Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

SUBRAMANIAN, Parmeshwar

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M.A.Sc.(Mechanical Engineering)

GRADE - DEGREE

Mechanical Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

THE STOCHASTIC ANALYSIS OF TECHNICAL REDUNDANT SYSTEMS

Balbir S. Dhillon

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

D. Necsulescu

T. Mussivand

J.-M. De Koninck, Ph.D.

Jean-Marie De Koninck

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

THE STOCHASTIC ANALYSIS OF TECHNICAL REDUNDANT SYSTEMS

BY

PARMESHWAR SUBRAMANIAN

A Thesis

presented to the University of Ottawa

September 2000

in partial fulfillment of the

requirements for the degree of

MASTERS OF APPLIED SCIENCES

in

MECHANICAL ENGINEERING

Ottawa-Carleton Institute for

Mechanical and Aeronautical Engineering

0-612-67867-9

Acknowledgments

I am highly indebted to my Thesis supervisor, Dr. B .S . Dhillon, for his patience and continuous guidance in my work. I express my gratitude to my parents, my uncles Messrs. Johnson and Raghunathan for their continued support and help. I would also like to thank M/s Kathleen Harris who works as a journalist for the Ottawa Sun for editing my thesis.

Above all, I thank God for making my dream come true.

Abstract

Today computers are widely used and their failures can lead to catastrophe. Redundancy is used to improve computer system reliability. This study presents a qualitative analysis of triple modular redundant (TMR) systems with redundant voters that can help to reduce the risk of failures. The voters examine the output of triple modular systems. Markov technique was employed to perform the analysis. Generalized formulas for reliability and mean time to failures of TMR systems with N voters have been developed. In addition, the system state probabilities in s-domain and steady state probabilities have been presented. The study clearly demonstrates improvements in system reliability, mean time to failure and steady state availability by the introduction of repair and redundancy in these systems.

Terms and Definitions

Availability	is the probability that, at any given instant in time, the system is functioning normally and available for use.
Dependability	describes the overall quality of service provided by the system.
Error	is the occurrence of an incorrect value in some unit of information with the system.
Failure	is a deviation from the expected performance of the system.
Fault	is a physical defect, flaw or imperfect occurring in the hardware or software of the system.
Fault Avoidance	is the process whereby attempts are made to prevent hardware failures and software errors before they occur in the system.
Fault Containment	is the process of confining or restricting the effects of a fault to a limited scope or locality of the system.

Fault Masking	is the process of preventing faults from causing errors.
Fault-Tolerant Computing	is the process of performing computations, such as calculations, on a computer in a fault-tolerant manner.
Fault-Tolerant System	is one that is capable of correct operation despite the presence of certain hardware failures and software errors.
Redundancy	is defined as the existence of more than one means of performing a stated function.
Maintainability	is the probability that, within a specified time, an inoperable system will be restored to a operational state.
Performability	is the probability that, for a given instant in time, the system is performing at or above a specified level of performance.
Reliability	is the probability that, throughout an interval of time, the system will continue to perform correctly.
Safety	is the probability that the system will either perform correctly or discontinue in a well defined and safe manner.
Testability	is the ability to test for certain attributes or conditions with the system.

System Failure	is the external state manifestation of an error such that the entire computing system has to be stopped.
Intermittent Fault	is a fault that is only occasionally present due to unstable hardware or software states.
Permanent Fault	is a fault that is continuous and stable. In hardware, permanent failure reflects an irreversible physical change. The word hard is used interchangeably with permanent.
Transient Fault	is a fault resulting from temporary environmental conditions. The word soft is used interchangeably with transient.
Mean Time to Failure	is defined as the average time for a system to fail.
Mean Time to Detection	is defined as the average time required to trace an error.
Mean Time in Between Failures	is defined as the average time in between two failures of a system.
Mean Time to Repair	is defined as the average time it takes to correct a fault after an error is detected.

Mean Time to Restart	**is defined as the average time required for a system to start after the error is detected and corrected.**

Title

Table of Contents Pg.No.

List of Tables

List of Figures

Chapter 1

1. Introduction

The remarkable progress of modern technology enables us to manufacture complex systems which play an important role in our society. Examples of such systems are a traffic control system, a communication system, a banking system, a seat reservation system, and an electronic switching system. Breakdown of such systems are costly, and dangerous. It is therefore of great importance to develop systems with high reliability.

1.1 Overview of Fault Tolerance

The concept of fault tolerance was introduced in the late 1960's in an effort to build more reliable and safe systems. Over the years, the concept of fault tolerance was accepted as an important factor in design of systems and since 1971 the International Symposium on Fault-Tolerant Computing is being held every year. The fault tolerance in a system may be defined as that system's built-in capability (redundancy) to preserve the continued correct execution of its programs and input/output functions in the presence of certain operational faults[2].

Systems using redundancy techniques can tolerate a certain number and/or types of failure while continuing to maintain the required relationship between input and output conditions. This is more likely to occur when the failures of individual components are statistically independent. The assumption of statistically-independent failure of individual components is easily violated in practice. Thus it is necessary to include fault-tolerance in designing systems.

1.2. Fault Tolerance Techniques

There are two fundamentally different approaches that can be taken to increase the reliability of a computing system. The first approach is called fault prevention and the second is fault tolerance. Fault prevention is almost impossible to achieve because its goal is to reduce the probability of system failure to an acceptably low value. In the fault tolerance approach, we incorporate redundancy in hardware systems. More specifically, fault tolerance is not a replacement but rather a supplement to the most important reliable system design. Fault tolerant techniques for hardware redundancy are discussed below.

Hardware redundancy is characterized by the addition of extra hardware resources, such as duplicated processors generally for the purpose of detecting or tolerating faults. There are three types of hardware redundancies:(i) static, (ii) dynamic, and (iii) hybrid redundancy.

Static hardware redundancy techniques use fault masking to hide faults and prevent errors from occurring due to faults. In this way, static designs inherently tolerate faults without the need for configuration or explicit fault detection and recovery. The most widely used form of static redundancy is triple modular redundancy (TMR). With TMR, the hardware is triplicated, the output is determined by taking a vote and choosing the majority or quorum value. The voting mechanism itself can be in the form of a hardware device or software polling routine, and can occur at multiple points in the system.

Dynamic hardware redundancy techniques attempt to detect and remove faulty hardware from operation. In this way, dynamic designs do not mask faults but instead use fault detection, location and recovery techniques to reconfigure the system for further operation. The dynamic approach is usually used in systems capable of tolerating temporary erroneous results while system experiences reconfiguration. There are four types of dynamic hardware redundancy:(i) duplication with comparison, (ii) standby sparing, (iii) pair and a spare, and (iv) watchdog timers.

With the **duplication with comparison** scheme, the hardware is duplicated in order to perform the same operations in parallel. The results of these parallel operation are compared and if they differ, an error condition is generated, thereby providing a level of fault detection. In the **standby sparing scheme**, one module operates while one or more modules are on standby. When a faulty module is detected and located, it is immediately replaced in operation by one of the spare modules during system reconfiguration.

The **pair and a spare** scheme is essentially a combination of the duplication with comparison and standby sparing schemes. That is, two modules operate in parallel, each of which having one or more standby spares available for replacement in the event of a module fault. In the event of a fault, one or both of the operating modules are replaced depending on the strategy. Finally, the **watchdog timer** scheme is sometimes an effective technique for detecting faults, and consists of timers which must be accessed within a prescribed amount of time. In this way, any hardware component not responding for an excessive amount of time can be considered faulty. Watchdog timers can also be used to detect software errors, such as an infinite loop which would prevent prompt timer access.

Hybrid Redundancy combines the static and the dynamic redundancy approaches. More specifically it consists of a TMR system with a set of spare modules. When one of the TMR modules fails, it is replaced by a spare and the operation continues. It has a voter and a disagreement detector which detects if the output of any of the three active modules is different from the voter output and that particular module is replaced by a spare or a switching system. The voter output will be correct as long as at least two modules have correct outputs. If all the spares are used up the system becomes a standard TMR system.

The hybrid system cannot tolerate more than two failed modules at one time. For example, if there are two failed modules in the hybrid system voter will switch out the fault free module and switch a spare in. Since the majority of modules are malfunctioning, the system will reject all good spares until all of them are used up and the system will crash. A comprehensive list of references on fault-tolerance are given in Refs. [50].

1.3 Review of Publications on Fault Tolerance

Today fault tolerance is a widely used approach in technical redundant systems. This includes the world of communications, software, hardware, nuclear, aerospace and medical equipment. Over the last three decades, there have been a large number of publications on fault tolerance as shown in Figure 1. These publications are classified into four categories as shown in Table 1.1. Each of this category has been reviewed below.

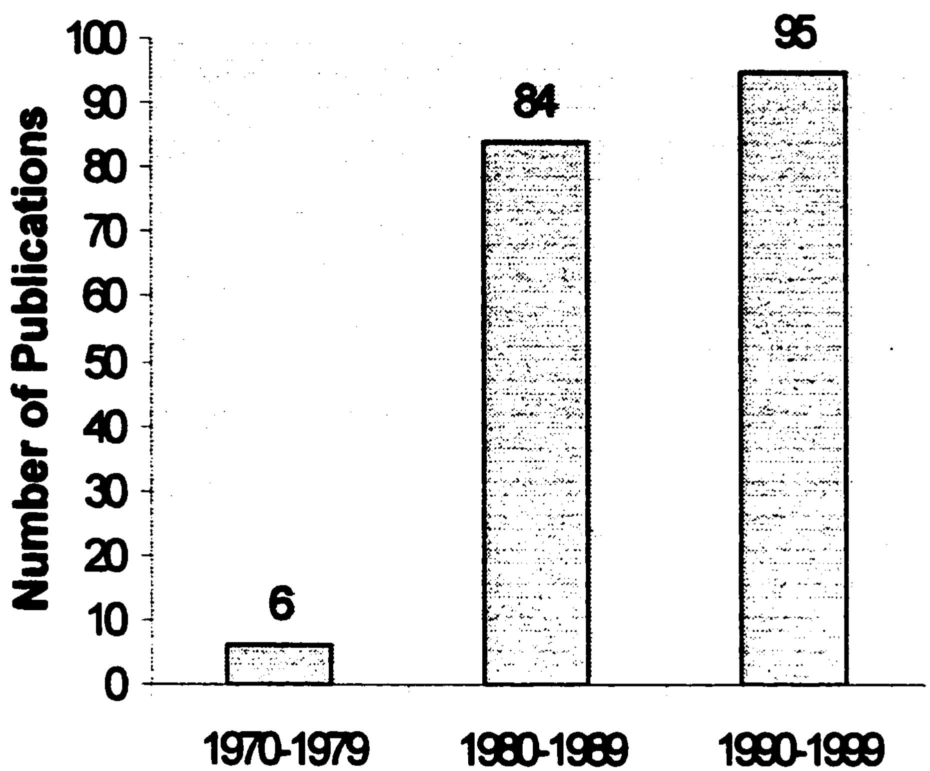


Fig. 1 Publications on Fault Tolerance

Hardware Modeling of Fault Tolerance

Hardware modeling of fault tolerance includes publications on modeling techniques developed by various authors that describe failures such as transient, permanent and correlated faults using the Markovian technique. This technique is used to analyze system reliability, system availability, and mean time of failures of various systems. A publication of a semi-Markov model has also been presented but this technique has not been explored due requirement of enormous data. Lastly, a few papers involving data collection have also been discussed.

Hardware Fault Tolerance

This category contains a few publications which describe fault tolerant schemes like, automated fault tolerance design, checkpointing, adjudication, and triple modular redundancy. Expressions for expected completion time of triple modular redundant systems have been derived.

Software Modeling of Fault Tolerance

Software modeling of fault tolerance includes publications associated with the Markovian modeling techniques for building highly reliable software with multiple identical versions. Optimization of these versions have been discussed.

Software Fault Tolerance

This category contains a number of publications on software fault tolerant techniques, like N-version programing, recovery block, and N self-checking programming. These techniques have been used to derive a voting strategy which compare identical sub-systems in an effort to improve system reliability.

Table 1.1 Classification of Literature

Category	References
Hardware Modeling of Fault Tolerance	29,19,110,206,213,166,199,120,31,207,78,119,176,104,51,64,15,151,149,16, 105,165,142,162,159,35,102,111,112,33,28,208,49,50,84,104,83,4,205,88,198, 228,89,186,138,22,188,141,97,32,60,62,124,52,182, 204,239,109,103,59,53,85, 229,91,90,212,211,39,23,155,61,58,100,55,54,56,87,132,14,170,226,190,5,68, 184,169
Hardware Fault Tolerance	167,2,148,106,13,73,210,108,230,30,12,127,191,192,232,168,178,181,135,75, 79,158,157,180,76,18
Software Modeling of Fault Tolerance	24,52,200,201,202,66,140,139,240,177,113,177,113,145,11,21,57,183,115,241, 137,215,216,36,37
Software Fault Tolerance	214,77,7,8,9,17,3,38,224,92,144,160,161,203,187,133,134,65,114,189,218,20, 193,86,172,173,6,94,95,118,174,136,153,180,154,1,45,46,47,96,134,194,48,34, 42,121,164,195,220,221,27,26,41,44,99,101,143,179, 225,217,219,231,222, 171,235,233,234,227,223,40,63,67,82,80,116,117,209,238,236,237,72,98,107, 126,156,196,197,125,81,117,122,123,152

1.3.1 Hardware Modeling Techniques for Fault Tolerance

Breuer, A. M., Ball, M. [29], in one of the earliest publications on fault tolerance in 1973 presented fundamental results relating to the problem of characterization and detection of intermittent faults in digital circuits. The results showed that no single test sequence could detect all possible intermittent faults.

Four years later, in 1977, Zvi, K., Koren, I. [120], improved this method for locating intermittent faults in combinational networks by a minimal diagnosis experiment. This method yields a minimal diagnosis and in addition the maximum possible diagnostic resolution.

In the year 1981, Pedar, A., Sharma, V. V.S.[176], used an approach based on multi-objective phased-mission analysis to evaluate the effectiveness of a distributed avionics architecture. It also viewed the computing system as a multi state s-coherent structure and various levels of performance were evaluated.. In the same year, Malaiya, K. Y. [151], extended the above approach by introducing two measures using linear correlation to describe the s-dependence of intermittent failures. One measure related the linear correlation for the entire period during which faults were active in different modules. The other measure related to the closeness in the time of the instance faults became active in different modules.

Five years later, in 1986, two publications appeared on intermittent failures in computer systems. Iyer, R. K. et. al. [102], proposed a methodology for recognizing the symptoms of persistent problems in large systems. The author used the system error rate to identify the error states among which relationships existed and quantified the results. In the same year, Sumita. et. al. [208], conducted a study which analyzed the performance of software subsystems under the influence of hardware failures. Expressions of various time dependent performance measures were derived.

In 1987, Goyal, A. et. al.[83], described the stochastic technique as an increasingly important factor in effectively minimizing intermittent and permanent failures in commercial computer systems. In the same year, Bengt, O.[22], described a Markov approach to include temporary faults in the reliability model of a repairable dual redundant system. It gives considerably higher mean time between system failure than the traditional permanent fault model.

Butler, G. R., White, A .L.[32], in 1988, described a program called SURE (Semi-Markov Unreliability Range Evaluator). The SURE reliability analysis method utilizes a fast bounding theorem based on means and variances and a fast bounding theorem based on means and percentiles. This method requires enormous failure and repair data for calculating the reliability of repairable systems particularly ones with intermittent failure.

In 1989, Dugan, B.J., Trivedi, K. S. [60], extended the Markov approach[22] by introducing fault coverage in a fault tolerant system. Fault coverage refers to the ability of a system to automatically recover from the occurrence of a fault during normal system operation.

A total of five publications appeared a year later on data measurement and data collection in addition to the modeling techniques. For example, Jurgen, D. [109], in the first, presented a modeling approach to investigate the interdependencies between memory faults and system performance. The results demonstrated that the performance decrease caused by memory errors depended on system workload and operating system characteristics. Dungan, B.J.[59], in the second, summarized the presentations of the work being done in the field of measurement and modeling of computer systems. In the third, Gray, J. [85], summarized a census of customer outages reported by Tandem. The census showed a clear improvement in the reliability of hardware and maintenance. It indicated that software was the main source of outages (62%) followed by systems operation (15%). In the fourth, Chillarege, R., Siewiorek, D. P [39], emphasized the need for reliability modeling and data collection in validating design, development and manufacture of a product. Lastly, McAllister, F.D., Wakerly, F. J. [154], presented a voting measurement and data collection strategy in a fault tolerant software system where an attempt was made to differentiate between correctness and agreement.

In 1992, Laprie, C. J., Kanoun, K.[132], developed a new model for calculating system's reliability and availability with included the various classes of faults like intermittent, temporary, and permanent faults. It also showed that classical reliability theory can be generalized in order to cover both hardware and software viewpoints.

A year later, Nitin, H .V., Dhiraj, K. P. [170], extended the above modeling technique to a modular redundant system consisting of multiple identical modules and an arbiter are considered. Strategies were formulated that achieved a maximal combination of safety and reliability. In the same year, Walter,A.B., Jai, M.[226], improved the above approach by introducing two data organization schemes based on maximum distance separable error correcting codes. The multiple check disk approach to improve reliability was found to be an excellent option.

Lastly, Shakil, A. [190], in 1996, developed recursive expressions for mean time to failure and mean time in between failures for repairable systems, considering perfect and imperfect fault-coverage. The study showed that mean time to failure and mean time between failures increases with higher coverage.

1.3.2 Hardware Fault Tolerance

Tamir, Y., Gagni, E. [210], in 1987, presented a new hardware fault tolerance scheme for large multi-computers. The author's scheme was based on simultaneous execution of identical copies of the application on two subnetworks of the system. The scheme provided a high probability of detecting errors caused by hardware faults without requiring the hardware to be self checked.

Sheth, A. [192], two years later, extended the above hardware fault tolerance scheme to a large database with a simple model and showed the effects of fault-tolerance techniques and system design on increased the system availability.

In 1990, two new hardware techniques appeared. Laura, L. [135], presented a technique called the automated fault tolerance design, simulation, and analysis tool which enabled users to evaluate reliability and fault tolerant characteristics of complex and adaptive multiprocessing systems using graphic and text user interfaces. In the same year, Giandomenico, D. F. Strigini, L. [75], gave a rigorous definition of the adjudication problem, summarized the existing literature on the topic, investigated the use of probabilistic knowledge about errors and faults in the sub-components of a fault tolerant component to obtain good adjudication functions. The concept of an optimal adjudicator(a concept more general than the voter) has been introduced and discussed in this publication.

In 1991, Meyer, J. F., Schlichting, R.D. [158], classified the four methods of hardware fault tolerance:

- Restarts
- Triple Modular redundancy
- Checkpointing, centralized and distributed
- Resilient

for each of the above fault tolerance methods, an expression for the expected completion time, as a function of the number of processors in the system, and numerical results were provided

Software Modeling Techniques for Fault Tolerance

Don, O .K., Kua, H. K. [52], in the year 1988, used exponential and weibull distributions for calculating the reliability of an electronic network configuration. This paper presented the results of a proposed reliability model of the IEEE 14 bus network configuration. In the same year, Siegrist, K.[200,201], improved on the above technique by explaining that the system reliability can be measured in terms of module reliability and transition probabilities by transferring control from one module to the other according to a Markov process.

The following year three new software modeling techniques appeared. Fevzi, B., Piotr, J., [66], reviewed available techniques for achieving fault tolerant programs and showed how to derive the failure probabilities of basic faults for achieving a fault tolerant program. Levendel, Y. [140,139], emphasized that system reliability is inversely related to the amount of unpaired defects in the system. The author proposed a methodology called partitioning to lower defects in system development. Zeil, J. S., [240], presented an approach called the perturbation approach to software testing which focuses on faults within arithmetic expressions appearing throughout the program. This technique opened up an interesting view about testing in which faults in the system that would have escaped detection could be detected.

In 1991 Eckhardt, E.D., Alper, K.C. [11], showed how multiple versions of independently developed software can be used for achieving higher reliability. The experimental results were based on 20 versions of an aerospace application developed and independently validated by 60 programmers from four universities. Belli, F. P.[21], extended the approach of reliability optimization to software redundancy. Optimization models have been formulated covering the modified recovery block scheme and multi version programming approaches.

Two years later, Robert, G., Kishor, T.[183], developed an analytical model for predicting the performance and reliability of mirrored disk subsystems and suggested a mechanism for optimal selection.

In 1995 two publications examined the impact of software faults. Lee, I . Iyer, R. K. [137] , described with a new Markov model the impact of software faults on the reliability of the overall system and Tian, J. [215], presented a tree based approach to examine the reliability of large software systems. These approaches have been demonstrated to be applicable and effective in testing of large software systems.

Lastly, Chen, S .M., Mills, S. [36], in 1996, presented a new binary Markov process model for random testing of software systems. In general it demonstrated that the effect of possible correlation between test runs cannot be ignored in estimating software reliability.

Software Fault Tolerance

Avizienius, A. [17], in 1985, presented an N-version approach to the tolerance of design faults. Principal requirements for the implementation of N-version software are summarized and goals of current research are presented and some potential benefits of the N–version approach have been identified.

Three publications appeared on software fault tolerant techniques in 1987. Laprie, J. C. et.al.[133], applied the N-version approach via the Markov technique on the three methods for tolerating software faults. These are the recovery block technique, the N–version programming, and the N self-checking programming technique. In the same year, Scott, R. K. et. al. [189], examined in detail the three methods of creating fault tolerant software systems, Recovery Block, N-Version Programming, and Consensus Recovery Block and a simple cost model that showed the relative costs of increasing software reliability using the three fault-tolerant methods were presented. Lastly, Tyrrell, A.M.[218], extended the N-version to a new software fault tolerance mechanism termed conversation which is used for error detection and fault recovery in distributed systems.

In 1989, two more papers appeared on software fault tolerant techniques. Kim, K. H.,Welch, H.O. [118], extended the concept of distributed execution of recovery blocks towards an approach for uniform treatment of hardware and software faults. The results of this study indicated the feasibility of achieving tolerance of hardware and software faults in a broad range of real time computer system by use of the above scheme. Lee, P. N., Tamboli, A.[136], presented a new concept by introducing concurrent correspondent modules as a strategy to implement software fault tolerance. It was based on the concurrent execution of the primary and its redundant correspondent versions.

In 1990, McAllister, F. D. et. al. [154], examined an independent N-version programming reliability model which distinguished between correctness and agreement. Authors proposed a voting strategy, viz, consensus voting, to treat cases when there was an agreement among incorrect outputs. The consensus voting strategy adapts the voting various version reliability and output-space cardinality characteristics. The results indicate that as more and more versions are added the overall reliability decreases. It shows In the same year, Sullivan, G. F. et. al.[194], improved upon the above methodology by presenting two programs for certification, the first program left a data trail called a certification trail, the second program solved the program again but this program had access to the trail of the first program. Then finally the two results where compared and the correct results were accepted.

In 1991, Shimeall, T., Leveson, N. G.,[193], described a large scale experiment comparing software fault tolerance and software fault elimination to improve software reliability.

In 1994, Noushin, A. et. al. [171], extended the N-version approach by presenting a framework illustrating the trade-off between the cost of using N–version programming and the improved reliability for a software system. The two models dealt with: a single task, and multiple software. These software systems consisted of several modules where each module performed a subtask and, by sequential execution of modules, a major task was performed.

Optimization models were used to select the optimal set of versions for each module such that system reliability was maximized.

Goseva, P.[82], a year later, presented a detailed, but efficiently solvable model of the N-version programming for evaluating the reliability and perform ability over a mission period. The hierarchical decomposition method was employed to reduce the model complexity and provide a modeling framework for evaluating the N-Version Programming failure.

Geoghegan. S. J., Avresky, D.[72], in 1996, proposed a formal approach for adding fault detection to software. An assertion based formalism was used to represent specifications and verify completeness and consistency. This technique embeds two types of software checks which can also detect hardware faults.

