and is based on the mean waiting time data obtained from semi Markov model. The transition rate matrix A is obtained by using (5.5) and is shown in Fig. 5.2. Using the elements of transition rate matrix

$$\Lambda = \begin{bmatrix} \lambda_1 & 0.0 & \cdots & 0.0 \\ 0.0 & \lambda_2 & & \vdots \\ \vdots & \vdots & & \vdots \\ 0.0 & 0.0 & \cdots & \lambda_{12} \end{bmatrix}$$

or

$$\Lambda = \begin{bmatrix} 0.5011 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.3814 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0101 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.4986 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.5022 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.7183 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5112 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.3518 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.137 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0152 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.901 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

Figure 5.1. Matrix $\Lambda$ for 12 states model.

$$A = \begin{bmatrix}
-\lambda_1 & \lambda_1 p_{12} & \lambda_1 p_{13} & \lambda_1 p_{14} & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 p_{1,11} & 0 \\
0 & -\lambda_2 & \lambda_2 p_{23} & 0 & \lambda_2 p_{25} & 0 & 0 & 0 & 0 & 0 & \lambda_2 p_{2,11} & 0 \\
0 & 0 & -\lambda_3 & 0 & \lambda_3 p_{35} & 0 & 0 & 0 & 0 & 0 & \lambda_3 p_{3,11} & 0 \\
0 & 0 & 0 & -\lambda_4 & \lambda_4 p_{45} & \lambda_4 p_{46} & 0 & 0 & 0 & 0 & \lambda_4 p_{4,11} & 0 \\
0 & 0 & 0 & 0 & -\lambda_5 & 0 & \lambda_5 p_{57} & \lambda_5 p_{58} & 0 & 0 & \lambda_5 p_{5,11} & 0 \\
0 & 0 & 0 & 0 & 0 & -\lambda_6 & 0 & \lambda_6 p_{68} & \lambda_6 p_{69} & 0 & \lambda_6 p_{6,11} & 0 \\
0 & \lambda_7 p_{72} & 0 & 0 & 0 & 0 & -\lambda_7 & 0 & \lambda_7 p_{79} & 0 & \lambda_7 p_{7,11} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda_8 & 0 & \lambda_8 p_{8,10} & \lambda_8 p_{8,11} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_9 p_{98} & -\lambda_9 & \lambda_9 p_{9,10} & \lambda_9 p_{9,11} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda_{10} & \lambda_{10} p_{10,11} & \lambda_{10} p_{10,12} \\
\lambda_{11} p_{11,1} & \lambda_{11} p_{11,2} & \lambda_{11} p_{11,3} & \lambda_{11} p_{11,4} & \lambda_{11} p_{11,5} & \lambda_{11} p_{11,6} & \lambda_{11} p_{11,7} & \lambda_{11} p_{11,8} & \lambda_{11} p_{11,9} & \lambda_{11} p_{11,10} & -S\lambda_{11} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

where

$$S = p_{11,1} + p_{11,2} + \text{--------} + p_{11,10}$$

Figure 5.2. Transition rate matrix for 12 states model.

the modified model for operational reliability is shown in Fig. 5.3.

The state equations in the matrix form is obtained as follows:

$$\dot{P}' = P'A$$

where

$$\dot{P}' = \frac{dp}{dt} = (\dot{P}_1\ \dot{P}_2\ \dot{P}_3 \text{ --------- } \dot{P}_{12})$$

and

$$P' = (P_1\ P_2\ P_3 \text{ ----------- } P_{12}) \tag{5.7}$$

and a prime denotes transpose.

Figure 5.4 gives the equations in matrix form.

These equations are solved for state probabilities by using fourth order Runge Kutta method. The operational reliabilities at various fault correction rate is shown in Fig. 5.5 and the values are tabulated in Table VIII (Appendix B).

The modified model for finding the operational availability is shown in Fig. 5.6. The operational availabilities at various input rates is shown in Fig. 5.7 and are tabulated in Table IX

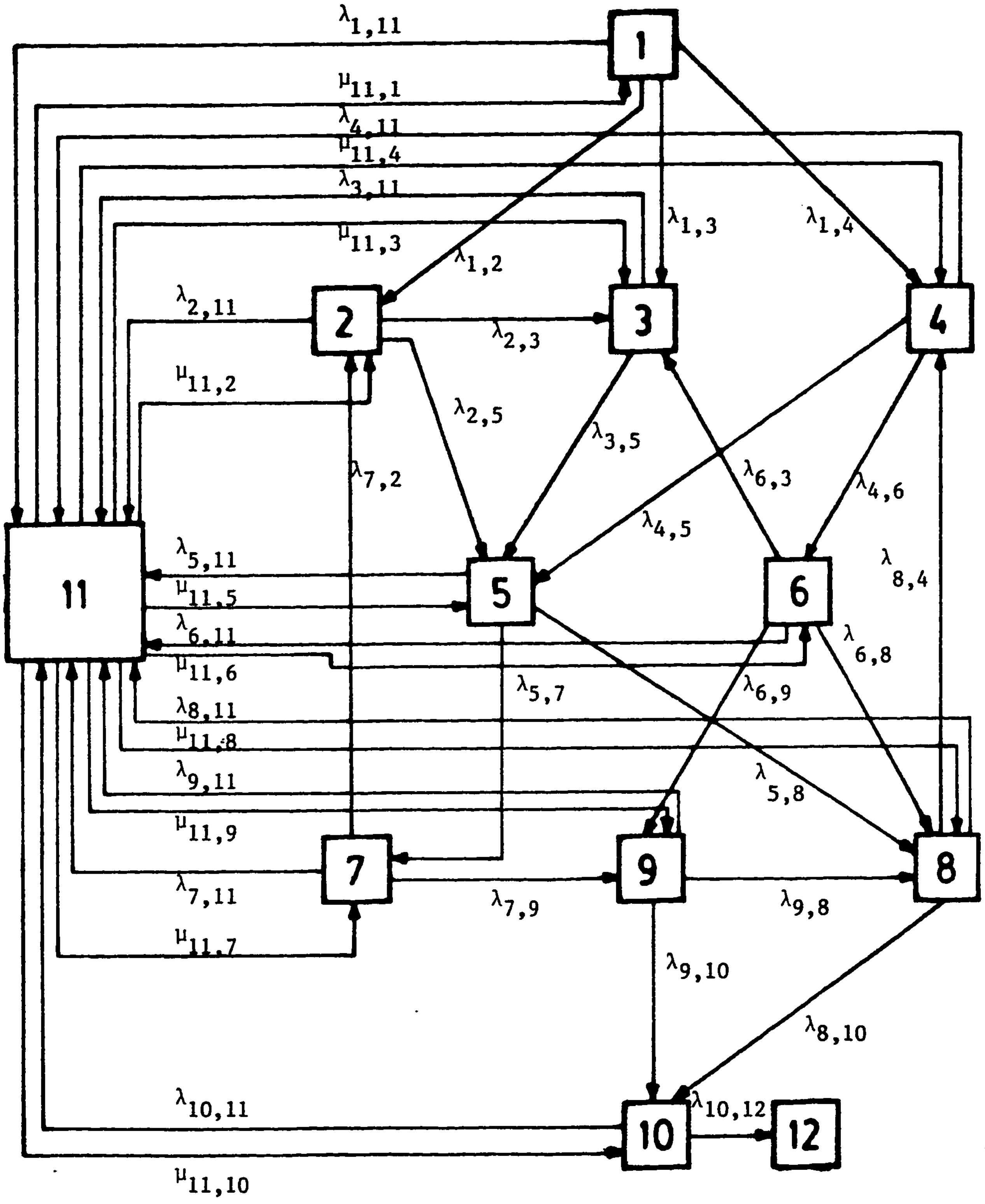

Figure 5.3. Modified 12 states model for operational reliability.

$$[\dot{P}] = \begin{bmatrix} -\lambda_1 P_1 + \mu_{11,1} P_{11} \\ -\lambda_2 P_2 + \lambda_{1,2} P_1 + \lambda_{7,2} P_7 + \mu_{11,2} P_{11} \\ -\lambda_3 P_3 + \lambda_{1,3} P_1 + \lambda_{2,3} P_2 + \mu_{11,3} P_{11} \\ -\lambda_4 P_4 + \lambda_{1,4} P_1 + \mu_{11,4} P_{11} \\ -\lambda_5 P_5 + \lambda_{2,5} P_2 + \lambda_{3,5} P_3 + \lambda_{4,5} P_4 + \mu_{11,5} P_{11} \\ -\lambda_6 P_6 + \lambda_{4,6} P_4 + \mu_{11,6} P_{11} \\ -\lambda_7 P_7 + \lambda_{5,7} P_5 + \mu_{11,7} P_{11} \\ -\lambda_8 P_8 + \lambda_{5,8} P_5 + \lambda_{6,8} P_6 + \lambda_{9,8} P_9 + \mu_{11,8} P_{11} \\ -\lambda_9 P_9 + \lambda_{6,9} P_6 + \lambda_{7,9} P_7 + \mu_{11,9} P_{11} \\ -\lambda_{10} P_{10} + \lambda_{8,10} P_8 + \lambda_{9,10} P_9 + \mu_{11,10} P_{11} \\ -S\lambda_{11} P_{11} + \lambda_{1,11} P_1 + \lambda_{2,11} P_2 + \text{------} + \lambda_{10,11} P_{10} \\ \lambda_{10,12} P_{10} \end{bmatrix}$$

where $[\dot{P}]' = [\dot{P}_1 \; \dot{P}_2 \; \dot{P}_3 \; \dot{P}_4 \; \dot{P}_5 \; \dot{P}_6 \; \dot{P}_7 \; \dot{P}_8 \; \dot{P}_9 \; \dot{P}_{10} \; \dot{P}_{11} \; \dot{P}_{12}]$

and $\lambda_{i,j} = \lambda_i \, p_{i,j}$ $\quad i = 1,2,\text{---} \, n$ $\quad$ and $S = p_{11,1} + p_{11,2} + p_{11,3} + \text{---} + p_{11,1}$

$\cdot \; \mu_{i,j} = \lambda_i \, p_{i,j}$ $\quad j = 1,2,\text{---} \, n$

Figure 5.4.

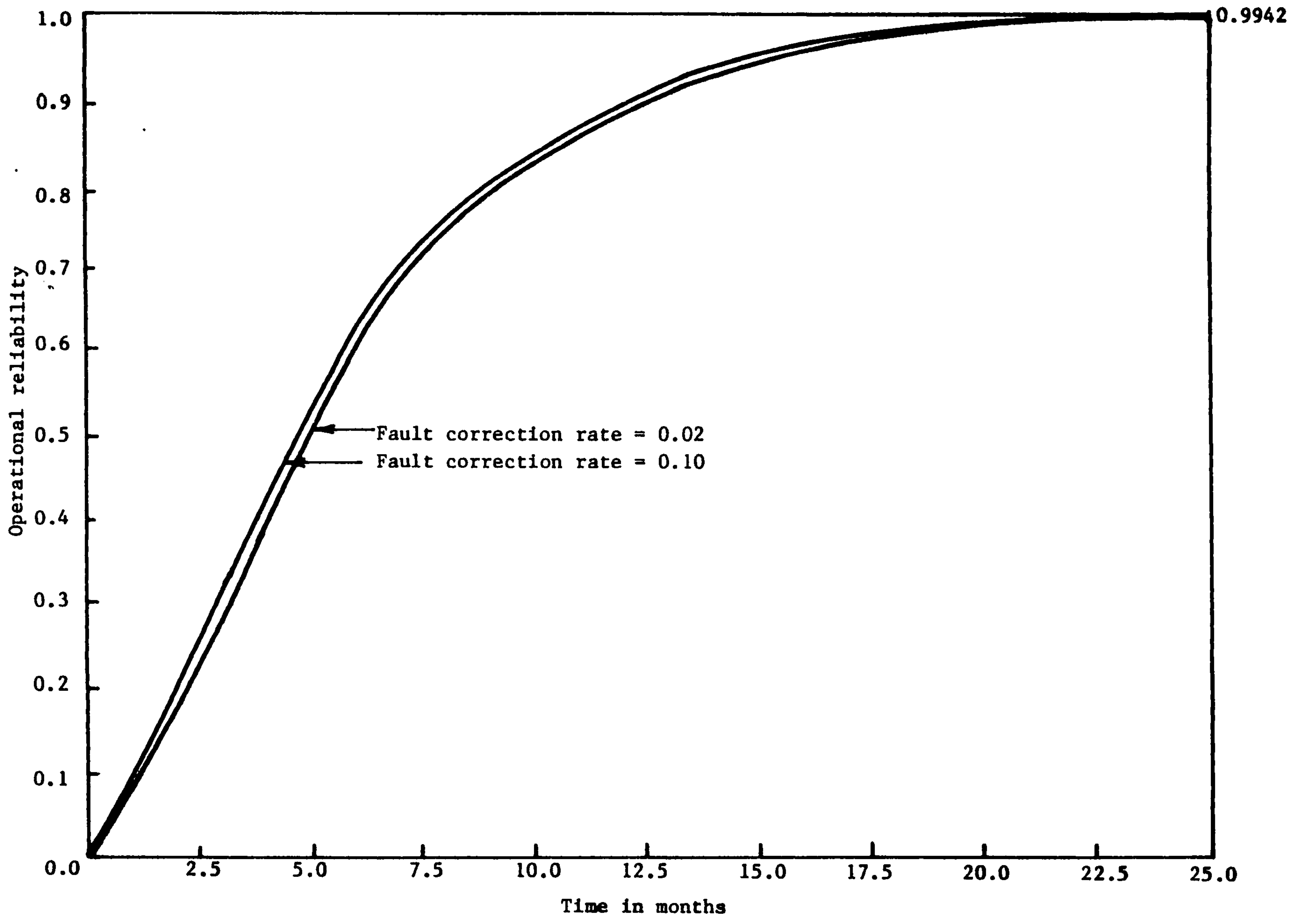

Figure 5.5. Operational reliabilities for 12 states modified model at different fault correction rates.

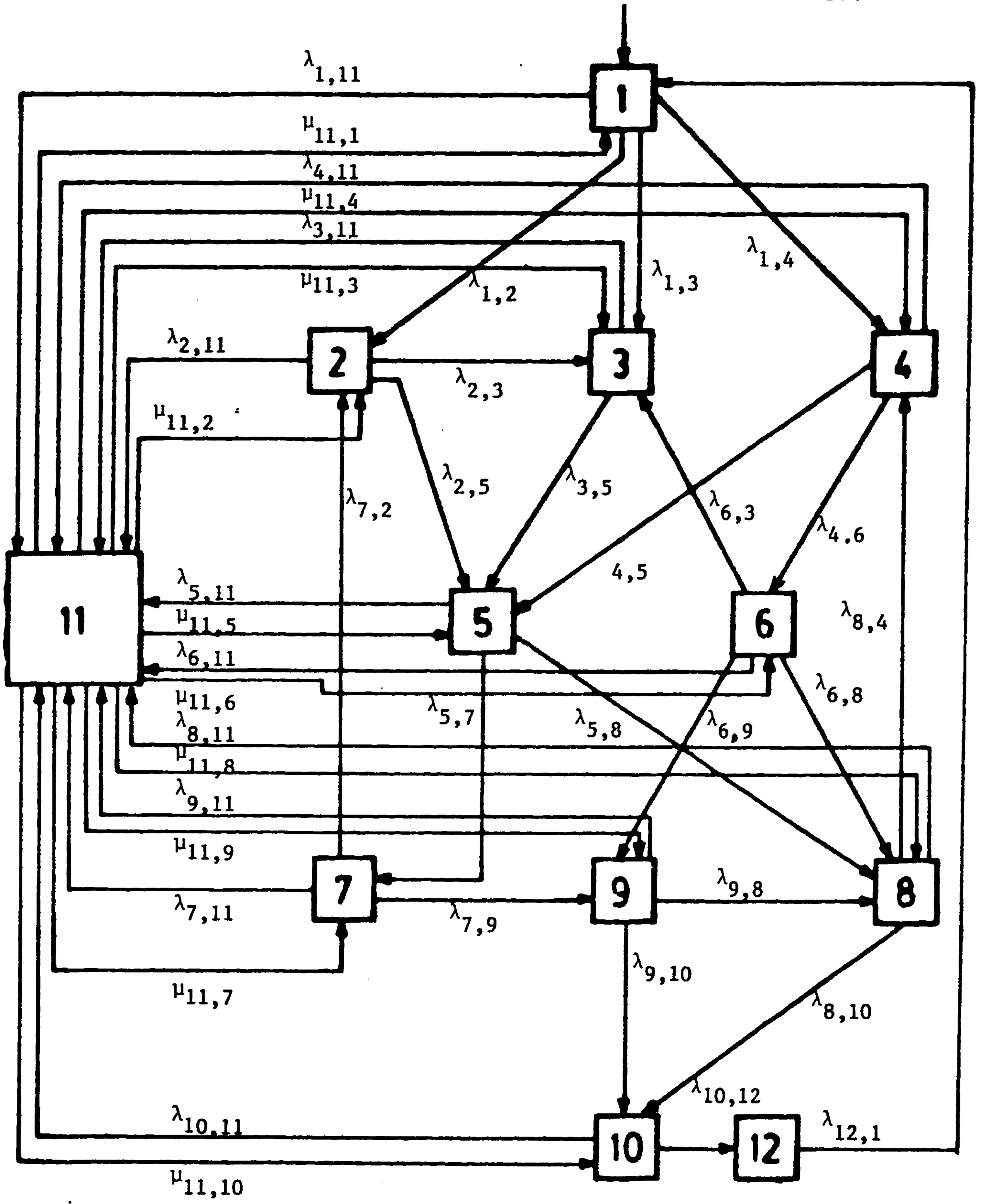

Figure 5.6. Modified 12 states model for operational availability.

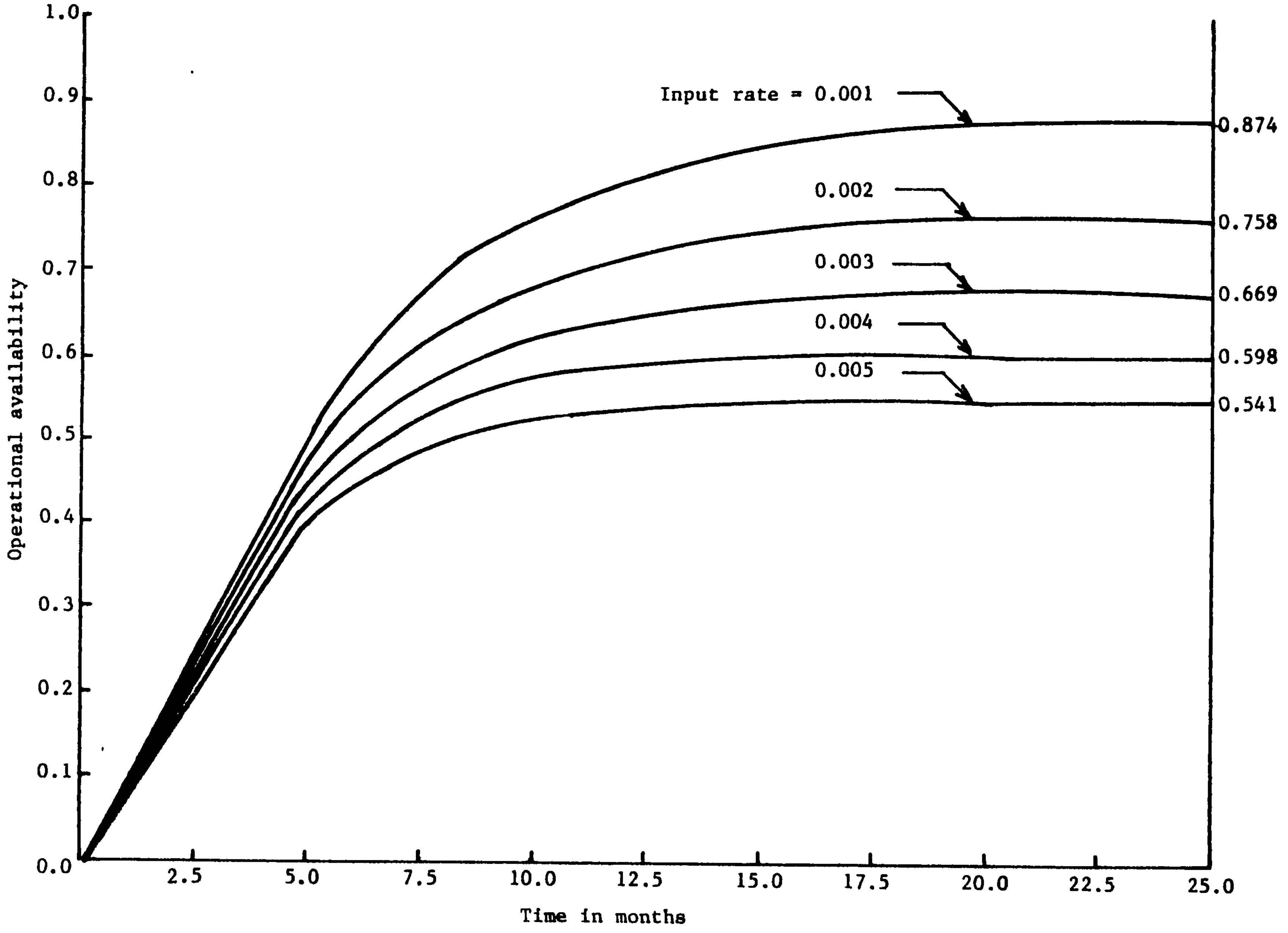

Figure 5.7. Operational availabilities for 12 states modified model at different input rates.

(Appendix B).

The computer program developed for the above mentioned estimations is given in Appendix C.

## 5.3 MODIFIED (2n+2) STATE MARKOV MODEL WITH FAULT CORRECTION

This model takes into account a non constant fault correction rate from state F which is the failure state. Using the method of stages an intermediate hypothetical state is added and a new model is obtained as in the previous chapter by employing similar technique.

The example treated in the last section is used here as well for illustration. The modified model is shown in Fig. 5.8 and the matrix $\Lambda$ for this model is shown in Fig. 5.9. After obtaining the transition rate matrix and state equations, operational reliability and availability are obtained as a function of time in Fig. 5.10 and Fig. 5.11 respectively and the corresponding values are tabulated in Tables X and XI respectively. Comparison between discrete and continuous values obtained are shown in Tables XII and XIII.

The computer program developed for the above mentioned estimations is given in Appendix C.

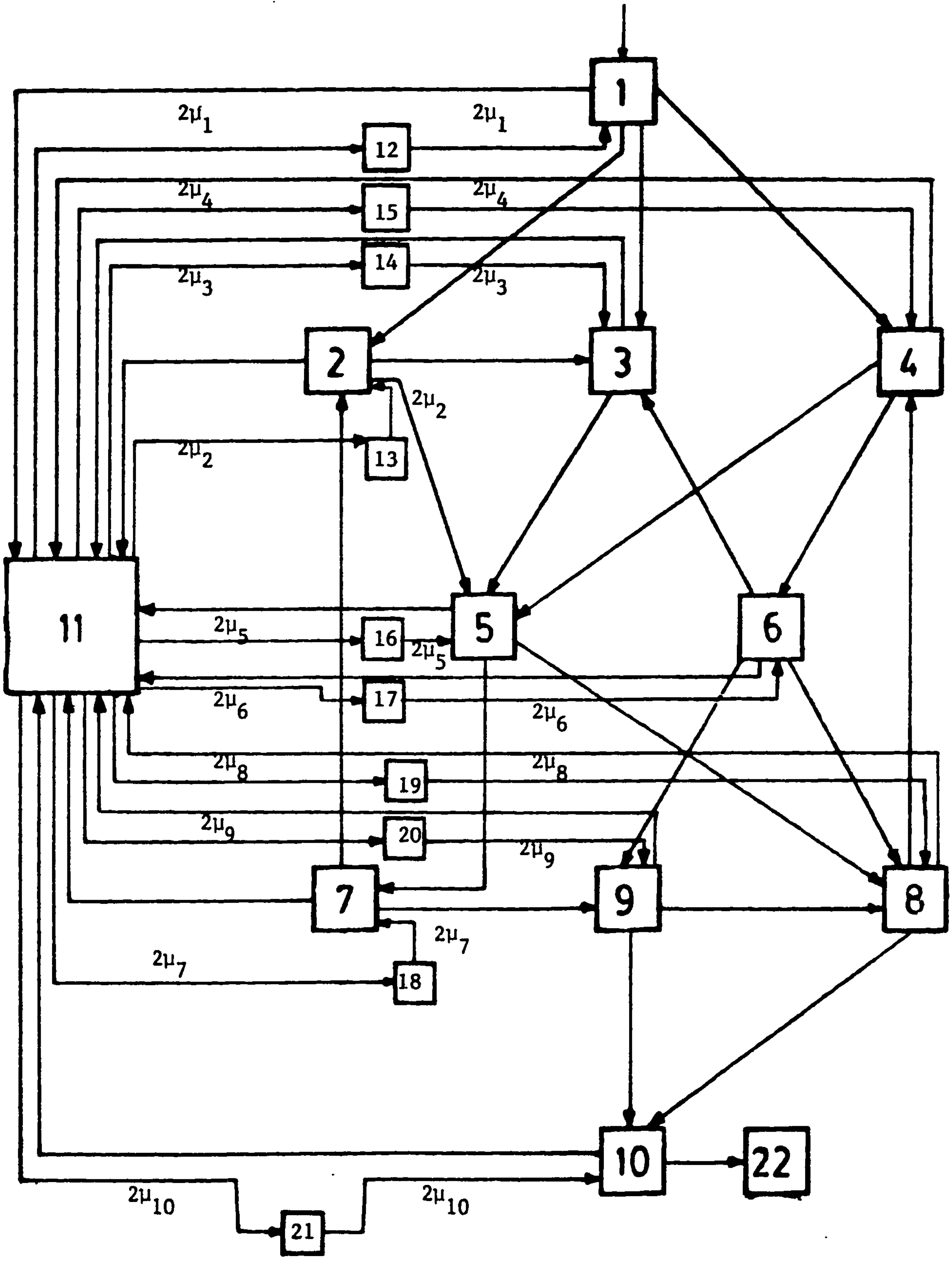

Figure 5.8. Modified model for 22 states.

$$\Lambda = \begin{bmatrix} \lambda_1 & 0.0 & \cdots & 0.0 \\ 0.0 & \lambda_2 & \cdots & 0.0 \\ \vdots & \vdots & & \vdots \\ 0.0 & 0.0 & \cdots & \lambda_{22} \end{bmatrix}$$

or

$$\left[\begin{array}{cccccccccccccccccccccc}
0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.38 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.72 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.51 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.35 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.11 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.11 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.11 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.11 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.11 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.11 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0
\end{array}\right]$$

Figure 5.9. Matrix $\Lambda$ for 22 states model.

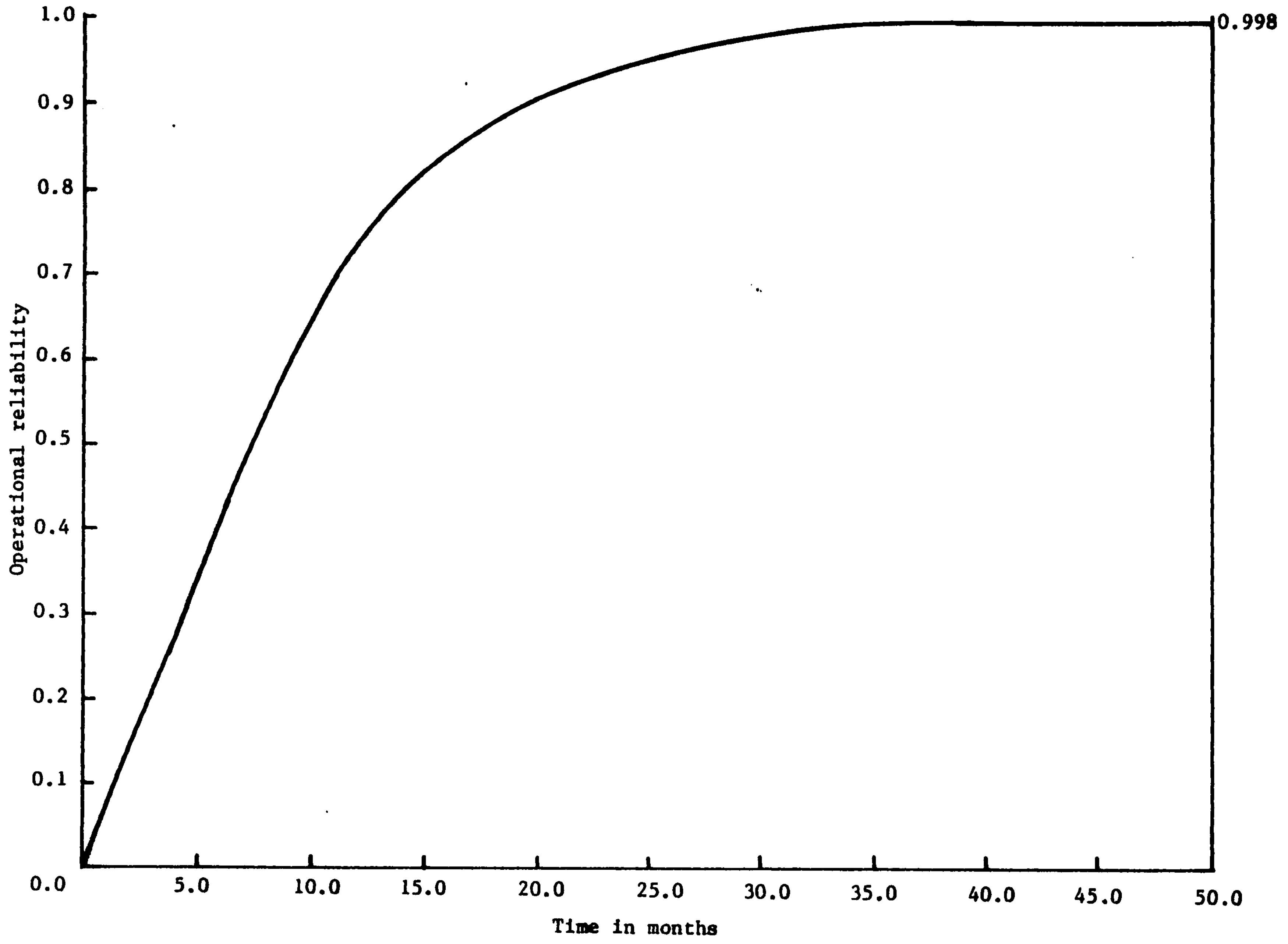

Figure 5.10. Operational reliability for 22 states modified model.

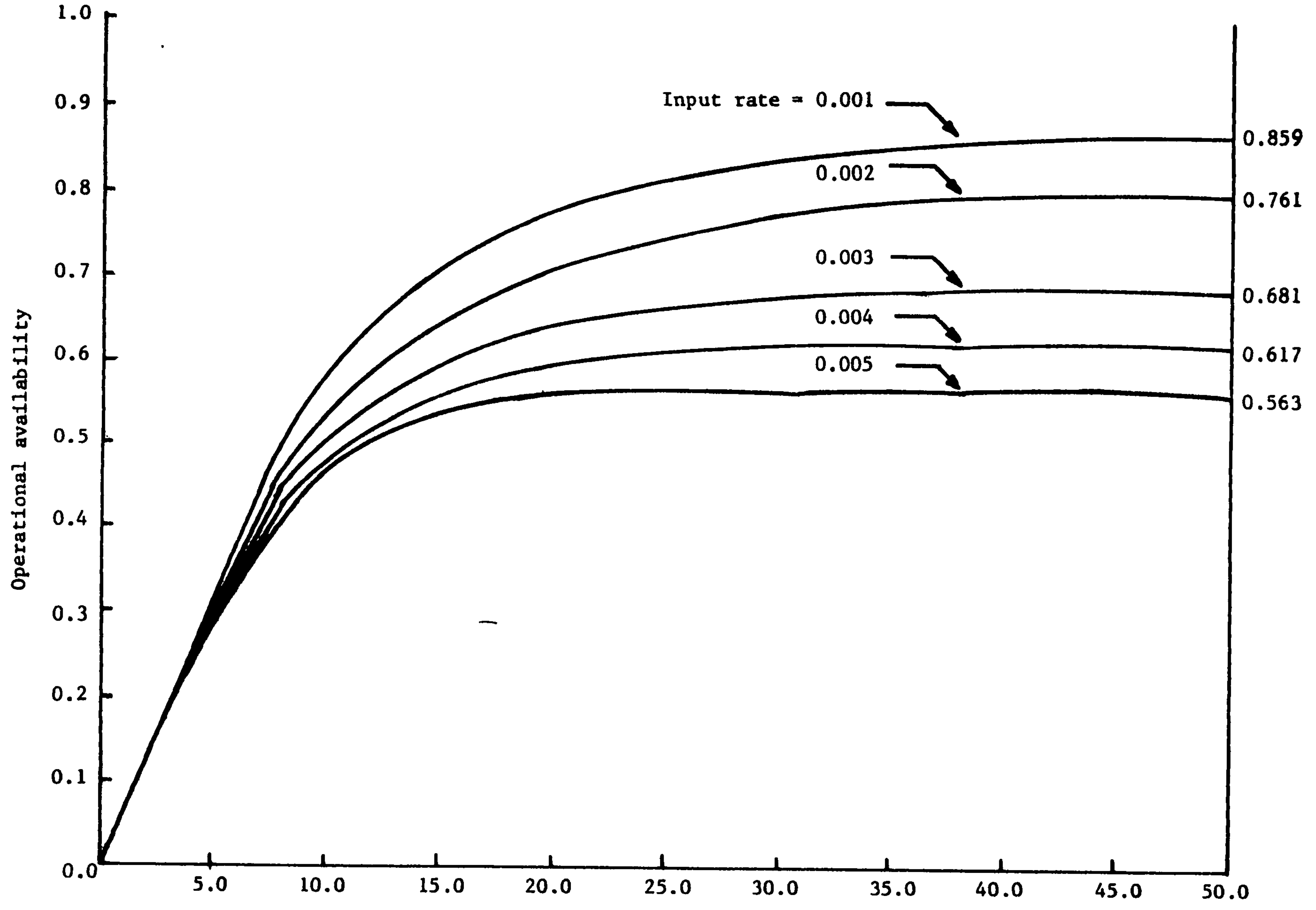

Figure 5.11. Operational availabilities at different input rates for 22 states modified model.

# 6. EXPECTED COST ESTIMATION

## 6.1 INTRODUCTION

Using the proposed models of the previous chapters, we can describe the overall failure process of a software system. This can be done by treating consequences of failures rather than merely the failures themselves. When module i, is occupied failure occurs with rate typical of subprogram i. Associated with such a failure is a random variable $Y_i$, representing the cost of that failure having a distribution function $G_i(t)$ and density function $g_i(t)$. The actual quantity of interest is the total cost of running the software for a time t.

The total expected program cost is given by the expression

$$Y(t) = \sum_{i=1}^{n} Y_i(t) + \sum_{\substack{i=1 \\ i \neq k}}^{n} \sum_{k=1}^{n} Y_{ik}(t) \tag{6.1}$$

where

$Y_{ik}$ represent the cost of a failure when module i calls module k (i to k transition), with cumulative distribution function $H_{ik}(t)$ and density function $h_{ik}(t)$. $Y_i(t)$ is the cost of failures of module i during (o,t), and $Y_{ik}(t)$ the total cost of failures of the $i \to k$ interface.

The expected cost due to failures in control transfer can be represented as follows:

$$Y_{ik}(t) = [Y_i(t) + Y_k(t)]\, \phi_{ik}(t)$$
$$i \neq k$$

$$i = 1,2, \text{----}\ n$$
$$k = 1,2, \text{---}\ n \qquad (6.2)$$

Thus the interval transition probability can be used effectively to represent the cost of failure for a transition within the program.

In this chapter we have estimated the expected cost of a program as a function of time using the above mentioned equations. Various computer techniques can be used to solve these equations and in this chapter digital simulation technique has been used.

The relationship between $Y_i(t)$ and $g_i(t)$ is defined as follows:

$$Y_i(t) = C_i\ g_i(t) \qquad (6.3)$$

where $C_i$ is a constant to relate the cost density function to the cost function. The value of $C_i$ being dependent upon the importance of a program module i. A similar relationship between the expected cost due to failures in control transfer and its density function is defined as below.

$$Y_{ik}(t) = d_{ik}\, h_{ik}(t) \tag{6.4}$$

Where $d_{ik}$ is a constant dependent upon $C_j$ where j varies from i to k.

It is clear now, that when the cost density function $g_i(t)$ of each program module is known the cost function can be determined by (6.3). Then by using (6.2) we can evaluate (6.1) and by using (6.4), $d_{ik}$ and $h_{ik}(t)$ can be determined.

## 6.2 THE TEN MODULE PROGRAM EXAMPLE

The actual cost estimation of ten module program example of last chapters is described in this section.