Lastly, in 1998, Marcelo, L., Bernard, C. [152], described a fault tolerant system that is based on two replicas of a self checking module and on an error-masking interface. The main contribution of this work relied on a fail safe design of the error masking interface, and on the analysis of the competitiveness of this fault tolerant scheme with respect to its reliability.

The main findings of the literature review on technical redundant systems were:

- Hardware fault tolerance techniques could be used to analyze system reliability, and system availability of redundant systems. Furthermore, mean time to failure of triple modular redundant systems have also been derived.
- Markovian techniques have been used successfully for building highly reliable redundant software systems.

- Repairable and non-repairable technical redundant systems are good methods to analyze failure probabilities via Markovian modeling.

1.4 Objectives

Triple Modular Redundancy (TMR) is one of the fault tolerant techniques. The main objective of this study was to analyze the effects of Triple Modular Redundant systems with N redundant voters in series on system reliability, availability, and system mean time to failure. To meet this objective repairable and nonrepairable TMR systems with N redundant voters in series are considered. Detailed reliability analysis using the Markov state space technique were performed on these systems.

The following is the main difference between the models considered in this study and those proposed by previous researchers:

• The models proposed by previous researchers consider the occurrence of fault tolerance in a TMR network with a voter. The models which are analyzed in this study consider the occurrence of failure irrespective of the number of redundant voters.

Under the above mentioned condition, generalized expressions for system reliability, system availability, mean time to failure, and the probability equations in the s domain of TMR systems with N voters were to be developed. In addition, with increase of redundant voters improvements in reliability, mean time to failure, and steady state availability of these systems were proposed to be studied.

1.5 Organization of Study

This study is presented in six chapters.

Chapter 1 gives a brief introduction of fault-tolerance and presents the literature review.

Chapter 2 is concerned with the reliability analysis of non-repairable TMR systems with N voters in series. In addition to the generalized expressions for system reliability and mean time to failure, improvements in system reliability and mean time to failure are studied as the number of voters are increased from zero to three.

Chapter 3 is concerned with the reliability analysis of repairable TMR systems with N voters in series. In addition to the generalized expressions for probabilities in the s domain, improvements in system reliability and mean time to failure are studied as the number of voters are increased from zero to three.

Chapter 4 is concerned with the availability analysis of fully failed TMR systems with N voters in series. In addition to generalized expressions for system availability for N voters, improvements in availability are shown as the number of voters are increased from one to four.

Chapter 5 is concerned with the availability analysis of a partially failed TMR systems having N redundant voters in series with intermittent states in repair. Improvements in system availability are shown as the number of voters are increased from one to three.

Chapter 6 discusses the conclusions and recommendations for future studies.

Chapter 2

TMR System with N Voters

This system represents a TMR network with N redundant voters in series as shown in Fig.2.1. A TMR system is one in which at least two out of three units have to be working for system success. This means that if any one of the units of a TMR network fails the voter detects the failure and that unit is phased out. More specifically, a voter checks the output of the two working units and decides if the value is good, if not one of the units is phased out and the third unit is commissioned. The system will operate as long as two units and one voter is working. This is static type of redundancy which could be widely used in the various industries.

This chapter presents a total of four models representing TMR systems with increasing number of voters from zero to three. The reliability analysis of TMR systems with redundant voters in series are presented. Similarly, generalized expressions for reliability and mean time to failure for the above system with N voters are developed.

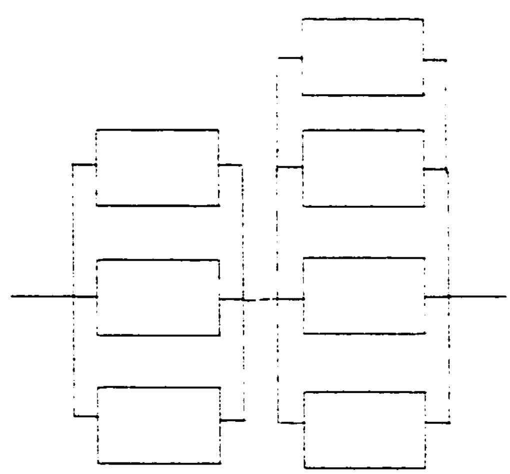


Fig. 2.1 A TMR Subsystem with N Voters

Assumptions

The following assumptions are associated with all the models studied in this chapter:

- Failure rates are exponential.
- Failures are statistically independent.
- All TMR subsystem units are identical.
- All Voter units are identical
- At least two units and one voter must operate for the system success.

Notations

The following symbols are associated with all the TMR system models in the chapter:

λ is the single unit failure rate.

λ_v is the unit voter failure rate.

t is time.

R(t) is the reliability of the system in [0,t].

s is the Laplace transform variable.

MTTF is the system mean time to failure.

n is the number of states.

$P_i(t)$ is the probability that the system is in state i at time t, for i=0,1,2,.....,n+1.

N is the number of voters.

2.1 Model I

For N=0, this section presents the reliability analysis of a Triple Modular Redundant System shown in Fig.2.1. by using the Markov state technique. The system state space diagram is shown in Fig.2.1.1 At time t=0 all the three units start working normally.

The numeral in each box in Fig.2.1.1 denotes the state number. At least two units must function normally to ensure the system success.

Fig.2.1.1. State space diagram of a TMR subsystem.

The following equations are associated with Figure. 2.1.1.

$$\frac{dP_o(t)}{dt} + 3\lambda P_o(t) = 0 \tag{2.1}$$

$$\frac{dP_1(t)}{dt} + 2\lambda P_1(t) = P_o(t)3\lambda \tag{2.2}$$

$$\frac{dP_2(t)}{dt} = 2\lambda P_1(t) \tag{2.3}$$

At $t=0$, $P_o(0)=1$ and $P_1(0)=P_2(0)=0$.

By using the Crammer's rule and Equations (2.1)-(2.3), probabilities in the s domain are:

$$P_o(s) = \frac{1}{(s+3\lambda)} \tag{2.4}$$

$$P_1(s) = \frac{3\lambda}{(s+3\lambda)(s+2\lambda)} \tag{2.5}$$

$$P_2(s) = \frac{6\lambda^2}{s(s+3\lambda)(s+2\lambda)} \tag{2.6}$$

Using Equations (2.4)-(2.6), we get the following state probabilities:

$$P_o(t) = \exp(C_1 t) \tag{2.7}$$

$$P_1(t) = \frac{3\lambda \exp(C_1 t)}{C_1(C_1 - C_2)} + \frac{3\lambda \exp(C_2 t)}{C_2(C_2 - C_1)} \tag{2.8}$$

$$P_2(t) = \frac{6\lambda^2}{C_1 C_2} + \frac{6\lambda^2 \exp(C_1 t)}{C_1(C_1 - C_2)} + \frac{6\lambda^2 \exp(C_2 t)}{C_2(C_2 - C_1)} \tag{2.9}$$

Where $k_1 = 3\lambda$, and $k_2 = 2\lambda$.

The roots of the Equation $s^2+s(k_2+k_1)+(k_1k_2)$ are $C_1 = -3\lambda$, and $C_2= -2\lambda$.

The reliability of the system is

$$R(t) = P_o(t) + P_1(t) \tag{2.10}$$

The system mean time to failure is given by

$$\mathrm{MTTF} = \int_0^\infty R(t)\, dt$$

$$= \frac{k_2 + 3\lambda}{k_1 k_2} \tag{2.11}$$

The plots of state probabilities and system reliability are shown in Figures. 2.1.2 and 2.1.3 for the specified model parameters.

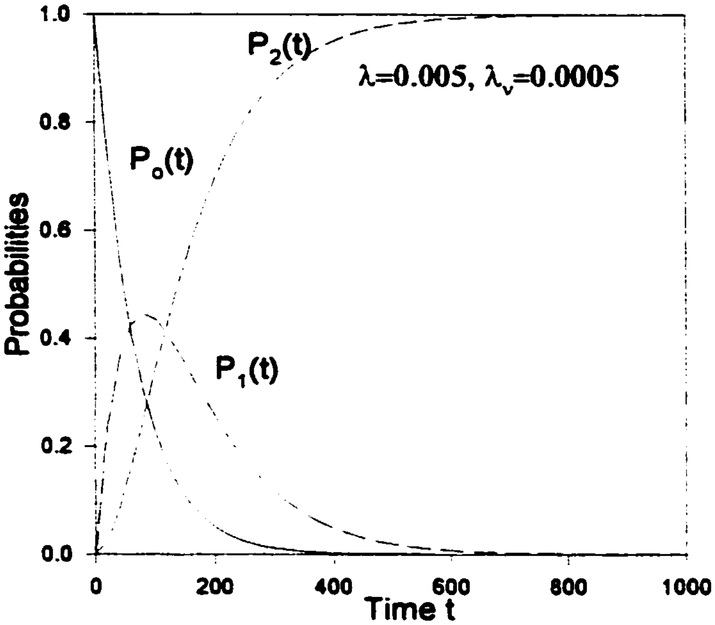


Fig.2.1.2 State probablities of a TMR subsystem

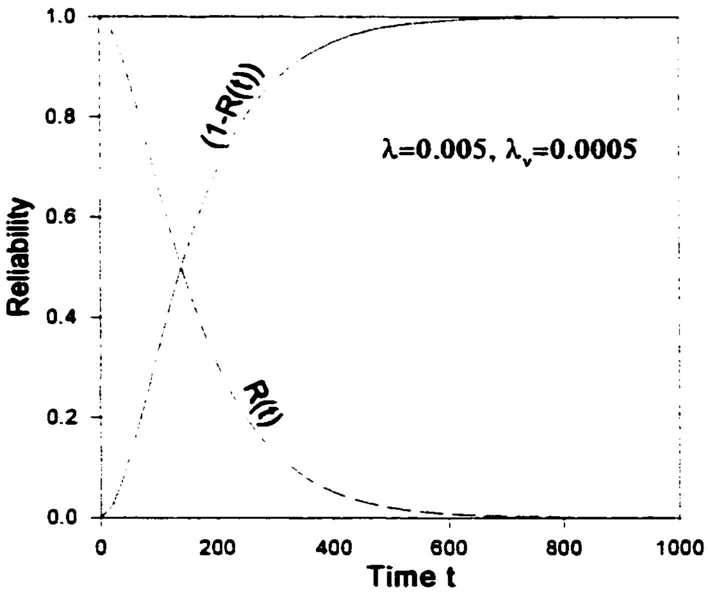


Fig.2.1.3 Relaibility plots of a TMR subsystem.

2.2 Model II

In this case the system is composed of a TMR subsystem with one voter. More specifically, for N=1, in Fig.2.1, this section presents the reliability analysis of a TMR-Voter system by using the Markov state technique. The system state space diagram is shown in Fig.2.2.1. At time t=0 all the three units and a voter start working normally. The numeral in each box denotes the state. At least two units and one voter must function normally to ensure the system success.

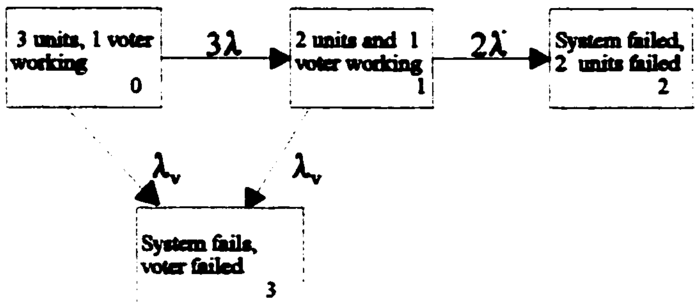


Fig.2.2.1. State space diagram of a TMR subsystem with one Voter.

The following equations are associated with Figure. 2.2.1.

$$\frac{dP_o(t)}{dt} + (3\lambda + \lambda_v)P_o(t) = 0 \tag{2.12}$$

$$\frac{dP_1(t)}{dt} + (2\lambda + \lambda_v)P_1(t) = P_o(t)3\lambda \tag{2.13}$$

$$\frac{dP_2(t)}{dt} = P_1(t)2\lambda \tag{2.14}$$

$$\frac{dP_3(t)}{dt} = \lambda_v(P_o(t) + P_1(t)) \tag{2.15}$$

At $t=0$, $P_o(0)=1$, and $P_1(0)=P_2(0)=P_3(0)=0$.

By the Crammer's rule and Equations (2.12)-(2.15), the probabilities in the s domain are:

$$P_o(s) = \frac{(s+k_4)}{(s+C_3)(s+C_4)} \qquad (2.16)$$

$$P_1(s) = \frac{3\lambda}{(s+C_3)(s+C_4)} \qquad (2.17)$$

$$P_2(s) = \frac{6\lambda^2}{(s+C_3)(s+C_4)} \qquad (2.18)$$

$$P_3(s) = \frac{s\lambda_v + B}{s(s+C_3)(s+C_4)} \qquad (2.19)$$

Equations (2.16)-(2.19) yield the following state probability results:

$$P_o(t) = \frac{(C_3+k_4)\exp(C_3 t)}{(C_3-C_4)} + \frac{(C_4+k_4)\exp(C_4 t)}{(C_4-C_3)} \qquad (2.20)$$

$$P_1(t) = \frac{3\lambda\exp(C_3 t)}{(C_3-C_4)} + \frac{3\lambda\exp(C_4 t)}{(C_4-C_3)} \qquad (2.21)$$

$$P_2(t) = \frac{6\lambda^2}{C_3 C_4} + \frac{6\lambda^2\exp(C_3 t)}{C_3(C_3-C_4)} + \frac{6\lambda^2\exp(C_4 t)}{C_4(C_4-C_3)} \qquad (2.22)$$

$$P_3(t) = \frac{B}{C_3 C_4} + \frac{(C_3\lambda_v + B)\exp(C_3 t)}{C_3(C_3-C_4)} + \frac{(C_4\lambda_v + B)\exp(C_4 t)}{C_4(C_4-C_3)} \qquad (2.23)$$

where

$k_3=3\lambda+\lambda_v$, $k_4=2\lambda+\lambda_v$, and $B=3\lambda\lambda_v+k_4\lambda_v$

The roots of the Equation $(s^2+s(k_3+k_4)+k_3k_4)$ are $C_3=-(3\lambda+\lambda_v)$, and $C_4=-(2\lambda+\lambda_v)$.

The reliability of the system is

$$R(t) = P_o(t) + P_1(t) \qquad (2.24)$$

The system mean time to failure is given by

$$\mathrm{MTTF} = \int_0^{\infty} R(t)\, dt$$

$$= \frac{k_4 + 3\lambda}{k_3 k_4} \qquad (2.25)$$

The plots of state probabilities, system reliability and mean time to failure are shown in Figures. 2.2.2, 2.2.3 and 2.2.4 for the specified model parameters, respectively.

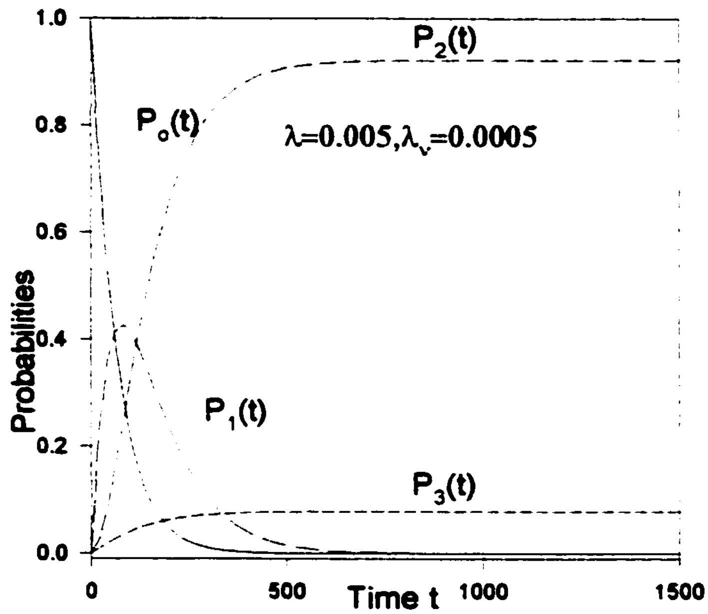


Fig.2.2.2 State probabilities of a TMR subsystem with one Voter.

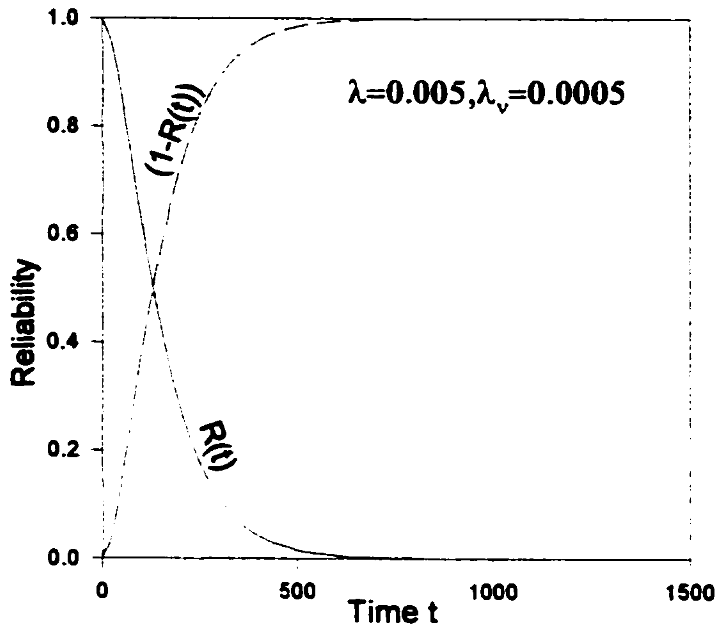


Fig.2.2.3 Reliability plots of a TMR subsystem with one voter.

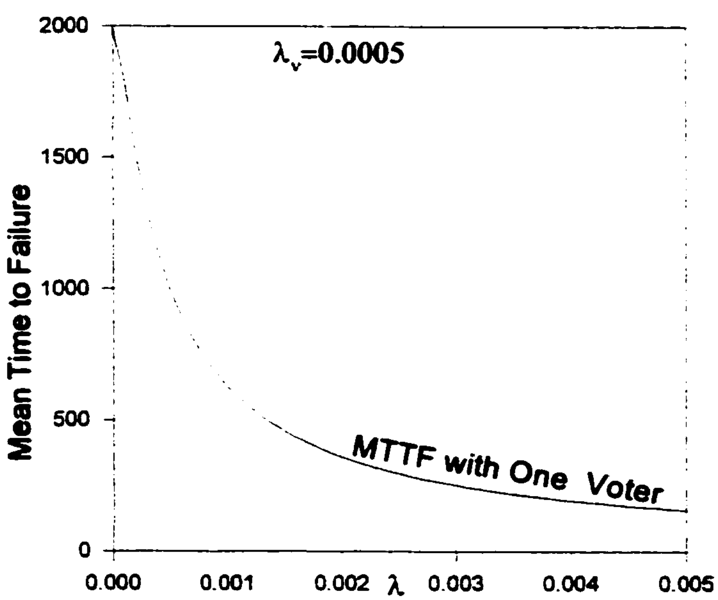


Fig.2.2.4 Mean time to failure of a TMR subsystem with one Voter.

2.3 Model III

This model represents a TMR subsystem with two parallel voters in series. Thus for N=2, in Fig.2.1, this section presents the reliability analysis of this model using the Markov state space technique. The system space diagram is shown in Fig.2.3.1 At time t=0 all the three units and two voters start working normally. The numeral in each box denotes the state. At least two units and one voter must function normally to ensure the system success.

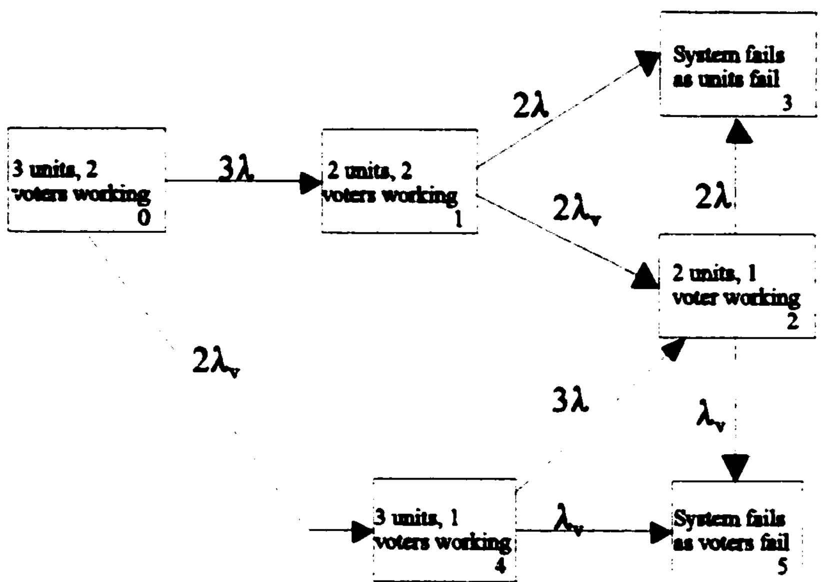


Fig.2.3.1.State space diagram of a TMR subsystem with two Voters in parallel.

The following equations are associated with Figure. 2.3.1.

$$\frac{dP_o(t)}{dt} + (3\lambda + 2\lambda_v)P_o(t) = 0 \tag{2.26}$$

$$\frac{dP_1(t)}{dt} + (2\lambda + 2\lambda_v)P_1(t) = P_o(t)3\lambda \tag{2.27}$$

$$\frac{dP_2(t)}{dt} + (2\lambda + \lambda_v)P_2(t) = P_1(t)2\lambda_v + P_4(t)3\lambda \tag{2.28}$$

$$\frac{dP_3(t)}{dt} = 2\lambda(P_1(t) + P_2(t)) \tag{2.29}$$

$$\frac{dP_4(t)}{dt} + (3\lambda + \lambda_v)P_4(t) = P_o(t)2\lambda_v \tag{2.30}$$

$$\frac{dP_5(t)}{dt} = \lambda_v(P_2(t) + P_4(t)) \tag{2.31}$$

At $t=0$, $P_o(0)=1$, and $P_1(0)=P_2(0)=P_3(0)=P_4(0)=P_5(0)=0$.