We define ten density functions for the ten program modules as follows:

$$g_1(t) = \frac{1}{20} e^{-t/20} \quad 0 < t \leq +\infty \text{ exponential density function}$$

$$g_2(t) = g_4(t) = \frac{3t^2}{500} e^{-3t^3/1000} \quad 0 < t \leq +\infty \text{ Weibull density function}$$

$$g_3(t) = g_5(t) = \frac{t}{10} e^{-2t^2/500} \quad 0 < t \leq +\infty \text{ Weibull density function}$$

$$g_6(t) = \frac{t}{20} e^{-t^2/200} \quad 0 < t \leq +\infty \text{ Rayleigh density function}$$

$$g_7(t) = \frac{1}{2!} \frac{t}{20} e^{-t/20} \quad 0 < t \leq +\infty \text{ Gamma density function}$$

$$g_8(t) = g_9(t) = \frac{1}{0.5\sqrt{\pi}} (\frac{t}{20})^{.5} e^{-t/20} \quad 0 < t \leq +\infty \text{ Gamma density function}$$

$$g_{10}(t) = \frac{1}{3!} (\frac{t}{20})^2 e^{-t/20} \quad 0 < t \leq +\infty \text{ Gamma density function}$$

(6.5)

These density functions in turn give the cost function as given by (6.3) where i varies from 1 to 10. These density functions have been plotted and shown in Fig. 6.1 and Fig. 6.2. An arbitrary selection of $C_i$ is done as follows:

$$C_5 = C_9 = 2C, \quad C_1 = 40C$$

$$C_2 = C_3 = C_8 = C_{10} = C$$

$$C_4 = C_6 = C_7 = 3C \qquad (6.6)$$

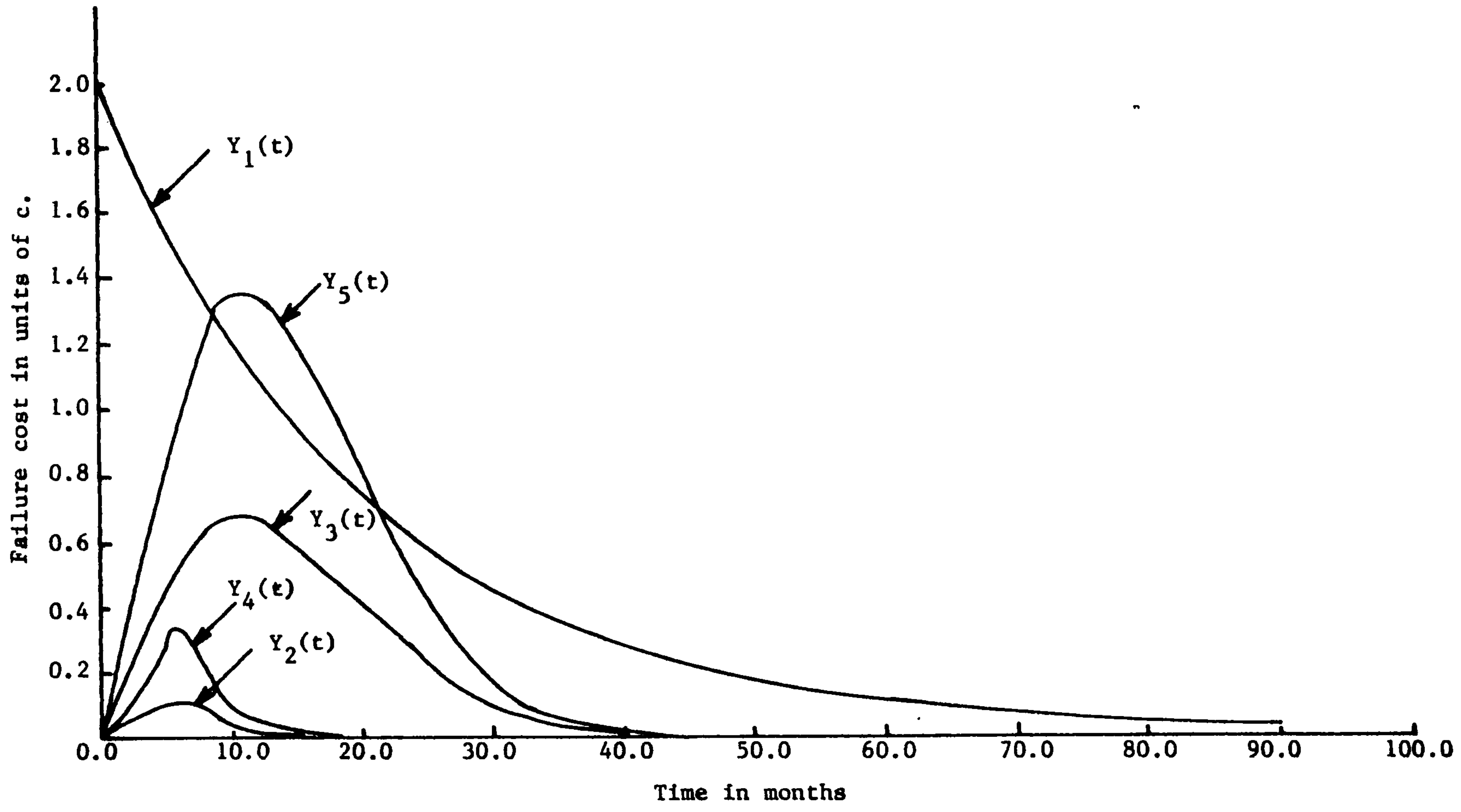

Figure 6.1. Cost functions for program modules 1-5.

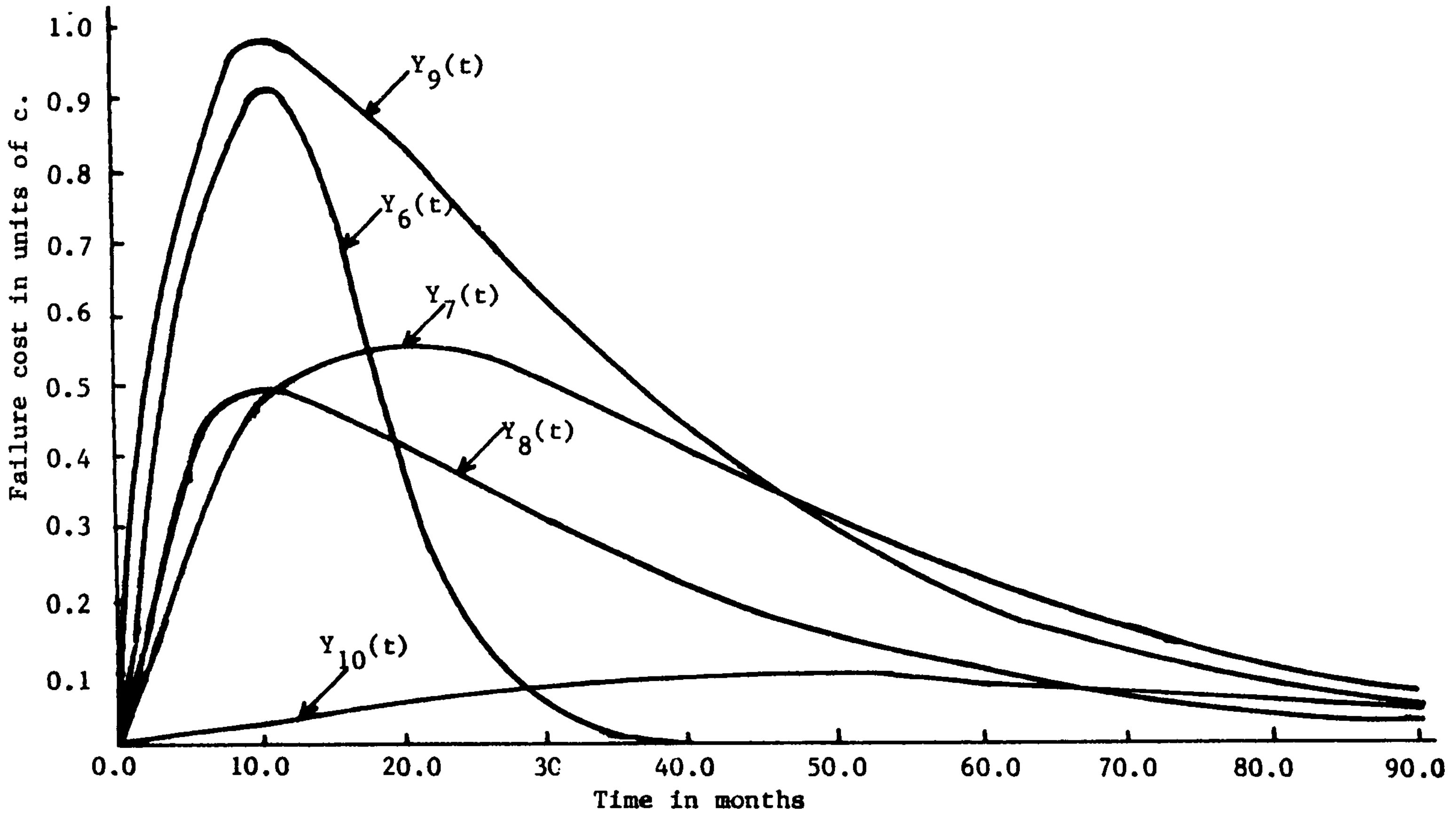

Figure 6.2. Cost functions for program modules 6-10.

Where C is a constant cost parameter in terms of money and other costs or consequences of failures. Putting the values found by (6.3) into (6.2) the expected cost due to failures in control transfer is estimated as a function of time and is plotted in Fig. 6.3 which shows the plot of $Y_{1,10}(t)$. It is worth noting that all the hypothetical states (states other than the module itself) are assumed to have zero cost of failure due to their physical non existence but their effect will be taken into account by interval transition probability. By using (6.1) the total expected program cost as a function of time is obtained and is shown in Fig. 6.4, some of the values are listed in Table XIV (Appendix B) and the program used is shown in Appendix C.

## 6.3 CONCLUSIONS

Two important quantities, the total expected program cost and expected cost due to failures in control transfer are estimated and plotted in this chapter. The presented technique can be useful for cost estimation of modular program structures. By defining the cost density function for individual module the cost function can be obtained quite effectively by using (6.3) where the constant $C_i$ can be used to place more emphasis on some important program modules. Therefore, its selection can be useful in establishing a module importance within a software e.g. in a computer controlled aeroplane landing system it may be more important for the software to cause a

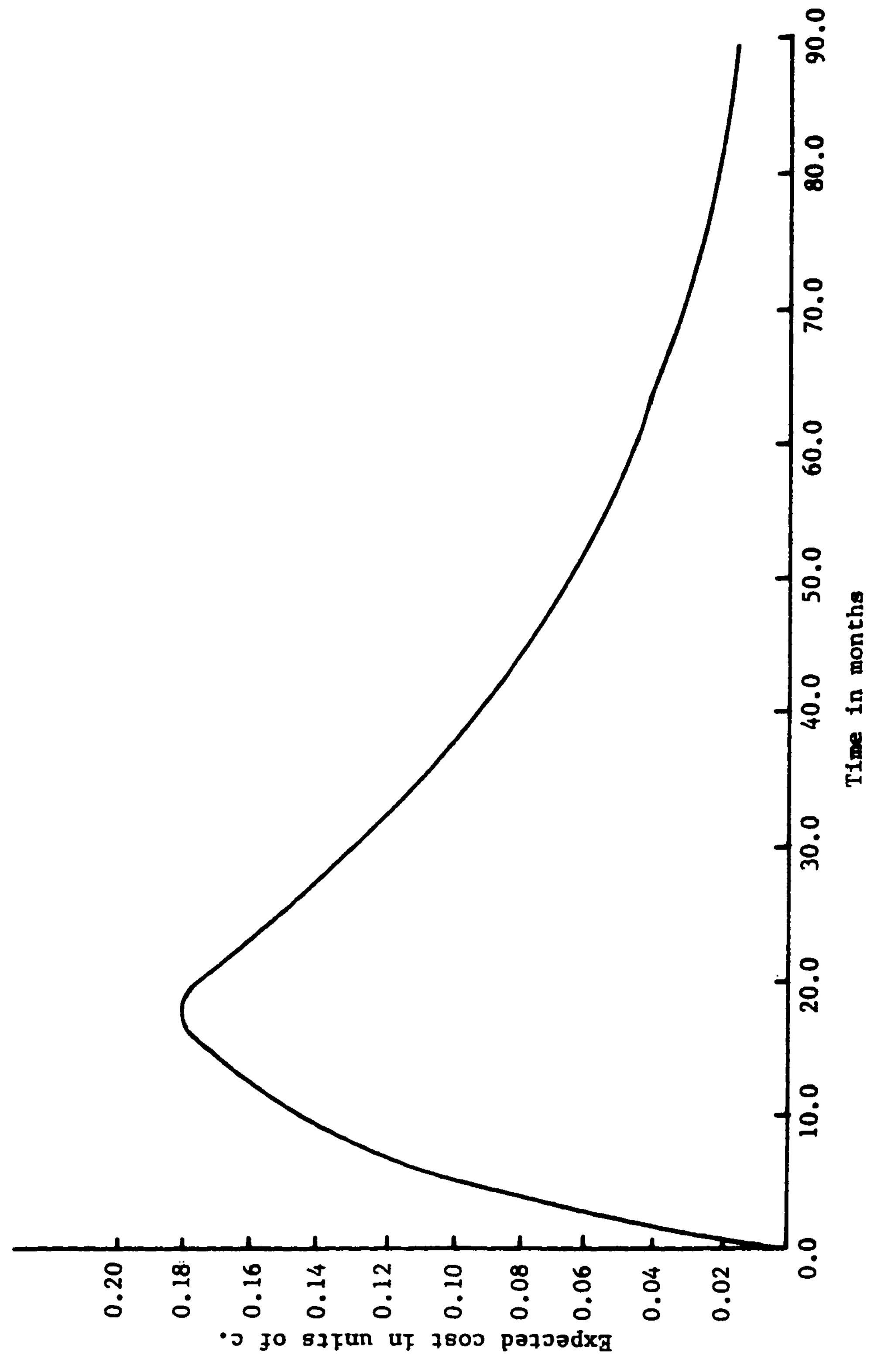

Figure 6.3. Expected cost function due to failures in control transfer for 10 module program example .

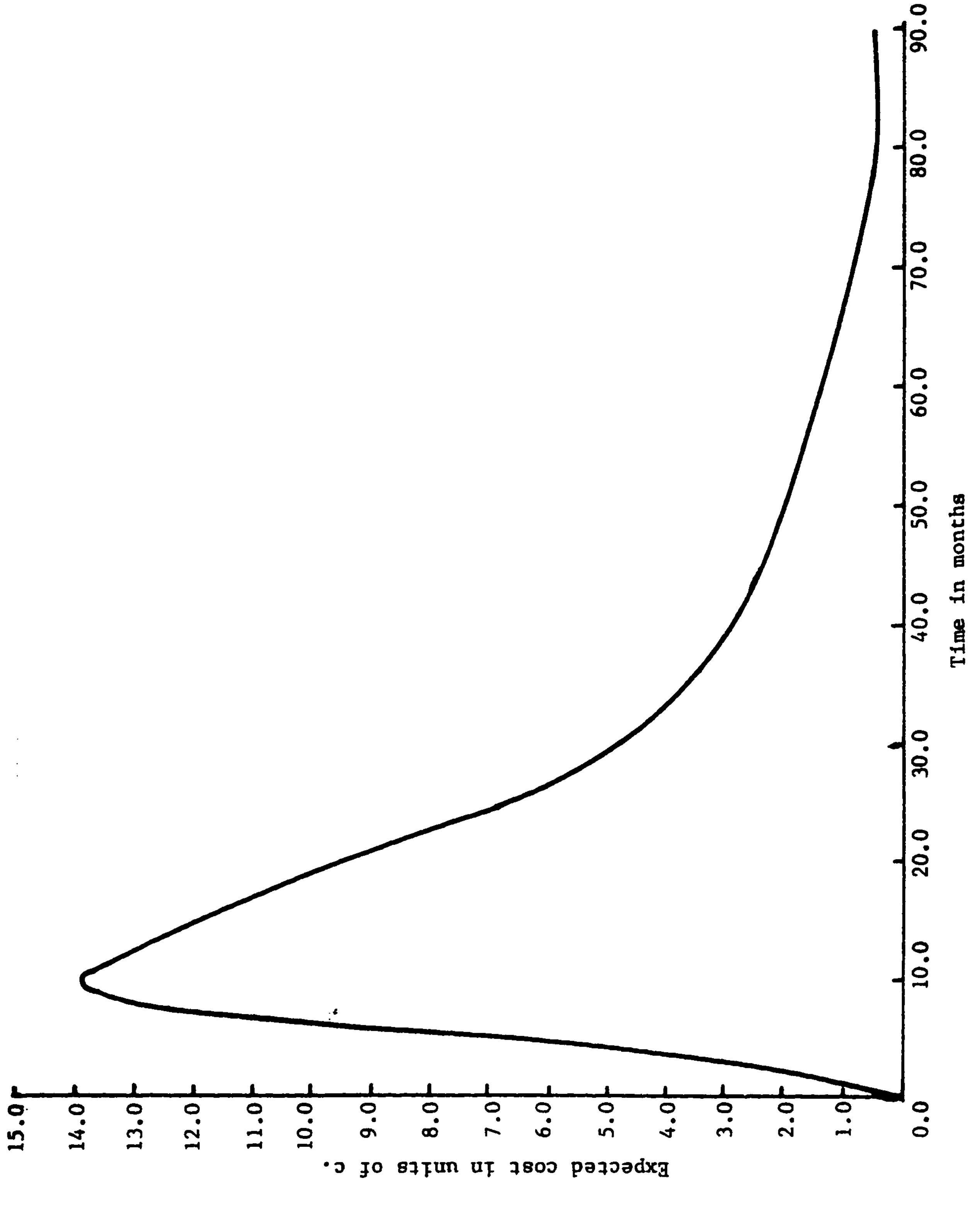

Figure 6.4. Total expected cost function for 10 module program example.

proper ground impact than to give accurate readings at the final moments of landing (approximate readings may be sufficieht). Perhaps the most important portion of software would be that which performs safety operations during emergency such as fire or engine failure.

# 7. CONCLUSIONS AND FUTURE RECOMMENDATIONS

In this thesis Markov and semi Markov models have been developed for quantitative evaluation of software reliability. The models have been presented first in discrete time format and then they are converted to continuous time domain. Two important figures of merit termed as operational reliability and operation availability are obtained. Operational reliability provides us a confidence level of a program, without considering the effect of various input conditions. Different sets of random input conditions is taken into account by 'input rate' and the confidence level thus achieved is termed operational availability. The models developed are tested using an actual ten module program and the defined quantities are evaluated as a function of time in both discrete and continuous time systems. Quantities like mean waiting time and interval transition probabilities are evaluated and discussed in semi Markov model. These quantitative estimations were shown to be useful to develop an effective testing strategy given limited testing resources.

In addition to this the overall failure process of the program is defined. This is done by treating consequences of failures rather than the failures themselves. Quantities like expected cost due to failures in control transfer and the total expected cost are evaluated.

The quantitative models which have been developed in this thesis can be further extended. The overall failure cost process can be modified and also extended. Monte Carlo simulation technique can be used to give a better picture of the overall failure cost process instead of the digital simulation technique presented in the thesis. Sensitivity analysis can also be carried out to check the effect of a particular module on the overall failure cost. The two quantities, reliability and failure cost of the program can then be optimized given module reliabilities and module failure cost thus causing an effective software design within constraints of limited time, budget and labor.

Furthermore, in the models discussed the 'input rate' could be taken to be described by any general density function rather than the exponential density function as assumed. This could be particularly realistic in situations where input data comes from a highly random environment. By introducing intermediate hypothetical states the non exponential density function describing the random input data, can be treated in the same way as it is done for non exponential fault correction process. More general density function could be simulated by the method of stages [43] e.g. density functions which give rise to a rising and falling hazard rate functions.

## APPENDIX 'A'

### Structured Programming:

The concept called structured programming has had such an impact on software development that it will probably be recorded in history as one of the great steps forward in programming technology (along with the subroutine and high-level language concepts).

Structured programming is defined as follows: "It is the attitude of writing code with the intent of communicating with people instead of machines".

Structured programs are constructed from sets of the five basic building blocks shown in Fig. A.1. The degree to which structured programming can be achieved is of course, dependent on the programming language used, of the widely used languages PL/I, PL/C, PASCAL, and ALGOL are most suited for structured programming. Structured programming can also be used with the COBOL, FORTRAN and BASIC languages, although with more difficulty and a few undesirable results.

Structured programming in FORTRAN and BASIC is quite difficult because they lack the IF THEN ELSE, DOWHILE, DOUNTIL, and CASE constructs. This forces the designer to use the GOTO statement. But we can make a FORTRAN program appear to be structured by simulating the IFTHENELSE and CASE constructs with a

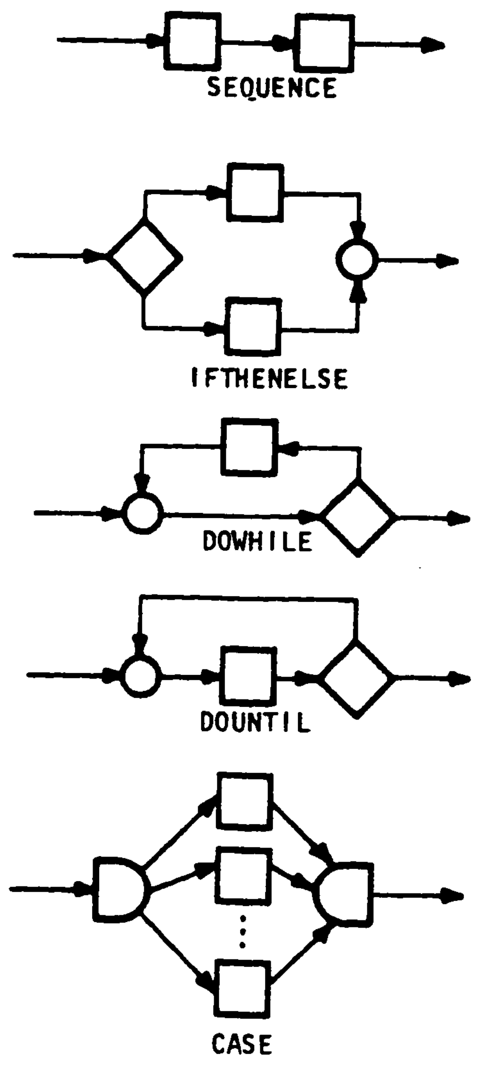

Figure A.1. The five structured programming constructs.

combination of indenting and comment statements. The example shown below is a FORTRAN IFTHENELSE construct.

```
      IF     I = 2          GO TO 4
                            GO TO 5
C          THEN
    4 A = 3
      B = 2
                            GO TO 10
C          ELSE
    5 A = 2
      B = 3
   10 CONTINUE
```

The code within the box has the appearances of a proper IF construct and the code outside of the box adds the necessary detail to make the construct a valid FORTRAN program.

## APPENDIX B

### TABLE I

### CHEUNG'S MODEL

IE PROBABILITIES OF BEING IN DIFFERENT STATES:

| IME | STATE 1 | STATE 2 | STATE 3 | STATE 4 | STATE 5 | STATE 6 |
|---|---|---|---|---|---|---|
| 1 | 0.0 | 0.60E+00 | 0.20E+00 | 0.20E+00 | 0.0 | 0.0 |
| 2 | 0.0 | 0.0 | 0.41E+00 | 0.0 | 0.45E+00 | 0.12E+00 |
| 3 | 0.0 | 0.0 | 0.35E-01 | 0.0 | 0.41E+00 | 0.0 |
| 4 | 0.0 | 0.10E+00 | 0.0 | 0.64E-01 | 0.34E-01 | 0.0 |
| 5 | 0.0 | 0.76E-01 | 0.70E-01 | 0.56E-01 | 0.55E-01 | 0.37E-01 |
| 6 | 0.0 | 0.64E-02 | 0.63E-01 | 0.70E-02 | 0.11E+00 | 0.33E-01 |
| 7 | 0.0 | 0.16E-01 | 0.14E-01 | 0.10E-01 | 0.67E-01 | 0.41E-02 |
| 8 | 0.0 | 0.26E-01 | 0.12E-01 | 0.16E-01 | 0.22E-01 | 0.58E-02 |
| 9 | 0.0 | 0.13E-01 | 0.20E-01 | 0.98E-02 | 0.26E-01 | 0.96E-02 |
| 10 | 0.0 | 0.51E-02 | 0.12E-01 | 0.38E-02 | 0.27E-01 | 0.57E-02 |
| 11 | 0.0 | 0.63E-02 | 0.52E-02 | 0.41E-02 | 0.15E-01 | 0.22E-02 |
| 12 | 0.0 | 0.59E-02 | 0.50E-02 | 0.40E-02 | 0.85E-02 | 0.24E-02 |
| 13 | 0.0 | 0.31E-02 | 0.48E-02 | 0.22E-02 | 0.82E-02 | 0.23E-02 |
| 14 | 0.0 | 0.19E-02 | 0.28E-02 | 0.14E-02 | 0.65E-02 | 0.13E-02 |
| 15 | 0.0 | 0.19E-02 | 0.17E-02 | 0.13E-02 | 0.39E-02 | 0.79E-03 |
| 16 | 0.0 | 0.14E-02 | 0.15E-02 | 0.97E-03 | 0.27E-02 | 0.73E-03 |
| 17 | 0.0 | 0.84E-03 | 0.12E-02 | 0.60E-03 | 0.23E-02 | 0.56E-03 |
| 18 | 0.0 | 0.62E-03 | 0.75E-03 | 0.43E-03 | 0.16E-02 | 0.35E-03 |
| 19 | 0.0 | 0.51E-03 | 0.53E-03 | 0.35E-03 | 0.11E-02 | 0.25E-03 |
| 20 | 0.0 | 0.36E-03 | 0.43E-03 | 0.25E-03 | 0.81E-03 | 0.20E-03 |
| 21 | 0.0 | 0.24E-03 | 0.31E-03 | 0.17E-03 | 0.62E-03 | 0.15E-03 |
| 22 | 0.0 | 0.18E-03 | 0.21E-03 | 0.12E-03 | 0.44E-03 | 0.97E-04 |
| 23 | 0.0 | 0.14E-03 | 0.15E-03 | 0.95E-04 | 0.31E-03 | 0.72E-04 |
| 24 | 0.0 | 0.97E-04 | 0.12E-03 | 0.67E-04 | 0.23E-03 | 0.55E-04 |
| 25 | 0.0 | 0.68E-04 | 0.83E-04 | 0.47E-04 | 0.17E-03 | 0.39E-04 |

## TABLE II

## CHEUNG'S MODEL

| IME | STATE 7 | STATE 8 | STATE 9 | STATE 10 | STATE 11 | STATE 12 |
|---|---|---|---|---|---|---|
| 1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.21E-01 | 0.0 |
| 3 | 0.21E+00 | 0.27E+00 | 0.35E-01 | 0.0 | 0.48E-01 | 0.0 |
| 4 | 0.15E+00 | 0.24E+00 | 0.10E+00 | 0.22E+00 | 0.86E-01 | 0.0 |
| 5 | 0.13E-01 | 0.29E-01 | 0.76E-01 | 0.26E+00 | 0.11E+00 | 0.22E+00 |
| 6 | 0.32E-01 | 0.42E-01 | 0.18E-01 | 0.88E-01 | 0.13E+00 | 0.47E+00 |
| 7 | 0.53E-01 | 0.69E-01 | 0.25E-01 | 0.46E-01 | 0.14E+00 | 0.56E+00 |
| 8 | 0.27E-01 | 0.41E-01 | 0.27E-01 | 0.72E-01 | 0.15E+00 | 0.60E+00 |
| 9 | 0.10E-01 | 0.16E-01 | 0.15E-01 | 0.53E-01 | 0.15E+00 | 0.67E+00 |
| 10 | 0.13E-01 | 0.17E-01 | 0.79E-02 | 0.25E-01 | 0.16E+00 | 0.73E+00 |
| 11 | 0.12E-01 | 0.17E-01 | 0.80E-02 | 0.19E-01 | 0.16E+00 | 0.75E+00 |
| 12 | 0.63E-02 | 0.94E-02 | 0.66E-02 | 0.19E-01 | 0.16E+00 | 0.77E+00 |
| 13 | 0.40E-02 | 0.57E-02 | 0.38E-02 | 0.12E-01 | 0.17E+00 | 0.79E+00 |
| 14 | 0.38E-02 | 0.53E-02 | 0.26E-02 | 0.74E-02 | 0.17E+00 | 0.80E+00 |
| 15 | 0.29E-02 | 0.41E-02 | 0.23E-02 | 0.61E-02 | 0.17E+00 | 0.81E+00 |
| 16 | 0.17E-02 | 0.25E-02 | 0.16E-02 | 0.49E-02 | 0.17E+00 | 0.81E+00 |
| 17 | 0.13E-02 | 0.18E-02 | 0.11E-02 | 0.32E-02 | 0.17E+00 | 0.82E+00 |
| 18 | 0.10E-02 | 0.15E-02 | 0.79E-03 | 0.22E-02 | 0.17E+00 | 0.82E+00 |
| 19 | 0.73E-03 | 0.11E-02 | 0.62E-03 | 0.17E-02 | 0.17E+00 | 0.82E+00 |
| 20 | 0.49E-03 | 0.70E-03 | 0.43E-03 | 0.13E-02 | 0.17E+00 | 0.83E+00 |
| 21 | 0.37E-03 | 0.52E-03 | 0.30E-03 | 0.88E-03 | 0.17E+00 | 0.83E+00 |
| 22 | 0.28E-03 | 0.40E-03 | 0.22E-03 | 0.64E-03 | 0.17E+00 | 0.83E+00 |
| 23 | 0.20E-03 | 0.28E-03 | 0.17E-03 | 0.48E-03 | 0.17E+00 | 0.83E+00 |
| 24 | 0.14E-03 | 0.20E-03 | 0.12E-03 | 0.35E-03 | 0.17E+00 | 0.83E+00 |
| 25 | 0.10E-03 | 0.15E-03 | 0.85E-04 | 0.25E-03 | 0.17E+00 | 0.83E+00 |

## TABLE III

### (N+2) STATE MODEL WITH FAULT CORRECTION

### STATE F PROBABILITIES AT VARIOUS FAULT CORRECTION RATES

| TIME | CORRECTION RATE | | | | |
|---|---|---|---|---|---|
| | 0.02 | 0.04 | 0.06 | 0.08 | 0.10 |
| .1000E+01 | 0.1000E-02 | 0.1000E-02 | 0.1000E-02 | 0.1000E-02 | 0.1000E-02 |
| .2000E+01 | 0.2078E-01 | 0.2058E-01 | 0.2038E-01 | 0.2018E-01 | 0.1998E-01 |
| .3000E+01 | 0.4390E-01 | 0.3963E-01 | 0.3544E-01 | 0.3132E-01 | 0.2729E-01 |
| .4000E+01 | 0.7333E-01 | 0.6208E-01 | 0.5257E-01 | 0.4475E-01 | 0.3857E-01 |
| .5000E+01 | 0.8458E-01 | 0.6342E-01 | 0.4741E-01 | 0.3551E-01 | 0.2670E-01 |
| .6000E+01 | 0.8252E-01 | 0.5338E-01 | 0.3465E-01 | 0.2305E-01 | 0.1614E-01 |
| .7000E+01 | 0.7811E-01 | 0.4475E-01 | 0.2698E-01 | 0.1797E-01 | 0.1350E-01 |
| .8000E+01 | 0.7345E-01 | 0.3848E-01 | 0.2275E-01 | 0.1569E-01 | 0.1215E-01 |
| .9000E+01 | 0.6670E-01 | 0.3164E-01 | 0.1787E-01 | 0.1196E-01 | 0.8815E-02 |
| .1000E+02 | 0.5913E-01 | 0.2526E-01 | 0.1352E-01 | 0.8753E-02 | 0.6329E-02 |
| .1100E+02 | 0.5223E-01 | 0.2043E-01 | 0.1070E-01 | 0.6986E-02 | 0.5186E-02 |
| .1200E+02 | 0.4591E-01 | 0.1660E-01 | 0.8566E-02 | 0.5608E-02 | 0.4156E-02 |
| .1300E+02 | 0.3998E-01 | 0.1331E-01 | 0.6671E-02 | 0.4282E-02 | 0.3105E-02 |
| .1400E+02 | 0.3463E-01 | 0.1063E-01 | 0.5181E-02 | 0.3282E-02 | 0.2371E-02 |
| .1500E+02 | 0.2996E-01 | 0.8539E-02 | 0.4094E-02 | 0.2593E-02 | 0.1886E-02 |
| .1600E+02 | 0.2586E-01 | 0.6859E-02 | 0.3229E-02 | 0.2032E-02 | 0.1467E-02 |
| .1700E+02 | 0.2226E-01 | 0.5485E-02 | 0.2523E-02 | 0.1566E-02 | 0.1119E-02 |
| .1800E+02 | 0.1914E-01 | 0.4388E-02 | 0.1976E-02 | 0.1217E-02 | 0.8677E-03 |
| .1900E+02 | 0.1644E-01 | 0.3518E-02 | 0.1556E-02 | 0.9530E-03 | 0.6789E-03 |
| .2000E+02 | 0.1411E-01 | 0.2818E-02 | 0.1222E-02 | 0.7419E-03 | 0.5251E-03 |
| .2100E+02 | 0.1210E-01 | 0.2255E-02 | 0.9572E-03 | 0.5754E-03 | 0.4048E-03 |
| .2200E+02 | 0.1037E-01 | 0.1805E-02 | 0.7513E-03 | 0.4482E-03 | 0.3144E-03 |
| .2300E+02 | 0.8884E-02 | 0.1446E-02 | 0.5904E-03 | 0.3496E-03 | 0.2444E-03 |
| .2400E+02 | 0.7610E-02 | 0.1157E-02 | 0.4633E-03 | 0.2720E-03 | 0.1891E-03 |
| .2500E+02 | 0.6517E-02 | 0.9265E-03 | 0.3634E-03 | 0.2115E-03 | 0.1463E-03 |

TABLE IV

(N+2) STATE MODEL WITH FAULT CORRECTION

OPERATIONAL RELIABILITIES AT VARIOUS FAULT CORRECTION RATES

| TIME | CORRECTION RATE | | | | |
|---|---|---|---|---|---|
| | 0.02 | 0.04 | 0.06 | 0.08 | 0.10 |
| 0.1000E+01 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.2000E+01 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.3000E+01 | 0.1970E-04 | 0.3940E-04 | 0.5910E-04 | 0.7880E-04 | 0.9850E-04 |
| 0.4000E+01 | 0.4604E-03 | 0.9129E-03 | 0.1358E-02 | 0.1794E-02 | 0.2223E-02 |
| 0.5000E+01 | 0.2208E+00 | 0.2226E+00 | 0.2242E+00 | 0.2256E+00 | 0.2269E+00 |
| 0.6000E+01 | 0.4771E+00 | 0.4816E+00 | 0.4852E+00 | 0.4882E+00 | 0.4906E+00 |
| 0.7000E+01 | 0.5693E+00 | 0.5775E+00 | 0.5837E+00 | 0.5882E+00 | 0.5917E+00 |
| 0.8000E+01 | 0.6217E+00 | 0.6343E+00 | 0.6428E+00 | 0.6486E+00 | 0.6526E+00 |
| 0.9000E+01 | 0.7016E+00 | 0.7186E+00 | 0.7290E+00 | 0.7355E+00 | 0.7397E+00 |
| 0.1000E+02 | 0.7646E+00 | 0.7855E+00 | 0.7972E+00 | 0.8040E+00 | 0.8082E+00 |
| 0.1100E+02 | 0.7999E+00 | 0.8240E+00 | 0.8362E+00 | 0.8429E+00 | 0.8469E+00 |
| 0.1200E+02 | 0.8301E+00 | 0.8564E+00 | 0.8685E+00 | 0.8747E+00 | 0.8783E+00 |
| 0.1300E+02 | 0.8598E+00 | 0.8873E+00 | 0.8989E+00 | 0.9045E+00 | 0.9077E+00 |
| 0.1400E+02 | 0.8827E+00 | 0.9106E+00 | 0.9214E+00 | 0.9265E+00 | 0.9293E+00 |
| 0.1500E+02 | 0.8999E+00 | 0.9276E+00 | 0.9374E+00 | 0.9419E+00 | 0.9444E+00 |
| 0.1600E+02 | 0.9151E+00 | 0.9419E+00 | 0.9507E+00 | 0.9546E+00 | 0.9567E+00 |
| 0.1700E+02 | 0.9282E+00 | 0.9538E+00 | 0.9617E+00 | 0.9650E+00 | 0.9668E+00 |
| 0.1800E+02 | 0.9390E+00 | 0.9630E+00 | 0.9699E+00 | 0.9728E+00 | 0.9743E+00 |
| 0.1900E+02 | 0.9479E+00 | 0.9703E+00 | 0.9763E+00 | 0.9787E+00 | 0.9799E+00 |
| 0.2000E+02 | 0.9556E+00 | 0.9762E+00 | 0.9814E+00 | 0.9834E+00 | 0.9845E+00 |
| 0.2100E+02 | 0.9621E+00 | 0.9810E+00 | 0.9854E+00 | 0.9871E+00 | 0.9880E+00 |
| 0.2200E+02 | 0.9677E+00 | 0.9848E+00 | 0.9886E+00 | 0.9900E+00 | 0.9907E+00 |
| 0.2300E+02 | 0.9724E+00 | 0.9878E+00 | 0.9910E+00 | 0.9922E+00 | 0.9928E+00 |
| 0.2400E+02 | 0.9764E+00 | 0.9902E+00 | 0.9929E+00 | 0.9939E+00 | 0.9944E+00 |
| 0.2500E+02 | 0.9798E+00 | 0.9922E+00 | 0.9945E+00 | 0.9953E+00 | 0.9957E+00 |

TABLE V

(N+2) STATE MODEL WITH FAULT CORRECTION

OPERATIONAL AVAILABILITIES AT VARIOUS INPUT RATES

FAULT CORRECTION RATE=0.02

| TIME | INPUT RATE | | | | |
|---|---|---|---|---|---|
| | 0.02 | 0.04 | 0.06 | 0.08 | 0.10 |
| 1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.492E-04 | 0.492E-04 | 0.492E-04 | 0.492E-04 | 0.492E-04 |
| 4 | 0.114E-02 | 0.113E-02 | 0.113E-02 | 0.113E-02 | 0.113E-02 |
| 5 | 0.223E+00 | 0.223E+00 | 0.223E+00 | 0.223E+00 | 0.223E+00 |
| 6 | 0.482E+00 | 0.478E+00 | 0.473E+00 | 0.469E+00 | 0.464E+00 |
| 7 | 0.574E+00 | 0.560E+00 | 0.546E+00 | 0.533E+00 | 0.519E+00 |
| 8 | 0.622E+00 | 0.597E+00 | 0.573E+00 | 0.550E+00 | 0.527E+00 |
| 9 | 0.696E+00 | 0.660E+00 | 0.625E+00 | 0.592E+00 | 0.561E+00 |
| 10 | 0.751E+00 | 0.702E+00 | 0.657E+00 | 0.614E+00 | 0.574E+00 |
| 11 | 0.777E+00 | 0.716E+00 | 0.660E+00 | 0.609E+00 | 0.561E+00 |
| 12 | 0.798E+00 | 0.727E+00 | 0.664E+00 | 0.606E+00 | 0.555E+00 |
| 13 | 0.819E+00 | 0.740E+00 | 0.671E+00 | 0.610E+00 | 0.556E+00 |
| 14 | 0.833E+00 | 0.748E+00 | 0.674E+00 | 0.611E+00 | 0.556E+00 |
| 15 | 0.842E+00 | 0.751E+00 | 0.674E+00 | 0.609E+00 | 0.553E+00 |
| 16 | 0.849E+00 | 0.754E+00 | 0.675E+00 | 0.609E+00 | 0.554E+00 |
| 17 | 0.855E+00 | 0.757E+00 | 0.677E+00 | 0.610E+00 | 0.555E+00 |
| 18 | 0.859E+00 | 0.759E+00 | 0.677E+00 | 0.611E+00 | 0.555E+00 |
| 19 | 0.862E+00 | 0.760E+00 | 0.678E+00 | 0.611E+00 | 0.556E+00 |
| 20 | 0.865E+00 | 0.761E+00 | 0.678E+00 | 0.611E+00 | 0.556E+00 |
| 21 | 0.866E+00 | 0.762E+00 | 0.679E+00 | 0.612E+00 | 0.557E+00 |
| 22 | 0.868E+00 | 0.763E+00 | 0.679E+00 | 0.612E+00 | 0.557E+00 |
| 23 | 0.869E+00 | 0.763E+00 | 0.680E+00 | 0.613E+00 | 0.558E+00 |
| 24 | 0.870E+00 | 0.764E+00 | 0.680E+00 | 0.613E+00 | 0.558E+00 |
| 25 | 0.871E+00 | 0.764E+00 | 0.681E+00 | 0.614E+00 | 0.559E+00 |