Using Equations (2.28)-(2.31) and the Crammer's rule, we get the following state probabilities in s domain:

$$P_o(s) = \frac{1}{s + 3\lambda + 2\lambda_v} \tag{2.32}$$

$$P_1(s) = \frac{3\lambda}{s + 2\lambda + 2\lambda_v} P_o(s) \tag{2.33}$$

$$P_2(s) = \frac{3\lambda}{s + 2\lambda + \lambda_v} P_4(s) + \frac{2\lambda_v}{s + 2\lambda + \lambda_v} P_1(s) \tag{2.34}$$

$$P_3(s) = \frac{2\lambda}{s} P_2(s) + \frac{2\lambda}{s} P_1(s) \tag{2.35}$$

$$P_4(s) = \frac{2\lambda_v}{s + 3\lambda + \lambda_v} P_o(s) \tag{2.36}$$

$$P_5(s) = \frac{\lambda_v}{s} P_2(s) + \frac{\lambda_v}{s} P_4(s) \tag{2.37}$$

Equations (2.32)-(2.37) yield the following state probabilities:

$$P_o(t) = \frac{(C_5 + k_6)(C_5 + k_7)(C_5 + k_8)\exp(C_5 t)}{(C_5 - C_6)(C_5 - C_7)(C_5 - C_8)} + \frac{(C_6 + k_6)(C_6 + k_7)(C_6 + k_8)\exp(C_6 t)}{(C_6 - C_5)(C_6 - C_7)(C_6 - C_8)} + \frac{(C_7 + k_6)(C_7 + k_7)(C_7 + k_8)\exp(C_7 t)}{(C_7 - C_5)(C_7 - C_6)(C_7 - C_8)} + \frac{(C_8 + k_6)(C_8 + k_7)(C_8 + k_8)\exp(C_8 t)}{(C_8 - C_5)(C_8 - C_6)(C_8 - C_7)} \tag{2.38}$$

$$P_1(t) = \frac{3\lambda(C_5^2 + C_5 B_1 + B_2)\exp(C_5 t)}{(C_5 - C_6)(C_5 - C_7)(C_5 - C_8)} + \frac{3\lambda(C_6^2 + C_6 B_1 + B_2)\exp(C_6 t)}{(C_6 - C_5)(C_6 - C_7)(C_6 - C_8)} + \frac{3\lambda(C_7^2 + C_7 B_1 + B_2)\exp(C_7 t)}{(C_7 - C_5)(C_7 - C_6)(C_7 - C_8)} + \frac{3\lambda(C_8^2 + C_8 B_1 + B_2)\exp(C_8 t)}{(C_8 - C_5)(C_8 - C_6)(C_8 - C_7)} \tag{2.39}$$

$$P_2(t) = \frac{(12\lambda\lambda_v C_5 + B_3)\exp(C_5 t)}{(C_5 - C_6)(C_5 - C_7)(C_5 - C_7)} + \frac{(12\lambda\lambda_v C_6 + B_3)\exp(C_6 t)}{(C_6 - C_5)(C_6 - C_7)(C_6 - C_8)} + \frac{(12\lambda\lambda_v C_7 + B_3)\exp(C_7 t)}{(C_7 - C_5)(C_7 - C_6)(C_7 - C_8)} + \frac{(12\lambda\lambda_v C_8 + B_3)\exp(C_8 t)}{(C_8 - C_5)(C_8 - C_6)(C_8 - C_7)} \tag{2.40}$$

$$P_3(t) = \frac{6\lambda^2 B_5}{C_5C_6C_7C_8} + \frac{6\lambda^2(C_5{}^2 + C_5B_4 + B_5)\exp(C_5t)}{C_5(C_5 - C_6)(C_5 - C_7)(C_5 - C_8)} +$$
$$\frac{6\lambda^2(C_6{}^2 + C_6B_4 + B_5)\exp(C_6t)}{C_6(C_6 - C_5)(C_6 - C_7)(C_6 - C_8)} +$$
$$\frac{6\lambda^2(C_7{}^2 + C_7B_4 + B_5)\exp(C_7t)}{C_7(C_7 - C_5)(C_7 - C_6)(C_7 - C_8)} +$$
$$\frac{6\lambda^2(C_8{}^2 + C_8B_4 + B_5)\exp(C_8t)}{C_8(C_8 - C_5)(C_8 - C_6)(C_8 - C_7)} \tag{2.41}$$

$$P_4(t) = \frac{2\lambda_v(C_5{}^2 + C_5B_6 + B_7)\exp(C_5t)}{(C_5 - C_6)(C_5 - C_7)(C_5 - C_8)} +$$
$$\frac{2\lambda_v(C_6{}^2 + C_6B_6 + B_7)\exp(C_6t)}{(C_6 - C_5)(C_6 - C_7)(C_6 - C_8)} +$$
$$\frac{2\lambda_v(C_7{}^2 + C_7B_6 + B_7)\exp(C_7t)}{(C_7 - C_5)(C_7 - C_6)(C_7 - C_8)} +$$
$$\frac{2\lambda_v(C_8{}^2 + C_8B_6 + B_7)\exp(C_8t)}{(C_8 - C_5)(C_8 - C_6)(C_8 - C_7)} \tag{2.42}$$

$$P_5(t) = \frac{2\lambda_v{}^2 B_9}{C_5C_6C_7C_8} + \frac{6\lambda_v{}^2(C_5{}^2 + C_5B_8 + B_9)\exp(C_5t)}{C_5(C_5 - C_6)(C_5 - C_7)(C_5 - C_8)} +$$
$$\frac{2\lambda_v{}^2(C_6{}^2 + C_6B_8 + B_9)\exp(C_6t)}{C_6(C_6 - C_5)(C_6 - C_7)(C_6 - C_8)} +$$
$$\frac{2\lambda_v{}^2(C_7{}^2 + C_7B_8 + B_9)\exp(C_7t)}{C_7(C_7 - C_5)(C_7 - C_6)(C_7 - C_8)} +$$
$$\frac{2\lambda_v{}^2(C_8{}^2 + C_8B_8 + B_9)\exp(C_8t)}{C_8(C_8 - C_5)(C_8 - C_6)(C_8 - C_7)} \tag{2.43}$$

The roots of the Equation $(s^4+s^3A+s^2A_1+sA_2+A_3)$ are $C_5= -(3\lambda+2\lambda_v)$, $C_6=-(2\lambda+2\lambda_v)$, $C_7=-(2\lambda+\lambda_v)$.

and $C_8=-(3\lambda+\lambda_v)$. The constants A to A_3, and B_1 to B_9 are defined in the appendices A and B.

The reliability of the system is

$$R(t) = P_o(t) + P_1(t) + P_2(t) + P_4(t) \tag{2.44}$$

The system mean time to failure is given by

$$\begin{aligned} MTTF &= \int_0^\infty R(t)\,dt \\ &= \frac{k_6k_7k_8 + 3\lambda k_7k_8 + 6\lambda\lambda_v(k_8 + k_6) + 2\lambda_v k_6k_7}{k_5k_6k_7k_8} \end{aligned} \tag{2.45}$$

The symbol k_5, k_6, k_7, and k_8 are defined in the appendix C.

The plots of state probability, system reliability and mean time to failure are shown in Figures.2.3.2, 2.3.3 and 2.3.4 for the specified values of the model parameters, respectively.

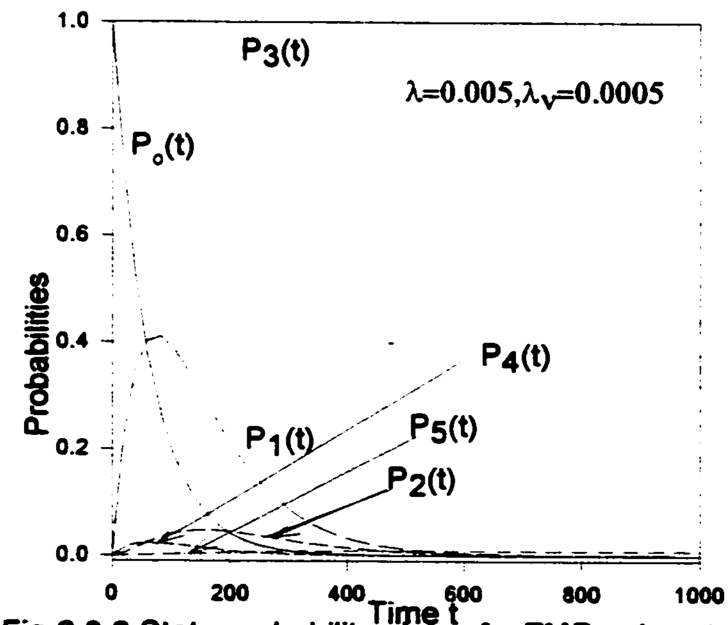
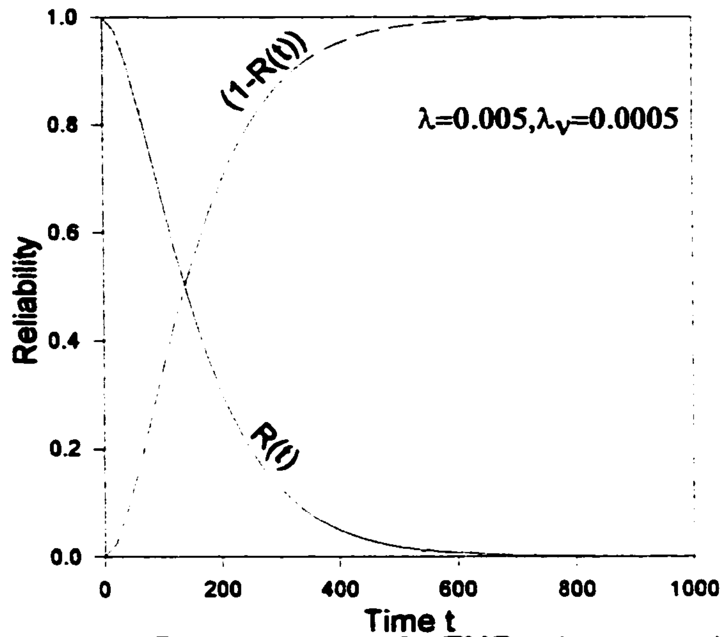


Fig.2.3.2 State probability plots of a TMR subsystem with two redundant Voters.



2.3.3 Reliability plots of a TMR subsystem with two redundant Voters.

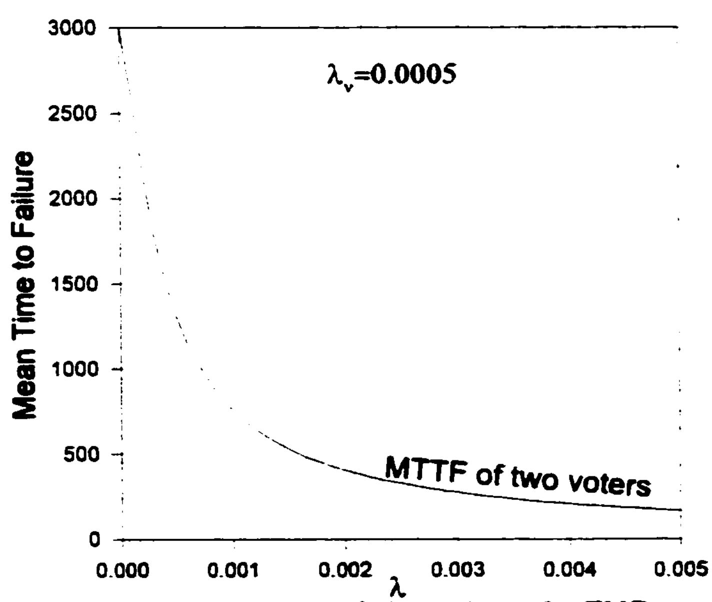


Fig.2.3.4 Mean time to failure plots of a TMR subsystem with two redundant Voters.

2.4 Model IV

This model represents a TMR subsystem with three redundant voters in series. Thus for N=3, in Fig.2.1, this section presents the reliability analysis of a TMR system with three voters. by using the Markov state space technique. The system transition diagram is shown in Fig.2.4.1. The numeral in each box denotes the state. At time t=0 all the three units and three voters start working normally. At least two units and one voter must function normally to ensure the system success.

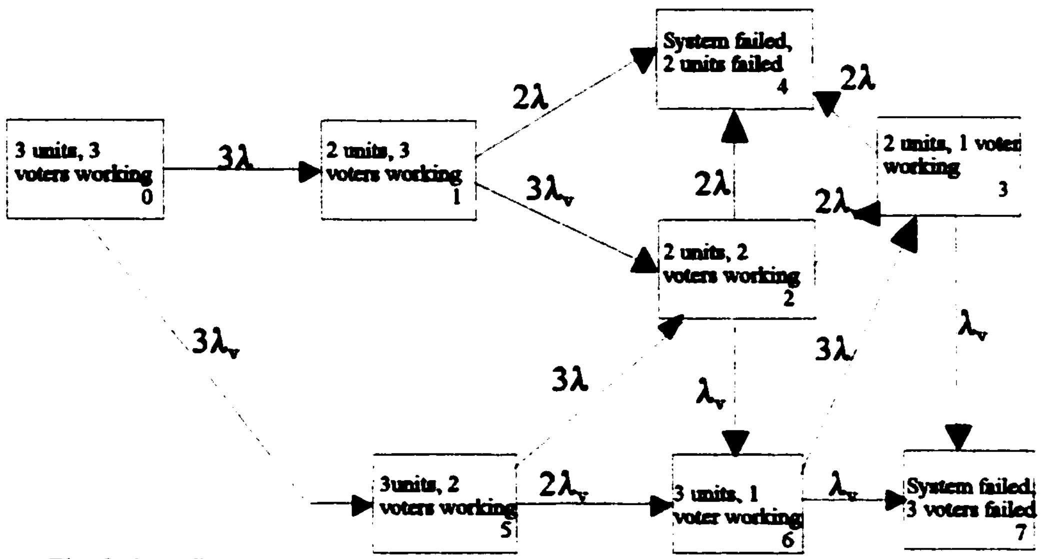


Fig.2.4.1. State space diagram of a TMR subsystem with three redundant Voters.

The following equations are associated with Figure. 2.4.1.

$$\frac{dP_0(t)}{dt} + (3\lambda + 3\lambda_v)P_0(t) = 0 \tag{2.46}$$

$$\frac{dP_1(t)}{dt} + (2\lambda + 3\lambda_v)P_1(t) = P_0(t)3\lambda \tag{2.47}$$

$$\frac{dP_2(t)}{dt} + (2\lambda + 2\lambda_v)P_1(t) = 3\lambda P_4(t) + 3\lambda_v P_1(t) \qquad (2.48)$$

$$\frac{dP_3(t)}{dt} + (2\lambda + \lambda_v)P_3(t) = 2\lambda_v P_2(t) + 3\lambda P_5(t) \qquad (2.49)$$

$$\frac{dP_4(t)}{dt} = 2\lambda(P_1(t) + P_2(t) + P_3(t)) \qquad (2.50)$$

$$\frac{dP_5(t)}{dt} + (3\lambda + 2\lambda_v)P_5(t) = 3\lambda_v P_o(t) \qquad (2.51)$$

$$\frac{dP_6(t)}{dt} + (3\lambda + \lambda_v)P_6(t) = 2\lambda_v P_5(t) \qquad (2.52)$$

$$\frac{dP_7(t)}{dt} = \lambda_v(P_6(t) + P_3(t)) \qquad (2.53)$$

At t=0, $P_o(0)=1$, and $P_1(0)=P_2(0)=P_3(0)=P_4(0)=P_5(0)=P_6(0)=P_7(0)=0$.

By solving Equations (2.46)-(2.53), we get the following state probabilities in s domain:

$$P_o(s) = \frac{1}{(s+k_1)} \qquad (2.54)$$

$$P_1(s) = \frac{3\lambda}{(s+k_1)(s+k_2)} \qquad (2.55)$$

$$P_2(s) = \frac{18\lambda^2(s+k_4)}{(s+k_1)(s+k_2)[(s(s+k_3)(s+k_4) + 6\lambda^2(s+k_4) + 12\lambda^2\lambda_v)]} \qquad (2.56)$$

$$P_3(s) = \frac{2\lambda_v P_2(s)}{(s+k_4)} \qquad (2.57)$$

$$P_4(s) = \frac{2\lambda}{s}[P_1(s) + P_2(s) + P_3(s)] \qquad (2.58)$$

$$P_5(s) = \frac{3\lambda_v}{(s+k_5)(s+k_1)} \tag{2.59}$$

$$P_6(s) = \frac{6\lambda_v^2}{(s+k_5)(s+k_6)(s+k_1)} \tag{2.60}$$

$$P_7(s) = \frac{\lambda_v}{s}[P_3(s) + P_6(s)] \tag{2.61}$$

Using Equations (2.54)-(2.61), we get the following state probabilities:

$$\begin{aligned}
P_o(t) = {} & \frac{(C_9 + k_{13})(C_9 + k_{10})(C_9{}^3 + C_9{}^2B_{10} + C_9B_{11} + B_{12})\exp(C_9 t)}{(C_9 - C_{10})(C_9 - C_{11})(C_9 - C_{12})(C_9 - C_{13})(C_9 - C_{14})} + \\
& \frac{(C_{10} + k_{13})(C_{10} + k_{10})(C_{10}{}^3 + C_{10}{}^2B_{10} + C_{10}B_{11} + B_{12})\exp(C_{10} t)}{(C_{10} - C_9)(C_{10} - C_{11})(C_{10} - C_{12})(C_{10} - C_{13})(C_{10} - C_{14})} + \\
& \frac{(C_{11} + k_{13})(C_{11} + k_{10})(C_{11}{}^3 + C_{11}{}^2B_{10} + C_{11}B_{11} + B_{12})\exp(C_{11} t)}{(C_{11} - C_9)(C_{11} - C_{10})(C_{11} - C_{12})(C_{11} - C_{13})(C_{11} - C_{14})} + \\
& \frac{(C_{12} + k_{13})(C_{12} + k_{10})(C_{12}{}^3 + C_{12}{}^2B_{10} + C_{12}B_{11} + B_{12})\exp(C_{12} t)}{(C_{12} - C_9)(C_{12} - C_{10})(C_{12} - C_{11})(C_{12} - C_{13})(C_{12} - C_{14})} + \\
& \frac{(C_{13} + k_{13})(C_{13} + k_{10})(C_{13}{}^3 + C_{13}{}^2B_{10} + C_{13}B_{11} + B_{12})\exp(C_{13} t)}{(C_{13} - C_9)(C_{13} - C_{10})(C_{13} - C_{11})(C_{13} - C_{12})(C_{13} - C_{14})} + \\
& \frac{(C_{14} + k_{13})(C_{13} + k_{10})(C_{14}{}^3 + C_{14}{}^2B_{10} + C_{14}B_{11} + B_{12})\exp(C_{14} t)}{(C_{14} - C_9)(C_{14} - C_{10})(C_{14} - C_{11})(C_{14} - C_{12})(C_{14} - C_{13})}
\end{aligned} \tag{2.62}$$

$$P_1(t) = \frac{3\lambda(C_9^4 + C_9^3B_{13} + C_9^2B_{14} + C_9B_{15} + B_{16})\exp(C_9t)}{(C_9 - C_{10})(C_9 - C_{11})(C_9 - C_{12})(C_9 - C_{13})(C_9 - C_{14})} + \frac{3\lambda(C_{10}^4 + C_{10}^3B_{13} + C_{10}^2B_{14} + C_{10}B_{15} + B_{16})\exp(C_{10}t)}{(C_{10} - C_9)(C_{10} - C_{11})(C_{10} - C_{12})(C_{10} - C_{13})(C_{10} - C_{14})} + \frac{3\lambda(C_{11}^4 + C_{11}^3B_{13} + C_{11}^2B_{14} + C_{11}B_{15} + B_{16})\exp(C_{11}t)}{(C_{11} - C_9)(C_{11} - C_{10})(C_{11} - C_{12})(C_{11} - C_{13})(C_{11} - C_{14})} + \frac{3\lambda(C_{12}^4 + C_{12}^3B_{13} + C_{12}^2B_{14} + C_{12}B_{15} + B_{16})\exp(C_{12}t)}{(C_{12} - C_9)(C_{12} - C_{10})(C_{12} - C_{11})(C_{12} - C_{13})(C_{12} - C_{14})} + \frac{3\lambda(C_{13}^4 + C_{13}^3B_{13} + C_{13}^2B_{14} + C_{13}B_{15} + B_{16})\exp(C_{13}t)}{(C_{13} - C_9)(C_{13} - C_{10})(C_{13} - C_{11})(C_{13} - C_{12})(C_{13} - C_{14})} + \frac{3\lambda(C_{14}^4 + C_{14}^3B_{13} + C_{14}^2B_{14} + C_{14}B_{15} + B_{16})\exp(C_{14}t)}{(C_{14} - C_9)(C_{14} - C_{10})(C_{14} - C_{11})(C_{14} - C_{12})(C_{14} - C_{13})} \tag{2.63}$$

$$P_2(t) = \frac{(18\lambda\lambda_v C_9^3 + C_9^2B_{17} + C_9B_{18} + B_{19})\exp(C_9t)}{(C_9 - C_{10})(C_9 - C_{11})(C_9 - C_{12})(C_9 - C_{13})(C_9 - C_{14})} + \frac{(18\lambda\lambda_v C_{10}^3 + C_{10}^2B_{17} + C_{10}B_{18} + B_{19})\exp(C_{10}t)}{(C_{10} - C_9)(C_{10} - C_{11})(C_{10} - C_{12})(C_{10} - C_{13})(C_{10} - C_{14})} + \frac{(18\lambda\lambda_v C_{11}^3 + C_{11}^2B_{17} + C_{11}B_{18} + B_{19})\exp(C_{11}t)}{(C_{11} - C_9)(C_{11} - C_{10})(C_{11} - C_{12})(C_{11} - C_{13})(C_{11} - C_{14})} + \frac{(18\lambda\lambda_v C_{12}^3 + C_{12}^2B_{17} + C_{12}B_{18} + B_{19})\exp(C_{12}t)}{(C_{12} - C_9)(C_{12} - C_{10})(C_{12} - C_{11})(C_{12} - C_{13})(C_{12} - C_{14})} + \frac{(18\lambda\lambda_v C_{13}^3 + C_{13}^2B_{17} + C_{13}B_{18} + B_{19})\exp(C_{13}t)}{(C_{13} - C_9)(C_{13} - C_{10})(C_{13} - C_{11})(C_{13} - C_{12})(C_{13} - C_{14})} + \frac{(18\lambda\lambda_v C_{14}^3 + C_{14}^2B_{17} + C_{14}B_{18} + B_{19})\exp(C_{14}t)}{(C_{14} - C_9)(C_{14} - C_{10})(C_{14} - C_{11})(C_{14} - C_{12})(C_{14} - C_{13})} \tag{2.64}$$

$$P_3(t) = \frac{(54\lambda\lambda_v C_9^2 + C_9 B_{20} + B_{21})\exp(C_9 t)}{(C_9 - C_{10})(C_9 - C_{11})(C_9 - C_{12})(C_9 - C_{13})(C_9 - C_{14})} +$$
$$\frac{(54\lambda\lambda_v C_{10}^2 + C_{10} B_{20} + B_{21})\exp(C_{10} t)}{(C_{10} - C_9)(C_{10} - C_{11})(C_{10} - C_{12})(C_{10} - C_{13})(C_{10} - C_{14})} +$$
$$\frac{(54\lambda\lambda_v C_{11}^2 + C_{11} B_{20} + B_{21})\exp(C_{11} t)}{(C_{11} - C_9)(C_{11} - C_{10})(C_{11} - C_{12})(C_{11} - C_{13})(C_{11} - C_{14})} +$$
$$\frac{(54\lambda\lambda_v C_{12}^2 + C_{12} B_{20} + B_{21})\exp(C_{12} t)}{(C_{12} - C_9)(C_{12} - C_{10})(C_{12} - C_{11})(C_{12} - C_{13})(C_{12} - C_{14})} +$$
$$\frac{(54\lambda\lambda_v C_{13}^2 + C_{13} B_{20} + B_{21})\exp(C_{13} t)}{(C_{13} - C_9)(C_{13} - C_{10})(C_{13} - C_{11})(C_{13} - C_{12})(C_{13} - C_{14})} +$$
$$\frac{(54\lambda\lambda_v C_{14}^2 + C_{14} B_{20} + B_{21})\exp(C_{14} t)}{(C_{14} - C_9)(C_{14} - C_{10})(C_{14} - C_{11})(C_{14} - C_{12})(C_{14} - C_{13})} \qquad (2.65)$$