TABLE VI

(2N+2) STATE MARKOV MODEL WITH FAULT CORRECTION

THE PROBABILITY OF BEING IN DIFFERENT STATES

| IME | STATE C (OP. REL.) | STATE F | TIME | STATE C (OP. REL.) | STATE F |
|---|---|---|---|---|---|
| 1 | 0.0 | 0.0 | 31 | 0.9655E+00 | 0.3873E-03 |
| 2 | 0.0 | 0.1998E-01 | 32 | 0.9682E+00 | 0.3558E-03 |
| 3 | 0.0 | 0.2727E-01 | 33 | 0.9706E+00 | 0.3269E-03 |
| 4 | 0.0 | 0.3811E-01 | 34 | 0.9729E+00 | 0.3004E-03 |
| 5 | 0.2190E+00 | 0.2567E-01 | 35 | 0.9750E+00 | 0.2761E-03 |
| 6 | 0.4727E+00 | 0.1439E-01 | 36 | 0.9770E+00 | 0.2538E-03 |
| 7 | 0.5610E+00 | 0.1145E-01 | 37 | 0.9787E+00 | 0.2333E-03 |
| 8 | 0.6088E+00 | 0.1024E-01 | 38 | 0.9804E+00 | 0.2145E-03 |
| 9 | 0.6837E+00 | 0.7206E-02 | 39 | 0.9819E+00 | 0.1972E-03 |
| 10 | 0.7416E+00 | 0.5072E-02 | 40 | 0.9832E+00 | 0.1813E-03 |
| 11 | 0.7721E+00 | 0.4288E-02 | 41 | 0.9845E+00 | 0.1667E-03 |
| 12 | 0.7981E+00 | 0.3584E-02 | 42 | 0.9857E+00 | 0.1533E-03 |
| 13 | 0.8242E+00 | 0.2810E-02 | 43 | 0.9867E+00 | 0.1410E-03 |
| 14 | 0.8442E+00 | 0.2310E-02 | 44 | 0.9877E+00 | 0.1296E-03 |
| 15 | 0.8593E+00 | 0.2013E-02 | 45 | 0.9886E+00 | 0.1192E-03 |
| 16 | 0.8729E+00 | 0.1737E-02 | 46 | 0.9895E+00 | 0.1096E-03 |
| 17 | 0.8852E+00 | 0.1495E-02 | 47 | 0.9902E+00 | 0.1008E-03 |
| 18 | 0.8956E+00 | 0.1320E-02 | 48 | 0.9909E+00 | 0.9266E-04 |
| 19 | 0.9046E+00 | 0.1182E-02 | 49 | 0.9916E+00 | 0.8520E-04 |
| 20 | 0.9128E+00 | 0.1057E-02 | 50 | 0.9922E+00 | 0.7835E-04 |
| 21 | 0.9202E+00 | 0.9508E-03 | 51 | 0.9927E+00 | 0.7204E-04 |
| 22 | 0.9268E+00 | 0.8619E-03 | 52 | 0.9932E+00 | 0.6624E-04 |
| 23 | 0.9328E+00 | 0.7840E-03 | 53 | 0.9937E+00 | 0.6091E-04 |
| 24 | 0.9383E+00 | 0.7140E-03 | 54 | 0.9941E+00 | 0.5601E-04 |
| 25 | 0.9433E+00 | 0.6519E-03 | 55 | 0.9945E+00 | 0.5150E-04 |
| 26 | 0.9478E+00 | 0.5965E-03 | 56 | 0.9949E+00 | 0.4736E-04 |
| 27 | 0.9520E+00 | 0.5464E-03 | 57 | 0.9952E+00 | 0.4355E-04 |
| 28 | 0.9558E+00 | 0.5008E-03 | 58 | 0.9955E+00 | 0.4004E-04 |
| 29 | 0.9593E+00 | 0.4594E-03 | 59 | 0.9958E+00 | 0.3682E-04 |
| 30 | 0.9625E+00 | 0.4218E-03 | 60 | 0.9960E+00 | 0.3386E-04 |

TABLE VII

(2N+2) STATE SEMIMARKOV MODEL WITH FAULT CORRECTION

WAITING TIME IN EACH STATE,DESTINATION UNKNOWN

| STATE | WAITING TIME |
|---|---|
| 1 | 0.19956E+01 |
| 2 | 0.26216E+01 |
| 3 | 0.99010E+02 |
| 4 | 0.20573E+01 |
| 5 | 0.19911E+01 |
| 6 | 0.13921E+01 |
| 7 | 0.19561E+01 |
| 8 | 0.28424E+01 |
| 9 | 0.72969E+01 |
| 10 | 0.65682E+02 |
| 11 | 0.11111E+01 |
| 12 | 0.91111E+01 |
| 13 | 0.91111E+01 |
| 14 | 0.91111E+01 |
| 15 | 0.91111E+01 |
| 16 | 0.91111E+01 |
| 17 | 0.91111E+01 |
| 18 | 0.91111E+01 |
| 19 | 0.91111E+01 |
| 20 | 0.91111E+01 |
| 21 | 0.91111E+01 |
| 22 | 0.10000E+01 |

TABLE VIII

(N+2) STATE MODEL WITH FAULT CORRECTION

OPERATIONAL RELIABILITIES AT VARIOUS FAULT CORRECTION RATES:

| | CORRECTION RATE | | | | |
|---|---|---|---|---|---|
| TIME | 0.02 | 0.04 | 0.06 | 0.08 | 0.10 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.1000D+01 | 0.7055D-01 | 0.7158D-01 | 0.7197D-01 | 0.7218D-01 | 0.7231D-01 |
| 0.2000D+01 | 0.1982D+00 | 0.1998D+00 | 0.2003D+00 | 0.2005D+00 | 0.2007D+00 |
| 0.3000D+01 | 0.3228D+00 | 0.3244D+00 | 0.3249D+00 | 0.3252D+00 | 0.3254D+00 |
| 0.4000D+01 | 0.4358D+00 | 0.4374D+00 | 0.4379D+00 | 0.4382D+00 | 0.4383D+00 |
| 0.5000D+01 | 0.5346D+00 | 0.5361D+00 | 0.5366D+00 | 0.5368D+00 | 0.5370D+00 |
| 0.6000D+01 | 0.6189D+00 | 0.6203D+00 | 0.6207D+00 | 0.6210D+00 | 0.6211D+00 |
| 0.7000D+01 | 0.6897D+00 | 0.6909D+00 | 0.6914D+00 | 0.6916D+00 | 0.6917D+00 |
| 0.8000D+01 | 0.7484D+00 | 0.7495D+00 | 0.7499D+00 | 0.7501D+00 | 0.7502D+00 |
| 0.9000D+01 | 0.7967D+00 | 0.7977D+00 | 0.7980D+00 | 0.7982D+00 | 0.7982D+00 |
| 0.1000D+02 | 0.8361D+00 | 0.8370D+00 | 0.8373D+00 | 0.8374D+00 | 0.8375D+00 |
| 0.1100D+02 | 0.8681D+00 | 0.8689D+00 | 0.8691D+00 | 0.8693D+00 | 0.8693D+00 |
| 0.1200D+02 | 0.8941D+00 | 0.8947D+00 | 0.8950D+00 | 0.8951D+00 | 0.8951D+00 |
| 0.1300D+02 | 0.9150D+00 | 0.9156D+00 | 0.9158D+00 | 0.9159D+00 | 0.9159D+00 |
| 0.1400D+02 | 0.9319D+00 | 0.9324D+00 | 0.9325D+00 | 0.9326D+00 | 0.9327D+00 |
| 0.1500D+02 | 0.9455D+00 | 0.9459D+00 | 0.9460D+00 | 0.9461D+00 | 0.9461D+00 |
| 0.1600D+02 | 0.9563D+00 | 0.9567D+00 | 0.9568D+00 | 0.9569D+00 | 0.9569D+00 |
| 0.1700D+02 | 0.9651D+00 | 0.9654D+00 | 0.9655D+00 | 0.9655D+00 | 0.9655D+00 |
| 0.1800D+02 | 0.9721D+00 | 0.9723D+00 | 0.9724D+00 | 0.9724D+00 | 0.9725D+00 |
| 0.1900D+02 | 0.9777D+00 | 0.9779D+00 | 0.9779D+00 | 0.9780D+00 | 0.9780D+00 |
| 0.2000D+02 | 0.9822D+00 | 0.9823D+00 | 0.9824D+00 | 0.9824D+00 | 0.9824D+00 |
| 0.2100D+02 | 0.9857D+00 | 0.9859D+00 | 0.9859D+00 | 0.9860D+00 | 0.9860D+00 |
| 0.2200D+02 | 0.9886D+00 | 0.9887D+00 | 0.9888D+00 | 0.9888D+00 | 0.9888D+00 |
| 0.2300D+02 | 0.9909D+00 | 0.9910D+00 | 0.9910D+00 | 0.9910D+00 | 0.9911D+00 |
| 0.2400D+02 | 0.9927D+00 | 0.9928D+00 | 0.9928D+00 | 0.9929D+00 | 0.9929D+00 |
| 0.2500D+02 | 0.9940D+00 | 0.9941D+00 | 0.9941D+00 | 0.9942D+00 | 0.9942D+00 |

## TABLE IX

(N+2) STATE MODEL WITH FAULT CORRECTION

OPERATIONAL AVAILABILITIES AT VARIOUS INPUT RATES:

FAULT CORRECTION RATE=0.10

| TIME | INPUT RATE 0.001 | 0.002 | 0.003 | 0.004 | 0.005 |
|---|---|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.1000D+01 | 0.7174D-01 | 0.7103D-01 | 0.7033D-01 | 0.6965D-01 | 0.6897D-01 |
| 0.2000D+01 | 0.1970D+00 | 0.1925D+00 | 0.1882D+00 | 0.1840D+00 | 0.1800D+00 |
| 0.3000D+01 | 0.3161D+00 | 0.3049D+00 | 0.2944D+00 | 0.2843D+00 | 0.2748D+00 |
| 0.4000D+01 | 0.4215D+00 | 0.4017D+00 | 0.3833D+00 | 0.3660D+00 | 0.3499D+00 |
| 0.5000D+01 | 0.5114D+00 | 0.4818D+00 | 0.4547D+00 | 0.4298D+00 | 0.4069D+00 |
| 0.6000D+01 | 0.5862D+00 | 0.5464D+00 | 0.5105D+00 | 0.4781D+00 | 0.4488D+00 |
| 0.7000D+01 | 0.6472D+00 | 0.5974D+00 | 0.5532D+00 | 0.5139D+00 | 0.4787D+00 |
| 0.8000D+01 | 0.6964D+00 | 0.6371D+00 | 0.5853D+00 | 0.5398D+00 | 0.4997D+00 |
| 0.9000D+01 | 0.7357D+00 | 0.6676D+00 | 0.6090D+00 | 0.5583D+00 | 0.5141D+00 |
| 0.1000D+02 | 0.7668D+00 | 0.6909D+00 | 0.6264D+00 | 0.5713D+00 | 0.5239D+00 |
| 0.1100D+02 | 0.7913D+00 | 0.7084D+00 | 0.6390D+00 | 0.5803D+00 | 0.5303D+00 |
| 0.1200D+02 | 0.8104D+00 | 0.7216D+00 | 0.6480D+00 | 0.5864D+00 | 0.5345D+00 |
| 0.1300D+02 | 0.8254D+00 | 0.7314D+00 | 0.6544D+00 | 0.5906D+00 | 0.5372D+00 |
| 0.1400D+02 | 0.8370D+00 | 0.7387D+00 | 0.6590D+00 | 0.5934D+00 | 0.5389D+00 |
| 0.1500D+02 | 0.8460D+00 | 0.7441D+00 | 0.6622D+00 | 0.5953D+00 | 0.5399D+00 |
| 0.1600D+02 | 0.8529D+00 | 0.7481D+00 | 0.6644D+00 | 0.5965D+00 | 0.5406D+00 |
| 0.1700D+02 | 0.8583D+00 | 0.7510D+00 | 0.6659D+00 | 0.5973D+00 | 0.5409D+00 |
| 0.1800D+02 | 0.8625D+00 | 0.7532D+00 | 0.6670D+00 | 0.5978D+00 | 0.5411D+00 |
| 0.1900D+02 | 0.8657D+00 | 0.7547D+00 | 0.6677D+00 | 0.5981D+00 | 0.5413D+00 |
| 0.2000D+02 | 0.8681D+00 | 0.7559D+00 | 0.6682D+00 | 0.5983D+00 | 0.5413D+00 |
| 0.2100D+02 | 0.8700D+00 | 0.7567D+00 | 0.6686D+00 | 0.5984D+00 | 0.5413D+00 |
| 0.2200D+02 | 0.8715D+00 | 0.7573D+00 | 0.6688D+00 | 0.5984D+00 | 0.5413D+00 |
| 0.2300D+02 | 0.8726D+00 | 0.7577D+00 | 0.6689D+00 | 0.5985D+00 | 0.5413D+00 |
| 0.2400D+02 | 0.8734D+00 | 0.7581D+00 | 0.6690D+00 | 0.5985D+00 | 0.5413D+00 |
| 0.2500D+02 | 0.8740D+00 | 0.7583D+00 | 0.6692D+00 | 0.5985D+00 | 0.5413D+00 |

TABLE X

(2N+2) STATE MODEL WITH FAULT CORRECTION

OPERATIONAL RELIABILITIES AT VARIOUS FAULT CORRECTION RATES:

| TIME | CORRECTION RATE | | | | |
|---|---|---|---|---|---|
| | 0.02 | 0.04 | 0.06 | 0.08 | 0.10 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.2000D+01 | 0.1033D+00 | 0.1040D+00 | 0.1046D+00 | 0.1051D+00 | 0.1056D+00 |
| 0.4000D+01 | 0.2559D+00 | 0.2594D+00 | 0.2623D+00 | 0.2647D+00 | 0.2668D+00 |
| 0.6000D+01 | 0.3939D+00 | 0.4017D+00 | 0.4076D+00 | 0.4123D+00 | 0.4159D+00 |
| 0.8000D+01 | 0.5103D+00 | 0.5231D+00 | 0.5320D+00 | 0.5384D+00 | 0.5432D+00 |
| 0.1000D+02 | 0.6053D+00 | 0.6230D+00 | 0.6343D+00 | 0.6419D+00 | 0.6471D+00 |
| 0.1200D+02 | 0.6814D+00 | 0.7033D+00 | 0.7163D+00 | 0.7244D+00 | 0.7297D+00 |
| 0.1400D+02 | 0.7417D+00 | 0.7672D+00 | 0.7810D+00 | 0.7891D+00 | 0.7942D+00 |
| 0.1600D+02 | 0.7895D+00 | 0.8175D+00 | 0.8316D+00 | 0.8393D+00 | 0.8439D+00 |
| 0.1800D+02 | 0.8274D+00 | 0.8571D+00 | 0.8708D+00 | 0.8779D+00 | 0.8820D+00 |
| 0.2000D+02 | 0.8576D+00 | 0.8880D+00 | 0.9010D+00 | 0.9074D+00 | 0.9110D+00 |
| 0.2200D+02 | 0.8817D+00 | 0.9122D+00 | 0.9243D+00 | 0.9299D+00 | 0.9330D+00 |
| 0.2400D+02 | 0.9012D+00 | 0.9312D+00 | 0.9421D+00 | 0.9470D+00 | 0.9496D+00 |
| 0.2600D+02 | 0.9171D+00 | 0.9460D+00 | 0.9558D+00 | 0.9600D+00 | 0.9621D+00 |
| 0.2800D+02 | 0.9300D+00 | 0.9576D+00 | 0.9663D+00 | 0.9698D+00 | 0.9715D+00 |
| 0.3000D+02 | 0.9407D+00 | 0.9667D+00 | 0.9742D+00 | 0.9772D+00 | 0.9786D+00 |
| 0.3200D+02 | 0.9496D+00 | 0.9739D+00 | 0.9803D+00 | 0.9828D+00 | 0.9840D+00 |
| 0.3400D+02 | 0.9570D+00 | 0.9794D+00 | 0.9850D+00 | 0.9870D+00 | 0.9880D+00 |
| 0.3600D+02 | 0.9632D+00 | 0.9838D+00 | 0.9886D+00 | 0.9902D+00 | 0.9910D+00 |
| 0.3800D+02 | 0.9685D+00 | 0.9873D+00 | 0.9913D+00 | 0.9926D+00 | 0.9932D+00 |
| 0.4000D+02 | 0.9729D+00 | 0.9900D+00 | 0.9933D+00 | 0.9944D+00 | 0.9949D+00 |
| 0.4200D+02 | 0.9767D+00 | 0.9921D+00 | 0.9949D+00 | 0.9958D+00 | 0.9962D+00 |
| 0.4400D+02 | 0.9799D+00 | 0.9938D+00 | 0.9961D+00 | 0.9968D+00 | 0.9971D+00 |
| 0.4600D+02 | 0.9827D+00 | 0.9951D+00 | 0.9970D+00 | 0.9976D+00 | 0.9979D+00 |
| 0.4800D+02 | 0.9851D+00 | 0.9962D+00 | 0.9977D+00 | 0.9982D+00 | 0.9984D+00 |
| 0.5000D+02 | 0.9882D+00 | 0.9975D+00 | 0.9978D+00 | 0.9983D+00 | 0.9985D+00 |

## TABLE XI

(2N+2) STATE MODEL WITH FAULT CORRECTION

OPERATIONAL AVAILABILITIES AT VARIOUS INPUT RATES:

FAULT CORRECTION RATE=0.02

| TIME | INPUT RATE | | | | |
|---|---|---|---|---|---|
| | 0.001 | 0.002 | 0.003 | 0.004 | 0.005 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.2000D+01 | 0.1023D+00 | 0.1013D+00 | 0.1003D+00 | 0.9930D-01 | 0.9833D-01 |
| 0.4000D+01 | 0.2502D+00 | 0.2447D+00 | 0.2393D+00 | 0.2341D+00 | 0.2291D+00 |
| 0.6000D+01 | 0.3804D+00 | 0.3676D+00 | 0.3554D+00 | 0.3438D+00 | 0.3327D+00 |
| 0.8000D+01 | 0.4873D+00 | 0.4656D+00 | 0.4454D+00 | 0.4264D+00 | 0.4086D+00 |
| 0.1000D+02 | 0.5717D+00 | 0.5407D+00 | 0.5122D+00 | 0.4858D+00 | 0.4614D+00 |
| 0.1200D+02 | 0.6371D+00 | 0.5969D+00 | 0.5604D+00 | 0.5273D+00 | 0.4971D+00 |
| 0.1400D+02 | 0.6871D+00 | 0.6384D+00 | 0.5948D+00 | 0.5557D+00 | 0.5205D+00 |
| 0.1600D+02 | 0.7253D+00 | 0.6689D+00 | 0.6190D+00 | 0.5749D+00 | 0.5357D+00 |
| 0.1800D+02 | 0.7545D+00 | 0.6913D+00 | 0.6361D+00 | 0.5878D+00 | 0.5454D+00 |
| 0.2000D+02 | 0.7769D+00 | 0.7077D+00 | 0.6481D+00 | 0.5965D+00 | 0.5516D+00 |
| 0.2200D+02 | 0.7942D+00 | 0.7200D+00 | 0.6567D+00 | 0.6024D+00 | 0.5555D+00 |
| 0.2400D+02 | 0.8076D+00 | 0.7291D+00 | 0.6628D+00 | 0.6064D+00 | 0.5580D+00 |
| 0.2600D+02 | 0.8181D+00 | 0.7360D+00 | 0.6673D+00 | 0.6092D+00 | 0.5597D+00 |
| 0.2800D+02 | 0.8265D+00 | 0.7414D+00 | 0.6706D+00 | 0.6112D+00 | 0.5607D+00 |
| 0.3000D+02 | 0.8332D+00 | 0.7455D+00 | 0.6731D+00 | 0.6126D+00 | 0.5615D+00 |
| 0.3200D+02 | 0.8386D+00 | 0.7488D+00 | 0.6750D+00 | 0.6136D+00 | 0.5620D+00 |
| 0.3400D+02 | 0.8431D+00 | 0.7514D+00 | 0.6765D+00 | 0.6144D+00 | 0.5624D+00 |
| 0.3600D+02 | 0.8467D+00 | 0.7534D+00 | 0.6776D+00 | 0.6150D+00 | 0.5627D+00 |
| 0.3800D+02 | 0.8497D+00 | 0.7551D+00 | 0.6786D+00 | 0.6155D+00 | 0.5629D+00 |
| 0.4000D+02 | 0.8522D+00 | 0.7565D+00 | 0.6793D+00 | 0.6159D+00 | 0.5631D+00 |
| 0.4200D+02 | 0.8543D+00 | 0.7577D+00 | 0.6799D+00 | 0.6162D+00 | 0.5632D+00 |
| 0.4400D+02 | 0.8561D+00 | 0.7586D+00 | 0.6804D+00 | 0.6165D+00 | 0.5633D+00 |
| 0.4600D+02 | 0.8575D+00 | 0.7594D+00 | 0.6809D+00 | 0.6167D+00 | 0.5634D+00 |
| 0.4800D+02 | 0.8588D+00 | 0.7601D+00 | 0.6812D+00 | 0.6169D+00 | 0.5635D+00 |
| 0.5000D+02 | 0.8591D+00 | 0.7610D+00 | 0.6814D+00 | 0.6170D+00 | 0.5635D+00 |

## TABLE XII

COMPARASION OF OPERATIONAL RELIABILITIES
FOR DISCRETE AND CONTINUOUS TIME MODELS:

| TIME | OPERATIONAL RELIABILITIES AT VARIOUS FAULT CORRECTION RATES | | | | | |
|---|---|---|---|---|---|---|
| | 0.06 | | 0.08 | | 0.10 | |
| | DISC. | CONT. | DISC. | CONT. | DISC. | CONT. |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1.0 | 0.0 | 0.7197D-01 | 0.0 | 0.7218D-01 | 0.0 | 0.7231D-01 |
| 2.0 | 0.5910D-04 | 0.2003D+00 | 0.7880D-04 | 0.2005D+00 | 0.9850D-04 | 0.2007D+00 |
| 3.0 | 0.1358D-02 | 0.3249D+00 | 0.1794D-02 | 0.3252D+00 | 0.2223D-02 | 0.3254D+00 |
| 4.0 | 0.2242D+00 | 0.4379D+00 | 0.2256D+00 | 0.4382D+00 | 0.2269D+00 | 0.4383D+00 |
| 5.0 | 0.4852D+00 | 0.5366D+00 | 0.4882D+00 | 0.5368D+00 | 0.4906D+00 | 0.5370D+00 |
| 6.0 | 0.5837D+00 | 0.6207D+00 | 0.5882D+00 | 0.6210D+00 | 0.5917D+00 | 0.6211D+00 |
| 7.0 | 0.6428D+00 | 0.6914D+00 | 0.6486D+00 | 0.6916D+00 | 0.6526D+00 | 0.6917D+00 |
| 8.0 | 0.7290D+00 | 0.7499D+00 | 0.7355D+00 | 0.7501D+00 | 0.7397D+00 | 0.7502D+00 |
| 9.0 | 0.7972D+00 | 0.7980D+00 | 0.8040D+00 | 0.7982D+00 | 0.8082D+00 | 0.7982D+00 |
| 10.0 | 0.8362D+00 | 0.8373D+00 | 0.8429D+00 | 0.8374D+00 | 0.8469D+00 | 0.8375D+00 |
| 11.0 | 0.8685D+00 | 0.8691D+00 | 0.8747D+00 | 0.8693D+00 | 0.8783D+00 | 0.8693D+00 |
| 12.0 | 0.8989D+00 | 0.8950D+00 | 0.9045D+00 | 0.8951D+00 | 0.9077D+00 | 0.8951D+00 |
| 13.0 | 0.9214D+00 | 0.9158D+00 | 0.9265D+00 | 0.9159D+00 | 0.9293D+00 | 0.9159D+00 |
| 14.0 | 0.9374D+00 | 0.9325D+00 | 0.9419D+00 | 0.9326D+00 | 0.9444D+00 | 0.9327D+00 |
| 15.0 | 0.9507D+00 | 0.9460D+00 | 0.9546D+00 | 0.9461D+00 | 0.9567D+00 | 0.9461D+00 |
| 16.0 | 0.9617D+00 | 0.9568D+00 | 0.9650D+00 | 0.9569D+00 | 0.9668D+00 | 0.9569D+00 |
| 17.0 | 0.9699D+00 | 0.9655D+00 | 0.9728D+00 | 0.9655D+00 | 0.9743D+00 | 0.9655D+00 |
| 18.0 | 0.9763D+00 | 0.9724D+00 | 0.9787D+00 | 0.9724D+00 | 0.9799D+00 | 0.9725D+00 |
| 19.0 | 0.9814D+00 | 0.9779D+00 | 0.9834D+00 | 0.9780D+00 | 0.9845D+00 | 0.9780D+00 |
| 20.0 | 0.9854D+00 | 0.9824D+00 | 0.9871D+00 | 0.9824D+00 | 0.9880D+00 | 0.9824D+00 |
| 21.0 | 0.9886D+00 | 0.9859D+00 | 0.9900D+00 | 0.9860D+00 | 0.9907D+00 | 0.9860D+00 |
| 22.0 | 0.9910D+00 | 0.9888D+00 | 0.9922D+00 | 0.9888D+00 | 0.9928D+00 | 0.9888D+00 |
| 23.0 | 0.9929D+00 | 0.9910D+00 | 0.9939D+00 | 0.9910D+00 | 0.9944D+00 | 0.9911D+00 |
| 24.0 | 0.9945D+00 | 0.9928D+00 | 0.9953D+00 | 0.9929D+00 | 0.9957D+00 | 0.9929D+00 |
| 25.0 | 0.9951D+00 | 0.9941D+00 | 0.9957D+00 | 0.9942D+00 | 0.9959D+00 | 0.9942D+00 |

TABLE XIII

COMPARASION OF OPERATIONAL AVAILABILITIES
FOR DISCRETE AND CONTINUOUS TIME MODELS:

| TIME | OPERATIONAL AVAILABILITIES AT VARIOUS INPUT RATES | | | | | |
|---|---|---|---|---|---|---|
| | 0.06 | | 0.08 | | 0.10 | |
| | DISC. | CONT. | DISC. | CONT. | DISC. | CONT. |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1.0 | 0.0 | 0.7033D-01 | 0.0 | 0.6965D-01 | 0.0 | 0.6897D-01 |
| 2.0 | 0.4920D-04 | 0.1882D+00 | 0.4920D-04 | 0.1840D+00 | 0.4920D-04 | 0.1800D+00 |
| 3.0 | 0.1130D-02 | 0.2944D+00 | 0.1130D-02 | 0.2843D+00 | 0.1130D-02 | 0.2748D+00 |
| 4.0 | 0.2230D+00 | 0.3833D+00 | 0.2230D+00 | 0.3660D+00 | 0.2230D+00 | 0.3499D+00 |
| 5.0 | 0.4730D+00 | 0.4547D+00 | 0.4690D+00 | 0.4298D+00 | 0.4640D+00 | 0.4069D+00 |
| 6.0 | 0.5460D+00 | 0.5105D+00 | 0.5330D+00 | 0.4781D+00 | 0.5190D+00 | 0.4488D+00 |
| 7.0 | 0.5730D+00 | 0.5532D+00 | 0.5500D+00 | 0.5139D+00 | 0.5270D+00 | 0.4787D+00 |
| 8.0 | 0.6250D+00 | 0.5853D+00 | 0.5920D+00 | 0.5398D+00 | 0.5610D+00 | 0.4997D+00 |
| 9.0 | 0.6570D+00 | 0.6090D+00 | 0.6140D+00 | 0.5583D+00 | 0.5740D+00 | 0.5141D+00 |
| 10.0 | 0.6600D+00 | 0.6264D+00 | 0.6090D+00 | 0.5713D+00 | 0.5610D+00 | 0.5239D+00 |
| 11.0 | 0.6640D+00 | 0.6390D+00 | 0.6060D+00 | 0.5803D+00 | 0.5550D+00 | 0.5303D+00 |
| 12.0 | 0.6710D+00 | 0.6480D+00 | 0.6100D+00 | 0.5864D+00 | 0.5560D+00 | 0.5345D+00 |
| 13.0 | 0.6740D+00 | 0.6544D+00 | 0.6110D+00 | 0.5906D+00 | 0.5560D+00 | 0.5372D+00 |
| 14.0 | 0.6740D+00 | 0.6590D+00 | 0.6090D+00 | 0.5934D+00 | 0.5530D+00 | 0.5389D+00 |
| 15.0 | 0.6750D+00 | 0.6622D+00 | 0.6090D+00 | 0.5953D+00 | 0.5540D+00 | 0.5399D+00 |
| 16.0 | 0.6770D+00 | 0.6644D+00 | 0.6100D+00 | 0.5965D+00 | 0.5550D+00 | 0.5406D+00 |
| 17.0 | 0.6770D+00 | 0.6659D+00 | 0.6110D+00 | 0.5973D+00 | 0.5550D+00 | 0.5409D+00 |
| 18.0 | 0.6780D+00 | 0.6670D+00 | 0.6110D+00 | 0.5978D+00 | 0.5560D+00 | 0.5411D+00 |
| 19.0 | 0.6780D+00 | 0.6677D+00 | 0.6110D+00 | 0.5981D+00 | 0.5560D+00 | 0.5413D+00 |
| 20.0 | 0.6790D+00 | 0.6682D+00 | 0.6120D+00 | 0.5983D+00 | 0.5570D+00 | 0.5413D+00 |
| 21.0 | 0.6790D+00 | 0.6686D+00 | 0.6120D+00 | 0.5984D+00 | 0.5570D+00 | 0.5413D+00 |
| 22.0 | 0.6800D+00 | 0.6688D+00 | 0.6130D+00 | 0.5984D+00 | 0.5580D+00 | 0.5413D+00 |
| 23.0 | 0.6800D+00 | 0.6689D+00 | 0.6130D+00 | 0.5985D+00 | 0.5580D+00 | 0.5413D+00 |
| 24.0 | 0.6810D+00 | 0.6690D+00 | 0.6140D+00 | 0.5985D+00 | 0.5590D+00 | 0.5413D+00 |
| 25.0 | 0.6810D+00 | 0.6692D+00 | 0.6140D+00 | 0.5985D+00 | 0.5590D+00 | 0.5413D+00 |

TABLE XIV

THE EXPECTED COST OF FAILURE IN TERMS OF CONSTANT C:

| TIME | COST*C | TIME | COST*C | TIME | COST*C |
|---|---|---|---|---|---|
| 1.00 | 0.2000D+01 | 31.00 | 0.4463D+01 | 61.00 | 0.1282D+01 |
| 2.00 | 0.5217D+01 | 32.00 | 0.4206D+01 | 62.00 | 0.1233D+01 |
| 3.00 | 0.7589D+01 | 33.00 | 0.3976D+01 | 63.00 | 0.1186D+01 |
| 4.00 | 0.9528D+01 | 34.00 | 0.3768D+01 | 64.00 | 0.1141D+01 |
| 5.00 | 0.1110D+02 | 35.00 | 0.3580D+01 | 65.00 | 0.1097D+01 |
| 6.00 | 0.1232D+02 | 36.00 | 0.3410D+01 | 66.00 | 0.1055D+01 |
| 7.00 | 0.1319D+02 | 37.00 | 0.3255D+01 | 67.00 | 0.1014D+01 |
| 8.00 | 0.1370D+02 | 38.00 | 0.3113D+01 | 68.00 | 0.9747D+00 |
| 9.00 | 0.1390D+02 | 39.00 | 0.2981D+01 | 69.00 | 0.9369D+00 |
| 10.00 | 0.1388D+02 | 40.00 | 0.2860D+01 | 70.00 | 0.9004D+00 |
| 11.00 | 0.1373D+02 | 41.00 | 0.2746D+01 | 71.00 | 0.8653D+00 |
| 12.00 | 0.1350D+02 | 42.00 | 0.2639D+01 | 72.00 | 0.8315D+00 |
| 13.00 | 0.1321D+02 | 43.00 | 0.2538D+01 | 73.00 | 0.7989D+00 |
| 14.00 | 0.1284D+02 | 44.00 | 0.2442D+01 | 74.00 | 0.7675D+00 |
| 15.00 | 0.1240D+02 | 45.00 | 0.2351D+01 | 75.00 | 0.7372D+00 |
| 16.00 | 0.1190D+02 | 46.00 | 0.2264D+01 | 76.00 | 0.7081D+00 |
| 17.00 | 0.1134D+02 | 47.00 | 0.2180D+01 | 77.00 | 0.6801D+00 |
| 18.00 | 0.1075D+02 | 48.00 | 0.2100D+01 | 78.00 | 0.6531D+00 |
| 19.00 | 0.1013D+02 | 49.00 | 0.2023D+01 | 79.00 | 0.6272D+00 |
| 20.00 | 0.9514D+01 | 50.00 | 0.1948D+01 | 80.00 | 0.6022D+00 |
| 21.00 | 0.8901D+01 | 51.00 | 0.1876D+01 | 81.00 | 0.5782D+00 |
| 22.00 | 0.8306D+01 | 52.00 | 0.1807D+01 | 82.00 | 0.5550D+00 |
| 23.00 | 0.7737D+01 | 53.00 | 0.1740D+01 | 83.00 | 0.5328D+00 |
| 24.00 | 0.7199D+01 | 54.00 | 0.1676D+01 | 84.00 | 0.5114D+00 |
| 25.00 | 0.6697D+01 | 55.00 | 0.1613D+01 | 85.00 | 0.4909D+00 |
| 26.00 | 0.6232D+01 | 56.00 | 0.1553D+01 | 86.00 | 0.4711D+00 |
| 27.00 | 0.5806D+01 | 57.00 | 0.1495D+01 | 87.00 | 0.4521D+00 |
| 28.00 | 0.5417D+01 | 58.00 | 0.1439D+01 | 88.00 | 0.4338D+00 |
| 29.00 | 0.5066D+01 | 59.00 | 0.1385D+01 | 89.00 | 0.4162D+00 |
| 30.00 | 0.4748D+01 | 60.00 | 0.1332D+01 | 90.00 | 0.3993D+00 |

```
C                                  APPENDIX C
C                                  ----------
C
C
C      PROGRAM FOR 22 STATES EXAMPLE FOR (2N+2) STATE MARKOV MODEL
C
C
       DIMENSION B(22,22),C(22,22),D(1,22),E(1,22),F(22,22),P(60,3)
C
C
C      B=STATE TRANSITION PROBABILITY MATRIX
C      E=STATE PROBABILITY VECTOR
C      P=MATRIX FOR DIFFERENT STATE PROBABILITY VECTOR AT DIFFERENT
C      FAULT CORRECTION RATES
C      D & F ARE WORK MATRICES
C
C
       K=60
       KK=3
       K1=KK-1
       KS=23
       KI=KS-1
       DO 97 I=1,K
       DO 97 J=1,KK
       P(I,J)=0.0
   97  CONTINUE
       DO 20 I=1,KI
       DO 20 J=1,KI
   20  B(I,J)=0.0
       DO 181 IC=1,KI
  181  D(1,IC)=0.0
       D(1,1)=1.0
       DO 110 I=1,K
  110  P(I,1)=I
C
C    *****    READ & WRITE B    *****
C
       DO 115 I=1,11
       READ(11,170)(B(I,J),J=1,11)
       WRITE(6,170)(B(I,J),J=1,11)
  115  CONTINUE
       WRITE(6,109)
       DO 116 I=1,11
       READ(11,170)(B(I,J),J=12,22)
       WRITE(6,170)(B(I,J),J=12,22)
  116  CONTINUE
       WRITE(6,109)
       DO 117 I=12,22
       READ(11,170)(B(I,J),J=1,11)
       WRITE(6,170)(B(I,J),J=1,11)
  117  CONTINUE
       WRITE(6,109)
       DO 118 I=12,22
       READ(11,170)(B(I,J),J=12,22)
       WRITE(6,170)(B(I,J),J=12,22)
  118  CONTINUE
```