$$P_5(t) = \frac{3\lambda_v (C_9^3 + C_9^2 B_{22} + C_9 B_{23} + B_{24})(C_9 + k_{14})\exp(C_9 t)}{(C_9 - C_{10})(C_9 - C_{11})(C_9 - C_{12})(C_9 - C_{13})(C_9 - C_{14})} +$$
$$\frac{3\lambda_v (C_{10}^3 + C_{10}^2 B_{22} + C_{10} B_{23} + B_{24})(C_{10} + k_{14})\exp(C_{10} t)}{(C_{10} - C_9)(C_{10} - C_{11})(C_{10} - C_{12})(C_{10} - C_{13})(C_{10} - C_{14})} +$$
$$\frac{3\lambda_v (C_{11}^3 + C_{11}^2 B_{22} + C_{11} B_{23} + B_{24})(C_{11} + k_{14})\exp(C_{11} t)}{(C_{11} - C_9)(C_{11} - C_{10})(C_{11} - C_{12})(C_{11} - C_{13})(C_{11} - C_{14})} +$$
$$\frac{3\lambda_v (C_{12}^3 + C_{12}^2 B_{22} + C_{12} B_{23} + B_{24})(C_{12} + k_{14})\exp(C_{12} t)}{(C_{12} - C_9)(C_{12} - C_{10})(C_{12} - C_{11})(C_{12} - C_{13})(C_{12} - C_{14})} +$$
$$\frac{3\lambda_v (C_{13}^3 + C_{13}^2 B_{22} + C_{13} B_{23} + B_{24})(C_{13} + k_{14})\exp(C_{13} t)}{(C_{13} - C_9)(C_{13} - C_{10})(C_{13} - C_{11})(C_{13} - C_{12})(C_{13} - C_{14})} +$$
$$\frac{3\lambda_v (C_{14}^3 + C_{14}^2 B_{22} + C_{14} B_{23} + B_{24})(C_{14} + k_{14})\exp(C_{14} t)}{(C_{14} - C_9)(C_{14} - C_{10})(C_{14} - C_{11})(C_{14} - C_{12})(C_{14} - C_{13})} \qquad (2.66)$$

$$
\begin{aligned}
P_6(t) = {} & \frac{6\lambda_v^2(C_9^3 + C_9^2B_{25} + C_9B_{26} + B_{27})\exp(C_9t)}{(C_9 - C_{10})(C_9 - C_{11})(C_9 - C_{12})(C_9 - C_{13})(C_9 - C_{14})} + \\
& \frac{6\lambda_v^2(C_{10}^3 + C_{10}^2B_{25} + C_{10}B_{26} + B_{27})\exp(C_{10}t)}{(C_{10} - C_9)(C_{10} - C_{11})(C_{10} - C_{12})(C_{10} - C_{13})(C_{10} - C_{14})} + \\
& \frac{6\lambda_v^2(C_{11}^3 + C_{11}^2B_{25} + C_{11}B_{26} + B_{27})\exp(C_{11}t)}{(C_{11} - C_9)(C_{11} - C_{10})(C_{11} - C_{12})(C_{11} - C_{13})(C_{11} - C_{14})} + \\
& \frac{6\lambda_v^2(C_{12}^3 + C_{12}^2B_{25} + C_{12}B_{26} + B_{27})\exp(C_{12}t)}{(C_{12} - C_9)(C_{12} - C_{10})(C_{12} - C_{11})(C_{12} - C_{13})(C_{12} - C_{14})} + \\
& \frac{6\lambda_v^2(C_{13}^3 + C_{13}^2B_{25} + C_{13}B_{26} + B_{27})\exp(C_{13}t)}{(C_{13} - C_9)(C_{13} - C_{10})(C_{13} - C_{11})(C_{13} - C_{12})(C_{13} - C_{14})} + \\
& \frac{6\lambda_v^2(C_{14}^3 + C_{14}^2B_{25} + C_{14}B_{26} + B_{27})\exp(C_{14}t)}{(C_{14} - C_9)(C_{14} - C_{10})(C_{14} - C_{11})(C_{14} - C_{12})(C_{14} - C_{13})}
\end{aligned}
\tag{2.67}
$$

$$
\begin{aligned}
P_7(t) = {} & \frac{6\lambda_v^3B_{30}}{C_9C_{10}C_{11}C_{12}C_{13}C_{14}} + \frac{6\lambda_v^3(C_9^3 + C_9^2B_{28} + C_9B_{29} + B_{30})\exp(C_9t)}{C_9(C_9 - C_{10})(C_9 - C_{11})(C_9 - C_{12})(C_9 - C_{13})(C_9 - C_{14})} + \\
& \frac{6\lambda_v^3(C_{10}^3 + C_{10}^2B_{28} + C_{10}B_{29} + B_{30})\exp(C_{10}t)}{C_{10}(C_{10} - C_9)(C_{10} - C_{11})(C_{10} - C_{12})(C_{10} - C_{13})(C_{10} - C_{14})} + \\
& \frac{6\lambda_v^3(C_{11}^3 + C_{11}^2B_{28} + C_{11}B_{29} + B_{30})\exp(C_{11}t)}{C_{11}(C_{11} - C_9)(C_{11} - C_{10})(C_{11} - C_{12})(C_{11} - C_{13})(C_{11} - C_{14})} + \\
& \frac{6\lambda_v^3(C_{12}^3 + C_{12}^2B_{28} + C_{12}B_{29} + B_{30})\exp(C_{12}t)}{C_{12}(C_{12} - C_9)(C_{12} - C_{10})(C_{12} - C_{11})(C_{12} - C_{13})(C_{12} - C_{14})} + \\
& \frac{6\lambda_v^3(C_{13}^3 + C_{13}^2B_{28} + C_{13}B_{29} + B_{30})\exp(C_{13}t)}{C_{13}(C_{13} - C_9)(C_{13} - C_{10})(C_{13} - C_{11})(C_{13} - C_{12})(C_{13} - C_{14})} + \\
& \frac{6\lambda_v^3(C_{14}^3 + C_{14}^2B_{28} + C_{14}B_{29} + B_{30})\exp(C_{14}t)}{C_{14}(C_{14} - C_9)(C_{14} - C_{10})(C_{14} - C_{11})(C_{14} - C_{12})(C_{14} - C_{13})} +
\end{aligned}
\tag{2.68}
$$

$$
\begin{aligned}
P_4(t) = {} & \frac{6\lambda^2 B_{34}}{C_9 C_{10} C_{11} C_{12} C_{13} C_{14}} + \frac{6\lambda^2 (C_9^4 + C_9^3 B_{31} + C_9^2 B_{32} + C_9 B_{33} + B_{34}) \exp(C_9 t)}{C_9 (C_9 - C_{10})(C_9 - C_{11})(C_9 - C_{12})(C_9 - C_{13})(C_9 - C_{14})} + \\
& \frac{6\lambda^2 (C_{10}^4 + C_{10}^3 B_{31} + C_{10}^2 B_{32} + C_{10} B_{33} + B_{34}) \exp(C_{10} t)}{C_{10} (C_{10} - C_9)(C_{10} - C_{11})(C_{10} - C_{12})(C_{10} - C_{13})(C_{10} - C_{14})} + \\
& \frac{6\lambda^2 (C_{11}^4 + C_{11}^3 B_{31} + C_{11}^2 B_{32} + C_{11} B_{33} + B_{34}) \exp(C_{11} t)}{C_{11} (C_{11} - C_9)(C_{11} - C_{10})(C_{11} - C_{12})(C_{11} - C_{13})(C_{11} - C_{14})} + \\
& \frac{6\lambda^2 (C_{12}^4 + C_{12}^3 B_{31} + C_{12}^2 B_{32} + C_{12} B_{33} + B_{34}) \exp(C_{12} t)}{C_{12} (C_{12} - C_9)(C_{12} - C_{10})(C_{12} - C_{11})(C_{12} - C_{13})(C_{12} - C_{14})} + \\
& \frac{6\lambda^2 (C_{13}^4 + C_{13}^3 B_{31} + C_{13}^2 B_{32} + C_{13} B_{33} + B_{34}) \exp(C_{13} t)}{C_{13} (C_{13} - C_9)(C_{13} - C_{10})(C_{13} - C_{11})(C_{13} - C_{12})(C_{13} - C_{14})} + \\
& \frac{6\lambda^2 (C_{14}^4 + C_{14}^3 B_{31} + C_{14}^2 B_{32} + C_{14} B_{33} + B_{34}) \exp(C_{14} t)}{C_{14} (C_{14} - C_9)(C_{14} - C_{10})(C_{14} - C_{11})(C_{14} - C_{12})(C_{14} - C_{13})}
\end{aligned}
\tag{2.69}
$$

The roots of the Equation $(s^6+s^5A_4+s^4A_5+s^3A_6+s^2A_7+sA_8+A_9)$ are $C_9=-(3\lambda+3\lambda_v)$, $C_{10}=-(2\lambda+3\lambda_v)$, $C_{11}=-(2\lambda+2\lambda_v)$, $C_{12}=-(2\lambda+\lambda_v)$, $C_{13}=-(3\lambda+2\lambda_v)$, and $C_{14}=-(3\lambda+\lambda_v)$. The roots of the above equation are obtained by solving the above equation. The constants A_4 to A_9, and B_{10} to B_{34} are defined in the appendices A and appendix B, respectively.

The system reliability of the system is

$$R(t) = P_o(t) + P_1(t) + P_2(t) + P_3(t) + P_5(t) + P_6(t) \tag{2.70}$$

The system mean time to failure is given by

$$
\begin{aligned}
MTTF &= \int_0^\infty R(t)\,dt \\
&= \frac{\begin{aligned}(k_{10}k_{11}k_{12}k_{13}k_{14} + 3\lambda k_{11}k_{12}k_{13}k_{14} + 9\lambda\lambda_v k_{12}k_{14}(k_{13}+k_{10}) + {} \\ 18\lambda\lambda_v^2(k_{14}k_{13} + k_{11}k_{10} + k_{14}k_{10}) + 6\lambda_v^2 k_{12}k_{11}k_{10})\end{aligned}}{k_9 k_{10} k_{11} k_{12} k_{13} k_{14}}
\end{aligned}
\tag{2.71}
$$

The symbols k_9, k_{10}, k_{11}, k_{12}, k_{13}, and k_{14} are defined in the appendix C.

The plots of state probabilities, system reliability and mean time to failures are shown in Figures. 2.4.2, 2.4.3 and 2.4.4 for the specified values of the model parameters, respectively.

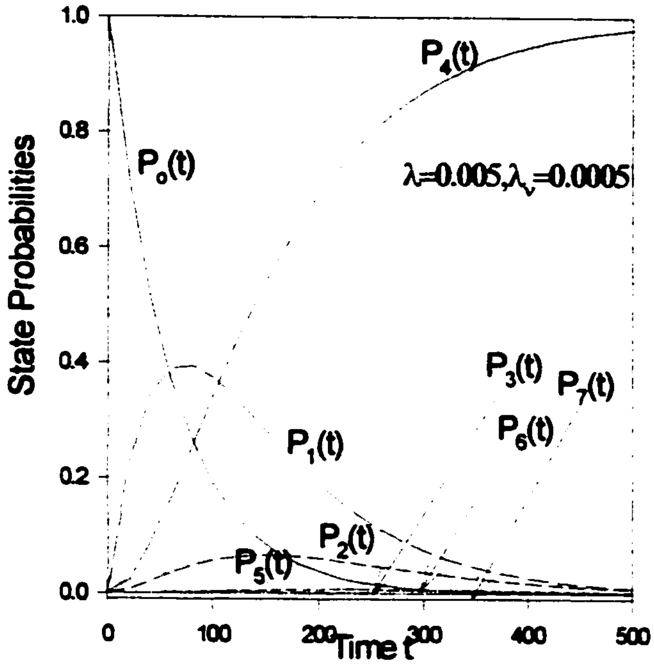


Fig.2.4.2 State probability plots of a TMR subsystem with three redundant Voters in series.

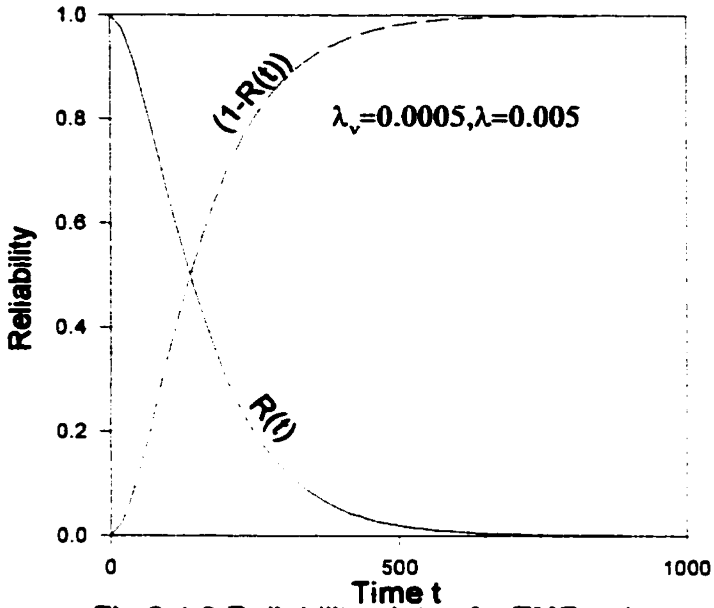


Fig.2.4.3 Reliability plots of a TMR subsystem with three redundant Voters in series.

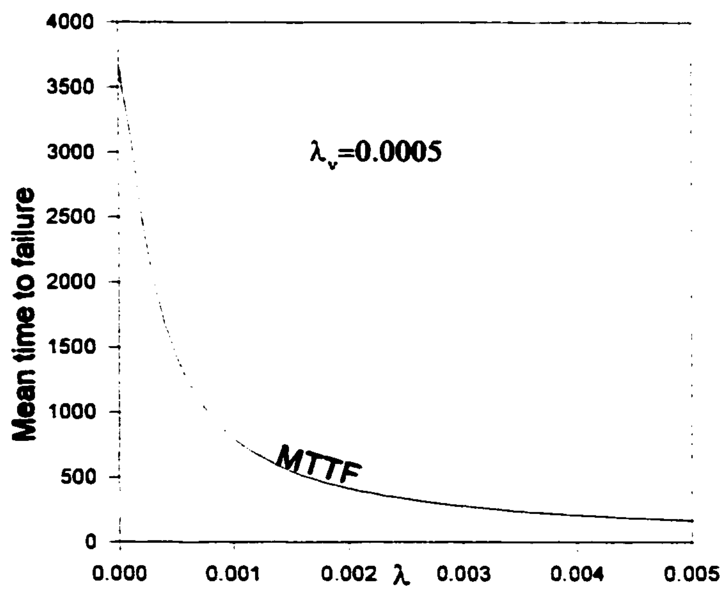


Fig.2.4.4 Mean time to failure of a TMR subsystem with three redundant Voters in series.

2.5 Generalized Formulas for Reliability and MTTF of TMR Subsystems with N redundant voters in Series

In the previous sections mean time to failure and reliability formulas for TMR systems with zero, one, two and three voters were developed. This section presents generalized mean time to failure and reliability formulas for a TMR subsystem with N redundant voters in series as shown in Fig.2.1.

In similar manner to the previous four cases, we get the following generalized expression for system reliability:

$$R_N(t) = [3e^{-2\lambda t} - 2e^{-3\lambda t}][1-(1-e^{-\lambda_v t})^N] \tag{2.72}$$

where RN(t) is the reliability of TMR subsystem with N redundant voters in series.

λ is the single unit failure rate.

λ_v is the voter failure rate.

N is the number of voters.

The generalized formula for the system mean time to failure is as follows:

$$MTTF = \int_0^{\infty} R_N(t)\,dt$$

$$= \int_0^{\infty} [3e^{-2\lambda t} - 2e^{-3\lambda_v t})][1-(1-e^{(-\lambda_v t)})^N]$$

$$MTTF_N = \sum_{j=1}^{N} \binom{N}{j} (-1)^{j+1} \left[\frac{3}{\lambda(2\lambda + \lambda_v j)} + \frac{2}{\lambda(3\lambda + \lambda_v j)} \right] \tag{2.73}$$

where $MTTF_N$ is the mean time to failure of TMR subsystem with N redundant voters in series.

The Figures 2.5.1 and 2.5.2 which show plots of Equations (2.72) and (2.73) for the specified values of the model parameters. Tables 2.1 and 2.2 present a numerical comparison for system reliability and system mean time to failure for one, two, and three redundant voters in series.

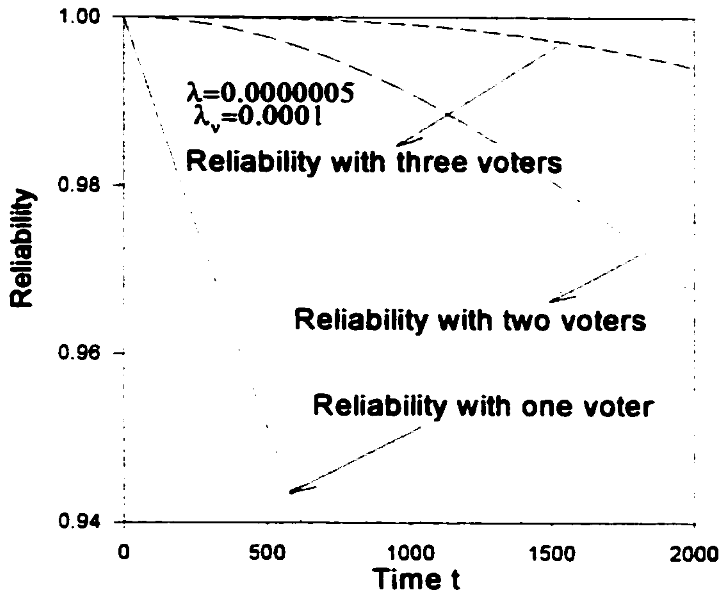


Fig.2.5.1 System reliability plots for voters from one to three.

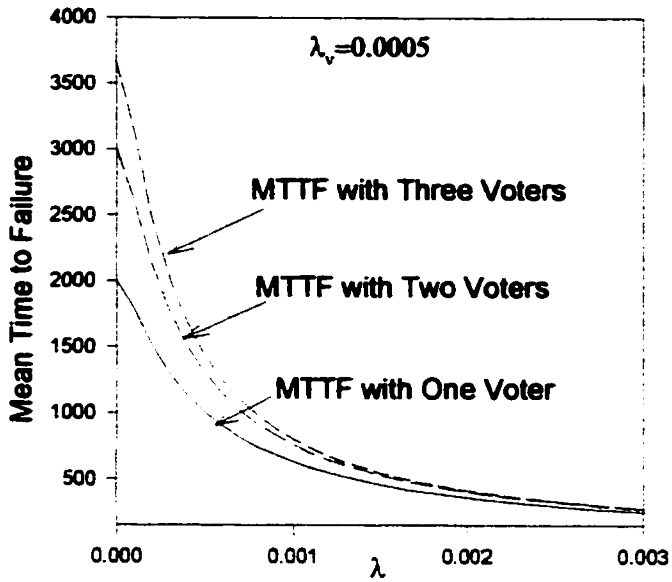


Fig.2.5.3 System mean time to failure for one to three.

Table 2.1 Numerical comparison of TMR subsystem reliabilities for one, two and three redundant voters in series.

Time	$R_1(t)$*	$R_2(t)$*	$R_3(t)$*
0	1.0000	1.0000	1.0000
200	0.9802	0.9996	0.9999
400	0.9608	0.9985	0.9999
600	0.9418	0.9966	0.9988
800	0.9231	0.9941	0.9995
1000	0.9048	0.9909	0.9991

* $R_1(t)$ is the system reliability of a TMR subsystem with one voter in series.

$R_2(t)$ is the system reliability of a TMR system with two voters in series.

$R_3(t)$ is the system reliability of a TMR system with three voters in series.

Table 2.2 Numerical comparison of a TMR subsystem mean time to failure for one, two, and three redundant voters in series

λ	$MTTF_1$*	$MTTF_2$*	$MTTF_3$*
0.000	2000	3000	3667
0.001	629	757	799
0.002	359	404	413
0.003	251	274	277
0.004	193	207	208
0.005	157	166	166

* $MTTF_1$ is the mean time to failure of a TMR subsystem with one voter in series.

$MTTF_2$ is the mean time to failure of a TMR subsystem with two voters in series.

$MTTF_3$ is the mean time to failure of a TMR subsystem with three voters in series.

Chapter 3

Repairable TMR systems with Redundant Voters in Series

In chapter 2 non-repairable TMR subsystems with voters were discussed. In this chapter the system presents a repairable TMR subsystem with redundant voters in series. This means when any one of the units of the TMR subsystem fails or the voters fail the failed items are repaired.

This chapter presents a total of four models representing repairable TMR systems when the number of voters are increased from zero to three. The reliability analysis of repairable TMR systems with redundant voters in series is presented. Similarly, expressions for mean time to failure and generalized expressions for probabilities in the s domain are presented.

Assumptions

The following assumptions are associated with all the repairable TMR models presented in this chapter:

- Failure and repair rates are exponential.
- Failures are statistically independent.
- All TMR subsystem units are identical.
- All Voter units are identical.
- Repaired system is as good as replaceable.
- At least two units and or voter must operate for system success.

Notations

The following symbols are associated with all the repairable TMR models:

λ is the single unit failure rate for a single unit.

λ_v is the voter unit failure rate.

μ_1 is the failed unit repair rate.

μ_2 is the failed voter repair rate.

t is time.

s is the Laplace transform variable.

R(t) is the reliability of the system.

MTTF is the mean time to failure.

n is the number of states.

$P_i(t)$ is the probability that the system is in state i at time t, for i=0,1,2,....,n+1.

N is the number of voters.

3.1 Model I

For N=0, in Fig.2.1, this section presents the reliability analysis of a repairable Triple Modular Redundant subsystem by using the Markov state technique. The system state space diagram is shown in Fig.3.1.1 At time t=0 all the three units start working normally. At least two units must function normally to ensure the system success. The failed unit is repaired. The numeral in each box in Fig.3.1.1 denotes the state.

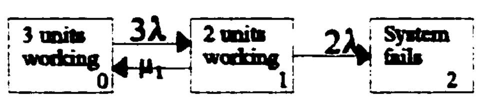


Fig.3.1.1. State space diagram of a repairable TMR subsystem.

The following equations are associated with Figure. 3.1.1.

$$\frac{dP_o(t)}{dt} + 3\lambda P_o(t) = \mu_1 P_1(t) \tag{3.1}$$

$$\frac{dP_1(t)}{dt} + (\mu_1 + 2\lambda)P_1(t) = 3\lambda P_o(t) \tag{3.2}$$

$$\frac{dP_2(t)}{dt} = 2\lambda P_1(t) \tag{3.3}$$

At t=0, $P_o(0)=1$, and $P_1(0)=P_2(0)=0$.

By using the Crammer's rule and solving the Equations (3.1)-(3.3) the probabilities in the s domain are:

$$P_o(s) = \frac{1+\mu_1 P_1(s)}{s+3\lambda} \tag{3.4}$$

$$P_1(s) = \frac{3\lambda P_o(s)}{s+2\lambda+\mu_1} \tag{3.5}$$

$$P_2(s) = \frac{2\lambda P_1(s)}{s} \tag{3.6}$$

Using Equations (3.4)-(3.6), we get the following state probabilities:

$$P_o(t) = \frac{(C_{15}+k_2)\exp(C_{15}t)}{(C_{15}-C_{16})} + \frac{(C_{16}+k_2)\exp(C_{16}t)}{(C_{16}-C_{15})} \tag{3.7}$$

$$P_1(t) = \frac{3\lambda\exp(C_{15}t)}{(C_{15}-C_{16})} + \frac{3\lambda\exp(C_{16}t)}{(C_{16}-C_{15})} \tag{3.8}$$

$$P_2(t) = \frac{6\lambda^2}{C_{15}C_{16}} + \frac{6\lambda^2\exp(C_{15}t)}{C_{15}(C_{15}-C_{16})} + \frac{6\lambda^2\exp(C_{16}t)}{C_{16}(C_{16}-C_{15})} \tag{3.9}$$

where

$k_{15}=3\lambda$, $k_{16}=2\lambda+\mu_1$

The roots of the equation $s^2+s(k_{16}+k_{15})+(k_{15}k_{16}-3\lambda\mu_1)$ are

$$C_{15}, C_{16} = \frac{-(k_{15}+k_{16}) \pm \sqrt{(k_{15}+k_{16})^2 - 4(k_{15}k_{16} - 3\lambda\mu_1)}}{2}$$

The reliability of the system is

$$R(t) = P_o(t) + P_1(t) \tag{3.10}$$

The system mean time to failure is

$$MTTF = \int_0^\infty R(t)\,dt$$

$$= \frac{k_{16} + 3\lambda}{k_{15}k_{16} - 3\lambda\mu_1} \tag{3.11}$$

The plots of state probabilities and system reliability are shown in Figures. 3.1.2 and 3.1.3 for the specified values of the model parameters.

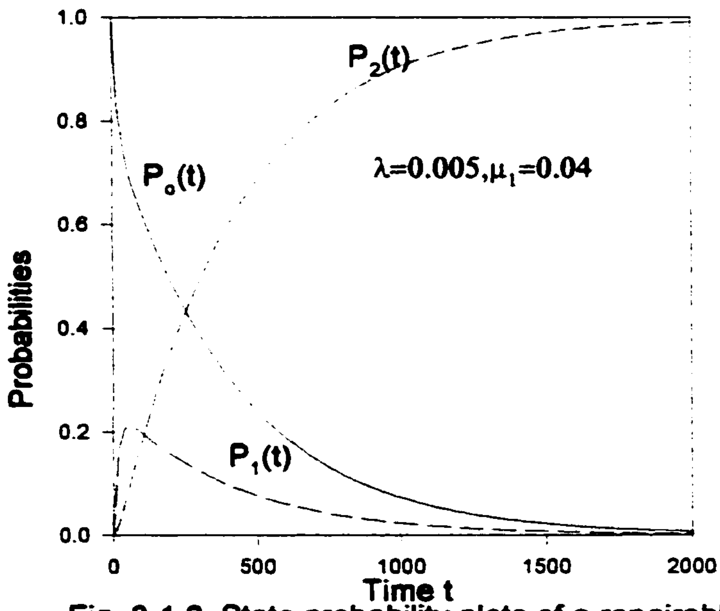


Fig. 3.1.2. State probability plots of a repairable TMR subsystem.

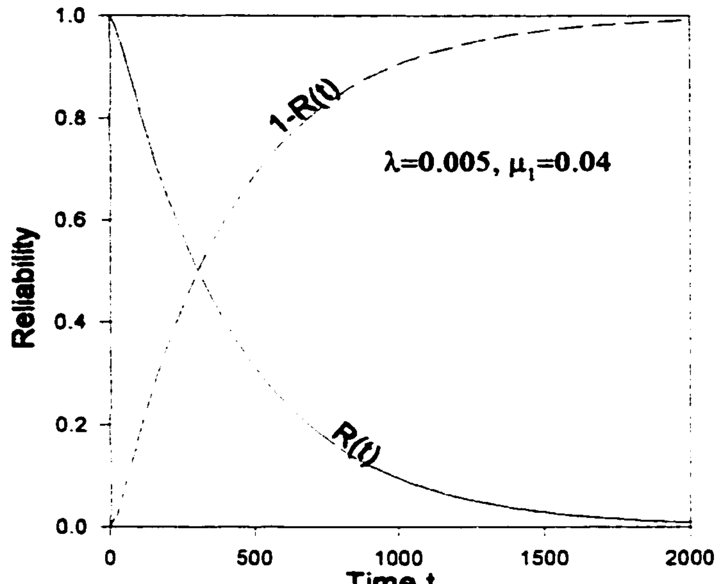


Fig.3.1.3. Reliability plots of a repairable TMR subsystem.

3.2 Model II

In this case the system is composed of a repairable TMR subsystem with one voter in series. More specifically, for N=1 in Fig.2.1, this section presents the reliability analysis of a repairable TMR-Voter system by using the Markov state technique. The system state space diagram is shown in Fig.3.2.1. At time t=0 all the three units and voter start working normally. The numeral in each box denotes the state number At least two units and a voter must function normally to ensure the system success. The failed unit is repaired.

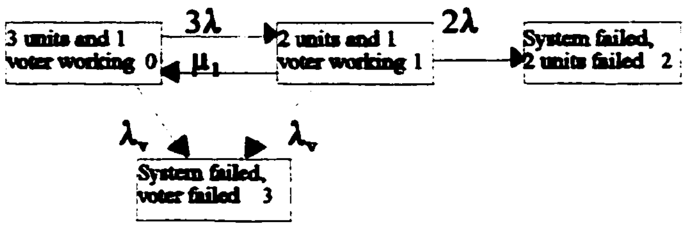


Fig.3.2.1. State space diagram of a repairable TMR subsystem with one Voter.

The following equations are associated with Figure. 3 .2.1.

$$\frac{dP_o(t)}{dt} + (3\lambda + \lambda_v)P_o(t) = P_1(t)\mu_1 \tag{3.12}$$

$$\frac{dP_1(t)}{dt} + (2\lambda + \lambda_v + \mu_1)P_1(t) = 3\lambda P_o(t) \tag{3.13}$$

$$\frac{dP_2(t)}{dt} = 2\lambda P_1(t) \tag{3.14}$$

$$\frac{dP_3(t)}{dt} = \lambda_v(P_o(t) + P_1(t)) \tag{3.15}$$

At $t=0$, $P_o(0)=1$, and $P_1(0)=P_2(0)=P_3(0)=0$.

By using the Crammer's rule and solving Equations (3.12)-(3.15), the probabilities in the s domain are:

$$P_o(s) = \frac{1 + \mu_1 P_1(s)}{s + 3\lambda + \lambda_v} \tag{3.16}$$

$$P_1(s) = \frac{3\lambda P_o(s)}{s + 2\lambda + \lambda_v + \mu_1} \tag{3.17}$$

$$P_2(s) = \frac{2\lambda P_1(s)}{s} \tag{3.18}$$

$$P_3(s) = \frac{\lambda_v(P_o(s) + P_1(s))}{s} \tag{3.19}$$

Using the Equations (3.16)-(3.19), we get state probabilities:

$$P_o(t) = \frac{(C_{17} + k_2)\exp(C_{17}t)}{(C_{17} - C_{18})} + \frac{(C_{18} + k_2)\exp(C_{18}t)}{(C_{18} - C_{17})} \tag{3.20}$$

$$P_1(t)=\frac{3\lambda\exp(C_{17}t)}{(C_{17}-C_{18})}+\frac{3\lambda\exp(C_{18}t)}{(C_{18}-C_{17})} \tag{3.21}$$

$$P_2(t)=\frac{6\lambda^2}{C_{17}C_{18}}+\frac{6\lambda^2\exp(C_{17}t)}{C_{17}(C_{17}-C_{18})}+\frac{6\lambda^2\exp(C_{18}t)}{C_{18}(C_{18}-C_{17})} \tag{3.22}$$

$$P_3(t)=\frac{B_{35}}{C_{17}C_{18}}+\frac{(\lambda_v C_{17}+B_{35})\exp(C_{17}t)}{C_{17}(C_{17}-C_{18})}+\frac{(\lambda_v C_{18}+B_{35})\exp(C_{18}t)}{C_{18}(C_{18}-C_{17})} \tag{3.23}$$

where

$k_{17}=3\lambda+\lambda_v$, and $k_{18}=2\lambda+\mu_1+\lambda_v$.

The roots of the equation $(s^2+s(k_{17}+k_{18})+(k_{17}k_{18}-3\lambda\mu_1)$ are

$$C_{17},C_{18}=\frac{-(k_{18}+k_{17})\pm\sqrt{(k_{18}+k_{17})^2-4(k_{17}k_{18}-3\lambda\mu_1)}}{2}$$

The constant B_{35} is defined in the appendix B.

The reliability of the system is

$$R(t)=P_o(t)+P_1(t) \tag{3.24}$$

The system mean time to failure is given by

$$\begin{aligned}MTTF &= \int_0^{\infty} R(t)\,dt \\ &= \frac{k_{18}+3\lambda}{k_{17}k_{18}-3\lambda\mu_1}\end{aligned} \tag{3.25}$$

The plots of state probabilities, system reliability and mean time to failure are shown in Figures. 3.2.2, 3.2.3 and 3.2.4 for the specified values of the model parameters, respectively.

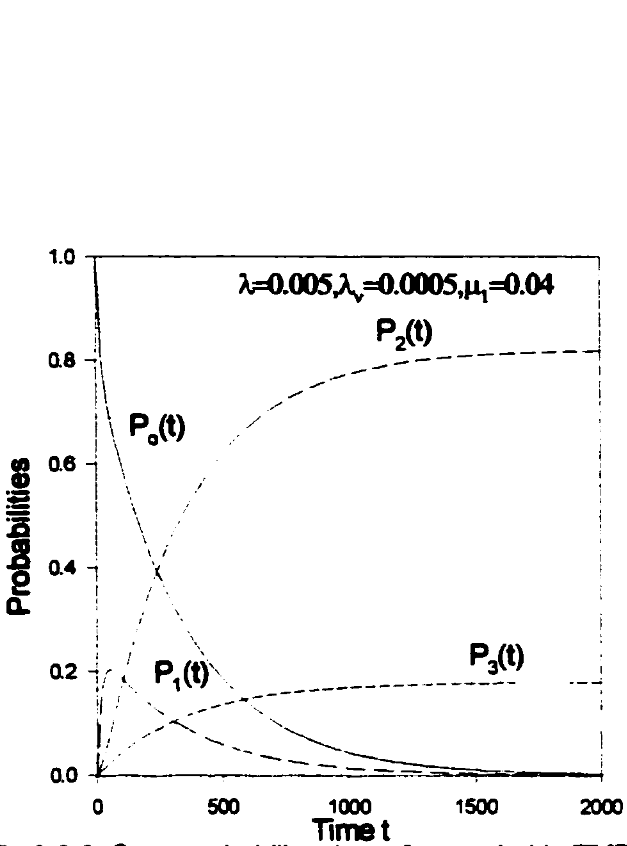


Fig.3.2.2. State probability plots of a repairable TMR subsystem with one Voter.

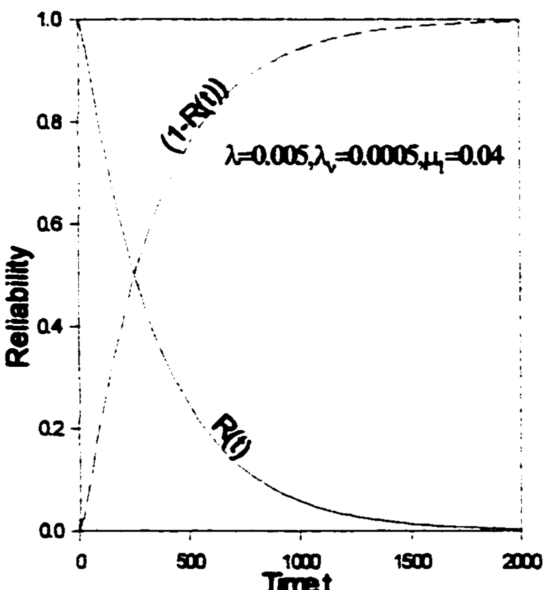


Fig.3.2.3 .Reliability plots of a repairable TMR subsystem with one Voter.

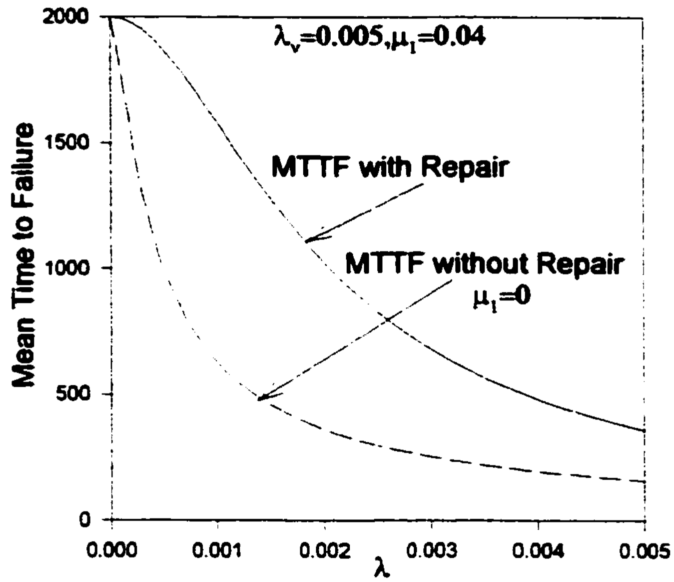


Fig.3.2.4. Mean time to failure plots of a repairable TMR subsystem with one Voter.

3.3 Model III

This model represents a repairable TMR subsystem with two voters in a parallel subsystem. Thus for N=2, in Fig.2.1, this section presents the reliability analysis of this model using the Markov state technique. The system state space diagram is shown in Fig.3.3.1. At time t=0 all the three units start working normally. The numeral in each box denotes the state number At least two units must function normally to ensure the system success. The failed unit and failed voter are repaired.

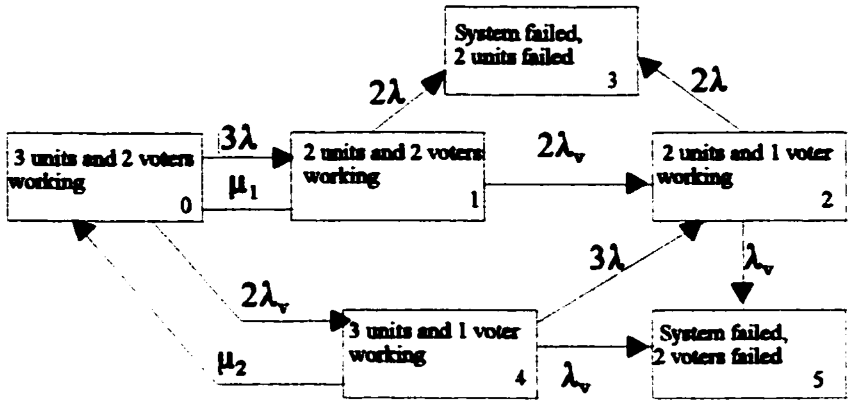


Fig.3.3.1 State space diagram of a repairable TMR subsystem with two Voters in parallel.

The following equations are associated with Figure. 3.3.1.

$$\frac{dP_o(t)}{dt} + (3\lambda + 2\lambda_v)P_o(t) = \mu_1 P_1(t) + \mu_2 P_4(t) \tag{3.26}$$

$$\frac{dP_1(t)}{dt} + (2\lambda_v + 2\lambda + \mu_1)P_1(t) = 3\lambda P_o(t) \tag{3.27}$$

$$\frac{dP_2(t)}{dt} + (2\lambda + \lambda_v)P_2(t) = 3\lambda P_4(t) + 2\lambda_v P_1(t) \tag{3.28}$$

$$\frac{dP_3(t)}{dt} = 2\lambda(P_1(t) + P_2(t)) \tag{3.29}$$

$$\frac{dP_4(t)}{dt} + (3\lambda + \mu_2 + \lambda_v)P_4(t) = 2\lambda_v P_o(t) \tag{3.30}$$

$$\frac{dP_5(t)}{dt} = \lambda_v(P_2(t) + P_4 t)) \tag{3.31}$$

At $t=0$, $P_o(0)=1$, and $P_1(0)=P_2(0)=P_3(0)=P_4(0)=P_5(0)=0$.

Using Equations (3.26)-(3.31) and the Crammer's rule, we get the following state probabilities in the s domain:

$$P_o(s) = \frac{1+\mu_1 P_1(s) + \mu_2 P_4(s)}{s + 3\lambda + 2\lambda_v} \tag{3.32}$$

$$P_1(s) = \frac{P_o(s)3\lambda}{s + 2\lambda + 2\lambda_v + \mu_1} \tag{3.33}$$

$$P_2(s) = \frac{3\lambda P_4(s) + 2\lambda_v P_1(s)}{s + 2\lambda + \lambda_v} \tag{3.34}$$

$$P_3(s) = \frac{2\lambda (P_1(s) + P_2(s))}{s} \tag{3.35}$$

$$P_4(s) = \frac{2\lambda_v P_o(s)}{s + 3\lambda + \mu_2} \tag{3.36}$$

$$P_5(s) = \frac{\lambda_v (P_2(s) + P_4(s))}{s} \tag{3.37}$$

Equations (3.32)-(3.37) yield the following state probabilities:

$$\begin{aligned} P_o(t) = {} & \frac{(C_{19}+k_{20})(C_{19}+k_{21})(C_{19}+k_{22})\exp(C_{19}t)}{(C_{19}-C_{20})(C_{19}-C_{21})(C_{19}-C_{22})} + \\ & \frac{(C_{20}+k_{20})(C_{20}+k_{21})(C_{20}+k_{22})\exp(C_{20}t)}{(C_{20}-C_{19})(C_{20}-C_{21})(C_{20}-C_{22})} + \\ & \frac{(C_{21}+k_{20})(C_{21}+k_{21})(C_{21}+k_{22})\exp(C_{21}t)}{(C_{21}-C_{19})(C_{21}-C_{20})(C_{21}-C_{22})} + \\ & \frac{(C_{22}+k_{20})(C_{22}+k_{21})(C_{22}+k_{22})\exp(C_{22}t)}{(C_{22}-C_{19})(C_{22}-C_{20})(C_{22}-C_{21})} \end{aligned} \tag{3.38}$$

$$P_1(t) = \frac{3\lambda(C_{19}^2 + C_{19}B_{36} + B_{37})\exp(C_{19}t)}{(C_{19} - C_{20})(C_{19} - C_{21})(C_{19} - C_{22})} +$$
$$\frac{3\lambda(C_{20}^2 + C_{20}B_{36} + B_{37})\exp(C_{20}t)}{(C_{20} - C_{19})(C_{20} - C_{21})(C_{20} - C_{22})} +$$
$$\frac{3\lambda(C_{21}^2 + C_{21}B_{36} + B_{37})\exp(C_{21}t)}{(C_{21} - C_{19})(C_{21} - C_{20})(C_{21} - C_{22})} +$$
$$\frac{3\lambda(C_{22}^2 + C_{22}B_{36} + B_{37})\exp(C_{22}t)}{(C_{22} - C_{19})(C_{22} - C_{20})(C_{22} - C_{21})} \tag{3.39}$$

$$P_2(t) = \frac{(12\lambda\lambda_v C_{19} + B_{38})\exp(C_{19}t)}{(C_1 - C_2)(C_1 - C_3)(C_1 - C_4)} +$$
$$\frac{(12\lambda\lambda_v C_{20} + B_{38})\exp(C_{20}t)}{(C_{20} - C_{19})(C_{20} - C_{21})(C_{20} - C_{22})} +$$
$$\frac{(12\lambda\lambda_v C_{21} + B_{38})\exp(C_{21}t)}{(C_{21} - C_{19})(C_{21} - C_{20})(C_{21} - C_{22})} +$$
$$\frac{(12\lambda\lambda_v C_{22} + B_{38})\exp(C_{22}t)}{(C_{22} - C_{19})(C_{22} - C_{20})(C_{22} - C_{21})} \tag{3.40}$$

$$P_3(t) = \frac{6\lambda^2 B_{40}}{C_{19}C_{20}C_{21}C_{22}} + \frac{6\lambda^2(C_{19}^2 + C_{19}B_{39} + B_{40})\exp(C_{19}t)}{C_{19}(C_{19} - C_{20})(C_{19} - C_{21})(C_{19} - C_{22})} +$$
$$\frac{6\lambda^2(C_{20}^2 + C_{20}B_{39} + B_{40})\exp(C_{20}t)}{C_{20}(C_{20} - C_{19})(C_{20} - C_{21})(C_{20} - C_{22})} +$$
$$\frac{6\lambda^2(C_{21}^2 + C_{21}B_{39} + B_{40})\exp(C_{21}t)}{C_{21}(C_{21} - C_{19})(C_{21} - C_{20})(C_{21} - C_{22})} +$$
$$\frac{6\lambda^2(C_{22}^2 + C_{22}B_{39} + B_{40})\exp(C_{22}t)}{C_{22}(C_{22} - C_{19})(C_{22} - C_{20})(C_{22} - C_{21})} \tag{3.41}$$

$$P_4(t)=\frac{2\lambda_v(C_{19}^2+C_{19}B_{41}+B_{42})\exp(C_{19}t)}{(C_{19}-C_{20})(C_{19}-C_{21})(C_{19}-C_{22})}+$$
$$\frac{2\lambda_v(C_{20}^2+C_{20}B_{41}+B_{42})\exp(C_{20}t)}{(C_{20}-C_{19})(C_{20}-C_{21})(C_{20}-C_{22})}+$$
$$\frac{2\lambda_v(C_{21}^2+C_{21}B_{41}+B_{42})\exp(C_{21}t)}{(C_{21}-C_{19})(C_{21}-C_{20})(C_{21}-C_{22})}+$$
$$\frac{2\lambda_v(C_{22}^2+C_{22}B_{41}+B_{42})\exp(C_{22}t)}{(C_{22}-C_{19})(C_{22}-C_{20})(C_{22}-C_{21})} \qquad (3.42)$$

$$P_5(t)=\frac{2\lambda_v^2B_{44}}{C_{19}C_{20}C_{21}C_{22}}+\frac{2\lambda_v^2(C_{19}^2+C_{19}B_{43}+B_{44})\exp(C_{19}t)}{C_{19}(C_{19}-C_{20})(C_{19}-C_{31})(C_{19}-C_{22})}+$$
$$\frac{2\lambda_v^2(C_{20}^2+C_{20}B_{43}+B_{44})\exp(C_{20}t)}{C_{20}(C_{20}-C_{19})(C_{20}-C_{21})(C_{20}-C_{22})}+$$
$$\frac{2\lambda_v^2(C_{21}^2+C_{21}B_{43}+B_{44})\exp(C_{21}t)}{C_{21}(C_{21}-C_{19})(C_{21}-C_{20})(C_{21}-C_{22})}+$$
$$\frac{2\lambda_v^2(C_{22}^2+C_{22}B_{43}+B_{44})\exp(C_{22}t)}{C_{22}(C_{22}-C_{19})(C_{22}-C_{20})(C_{22}-C_{21})} \qquad (3.43)$$

The symbols k_{20}, k_{21}, and k_{22} are defined in the appendix C.

C_{19}, C_{20}, C_{21}, and C_{22} are the roots of the Equation ($s^4+A_{10}s^3+A_{11}s^2+A_{12}s+A_{13}=0$), obtained by solving the above equation using the softwares maple and mathematica[43,25]. The constants A_{10} to $A_{13.}$ and B_{36} to B_{44} are defined in the appendix A and appendix B.

$$C_{19},C_{20}=-\frac{1}{4}A_{10}+\frac{1}{2}3^{1/2}L^{1/2}\pm\frac{1}{12}\begin{bmatrix}[18A_{10}^2M^{1/3}L^{1/2}-48A_{11}M^{1/3}L^{1/2}\\ -6M^{2/3}L^{1/2}+72A_{10}A_{12}L^{1/2}-288A_{13}L^{1/2}\\ -24A_{11}^2L^{1/2}-18A_{10}^3M^{1/3}3^{1/2}+\\ 72A_{10}A_{11}M^{1/3}3^{1/2}-144A_{10}A_{11}M^{1/3}3^{1/2}]/M^{1/3}L^{1/2}\end{bmatrix}^{1/2}$$

$$C_{21},C_{22} = -\frac{1}{4}A_{10} + \frac{1}{12}3^{1/2}L^{1/2} \pm \frac{6^{1/2}}{12}\begin{bmatrix}[3A_{10}{}^{2}M^{1/3}L^{1/2} - 8A_{11}M^{1/3}L^{1/2} \\ -M^{2/3}L^{1/2} + 12A_{10}A_{12}L^{1/2} - 48A_{13}L^{1/2} \\ -4A_{11}{}^{2}L^{1/2} + 3A_{10}{}^{3}M^{1/3}3^{1/2} + \\ 12A_{10}A_{11}M^{1/3}3^{1/2} - 24A_{12}M^{1/3}3^{1/2}]/M^{1/3}L^{1/2}\end{bmatrix}^{1/2}$$

The symbols M and L are defined in the appendix C.