(CONTINUED)

```
C
C   *****   CHANGE CARDS *****
C
      DO 80 M=1,KI
      DO 80 N=1,KI
  80  C(M,N)=B(M,N)
      M=1
      DO 70 I=1,K
CC    WRITE(6,101)M
      DO 78 II=1,KI
      DO 78 JJ=1,KI
  78  F(II,JJ)=0.0
      CALL GMPRD(D,B,E,1,KI,KI)
CC    WRITE(6,55)M,(II,II=1,11)
CC    WRITE(6,57)(E(1,J),J=1,11)
      CALL GMPRD(B,C,F,KI,KI,KI)
      DO 77 II=1,KI
      DO 77 JJ=1,KI
  77  B(II,JJ)=F(II,JJ)
      M=M+1
      P(I,2)=E(1,22)
      P(I,3)=E(1,11)
  70  CONTINUE
      KP=K/2
      KC=KP+1
      DO 57 I=1,KP
      WRITE(10,121)I,(P(I,J),J=2,3),KC,(P(KC,J),J=2,3)
      KC=KC+1
   57 CONTINUE
      CALL PLOT(1,P,K,KK,0,1)
  170 FORMAT(11F7.4)
  121 FORMAT(1X,I2,4X,2E12.4,10X,I2,4X,2E12.4)
 189  FORMAT(2X,////' NEXT 9*11 ELEMENTS:',//)
 109  FORMAT(2X,////' NEXT 11*9 ELEMENTS:',//)
 108  FORMAT(2X,////' NEXT 9*9 ELEMENTS:',//)
      STOP
      END
```

```
C
C
C     PROGRAM FOR 22 STATES EXAMPLE FOR SEMI MARKOV MODEL
C
C
      DIMENSION B(22,22),C(22,22,90),H(22,22),HH(22,22,90),WT(22),
     .W(22,22,91),WW(22,91),SM(22,22),CC(22,22),PHI1(22,22),
     .SM2(22,22,90),PHI(22,22,91),SH(91,4)
C
C
C
C     B=STATE TRANSITION PROBABILITY MATRIX
C     C=CORE MATRIX
C     H=HOLDING TIME MASS FUNCTION MATRIX
C     HH=H AT DIFFERENT VALUES OF TIME
C     WT=WAITING TIME VECTOR
C     WW=COMPLEMENTARY CUMULATIVE PROBABILITY DISTRIBUTION FOR WAITING
C     TIME
C     SM2=THE SIGMA TERM IN MAIN EQUATION
C     PHI=INTERVAL TRANSITION PROBABILITY MATRIX
C     SH=INTERVAL TRANSITION PROBABILITIES FOR DIFFERENT STATES
C     W,SM & PHI1 ARE THE WORK MATRICES
C
C
C
      KT=90
      KI=22
      KT1=KT+1
      DO 20 I=1,KI
      DO 20 J=1,KI
  20  B(I,J)=0.0
C
C     ***** READ & WRITE B *****
C
      DO 115 I=1,11
      READ(11,170)(B(I,J),J=1,11)
      WRITE(6,170)(B(I,J),J=1,11)
  115 CONTINUE
      WRITE(6,109)
      DO 116 I=1,11
      READ(11,170)(B(I,J),J=12,22)
      WRITE(6,170)(B(I,J),J=12,22)
  116 CONTINUE
      WRITE(6,109)
      DO 117 I=12,22
      READ(11,170)(B(I,J),J=1,11)
      WRITE(6,170)(B(I,J),J=1,11)
  117 CONTINUE
      WRITE(6,109)
      DO 118 I=12,22
      READ(11,170)(B(I,J),J=12,22)
      WRITE(6,170)(B(I,J),J=12,22)
  118 CONTINUE
```

(CONTINUED)

```
C
C     *****   READ & WRITE H *****
C
      WRITE(6,220)
      WRITE(6,230)
      DO 35 I=1,11
      READ(11,170)(H(I,J),J=1,11)
      WRITE(6,170)(H(I,J),J=1,11)
   35 CONTINUE
      WRITE(6,109)
      DO 36 I=1,11
      READ(11,170)(H(I,J),J=12,22)
      WRITE(6,170)(H(I,J),J=12,22)
   36 CONTINUE
      WRITE(6,189)
      DO 37 I=12,22
      READ(11,170)(H(I,J),J=1,11)
      WRITE(6,170)(H(I,J),J=1,11)
   37 CONTINUE
      WRITE(6,108)
      DO 38 I=12,22
      READ(11,170)(H(I,J),J=12,22)
      WRITE(6,170)(H(I,J),J=12,22)
   38 CONTINUE
C
C     *********   COMPUTE HH(I,J,N)  **********
C
      DO 39 M=1,KT
      DO 39 M1=1,KI
      DO 39 M2=1,KI
      IF(H(M1,M2).EQ.0.0)GO TO 300
      GO TO 301
  300 IF(M.EQ.1)GO TO 302
      GO TO 301
  302 HH(M1,M2,M)=1.0
      GO TO 39
  301 HH(M1,M2,M)=(1.0-H(M1,M2))*(H(M1,M2)**(M-1))
   39 CONTINUE
C
C     *********  OBTAIN THE CORE MATRIX C(M)=C(I,J,M)  ***********
C
      DO 40 M=1,KT
      DO 40 M1=1,KI
      DO 40 M2=1,KI
   40 C(M1,M2,M)=HH(M1,M2,M)*B(M1,M2)
C
C     *********  OBTAIN THE WAITNG TIME IN EACH STATE,DESTINATION UNKNOWN  ***
C
      WRITE(12,500)
      DO 350 IS=1,KI
      WT(IS)=0.
      DO 350 J=1,KI
      WT(IS)=B(IS,J)*(1/(1-H(IS,J)))+WT(IS)
  350 CONTINUE
      DO 502 I=1,KI
  502 WRITE(12,501)I,WT(I)
```

```
C
C      *********  OBTAIN )WI(N)=WW(I,N)  ************
C
       DO 41 N1=1,KT1
       N=N1-1
       DO 41 N2=1,KI
       DO 41 N3=1,KI
       IF(H(N2,N3).EQ.0.0)GO TO 310
       GO TO 311
   310 IF(N1.EQ.1)GO TO 312
       GO TO 311
   312 W(N2,N3,N1)=0.0
       GO TO 41
   311 W(N2,N3,N1)=B(N2,N3)*(H(N2,N3)**N)
    41 CONTINUE
       DO 43 N=1,KT1
       DO 43 N1=1,KI
    43 WW(N1,N)=0.
       DO 42 N=1,KT1
       DO 42 N1=1,KI
       DO 42 N2=1,KI
    42 WW(N1,N)=W(N1,N2,N)+WW(N1,N)
C
C      *********  INITIALIZE PHI(N)=PHI(I,J,N)  *********
C
       DO 53 I=1,KI
       DO 53 J=1,KI
       PHI(I,J,1)=0.
       IF(I.NE.J)GO TO 53
       PHI(I,J,1)=1
    53 CONTINUE
C
C      **********  OBTAIN THE SIGMA 'SM'  TERMS    ************
C
       DO 72 N=1,KT
       DO 72 I=1,KI
       DO 72 J=1,KI
    72 SM2(I,J,N)=0.
       DO 50 N=1,KT
       K1=N
       DO 52 K=1,N
       DO 71 I=1,KI
       DO 71 J=1,KI
       CC(I,J)=0.
       PHI1(I,J)=0.
       CC(I,J)=C(I,J,K)
    71 PHI1(I,J)=PHI(I,J,K1)
       CALL GMPRD(CC,PHI1,SM,KI,KI,KI)
       DO 73 I=1,KI
       DO 73 J=1,KI
    73 SM2(I,J,N)=SM2(I,J,N)+SM(I,J)
       K1=K1-1
    52 CONTINUE
```

(CONTINUED)

```
C
C     **********   OBTAIN THE TERM PHI(I,J,N)   *********
C
      NN=N+1
      DO 70 I=1,KI
      DO 70 J=1,KI
      IF(I.NE.J)GO TO 51
      PHI(I,J,NN)=WW(I,NN)+SM2(I,J,N)
      GO TO 70
   51 PHI(I,J,NN)=SM2(I,J,N)
   70 CONTINUE
   50 CONTINUE
      SF=0.0
      DO 510 I=1,KT1
      SH(I,1)=SF
      SF=SF+1
      SH(I,2)=PHI(10,10,I)
      SH(I,4)=PHI(1,10,I)
  510 SH(I,3)=PHI(11,11,I)
      CALL PLOT(1,SH,KT1,4,KT1,1)
  170 FORMAT(11F7.4)
  210 FORMAT(////2X,' THE 22*22 MATRIX OF B:',/2X,23('-'),//)
  220 FORMAT(////2X,' THE 22*22 MATRIX OF H:',/2X,23('-'),//)
  230 FORMAT(////2X,' FIRST 11*11 ELEMENTS:',//)
  189 FORMAT(////2X,' NEXT 11*11 ELEMENTS:',//)
  109 FORMAT(////2X,' NEXT 11*11 ELEMENTS:',//)
  108 FORMAT(////2X,' NEXT 11*11 ELEMENTS:',//)
  500 FORMAT(///2X,' WAITING TIME IN EACH STATE,DESTINATION UNKNOWN:',
     ./3X,47('-'),///7X,'STATE',7X,'WAITING TIME',/7X,5('-'),7X,12('-'))
  501 FORMAT(8X,I2,8X,E12.5)
      STOP
      END
```

```
C
C
C      PROGRAM FOR CONTINOUS TIME 12 STATES MODEL
C
C
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION B(12,12),P(250,7),X(12,250),XN(12),AS(12),FV(12)
      .,RK(12,12),Q(250,7),QQ(250,6),A(12,12)
C
C
C      B=STATE TRANSITION PROBABILITY MATRIX
C      P=MATRIX TO STORE STATE PROBABILITIES OF STATES 1-6
C      Q=MATRIX TO STORE STATE PROBABILITIES OF STATES 7-12
C      QQ=MATRIX TO STORE OPERATIONAL RELIABILITIES AT DIFFERENT FAULT
C      CORRECTION RATES
C      A=PROGRAM MODEL BRANCHING PROBABILITY MATRIX
C      X,XN,AS,FV & RK=WORK MATRICES FOR RUNGE KUTTA TECHNIQUE
C
C
       NL=250
       NU=12
       H=1.0
       NQ=7
       NI=1
       XD=0.02
       XI=0.02
       XII=10.0
       IX=10
       ITR=5
       ITR1=ITR+1
C
C      READ THE BRANCHING PROBABILITY MATRIX
C
       DO 19 I=1,NU
  19   READ(5,50)(A(I,J),J=1,NU)
C
C      INITIALIZE THE MATRICES
C
       DO 900 I=1,NL
       DO 900 J=1,ITR1
       QQ(I,J)=0.0
 900   QQ(I,1)=(I-1.0)/XII
       DO 777 IT=1,ITR
       DO 97 I=1,NL
       DO 97 J=1,7
       P(I,J)=0.0
       Q(I,J)=0.0
  97   CONTINUE
       DO 110 I=1,NL
       P(I,1)=I
 110   Q(I,1)=I
       DO 20 I=1,NU
       XN(I)=0.0
       AS(I)=0.0
       FV(I)=0.0
       DO 20 J=1,NU
       RK(I,J)=0.0
  20   B(I,J)=0.0
```
(CONTINUED)

```
      DO 27 I=1,NU
      DO 27 J=1,NL
   27 X(I,J)=0.0
C
C     OBTAIN THE BRANCHING PROBABILITY MATRIX
C
      R1=0.999
      R2=0.98
      R3=0.99
      R4=0.97
      R5=0.95
      R6=0.995
      R7=0.985
      R8=0.95
      R9=0.975
      R10=0.985
      B(1,2)=A(1,2)*R1
      B(1,3)=A(1,3)*R1
      B(1,4)=A(1,4)*R1
      B(1,11)=.001
      B(2,3)=A(2,3)*R2
      B(2,5)=A(2,5)*R2
      B(2,11)=.02
      B(3,5)=A(3,5)*R3
      B(4,5)=A(4,5)*R4
      B(4,6)=A(4,6)*R4
      B(4,11)=.03
      B(3,11)=.01
      B(5,7)=A(5,7)*R5
      B(5,8)=A(5,8)*R5
      B(6,3)=A(6,3)*R6
      B(6,7)=A(6,7)*R6
      B(6,8)=A(6,8)*R6
      B(6,9)=A(6,9)*R6
      B(7,2)=A(7,2)*R7
      B(7,9)=A(7,9)*R7
      B(8,4)=A(8,4)*R8
      B(8,10)=A(8,10)*R8
      B(9,8)=A(9,8)*R9
      B(9,10)=A(9,10)*R9
      B(10,12)=A(10,12)*R10
      B(10,10)=0.0
      B(5,11)=.05
      B(6,11)=.005
      B(7,11)=.015
      B(8,11)=.05
      B(9,11)=.025
      B(10,11)=.015
      B(11,11)=0.
      B(12,12)=1.
      DO 771 I=1,10
      B(11,I)=XD
771   B(11,11)=B(11,11)+B(11,I)
      B(11,11)=1-B(11,11)
      (CONTINUED)
```

```
C
C     START RUNGE KUTTA TECHNIQUE TO SOLVE DIFFERENTIAL EQUATIONS GIVEN
C     BY SUBROUTINE FNVL
C
      DO 111 I=1,NU
  111 X(I,1)=0.0
      X(1,1)=1.0
      DO 1 I=1,NU
    1 XN(I)=X(I,NI)
      K=NI
    2 L=1
      DO 3 I=1,NU
    3 AS(I)=XN(I)
      GO TO 4
    5 DO 6 I=1,NU
      RK(I,L)=H*FV(I)
    6 AS(I)=XN(I)+RK(I,L)/2
      K=K+1
      L=2
      GO TO 4
    7 DO 8 I=1,NU
      RK(I,L)=H*FV(I)
    8 AS(I)=XN(I)+RK(I,L)/2
      L=3
      GO TO 4
    9 DO 10 I=1,NU
      RK(I,L)=H*FV(I)
   10 AS(I)=XN(I)+RK(I,L)
      L=4
      GO TO 4
   11 DO 12 I=1,NU
      RK(I,L)=H*FV(I)
   12 XN(I)=XN(I)+(RK(I,1)+2.*RK(I,2)+2.*RK(I,3)+RK(I,4))/6.
      DO 13 I=1,NU
   13 X(I,K)=XN(I)
      IF(K.EQ.NL)GO TO 14
      GO TO 2
    4 CALL FNVL(AS,FV,B,XD)
      GO TO (5,7,9,11),L
   14 CONTINUE
      DO 200 I=1,NL
      DO 200 J=2,NQ
      JX1=J-1
      JX2=J+5
      P(I,J)=X(JX1,I)
      Q(I,J)=X(JX2,I)
  200 CONTINUE
      IQ=IT+1
      DO 401 I=1,NL
  401 QQ(I,IQ)=Q(I,7)
  777 XD=XD+XI
      (CONTINUED)
      IK=0
      DO 252 I=1,NL
      II=I-1
      IF((II/IX).NE.IK)GO TO 252
      WRITE(6,257)(QQ(I,J),J=1,ITR1)
      IK=IK+1
  252 CONTINUE
      CALL PLOT(3,QQ,NL,ITR1,NL,1)
  250 FORMAT(7F10.5)
  257 FORMAT(6E12.4)
   50 FORMAT(12F5.3)
      STOP
      END
      SUBROUTINE FNVL(S,FV,B,A)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION FV(12),S(12),B(12,12)
      BL1=0.5011
      BL2=0.3814
      BL3=0.0101
      BL4=0.4986
      BL5=0.5022
      BL6=0.7183
      BL7=0.5112
      BL8=0.3518
      BL9=0.137
      BL10=0.0152
      BL11=0.901
      BL12=1.0
      FV(1)=-BL1*S(1)+BL11*A*S(11)
      FV(2)=-BL2*S(2)+BL1*B(1,2)*S(1)+BL7*B(7,2)*S(7)+
     .A*S(11)*BL11
      FV(3)=-BL3*S(3)+BL1*B(1,3)*S(1)+BL2*B(2,3)*S(2)+BL6*B(6,3)*S(6)+
     .A*S(11)*BL11
      FV(4)=-BL4*S(4)+BL1*B(1,4)*S(1)+BL8*B(8,4)*S(8)+
     .A*S(11)*BL11
      FV(5)=-BL5*S(5)+BL2*B(2,5)*S(2)+BL3*B(3,5)*S(3)+
     .BL4*B(4,5)*S(4)+A*S(11)*BL11
      FV(6)=-BL6*S(6)+BL4*B(4,6)*S(4)+A*S(11)*BL11
      FV(7)=-BL7*S(7)+BL5*B(5,7)*S(5)+BL6*B(6,7)*S(6)+
     .A*S(11)*BL11
      FV(8)=-BL8*S(8)+BL5*B(5,8)*S(5)+BL6*B(6,8)*S(6)+
     .BL9*B(9,8)*S(9)+A*S(11)*BL11
      FV(9)=-BL9*S(9)+BL6*B(6,9)*S(6)+BL7*B(7,9)*S(7)+
     .A*S(11)*BL11
      FV(10)=-BL10*S(10)+BL8*B(8,10)*S(8)+BL9*B(9,10)*S(9)+
     .A*S(11)*BL11
      FV(11)=-BL11*10*A*S(11)+B(1,11)*BL1*S(1)+B(2,11)*BL2*S(2)
     .+B(3,11)*BL3*S(3)+B(4,11)*BL4*S(4)+B(5,11)*BL5*S(5)+B(6,11)*BL6
     .*S(6)+B(7,11)*BL7*S(7)+B(8,11)*BL8*S(8)+B(9,11)*BL9*S(9)
     .+B(10,11)*BL10*S(10)
      FV(12)=BL10*B(10,12)*S(10)
      RETURN
      END
```

```
C
C
C     PROGRAM FOR CONTINOUS TIME 22 STATE MODEL
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION B(22,22),P(1000),X(22,1000),XN(22),AS(22),FV(22)
     .,RK(22,22),QQ(1000,6)
C
C
C     B=STATE TRANSITION PROBABILITY MATRIX
C     P=MATRIX TO STORE STATE PROBABILITIES OF STATE 22
C     QQ=MATRIX TO STORE OPERATIONAL RELIABILITIES AT DIFFERENT FAULT
C     CORRECTION RATES
C     X,XN,AS,FV & RK=WORK MATRICES FOR RUNGE KUTTA TECHNIQUE
C
C
      KS=23
      KI=KS-1
      NL=1000
      NU=22
      KU=NU-1
      ITR=5
      XD=0.04
      H=1.0
      NQ=6
      NI=1
C
C     INITIALIZE THE MATRICES
C
      DO 20 I=1,KI
      DO 20 J=1,KI
   20 B(I,J)=0.0
      DO 25 J=1,NL
      QQ(J,1)=(J-1.0)/20.0
   25 P(J)=0.0
C
C   *****   READ & WRITE B   *****
C
      DO 115 I=1,11
      READ(11,170)(B(I,J),J=1,11)
      WRITE(6,170)(B(I,J),J=1,11)
  115 CONTINUE
      WRITE(6,109)
      DO 116 I=1,11
      READ(11,170)(B(I,J),J=12,22)
      WRITE(6,170)(B(I,J),J=12,22)
  116 CONTINUE
      WRITE(6,109)
      DO 117 I=12,22
      READ(11,170)(B(I,J),J=1,11)
      WRITE(6,170)(B(I,J),J=1,11)
  117 CONTINUE
```
(CONTINUED)

```
      WRITE(6,109)
      DO 118 I=12,22
      READ(11,170)(B(I,J),J=12,22)
      WRITE(6,170)(B(I,J),J=12,22)
  118 CONTINUE
      XT=XD
      DO 777 IT=1,ITR
C
C     START RUNGE KUTTA TECHNIQUE TO SOLVE DIFFERENTIAL EQUATIONS GIVEN -
C     BY SUBROUTINE FNVAL
C
      DO 111 I=1,NU
  111 X(I,1)=0.0
      X(1,1)=1.0
      DO 1 I=1,NU
    1 XN(I)=X(I,NI)
      K=NI
    2 L=1
      DO 3 I=1,NU
    3 AS(I)=XN(I)
      GO TO 4
    5 DO 6 I=1,NU
      RK(I,L)=H*FV(I)
    6 AS(I)=XN(I)+RK(I,L)/2
      K=K+1
      L=2
      GO TO 4
    7 DO 8 I=1,NU
      RK(I,L)=H*FV(I)
    8 AS(I)=XN(I)+RK(I,L)/2
      L=3
      GO TO 4
    9 DO 10 I=1,NU
      RK(I,L)=H*FV(I)
   10 AS(I)=XN(I)+RK(I,L)
      L=4
      GO TO 4
   11 DO 12 I=1,NU
      RK(I,L)=H*FV(I)
   12 XN(I)=XN(I)+(RK(I,1)+2.*RK(I,2)+2.*RK(I,3)+RK(I,4))/6.
      DO 13 I=1,NU
   13 X(I,K)=XN(I)
      IF(K.EQ.NL)GO TO 14
      GO TO 2
    4 CALL FNVAL(AS,FV,B,XT)
      GO TO (5,7,9,11),L
   14 CONTINUE
      DO 200 I=1,NL
      P(I)=X(22,I)
  200 CONTINUE
      IJ=1
      DO 779 I=12,KU
      B(11,I)=B(11,I)+XD
      B(I,IJ)=B(I,IJ)+XD
      B(I,I)=1.0-B(I,IJ)
  779 IJ=IJ+1
      (CONTINUED)
```

```
;
;
;     CHANGE VALUES OF B FOR DIFFERENT ITERATIONS
;
      B(11,11)=0.0
      DO 780 I=12,KU
  780 B(11,11)=B(11,11)+B(11,I)
      B(11,11)=1.0-B(11,11)
      IA=IT+1
      DO 782 I=1,NL
  782 QQ(I,IA)=P(I)
      XT=XT+XD
  777 CONTINUE
      IK=0
      DO 252 I=1,NL
      II=I-1
      IF((II/40).NE.IK)GO TO 252
      WRITE(6,253)(QQ(I,J),J=1,NQ)
      IK=IK+1
  252 CONTINUE
  253 FORMAT(6E12.4)
      CALL PLOT(1,QQ,NL,NQ,NL,1)
  170 FORMAT(11F7.4)
  109 FORMAT(2X,////' NEXT 11*11 ELEMENTS:',//)
      STOP
      END
      SUBROUTINE FNVAL(S,FV,B,A)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION FV(22),S(22),B(22,22)
      BL1=0.5011
      BL2=0.3814
      BL3=0.0101
      BL4=0.4986
      BL5=0.5022
      BL6=0.7183
      BL7=0.5112
      BL8=0.3518
      BL9=0.137
      BL10=0.0152
      BL11=0.901
      BL22=1.0
      BBL=1/9.11111
      FV(1)=-BL1*S(1)+BBL*A*S(12)
      FV(2)=-BL2*S(2)+BL1*B(1,2)*S(1)+BL7*B(7,2)*S(7)+
     .A*S(13)*BBL
      FV(3)=-BL3*S(3)+BL1*B(1,3)*S(1)+BL2*B(2,3)*S(2)+BL6*B(6,3)*S(6)+
     .A*S(14)*BBL
      FV(4)=-BL4*S(4)+BL1*B(1,4)*S(1)+BL8*B(8,4)*S(8)+
     .A*S(15)*BBL
      FV(5)=-BL5*S(5)+BL2*B(2,5)*S(2)+BL3*B(3,5)*S(3)+
     .BL4*B(4,5)*S(4)+A*S(16)*BBL
      FV(6)=-BL6*S(6)+BL4*B(4,6)*S(4)+A*S(17)*BBL
      (CONTINUED)
      FV(7)=-BL7*S(7)+BL5*B(5,7)*S(5)+BL6*B(6,7)*S(6)+
     .A*S(18)*BBL
      FV(8)=-BL8*S(8)+BL5*B(5,8)*S(5)+BL6*B(6,8)*S(6)+
     .BL9*B(9,8)*S(9)+A*S(19)*BBL
      FV(9)=-BL9*S(9)+BL6*B(6,9)*S(6)+BL7*B(7,9)*S(7)+
     .A*S(20)*BBL
      FV(10)=-BL10*S(10)+BL8*B(8,10)*S(8)+BL9*B(9,10)*S(9)+
     .A*S(21)*BBL
      FV(11)=-BL11*10*A*S(11)+B(1,11)*BL1*S(1)+B(2,11)*BL2*S(2)
     .+B(3,11)*BL3*S(3)+B(4,11)*BL4*S(4)+B(5,11)*BL5*S(5)+B(6,11)*BL6
     .*S(6)+B(7,11)*BL7*S(7)+B(8,11)*BL8*S(8)+B(9,11)*BL9*S(9)
     .+B(10,11)*BL10*S(10)
      DO 10 I=12,21
   10 FV(I)=-A*BBL*S(I)+A*BL11*S(11)
      FV(22)=BL10*B(10,22)*S(10)
      RETURN
      END
```