The reliability of the system is

$$R(t) = P_0(t) + P_1(t) + P_2(t) + P_4(t) \tag{3.44}$$

The system mean time to failure is given by

$$\begin{aligned} MTTF &= \int_0^\infty R(t)\,dt \\ &= \frac{k_{20}k_{21}k_{22} + 3\lambda k_{21}k_{22} + 2\lambda_v k_{20}k_{21} + 6\lambda\lambda_v(k_{22} + k_{20})}{k_{19}k_{20}k_{21}k_{22} - 3\lambda\mu_1 k_{21}k_{22} - 2\lambda_v\mu_2 k_{20}k_{21}} \end{aligned} \tag{3.45}$$

The symbol k_{19} is defined in the appendix C.

The plots of state probabilities, system reliabilities and mean time to failure are shown in Figures.3.3.2, 3.3.3 and 3.3.4 for the specified values of the model parameters, respectively.

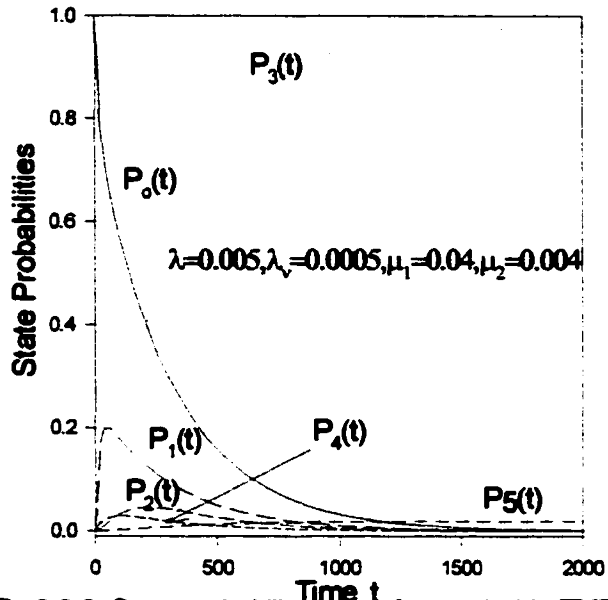


Fig.3.3.2. State probability plots of a repairable TMR subsystem with two Voters in parallel.

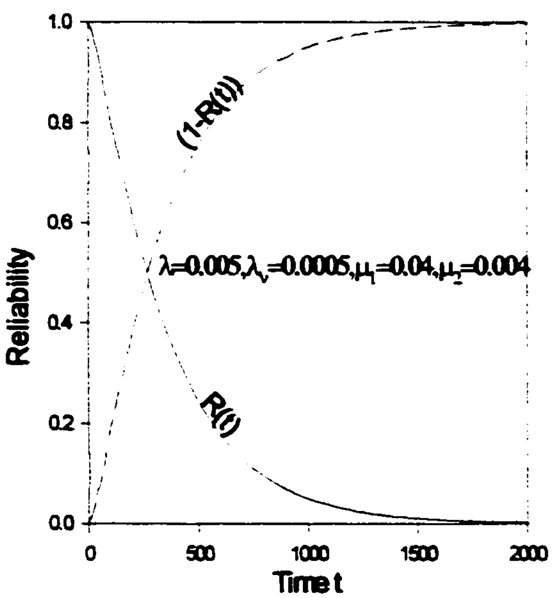


Fig.3.3.3. Reliability plots of a repairable TMR subsystem with two Voters in parallel.

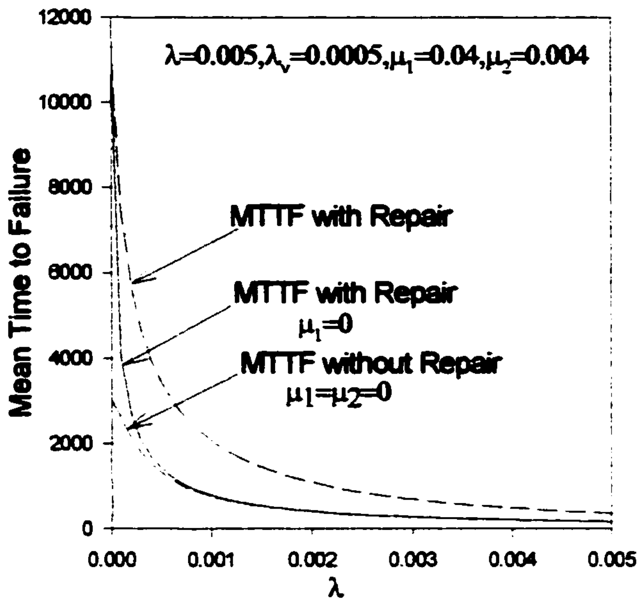


Fig.3.3.4. Mean time to failure plots of a repairable TMR subsystem with two Voters in parallel.

3.4 Model IV

This model represents a repairable TMR subsystem with three redundant voters in series. Thus for, N=3 in Fig.2.1, this section presents the reliability analysis of a repairable TMR system using the Markov state technique. The system transition diagram is shown in Fig.3.4.1.At time t=0 all the three units and three voters start working normally. The numeral in each box denotes the state number At least two units and one voter must function normally to ensure the system success. The failed unit and failed voters are repaired.

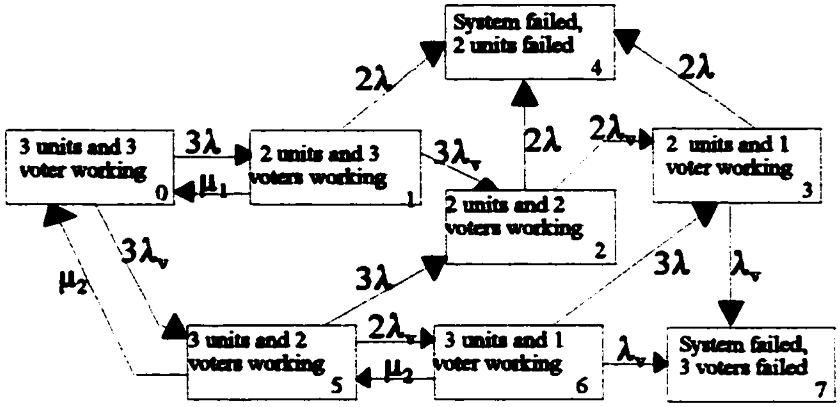


Fig.3.4.1 State space diagram of a TMR subsystem with three Voters in parallel.

The following equations are associated with Figure.3.4.1.

$$\frac{dP_o(t)}{dt} + (3\lambda + 3\lambda_v)P_o(t) = \mu_1 P_1(t) + \mu_2 P_5(t) \tag{3.46}$$

$$\frac{dP_1(t)}{dt} + (2\lambda + 3\lambda_v + \mu_1)P_1(t) = 3\lambda P_o(t) \tag{3.47}$$

$$\frac{dP_2(t)}{dt} + (2\lambda + 2\lambda_v)P_2(t) = 3\lambda P_5(t) + 2\lambda_v P_1(t) \tag{3.48}$$

$$\frac{dP_3(t)}{dt} + (2\lambda + \lambda_v)P_3(t) = 3\lambda P_6(t) + 2\lambda_v P_2(t) \tag{3.49}$$

$$\frac{dP_4(t)}{dt} = 2\lambda(P_1(t) + P_2(t) + P_3(t)) \tag{3.50}$$

$$\frac{dP_5(t)}{dt} + (3\lambda + 2\lambda_v + \mu_2)P_5(t) = 3\lambda_v P_o(t) + \mu_2 P_6(t) \tag{3.51}$$

$$\frac{dP_6(t)}{dt} + (3\lambda + \lambda_v + \mu_2)P_6(t) = 2\lambda_v P_5(t) \tag{3.52}$$

$$\frac{dP_7(t)}{dt} = \lambda_v(P_3(t) + P_6(t)) \tag{3.53}$$

At t=0, $P_o(0)=1$, and $P_1(0)=P_2(0)=P_3(0)=P_4(0)=P_5(0)=P_6(0)=P_7(0)=0$.

By solving Equations (3.46)-(3.53), yields the state probabilities in the s domain:

$$P_o(s) = \frac{(s+k_{24})(s^4 + s^3B_{45} + s^2B_{46} + sB_{47} + B_{48})}{(s^6 + s^5A_{14} + s^4A_{15} + s^3A_{16} + s^2A_{17} + sA_{18} + A_{19})} \tag{3.54}$$

$$P_1(s) = \frac{3\lambda(s^4 + s^3B_{49} + s^2B_{50} + sB_{51} + B_{52})}{(s^6 + s^5A_{14} + s^4A_{15} + s^3A_{16} + s^2A_{17} + sA_{18} + A_{19})} \tag{3.55}$$

$$P_2(s) = \frac{(s^3 18\lambda\lambda_v + s^2B_{53} + sB_{54} + B_{55})}{(s^6 + s^5A_{14} + s^4A_{15} + s^3A_{16} + s^2A_{17} + sA_{18} + A_{19})} \tag{3.56}$$

$$P_3(s) = \frac{(s^2 54\lambda\lambda_v + sB_{56} + B_{57})}{(s^6 + s^5A_{14} + s^4A_{15} + s^3A_{16} + s^2A_{17} + sA_{18} + A_{19})} \tag{3.57}$$

$$P_4(s) = \frac{6\lambda^2(s^4 + s^3B_{58} + s^2B_{59} + sB_{60} + B_{61})}{s(s^6 + s^5A_{14} + s^4A_{15} + s^3A_{16} + s^2A_{17} + sA_{18} + A_{19})} \tag{3.58}$$

$$P_5(s) = \frac{3\lambda_v(s^3 + s^2B_{62} + sB_{63} + B_{64})(s+k_{28})}{(s^6 + s^5A_{14} + s^4A_{15} + s^3A_{16} + s^2A_{17} + sA_{18} + A_{19})} \tag{3.59}$$

$$P_6(s) = \frac{6\lambda_v^2(s^3 + s^2B_{65} + sB_{66} + B_{67})}{(s^6 + s^5A_{14} + s^4A_{15} + s^3A_{16} + s^2A_{17} + sA_{18} + A_{19})} \tag{3.60}$$

$$P_7(s) = \frac{6\lambda_v^3(s^3 + s^2B_{68} + sB_{69} + B_{70})}{s(s^6 + s^5A_{14} + s^4A_{15} + s^3A_{16} + s^2A_{17} + sA_{18} + A_{19})} \tag{3.61}$$

Where

$P_i(s)$ is the Laplace transform of the system probability being in state i; for i=0,1,2,...,7.

The constants A_{14} to A_{19}, and B_{45} to B_{70} are defined in the appendix A and B.

The symbols k_{24} and k_{28} are defined in appendix C.

The Laplace transform of the system reliability is

$$R(s) = P_o(s) + P_1(s) + P_2(s) + P_3(s) + P_5(s) + P_6(s) \tag{3.62}$$

The system mean time to failure is given by

$$\text{MTTF} = \lim_{s \to 0} R(s)$$

$$= \frac{k_{24}B_{49} + 3\lambda B_{53} + B_{56} + B_{58} + 3\lambda_v k_{28} B_{65} + 6\lambda_v^2 B_{69}}{A_{19}} \tag{3.63}$$

The plots of the mean time to failures are shown in Figure.3.4.2, for the specified values of the model parameters.

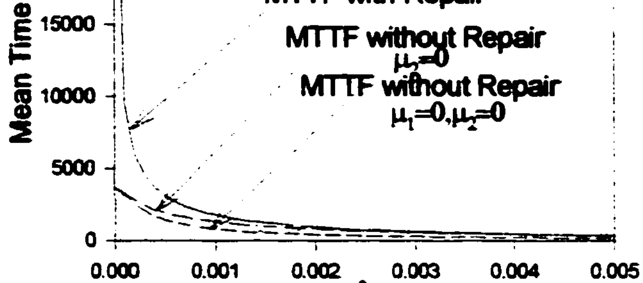


Fig.3.4.2. Mean time to failure plots of a repiarable TMR subsystem with three redundant Voters in series.

The plots of the state probability and reliability in Figures.3.4.3 and 3.4.4, for the specified values of the model parameters which have been obtained by solving the Equation ($s^6+A_{14}s^5+A_{15}s^4+A_{16}s^3+A_{17}s^2+A_{18}s^2+A_{19}= 0$) as shown in the previous models.

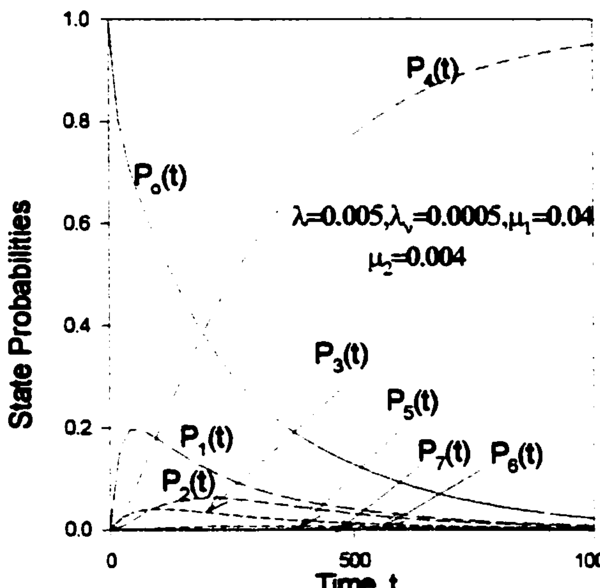


Fig.3.4.3. State probability plots of a repairable TMR subsystem with three redundant Voters in series.

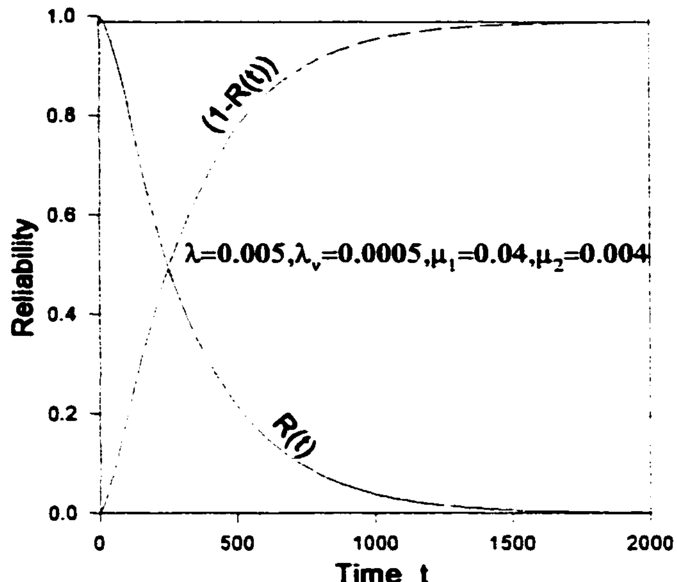


Fig.3.4.3 . Reliability plots of a repairable TMR subsystem with three redundant Voters in series.

The Figures. 3.4.5 and 3.4.6 show the improvement in system reliabilities and mean time to failure as the number of voters are increased. The graphs were obtained by using the softwares maple and mathematica for the specified values of the model parameters [45,23].

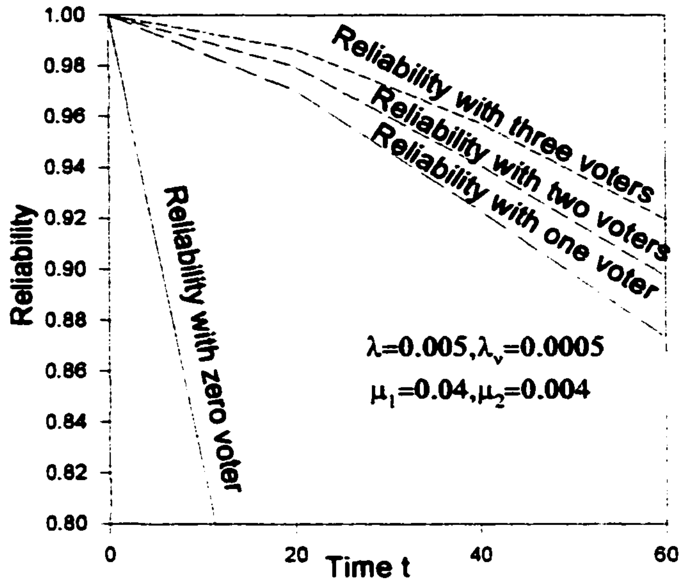


Figure 3.4.5. System reliability for varying number number of voters.

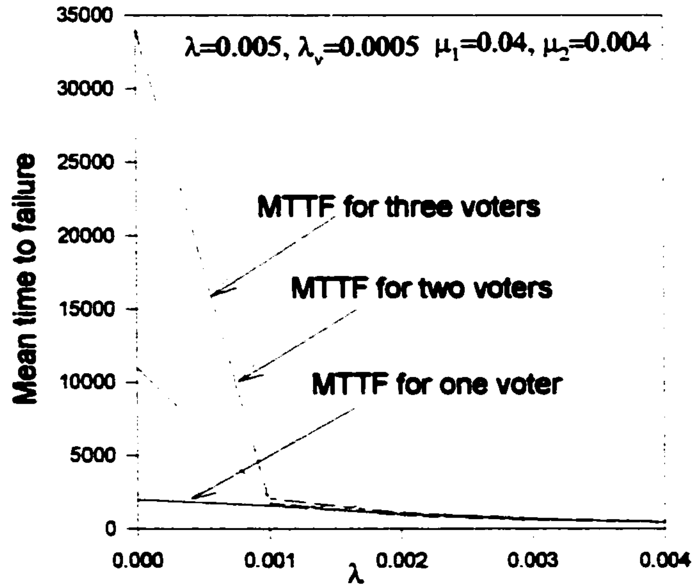


Figure. 3.4.6 . System mean time to failure for varying number of voters.

The Tables 3.1 and 3.2 show the improvement in system reliability and system mean time to failure as the number of voters are increased from zero to three.

Table 3.1 Numerical comparison of system reliabilities of repairable TMR subsystems with zero, one, two, and three redundant voters in series.

t	$R_0(t)$*	$R_1(t)$*	$R_2(t)$*	$R_3(t)$*
0.0000	1.0000	1.0000	1.0000	1.0000
20.0000	0.6439	0.9700	0.9796	0.9866
40.0000	0.3988	0.9229	0.9407	0.9573
60.0000	0.2469	0.8730	0.8972	0.9195
80.0000	0.1529	0.8245	0.8531	0.8759
100.0000	0.0947	0.7783	0.8089	0.8285

* $R_0(t)$ is the system reliability of a repairable TMR subsystem.

$R_1(t)$ is the system reliability of a repairable TMR subsystem with one voter in series..

$R_2(t)$ is the system reliability of a repairable TMR subsystem with two voters in series.

$R_3(t)$ is the system reliability of a repairable TMR subsystem with three voters in series.

Table 3.2 Numerical comparison of system mean time to failures for repairable TMR subsystems with one, two, and three redundant voters in series.

λ	$MTTF_1$*	$MTTF_2$*	$MTTF_3$*
0.0000	2000	11000	34076
0.0010	1583	2058	1756
0.0020	1025	1077	937
0.0030	679	684	614
0.0040	479	479	442
0.0050	358	359	337

* $MTTF_1$ is the system mean time to failure of a repairable TMR subsystem with one voter in series.

$MTTF_2$ is the system mean time to failure of a repairable TMR subsystem with two voters in series.

$MTTF_3$ is the system mean time to failure of a repairable TMR subsystem with three voters in series

3.5 Generalized Probabilities in s Domain for N Redundant Voters in Series

In the above sections 3.1 to 3.4 the s domain equations have been developed for voters zero to three. In a similar manner generalized expressions for the state probabilities in the s domain have been developed.

The generalized probability equations in the s domain for N redundant voters in series are:

$$P_o(s) = \frac{1 + P_1(s)\mu_1 + P_{N+2}(s)\mu_2}{s + 3\lambda + N\lambda_v} \text{ where } N = (0, 1, \ldots N-1), \mu_2 \text{ to exist } N \geq 2 \qquad (3.64)$$

$$P_1(s) = \frac{P_o(s)3\lambda}{s + 2\lambda + N\lambda_v + \mu_1} \text{ where } N = (0,1,2,\ldots N-1) \tag{3.65}$$

$$P_2(s) = \frac{P_1(s)N\lambda_v + P_{N+2}(s)3\lambda}{s + 2\lambda + (N-1)\lambda_v} \text{ where } N \geq 2,3,4,5,\ldots N-1. \tag{3.66}$$

$$P_N(s) = \frac{P_{N-1}(s)(N-1)\lambda_v + P_{N+2}(s)3\lambda}{s + 2\lambda + (N-2)\lambda_v} \text{ where } N \geq 3,4,5,\ldots N-1. \tag{3.67}$$

$$P_{N+2}(s) = \frac{P_o(s)N\lambda_v}{s + 3\lambda + (N-1)\lambda_v + \mu_2} \text{ where } N \geq 2,3,4,5,\ldots. N-1. \tag{3.68}$$

$$P_{N+2}(s) = \frac{P_o(s)(N-1)\lambda_v + P_{N+3}(s)\mu_2}{s + 3\lambda + (N-1)\lambda_v + \mu_2} \text{ where } N \geq 3,4,..n-1.,\mu_2 \text{ to exist } n \geq 3. \tag{3.69}$$

$$P_{N+3}(s) = \frac{P_5(s)(N-2)\lambda_v}{s + 3\lambda + (N-2)\lambda_v + \mu_2} \text{ where } N \geq 3,4,5,\ldots N-1. \tag{3.70}$$

where

Pi(s) is the Laplace transform of the system probability in state i; for i=0,1,2,..,N.

N is the number of voters.

When N=0 there are 2 up states.

When N=1 there are 2 up states.

When N=2 there are 4 up states.

When N=3 there are 6 up states and so on.

Chapter 4

Availability of Fully Failed TMR Systems with N Redundant Voters

In chapter 3 repairable TMR systems with redundant voters in series was discussed. In this chapter the system presents the availability analysis of a fully failed TMR subsystem with N redundant voters in series. This means that if anyone of the units of a TMR subsystem fails or the voters fail the system still operates. The function of the voter is to verify the output of the units and allow the best two units to operate.

The steady state availability analyses of systems are presented in the proceeding sections with the help of four models representing fully failed TMR subsystems having voters one to four. Generalized expressions for steady availability have been developed..

Assumptions

The following assumptions are associated with all the models studied under this chapter:

- Failure and repair rates are exponential.
- Failures are statistically independent.
- All units and voters are identical.
- Repaired system is as good as replaceable.
- At least two units and one voter must operate normally for system success.

Notations

The following symbols are associated with all the failed TMR system models in this chapter:

λ is the single unit failure rate.

λ_v is the voter failure rate.

μ_1 is the failed unit repair rate.

μ_2 is the voter repair rate.

AV is the system steady state availability.

UAV is the system steady state unavailability.

n is the number of states.

P_i is the steady state probability that the system is in state i, for i=0,1,2,3,...n.

N is the number of voters.

4.1 Model I

For N=1 in Fig 2.1, this section presents the availability analysis of a fully failed Triple Modular Redundant system as shown in Fig.2.1 using the stochastic technique. At time t=0 all the three units and one voter start working normally. At least two units must function normally to ensure the system success. The failed system is repaired. The numeral in each box in Fig.4.1.1 denotes the state.

.

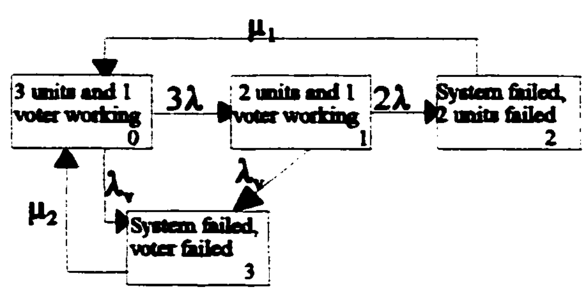


Fig.4.1.1 State space diagram of a fully failed TMR subsystem having one Voter and failed states in repair.

The calculations for steady-state probabilities for a Markov chain can be most easily done via balance equations[175]. Equating the inbound and outbound flows for Fig.4.1.1 gives a set of balanced algebraic equations that can be solved for the steady state probabilities.

$$P_o(3\lambda + \lambda_v) = \mu_1 P_2 + \mu_2 P_3 \tag{4.1}$$

$$P_1(2\lambda + \lambda_v) = P_o 3\lambda \tag{4.2}$$

$$P_2\mu_1 = P_1 2\lambda \tag{4.3}$$

$$P_3\mu_2 = \lambda_v(P_o + P_1) \tag{4.4}$$

Solving the Equations (4.1)-(4.4) in terms of P_o results in:

$$P_o = P_o \tag{4.5}$$

$$P_1 = \left[\frac{3\lambda}{2\lambda + \lambda_v}\right]P_o \tag{4.6}$$

$$P_2 = \left[\frac{6\lambda^2}{\mu_1(2\lambda + \lambda_v)}\right]P_o \tag{4.7}$$

$$P_3 = \frac{\lambda_v}{\mu_2}\left[1 + \frac{3\lambda}{2\lambda + \lambda_v}\right]P_o \tag{4.8}$$

Using the fact that $P_o+P_1+P_2+P_3=1$

$$P_o = \frac{(2\lambda + \lambda_v)\mu_1\mu_2}{(2\lambda + \lambda_v)\mu_1\mu_2 + 3\lambda\mu_1\mu_2 + 6\lambda^2\mu_2 + \lambda_v(5\lambda + \lambda_v)\mu_1} \tag{4.9}$$

by substituting equation 4.9 into equations 4.8, 4.7 and 4.6 the steady state probabilities P_1, P_2, and P_3 can be determined.

The steady state system availability is given by

$$AV = P_o + P_1 \tag{4.10}$$

The plots of steady state probabilities and availabilities are shown in Figures.4.1.2 and 4.1.3 for the specified values of the model parameters.

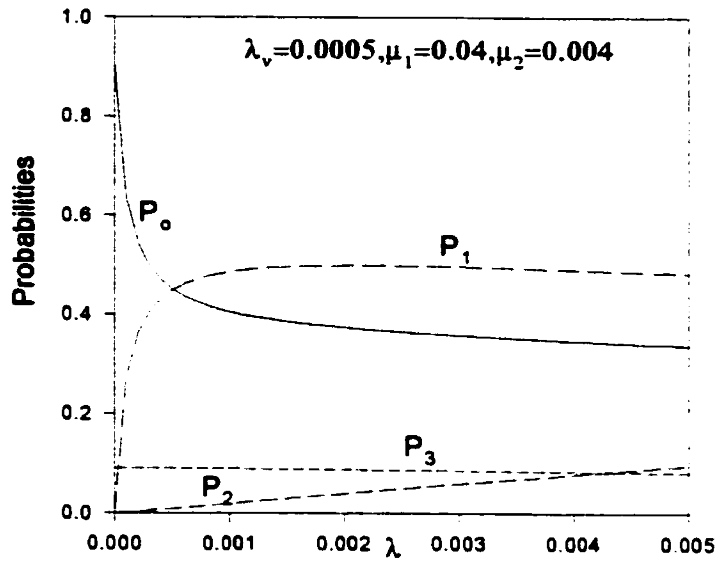


Fig.4.1.2 Steady state probablities plots of a fully failed TMR subsystem having one Voter and failed states in repair.

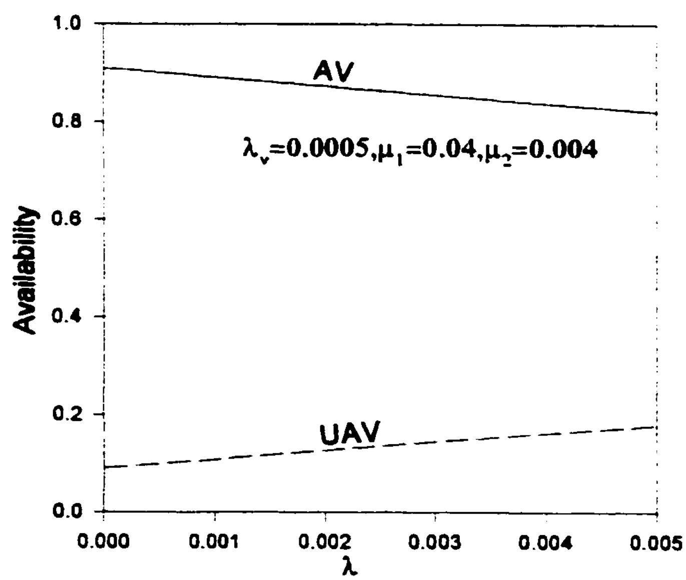


Fig.4.1.3 Availability plots of a fully failed TMR subsystem having one Voter and failed states in repair.

4.2 Model II

In this case the system is composed of a TMR subsystem and two redundant voters in series. More specifically, for N=2 in Fig 2.1, this section presents the availability analysis of a fully failed TMR subsystem with two redundant voters in series by using the stochastic technique. The state space diagram is shown in Fig.4.2.1. At time t=0 all the three units and two voters start working normally. The numeral in each box denotes the state. At least two units and one voter must must function normally to ensure the system success. The failed system is repaired.

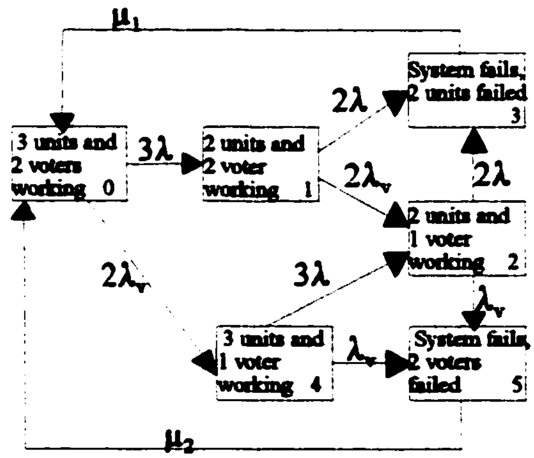


Fig.4.2.1 State space diagram of a fully failed TMR subsystem having two Voters and failed states in repair.

The calculations for steady-state probabilities for a Markov chain can be most easily done via balance equations[175]. Equating the inbound and outbound flows for Fig.4.2.1 gives a set of balanced algebraic equations that can be solved for the steady state probabilities.

$$P_o(3\lambda + 2\lambda_v) = \mu_1 P_3 + \mu_2 P_5 \qquad (4.11)$$

$$P_1(2\lambda + 2\lambda_v) = P_o 3\lambda \qquad (4.12)$$

$$P_2(2\lambda + \lambda_v) = P_1 2\lambda_v \tag{4.13}$$

$$P_3\mu_1 = 2\lambda(P_1 + P_2) \tag{4.14}$$

$$P_4(3\lambda + \lambda_v) = P_o 2\lambda_v \tag{4.15}$$

$$P_5\mu_2 = \lambda_v(P_2 + P_4) \tag{4.16}$$

Solving Equations (4.10)-(4.16) in terms of P_o results in:

$$P_o = P_o \tag{4.17}$$

$$P_1 = \left[\frac{3\lambda}{2\lambda + 2\lambda_v}\right]P_o \tag{4.18}$$

$$P_2 = \left[\frac{6\lambda\lambda_v}{(2\lambda + \lambda_v)(2\lambda + 2\lambda_v)}\right]P_o \tag{4.19}$$

$$P_3 = \frac{2\lambda}{\mu_1}\left[\frac{3\lambda}{2\lambda + 2\lambda_v} + \frac{6\lambda\lambda_v}{(2\lambda + 2\lambda_v)(2\lambda + \lambda_v)}\right]P_o \tag{4.20}$$

$$P_4 = \left[\frac{2\lambda_v}{3\lambda + \lambda_v}\right]P_o \tag{4.21}$$

$$P_5 = \frac{\lambda_v}{\mu_2}\left[\frac{6\lambda\lambda_v}{(2\lambda + \lambda_v)(2\lambda + 2\lambda_v)} + \frac{2\lambda_v}{3\lambda + \lambda_v}\right]P_o \tag{4.22}$$

Using the fact that $P_o+P_1+P_2+P_3+P_4+P_5=1$.

$$P_o = \left[\frac{D_1D_2D_3\mu_1\mu_2}{\begin{array}{c}D_1D_2D_3\mu_1\mu_2 + 3\lambda D_2D_3\mu_1\mu_2 + 6\lambda\lambda_v D_3\mu_1\mu_2 + 6\lambda^2 D_2D_3\mu_2 + \\ 12\lambda^2\lambda_v D_3\mu_2 + 2\lambda_v D_1D_2\mu_1\mu_2 + 6\lambda\lambda_v^2 D_3\mu_1 + 2\lambda\lambda_v^2 D_1D_2\mu_1\end{array}}\right] \tag{4.23}$$

The symbols D_1, D_2, and D_3 are defined in the appendix C.

By substituting equation 4.23 into equations 4.22, 4.21, 4.20, 4.19, and 4.18, the steady state probabilities P_1, P_2, P_3, P_4 and P_5 can be determined.

The system availability is given by

$$AV = P_o + P_1 + P_2 + P_4 \qquad (4.24)$$

The plots of steady state probabilities and availabilities are shown in Figures. 4.2.1 and 4.2.2 for the specified values of the model parameters.

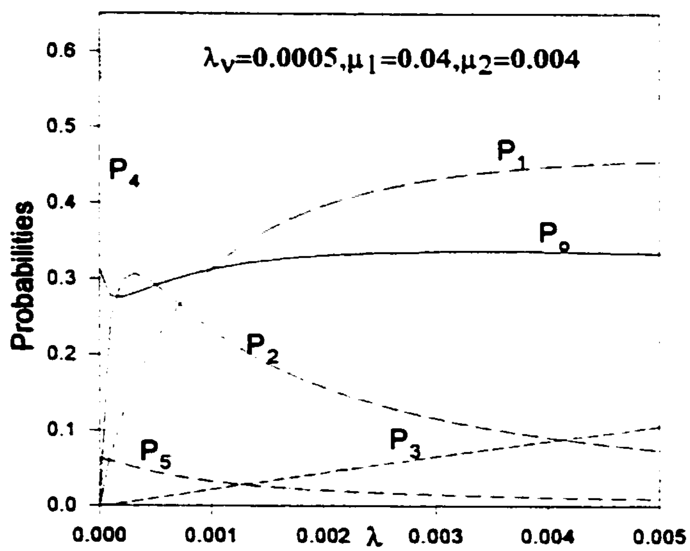


Fig.4.2.2 Steady state probability plots of a fully failed TMR subsystem having two Voters and failed states in repair.

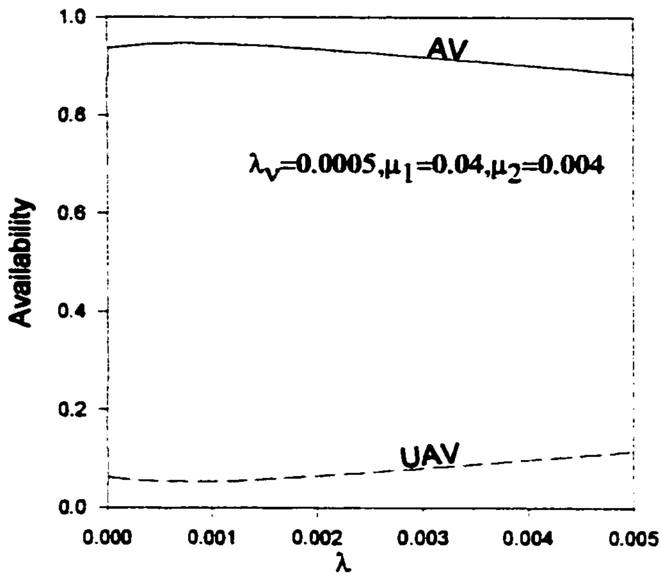


Fig.4.2.3 Availability plots of a fully failed TMR subsystem having two Voters and failed states in repair.

4.3 Model III

This model represents a fully failed TMR subsystem with three redundant voters in series. Thus for N=3 in Fig.2.1, this section presents the availability analysis of this model by using the stochastic technique. The system state space diagram is shown in Fig.4.3.1.At time t=0 all the three units and three voters start working normally. The numeral in each box denotes the state. At least two units and one voter must function normally to ensure the system success. The failed system is repaired.

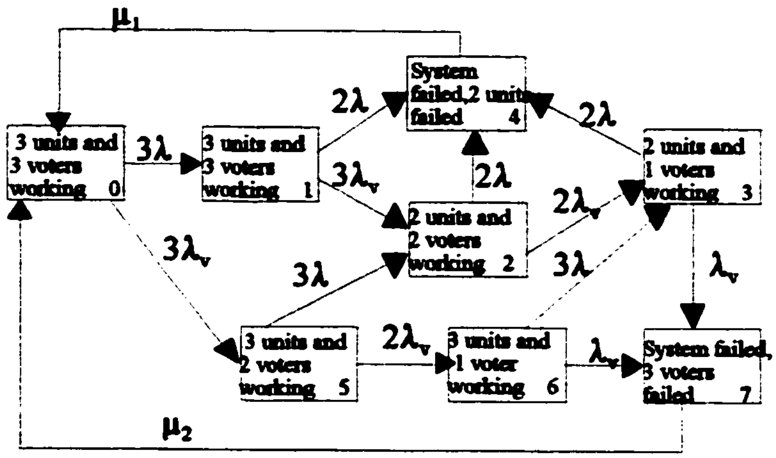


Fig.4.3.1 State space diagram of a fully failed TMR sub-system having three Voters and failed states in repair .

The calculations for steady-state probabilities for a Markov chain can be most easily done via balance equations[175]. Equating the inbound and outbound flows for Fig.4.3.1 gives a set of balanced algebraic equations that can be solved for the steady state probabilities.

$$P_o(3\lambda + 3\lambda_v) = P_4\mu_1 + P_7\mu_2 \tag{4.25}$$

$$P_1(2\lambda + 3\lambda_v) = P_o 3\lambda \tag{4.26}$$

$$P_2(2\lambda + 2\lambda_v) = P_1 3\lambda_v + P_5 3\lambda \tag{4.27}$$

$$P_3(2\lambda + \lambda_v) = P_2 2\lambda_v + P_6 3\lambda \tag{4.28}$$

$$P_4\mu_1 = 2\lambda(P_1 + P_2 + P_3) \tag{4.29}$$

$$P_5(3\lambda + 2\lambda_v) = P_o 3\lambda_v \tag{4.30}$$

$$P_6(3\lambda + \lambda_v) = P_5 2\lambda_v \tag{4.31}$$

$$P_7\mu_2 = \lambda_v(P_3 + P_6) \tag{4.32}$$

Solving the Equations (4.25)-(4.32) in terms of P_o yields:

$$P_o = P_o \tag{4.33}$$

$$P_1 = \left[\frac{3\lambda}{2\lambda + 3\lambda_v}\right]P_o \tag{4.34}$$

$$P_2 = \left[\frac{3\lambda_v}{2\lambda + 2\lambda_v}\right]\left[\frac{3\lambda}{2\lambda + 3\lambda_v}\right]P_o + \left[\frac{3\lambda}{2\lambda + 2\lambda_v}\right]\left[\frac{3\lambda_v}{3\lambda + 2\lambda_v}\right]P_o \tag{4.35}$$

$$P_3 = \left[\frac{2\lambda_v}{2\lambda + \lambda_v}\right]P_2 + \left[\frac{3\lambda}{2\lambda + \lambda_v}\right]P_6 \tag{4.36}$$

$$P_4 = \left[\frac{2\lambda}{\mu_1}\right][P_1 + P_2 + P_3] \tag{4.37}$$

$$P_5 = \left[\frac{3\lambda_v}{3\lambda + 2\lambda_v}\right]P_o \tag{4.38}$$

$$P_6 = \left[\frac{2\lambda_v}{3\lambda + \lambda_v}\right]P_5 \tag{4.39}$$

$$P_7 = \left[\frac{\lambda_v}{\mu_2}\right][P_3 + P_6] \tag{4.40}$$

Using the fact that $P_o+P_1+P_2+P_3+P_4+P_5+P_6+P_7=1$

we get

$$P_o = \left[\frac{1}{1+\frac{3\lambda}{D_4} + B_{71} + B_{72} + B_{73} + \frac{3\lambda_v}{D_7} + B_{74} + \frac{\lambda_v}{\mu_1}[B_{73} + B_{74}]}\right] \tag{4.41}$$

The symbols D_4 and D_7 are defined in the appendix C.

By substituting equation 5.42 into equations 5.41 to 5.34 the steady state probabilities P_1, P_2, P_3, P_4, P_5, P_6, and P_7 can be determined. The constants B_{71} to B_{74} are defined in the appendix B.

The system availability is

$$AV = P_o + P_1 + P_2 + P_3 + P_5 + P_6 \tag{4.42}$$

The plots of steady state probabilities and availabilities are shown in Figures. 4.3.2 and 4.3.3 for the specified values of the model parameters.

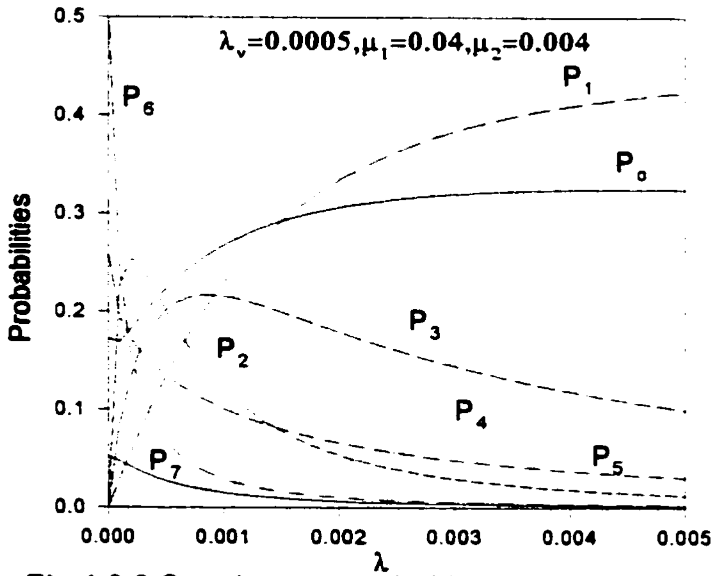


Fig.4.3.2 Steady state probability plots of a fully failed TMR subsystem having three Voters and failed states in repair.

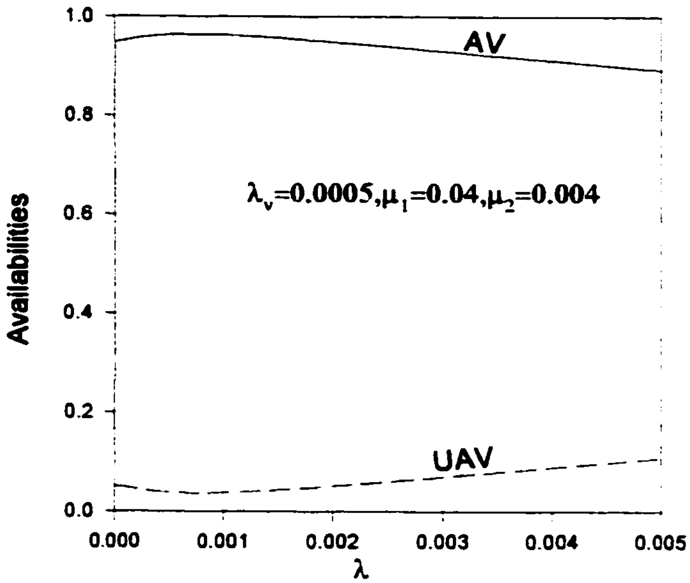


Fig.4.3.3 Availability plots of a fully failed TMR subsystem having three Voters and failed states in repair.

4.4 Model IV

This model represents a fully failed TMR subsystem with four redundant voters in series. Thus for N=4 in Fig.2.1, this section presents the reliability analysis of a fully failed TMR system with voters by using the stochastic technique. The system transition diagram is shown in Fig.4.4.1. At time t=0 all the three units and four voters start working normally. The numeral in each box denotes the state. At least two units and one voter must function normally to ensure the system success. The failed system is repaired.

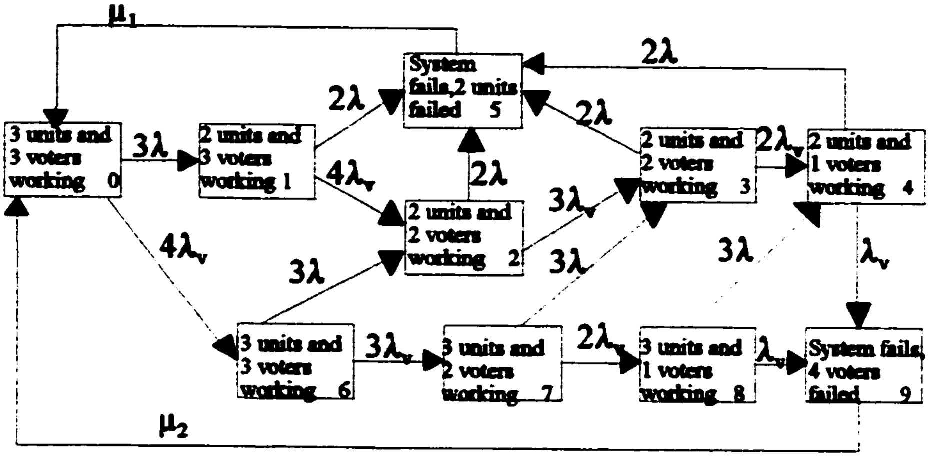


Fig.4.4.1 State transition diagram of a fully failed TMR subsystem having four Voters and failed states in repair.

The calculations for steady-state probabilities for a Markov chain can be most easily done via balance equations[175]. Equating the inbound and outbound flows for Fig.4.4.1 gives a set of balanced algebraic equations that can be solved for the steady state probabilities.

$$P_o(3\lambda + 4\lambda_v) = P_6\mu_1 + P_9\mu_2 \tag{4.43}$$

$$P_1(2\lambda + 4\lambda_v) = P_o 3\lambda \tag{4.44}$$

$$P_2(2\lambda + 3\lambda_v) = P_1 4\lambda_v + P_6 3\lambda \tag{4.45}$$

$$P_3(2\lambda + 2\lambda_v) = P_2 3\lambda_v + P_7 3\lambda \tag{4.46}$$

$$P_4(2\lambda + \lambda_v) = P_3 2\lambda_v + P_8 3\lambda \tag{4.47}$$

$$P_5\mu_1 = 2\lambda(P_1 + P_2 + P_3 + P_4) \tag{4.48}$$

$$P_6(3\lambda + 3\lambda_v) = P_o 4\lambda_v \tag{4.49}$$

$$P_7(3\lambda + 2\lambda_v) = P_6 3\lambda_v \tag{4.50}$$

$$P_8(3\lambda + \lambda_v) = P_7 2\lambda_v \tag{4.51}$$

$$P_9\mu_2 = \lambda_v(P_7 + P_4) \tag{4.52}$$

Solving the Equations (4.43)-(4.520 in terms of P_o results in:

$$P_o = P_o \quad (4.53)$$

$$P_1 = \left[\frac{3\lambda}{2\lambda + 4\lambda_v}\right]P_o \quad (4.54)$$

$$P_2 = \left[\frac{4\lambda_v}{2\lambda + 3\lambda_v}\right]P_1 + \left[\frac{3\lambda}{2\lambda + 3\lambda_v}\right]P_6 \quad (4.55)$$

$$P_3 = \left[\frac{3\lambda_v}{2\lambda + 2\lambda_v}\right]P_2 + \left[\frac{3\lambda}{2\lambda + 2\lambda_v}\right]P_7 \quad (4.56)$$

$$P_4 = \left[\frac{2\lambda_v}{2\lambda + \lambda_v}\right]P_3 + \left[\frac{3\lambda}{2\lambda + \lambda_v}\right]P_3 \quad (4.57)$$

$$P_5 = \left[\frac{2\lambda}{\mu_1}\right][P_1 + P_2 + P_3 + P_4] \quad (4.58)$$

$$P_6 = \left[\frac{4\lambda_v}{3\lambda + 3\lambda_v}\right]P_o \quad (4.59)$$

$$P_7 = \left[\frac{3\lambda_v}{3\lambda + 2\lambda_v}\right]P_6 \quad (4.60)$$

$$P_8 = \left[\frac{2\lambda_v}{3\lambda + \lambda_v}\right]P_7 \quad (4.61)$$

$$P_9 = \left[\frac{\lambda_v}{\mu_2}\right][P_4 + P_8] \quad (4.62)$$

Using the fact that $P_o+P_1+P_2+P_3+P_4+P_5+ P_6+P_7+P_8+P_9=1$

we get

$$P_o = \left[\frac{1}{1 + \frac{3\lambda}{D_9} + B_{75} + B_{76} + B_{77} + B_{78} + \frac{4\lambda_v}{D_{13}} + \frac{12\lambda_v^2}{D_{13}D_{14}} + \frac{24\lambda_v^3}{D_{13}D_{14}D_{15}} + B_{79}} \right] \qquad (4.63)$$

The symbols D_9, D_{13}, D_{14}, and D_{15} are defined in the appendix C. The constants B_{75} to B_{79} are defined in the appendix B.