PROGRAM FOR EXPECTED COST ESTIMATION

```
      DIMENSION C(10),G(10),YI(10),P1(901,6),P2(901,6),Q(90,2)
     . ,Y(10,10),PHI(10,10,90),YIK(10,10,90),YT(90,2),Y1(901)


      C=VECTOR TO STORE THE VALUES OF CONSTANTS
      G=VECTOR TO STORE THE TEN DENSITY FUNCTIONS
      YI=THE COST FUNCTIONS FROM RESPECTIVE DENSITY FUNCTIONS
      P1 & P2=MATRICES TO STORE THE TEN DENSITY FUNCTIONS VALUES AS
      A FUNCTION OF TIME
      Q=MATRIX TO STORE YIK FOR PLOTTING
      PHI=TRANSITION PROBABILITY MATRIX AT DIFFERENT VALUES OF TIME
      Y=SUM MATRIX OF Y(I) AND Y(K);IN REFERENCE TO THE ACTUAL FORMULA
      YIK=MATRIX OF EXPECTED LOSS DUE TO CONTROL TRANSFER
      YT=MATRIX OF TOTAL EXPECTED LOSS
      Y1=VECTOR OF EXPECTED LOSS DUE TO A PROGRAM MODULE


      CC=1.0
      NF=901
      NC=10
      NT=6
      NK=3
      NNT=90
      T=0.0
      MM=0.0
      MP=0.0
      MS=0.0
      SN=10.0
      NX=2
      C(1)=2*CC
      C(5)=2*CC
      C(9)=2*CC
      C(2)=CC
      C(3)=CC
      C(8)=CC
      C(10)=CC
      C(4)=3*CC
      C(6)=3*CC
      C(7)=3*CC

     INITIALIZE THE MATRICES

      DO 90 I=1,NC
   90 G(I)=0.0
      DO 100 I=1,NF
      DO 100 J=1,NT
      P1(I,J)=0.0
  100 P2(I,J)=0.0
```

(CONTINUED)

```
      DO 103 I=1,NNT
      DO 103 J=1,NX
      Q(I,J)=0.0
 103  CONTINUE
      IC=0.0
      IO=1
      DO 101 I=1,NF
      P1(I,1)=(I-1.0)/SN
      P2(I,1)=(I-1.0)/SN
      IE=I-1.0
      IF((IE/10.0).NE.IC)GO TO 101
      IC=IC+1.0
      IF(IO.GT.NNT)GO TO 101
      Q(IO,1)=(I-1.0)/SN
      IO=IO+1
 101  CONTINUE
      DO 104 I=1,NNT
      DO 104 J=1,NX
      YT(I,J)=0.0
 104  YT(I,1)=I-1.0
;
;     READ THE INTERVAL TRANSITION PROBABILITY MATRIX
;
      DO 150 IU=1,NNT
      DO 150 I=1,NC
 150  READ(12,151)(PHI(I,J,IU),J=1,NC)
 151  FORMAT(10F12.6)
      LP=1
      DO 50 II=1,NF
;
;     CHECK FOR UNDERFLOW
;
      EPS=10.0**(-25)
  70  IF(G(1).LT.EPS.AND.G(1).GT.0.0)GO TO 71
      G(1)=EXP(-T/20.)
  71  IF(G(2).LT.EPS.AND.G(2).GT.0.0)GO TO 72
      G(2)=(3*(T**2)/500.0)*EXP(-3*(T**3)/1000.0)
  72  IF(G(4).LT.EPS.AND.G(4).GT.0.0)GO TO 73
      G(4)=(3*(T**2)/500.0)*EXP(-3*(T**3)/1000.0)
  73  IF(G(3).LT.EPS.AND.G(3).GT.0.0)GO TO 74
      G(3)=(T/10.0)*EXP(-2*(T**2)/500.0)
  74  IF(G(5).LT.EPS.AND.G(5).GT.0.0)GO TO 75
      G(5)=(T/10.0)*EXP(-2*(T**2)/500.0)
  75  IF(G(6).LT.EPS.AND.G(6).GT.0.0)GO TO 76
      G(6)=(T/20.0)*EXP(-(T**2)/200.0)
  76  IF(G(7).LT.EPS.AND.G(7).GT.0.0)GO TO 77
      G(7)=(1/2.0)*(T/20.0)*EXP(-T/20.0)
  77  IF(G(8).LT.EPS.AND.G(8).GT.0.0)GO TO 78
      G(8)=(1/(0.5*(3.14159**0.5)))*((T/20.0)**0.5)*EXP(-T/20.0)
  78  IF(G(9).LT.EPS.AND.G(9).GT.0.0)GO TO 79
      G(9)=(1/(0.5*(3.14159**0.5)))*((T/20.0)**0.5)*EXP(-T/20.0)
  79  IF(G(10).LT.EPS.AND.G(10).GT.0.0)GO TO 88
      G(10)=(1/6.0)*((T/20.0)**2)*EXP(-T/20.0)
      (CONTINUED)
```

```
     OBTAIN YI & Y1

 88  DO 10 I=1,NC
     YI(I)=0.0
 10  YI(I)=C(I)*G(I)
     Y1(II)=0.0
     DO 11 I=1,NC
 11  Y1(II)=Y1(II)+YI(I)
     IK=II-1
     IF((IK/10.0).NE.MM)GO TO 122
     MM=MM+1.0
     WRITE(6,121)(YI(I),I=1,NC)
121  FORMAT(10F12.5)
122  CONTINUE

     OBTAIN P1 & P2

     DO 80 IS=2,NT
     ID=IS-1
 80  P1(II,IS)=YI(ID)
     DO 81 IS=2,NT
     ID=IS+4
 81  P2(II,IS)=YI(ID)

     OBTAIN Y

     DO 20 I=1,NC
     DO 20 K=1,NC
     Y(I,K)=0.0
     IF(I.EQ.K)GO TO 20
     Y(I,K)=YI(I)+YI(K)
 20  CONTINUE

     OBTAIN YIK

     IK=II-1
     IF((IK/10.0).NE.MP)GO TO 132
     MP=MP+1.0
     IF(LP.GT.NNT)GO TO 132
     DO 133 IA=1,NC
     DO 133 IB=1,NC
     YIK(IA,IB,LP)=PHI(IA,IB,LP)*Y(IA,IB)
133  CONTINUE
     LP=LP+1
132  CONTINUE
     T=II/10.0
 50  CONTINUE
     (CONTINUED)
```

```
C
C       OBTAIN YT
C
        IO=1
        DO 144 II=1,NF
        Y2=0.0
        IK=II-1.0
        IF((IK/10.0).NE.MS)GO TO 144
        MS=MS+1.0
        IF(IO.GT.NNT)GO TO 144
        Q(IO,2)=YIK(1,10,IO)
        DO 145 I=1,NC
        DO 145 K=1,NC
  145   Y2=Y2+YIK(I,K,IO)
        YT(IO,2)=Y1(II)+Y2
        IO=IO+1
  144   CONTINUE
        DO 701 IO=1,30
        S=IO
        IE=IO+30
        IH=IO+60
        SR=IE
        SRN=IH
  701   WRITE(11,700)S,YT(IO,2),SR,YT(IE,2),SRN,YT(IH,2)
  700   FORMAT(2X,3(F5.2,2X,E12.4,6X))
        DO 705 I=1,NNT
  705   WRITE(6,706)(Q(I,J),J=1,NX)
  706   FORMAT(2F10.5)
        CALL PLOT(1,P1,NF,NT,NF,1)
        CALL PLOT(2,P2,NF,NT,NF,1)
        CALL PLOT(3,Q,NNT,NX,NNT,1)
        CALL PLOT(4,YT,NNT,NX,NNT,1)
        STOP
        END
```

## REFERENCES

[1] J. Dickson, J. Hesse, A. Kientz and M. Shooman, "Quantitative analysis of software reliability", 1972 Annual Reliability Symposium Proceedings, IEEE, Jan. 1972.

[2] M. J. Ratynski: The Airforce computer program acquisition concept, SJCC, 1967.

[3] O. L. Williamson, G. G. Dorris, A. J. Ryberg, and W. E. Straight: Development studies of a software reliability program, NASA report, March 1970.

[4] R. L. London, "Software reliability through proving programs correct", 1971 International Symposium on Fault Tolerant Computing (IEEE), March 1971.

[5] E. W. Dijkstra, "Programming considered as a Human activity", Proc. of the IFIPS Congress 1965, p. 213-217.

[6] M. Shooman, "Probabilistic models for software reliability prediction", 1972 International Symposium on Fault Tolerant Computing, Newton, Mass., June 1972, IEEE Computer Society.

[7] A. J. Gross and M. Kamins, "Reliability assessment in the presence of reliability growth", RAND Corp., Santa Monica, California, Rept. RM-5346-PR, Sept. 1967.

[8] IBM Systems Assurance Dept., SAFEGUARD project, Federal Systems Division, "Safeguard code certification Experiment",

[9] R. E. Barlow, R. Proschan, E. M. Scheur, "A system debugging model", Operations Research Center, University of California at Berkeley, April 1969.

[10] D. M. Bender, "The prediction and measurement of system availability; A Bayesian Treatment", IEEE Trans. on Reliability, vol. R-17, Sept. 1968, pp. 127-138.

[11] P. L. Moranda and Z. Jelinski, "Final Report on software Reliability study", McDonnel Douglas Astronautics Company, MDC Report No. 63921, Dec. 1972.

[12] Myers, "Software Reliability, Principles and Practices", John Wiley & Sons, 1976.

[13] B. Littlewood, "MTBF is meaningless in software reliability", (letter) IEEE Trans. Reliability, Vol. R-24, 1975 Apr., p.82.

[14] P. B. Moranda and Z. Jelinski, "Software Reliability Research Conference on Statistical Methods for the Evaluation of Computer Systems Performance, Providence, R.I., Nov. 1971.

[15] M. Shooman, "Operational Testing and Software Reliability Estimation During Program Development", 1973 IEEE Symposium on Computer Software Reliability, N.Y.C. April 30-May 2, 1973.

[16] M. Shooman, "Software Reliability: Measurement and Models", 1975 Annual Reliability and Maintainbility Symposium, Washington, DC, Jan. 28-30, 1975.

[17] M. Shooman, "Probabilistic Models for Software Reliability Prediction", Conference on Statistical Methods for the Evaluation of Computer Systems Performance, Providence, R.I., Nov. 1971.

[18] J. D. Musa, "Software Reliability Measurement", The Journal of Systems and Software, Vol. I, 1980, pp. 223-241.

[19] B. Littlewood, "How to Measure Software Reliability and How Not to", IEEE Transaction on Reliability, Vol. R-28, No. 2, June 1979.

[20] B. Littlewood, "A Reliability model for Markov structured software", in Proc. 1975 Int. Conf. Reliable Software, Los Angeles, CA, Apr. 1975, pp. 204-207.

[21] B. Littlewood, "A Semi Markov model for Software Reliability with failure costs", in Proc. Symp. Comput. Software Eng. New York, N.Y., April 1976, pp. 281-300.

[22] B. Littlewood, "Software Reliability Model for Modular Program Structure", IEEE Transactions on Reliability, Vol. R-28, No. 3, August 1979.

[23] B. Littlewood, "A Critique of the Jelinski-Moranda Model for Software Reliability", IEEE 1981 Proceedings Annual Reliability & Maintainbility Symposium.

[24] M. Shooman & A. K. Trivedi, "A Many-state Markov Model for Computer Software Performance Parameters", IEEE Transactions on Reliability, Vol. R-25, No. 2, June 1976.

[25] J. D. Musa, "A Theory of Software Reliability and its applications", IEEE Trans. on Software Engineering, Vol. SE-1, No. 3, Sept. 1975.

[26] J. D. Musa, "Software Reliability Measurement", Presented at Software Life Cycle Management Workshop, Airline, Va, Aug. 1977.

[27] J. D. Musa, "The use of Software Reliability measures in Project Management", IEEE C-1338-3/78/0000-0493500.78, March 1978.

[28] J. D. Musa, "Software Reliability Measures Applied to Systems Engineering", Presented to National Computer Conference, AFIPS Conference Proceedings, Volume 48, Montvale, N.J., 1979.

[29] J. D. Musa, "Validity of Execution - Time Theory of Software Reliability", IEEE Trans. on Reliability, Volume R-28, No. 3, Aug. 1979.

[30] R. W. Wolverton and G. J. Schick, "Assessment of Software Reliability", TRW Systems Group, Report No. TRW-SS-72-04, Sept. 1972.

[31] Sukert, Capt Alan N., "A Software Reliability Modelling Study", RADC-TR-76-247, July 1976.

[32] Sukert, Capt Alan N., "An investigation of software Reliability models", Proc. 1977 Annual Reliability and Maintainability Symposium.

[33] L. E. James, J. E. Angus, J. B. Bowen and J. C. McDaniel, "Combined Hardware/Software Reliability Models", Final Report submitted to RADC, Contract F 30602-80-C-0085.

[34] R. D. Haynes and W. E. Thompson, "Combined Hardware and Software Availability", 1981 Proc. Annual Reliability AND Maintainability Symposium.

[35] J. E. Angus and L. E. James, "Combined Hardware/Software Reliability Models", 1982 Proc. Annual Reliability AND Maintainability Symposium.

[36] R. C. Cheung, "A user oriented software Reliability Model", IEEE Trans. on Software Engineering, Vol. SE-6, No. 2, March 1980.

[37] C. V. Ramamoorthy, R. C. Cheung and K. H. Kim, "Reliability and integrity of large computer programs", Comput. Syst. Rel., Infotech State of the Art Rep. 20, 1974.

[38] R. C. Cheung, "A structural theory for improving software reliability", Ph.D. dissertation, Dep. Elec. Eng. Comput. Sci., Univ. California, Berkeley, 1974.

[39] Supplement: NAPS document No. 03307; ASIS-NAPS; Contains detailed mathematical analysis, including formulas for computing the parameters of the approximating models.

[40] M. L. Shooman, "Probabilistic reliability: an engineering approach", McGraw-Hill, New York, 1968.

[41] F. P. Brooks, "The Mythical Man-month: Essays on Software Engineering", Reading, Mass.: Addison-Wesley, 1975.

[42] G. J. Myers, "Reliable Software Through Composite Design", New York: Petrocelli/Charter, 1975.

[43] C. Singh and R. Billinton, "System Reliability Modelling and evaluation", Hutchison, 1977.