By substituting equation 4.63 into equations 4.62 to 4.53 the steady state probabilities P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, and P_9 can be determined.

The system availability is

$$AV = P_o + P_1 + P_2 + P_3 + P_4 + P_6 + P_7 + P_8 \qquad (4.64)$$

The plots of steady state probabilities and availabilities are shown in Figures. 4.4.2 and 4.4.3 for the specified values of the model parameters.

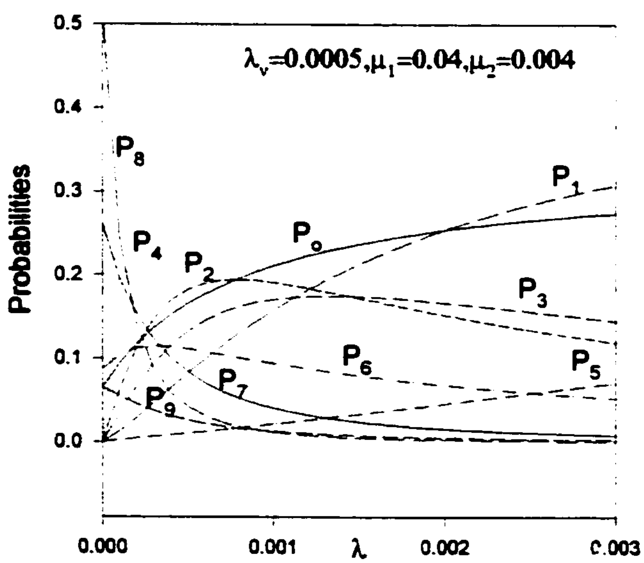


Fig.4.4.2 Steady state probability plots of a fully failed TMR subsystem having four Voters and failed states in repair.

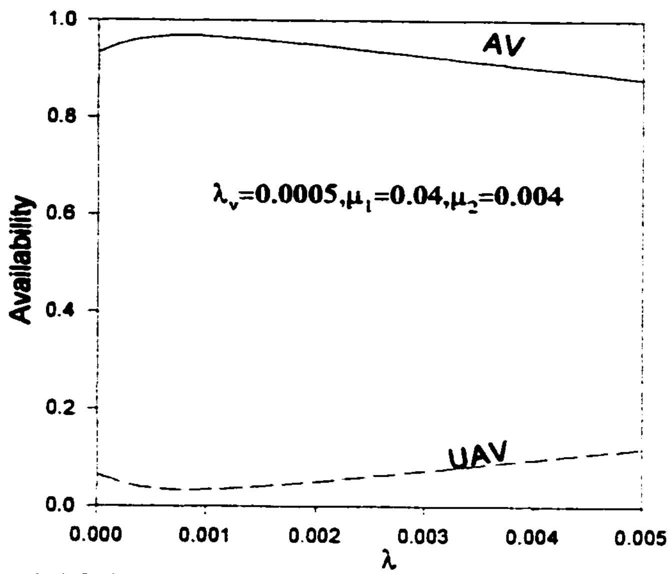


Fig.4.4.3 Availability plots of a fully failed TMR subsystem having four Voters and failed states in repair.

4.5 Generalized Formula for Availability of Fully Failed TMR Systems with N Voters in Series

The steady availability expressions of fully failed TMR systems with failed states in repair having one, two, three and four redundant voters in series have been developed in sections 4.1, 4.2, 4.3 and 4.4. In a similar manner a generalized formula for the availability of TMR subsystem with N redundant voters in series has been developed in this section.

The generalized equations for availability for N voters are:

$$P_1 = \left[\frac{3\lambda}{2\lambda + N\lambda_v}\right] P_o \quad \text{where } N \geq 1 \tag{4.65}$$

$$P_{N+1} = \left[\frac{2\lambda}{\mu_1}\right] [P_N + P_{N-1} \dots\dots\dots P_{N-i}] \quad \text{where } N \geq 1,\ i \leq N,\ N - i \geq 0, \tag{4.66}$$

$$P_N = \left[\frac{N\lambda_v}{2\lambda + (N-1)\lambda_v}\right] P_{N-1} + \left[\frac{N\lambda_v}{2\lambda + (N-1)\lambda_v}\right] P_{2N} \quad \text{where } N \geq 2 \tag{4.67}$$

$$P_{N+2} = \left[\frac{N\lambda_v}{3\lambda + (N-1)\lambda_v}\right] [P_{N-2} + P_{N-i} \dots] \quad \text{where } N \geq 2,\ N - i \leq 0 \tag{4.68}$$

$$P_{N+3} = \left[\frac{N\lambda_v}{3\lambda + (N-1)\lambda_v}\right] P_{N+2} \quad \text{where } N \geq 3,4,5 \tag{4.69}$$

$$P_{a+(N-1)d} = \left[\frac{\lambda_v}{\mu_2}\right] [P_0 + P_N] \quad \text{where } N = 1,\ a = 3,\ d = 2 \tag{4.70}$$

$$P_{a+(N-1)d} = \left[\frac{\lambda_v}{\mu_2}\right] [P_N + P_{N+1}] \quad \text{where } N \geq 2,\ a = 3,\ d = 2 \tag{4.71}$$

Using the fact that $P_o+P_1+P_2 \dots\dots\dots\dots\dots\dots +P_N=1$

$$P_o = \left[\frac{1}{1 + P_1 + P_2 + P_3 \ldots\ldots + P_N}\right] \tag{4.72}$$

after determining P_o from equation 4.72 . The steady state probabilities P_1, P_2, P_3, P_4...... P_n can be found by substituting equation 4.72 into equations 4.71, 4.70, 4.69, 4.68, 4.67, 4.66, and 4.65.

$$AV=P_o+P_1+P_2+P_3+\ldots\ldots+P_N. \tag{4.73}$$

The Figure 4.5.1 shows the improvement in steady state availability with increase in voters and the Table.4.1 shows the numerical comparison of the availability with increase in voters for the specified model parameters.

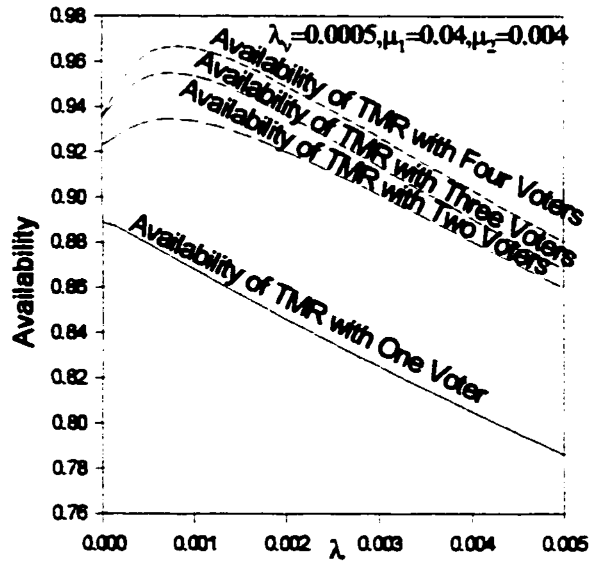


Fig.4.5.1 Steady state availability plots of fully failed TMR subsystem with increase in Voters.

Table. 4.1 Numerical comparison of steady state availabilities of fully failed TMR subsystems with one, two, three and four redundant voters in series.

λ	A_{V1}*	A_{V2}*	A_{V3}*	A_{V4}*
0.0000	0.8889	0.923	0.9362	0.9348
0.0005	0.8791	0.933	0.9534	0.9639
0.0010	0.8679	0.933	0.9535	0.9658
0.0015	0.8568	0.928	0.9463	0.9590
0.0020	0.8459	0.920	0.9363	0.9491
0.0025	0.8353	0.910	0.9252	0.9380
0.0030	0.8250	0.901	0.9137	0.9265
0.0035	0.8149	0.890	0.9021	0.9148
0.0040	0.8051	0.880	0.8906	0.9031
0.0045	0.7955	0.870	0.8793	0.8916
0.0050	0.7861	0.859	0.8681	0.8802

* A_{v1} is the system availability for a fully failed TMR subsystem with one voter in series.

A_{v2} is the system availability for a fully failed TMR subsystem with two redundant voters in series.

A_{v3} is the system availability for a fully failed TMR subsystem with three redundant voters in series.

A_{v4} is the system availability for a fully failed TMR subsystem with four voters redundant in series.

Chapter 5

Availability of a Fully/Partially Failed TMR System with Redundant Voters in Series

In chapter 4 availability analysis of a TMR system was discussed. In this chapter the system presents the steady state TMR network having redundant voters in series. More specifically, the failed system or individual items are repaired.

Thus, the chapter presents the availability analysis of TMR subsystems with one to three redundant voters in series. Improvements in availability are shown for TMR subsystems having one the three redundant voters in series.

Assumptions

The following assumptions are associated with all the models studied in this chapter:

- Failure and repair rates are exponential
- Failures are statistically independent
- All units and voters are identical
- Repaired system is as good as replaceable
- At least two units and one voter must operate for system success

Notations

The following symbols are associated with all the models studied in this chapter:

λ is the single unit failure rate.

λ_v is the voter failure rate.

μ_1 is the failed unit repair rate.

μ_2 is the voter repair rate.

t is time.

AV is the system steady state availability.

UAV is the system steady state unavailability.

n is the number of states.

P_i is the steady state probability that the system is in state i, for i=0,1,2,3,....,n..

N is the number of voters.

5.1 Model I

For N=1 in Fig 2.1, this section presents the steady state availability analysis of a TMR subsystem with one voter in series by using the stochastic technique. The state space diagram is shown in Fig.5.1.1. At time t=0 all the three units and one voter start working normally. At least two units and one voter must function normally to ensure the system success. The numeral in each box denotes the state number. The system failed states, and intermediate states are repaired.

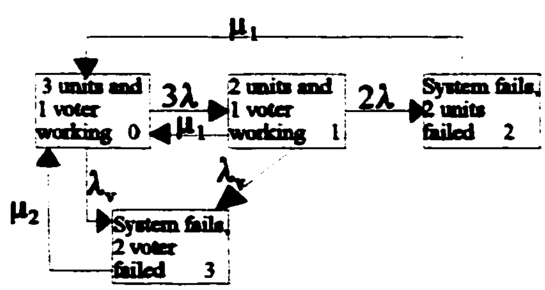


Fig.5.1.1 State space diagram of a partially failed TMR subsystem having one Voter with failed and intermittent states in repair.

The calculations for steady-state probabilities for a Markov chain can be most easily done via balance equations. Equating the inbound and outbound flows for the Figure. 5.1.1 gives a set of balanced algebraic equations that can be solved for the steady state probabilities.

$$P_o(3\lambda + \lambda_v) = \mu_1(P_1 + P_2) + \mu_2 P_3 \tag{5.1}$$

$$P_1(2\lambda + \lambda_v + \mu_1) = P_o 3\lambda \tag{5.2}$$

$$P_2\mu_1 = P_1 2\lambda \tag{5.3}$$

$$P_3\mu_2 = \lambda_v(P_o + P_1) \tag{5.4}$$

Solving the Equations (5.1)-(5.4) in terms of P_o results in:

$$P_o = P_o \tag{5.5}$$

$$P_1 = \left[\frac{3\lambda}{2\lambda + \lambda_v + \mu_1}\right]P_o \tag{5.6}$$

$$P_2 = \left[\frac{2\lambda}{\mu_1}\right]P_1 \tag{5.7}$$

$$P_3 = \left[\frac{\lambda_v}{\mu_2}\right][P_o + P_1] \tag{5.8}$$

Using the fact that $P_o+P_1+P_2+P_3=1$

$$P_o = \left[\frac{1}{1 + \frac{3\lambda}{D_{16}} + \frac{6\lambda^2}{D_{16}\mu_1} + \frac{\lambda_v}{\mu_2}\left[1 + \frac{3\lambda}{D_{16}}\right]}\right] \tag{5.9}$$

The symbol D_{16} is defined in the appendix C.

By substituting equation 5.9 into equations 5.8, 5.7 and 5.6 the steady state probabilities P_1, P_2 and P_3 can be determined.

The system steady availability is

$$AV = P_o + P_1 \tag{5.10}$$

The plots of steady state probabilities and availabilities are shown in Figures. 5.1.2 and 5.1.3 for the specified values of the model parameters.

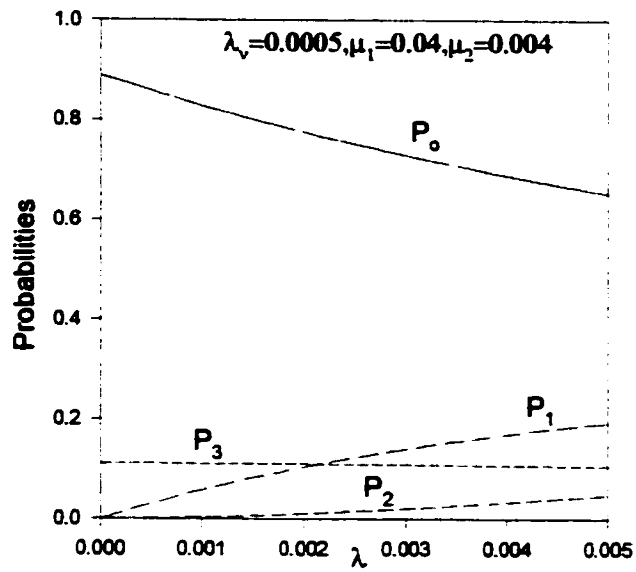


Fig.5.1.2 Steady state probabilities of partially failed TMR subsystem having one Voter with failed and intermittent states in repair.

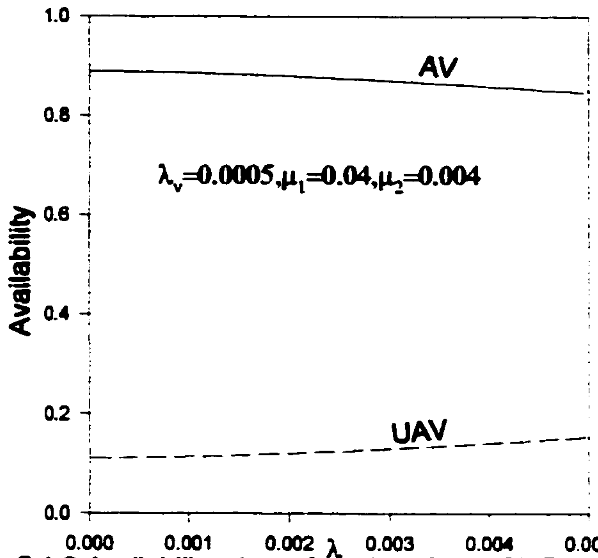


Fig.5.1.3 Availability plots of partially failed TMR subsystem having one Voter with failed and intermittent states in repair.

5.2 Model II

In this case the system represents a subsystem with two redundant voters in series. More specifically, for N=2 in Fig.2.1, this section presents the availability analysis of a partially failed TMR subsystem with voters by using the stochastic technique. The system state space diagram is shown in Fig.5.2.1. At time t=0 all the three units and two voters start working normally. The numeral in each box denotes the state At least two units and one voter must function normally to ensure the system success. The failed states, and intermediate states are repaired.

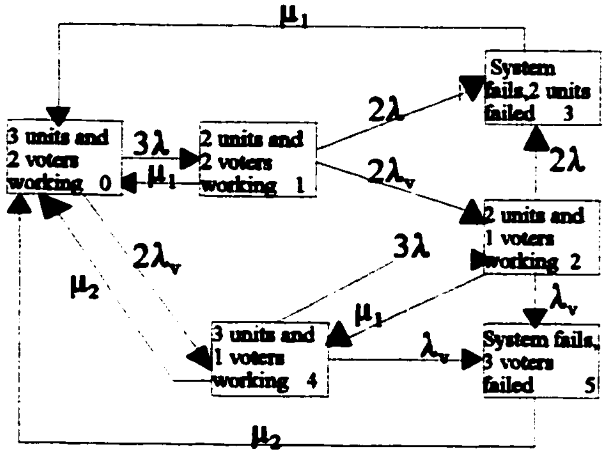


Fig.5.2.1 State space diagram of partially failed TMR subsystem having two Voters and with failed and intermittent states in repair.

The calculations for steady-state probabilities for a Markov chain can be most easily done via balance equations. Equating the inbound and outbound flows in Fig.5.2.1 gives a set of balanced algebraic equations that can be solved for the steady state probabilities.

$$P_o(3\lambda + 2\lambda_v) = \mu_1(P_1 + P_3) + \mu_2(P_5 + P_4) \qquad (5.11)$$

$$P_1(2\lambda + 2\lambda_v + \mu_1) = 3\lambda P_o \qquad (5.12)$$

$$P_2(2\lambda + \lambda_v + \mu_1) = P_1 2\lambda_v + P_4 3\lambda \qquad (5.13)$$

$$P_3\mu_2 = 2\lambda(P_1 + P_2) \qquad (5.14)$$

$$P_4(3\lambda + \lambda_v + \mu_2) = P_o 2\lambda_v + P_2\mu_1 \qquad (5.15)$$

$$P_5\mu_2 = \lambda_v(P_2 + P_4) \qquad (5.16)$$

Solving the Equations (5.11)-(5.160 in terms of P_o yields:

$$P_o = P_o \qquad (5.17)$$

$$P_1 = \left[\frac{3\lambda}{2\lambda + 2\lambda_v + \mu_1}\right]P_o \tag{5.18}$$

$$P_2 = \left[\frac{2\lambda_v}{2\lambda + \lambda_v + \mu_1}\right]P_1 + \left[\frac{3\lambda}{2\lambda + \lambda_v + \mu_1}\right]P_4 \tag{5.19}$$

$$P_3 = \left[\frac{2\lambda}{\mu_1}\right][P_1 + P_2] \tag{5.20}$$

$$P_4 = \left[\frac{2\lambda_v}{3\lambda + \lambda_v + \mu_2}\right]P_o \tag{5.21}$$

$$P_5 = \left[\frac{\lambda_v}{\mu_2}\right][P_2 + P_4] \tag{5.22}$$

Using the fact that $P_o+P_1+P_2+P_3+P_4+P_5=1$

$$P_o = \left[1 + \frac{3\lambda}{D_{17}} + B_{80} + B_{81} + \frac{2\lambda_v}{D_{19}} + B_{82}\right] \tag{5.23}$$

The constants B_{80} to B_{82} are defined in the appendix B. The symbols D_{17} and D_{19} are defined in appendix C.

By substituting equation 5.23 into equations 5.22, 5.21, 5.20, 5.19 and 5.18 the steady state probabilities P_1, P_2, P_3, P_4, and P_5 can be determined.

The system steady availability is

$$AV = P_o + P_1 + P_2 + P_4 \tag{5.24}$$

The plots of steady state probabilities and availabilities are shown in Figures. 5.2.2 and 5.2.3 for the specified values of the model parameters.

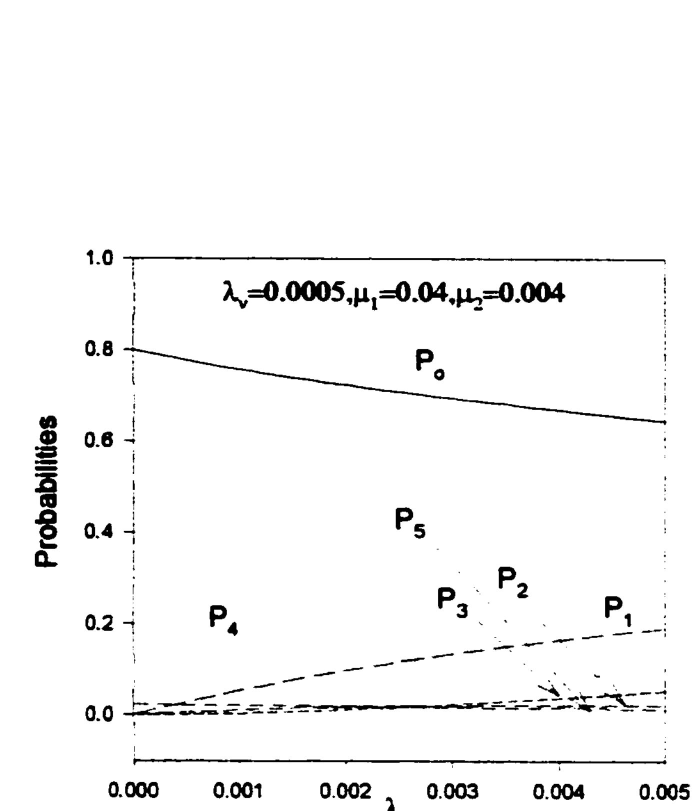


Fig.5.2.2 Steady state probablities of a partially failed TMR subsystem having two Voters with failed and intermittent states in repair.

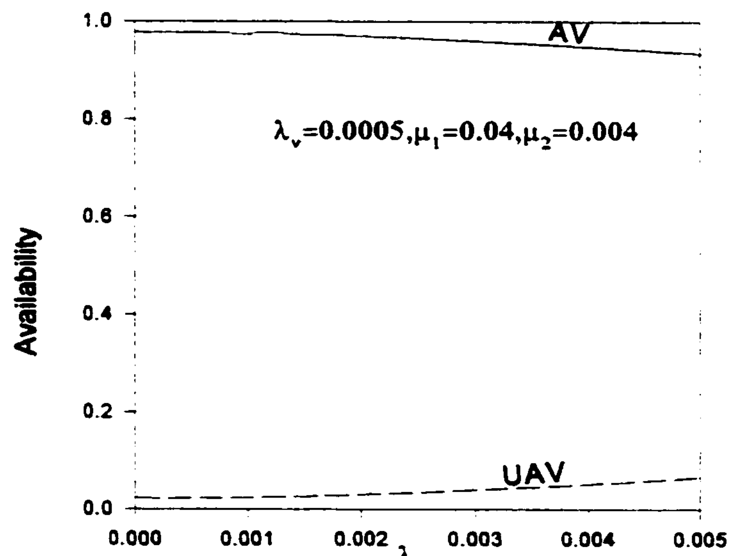


Fig.5.2.3 Availability plots of a partially failed TMR subsystem having two Voters with failed and intermittent states in repair.

5.3 Model III

This model represents a TMR subsystem with three redundant voters in series. Thus for N=3 in Fig 2.1, this section presents the steady availability analysis of a TMR subsystem with three voters by using the stochastic technique. The state space diagram is shown in Fig.5.3.1. At time t=0 all the three units and three voters start working normally. The numeral in each box denotes the state. At least two units and one voter must function normally to ensure the system success. The failed states, and intermediate states are repaired.

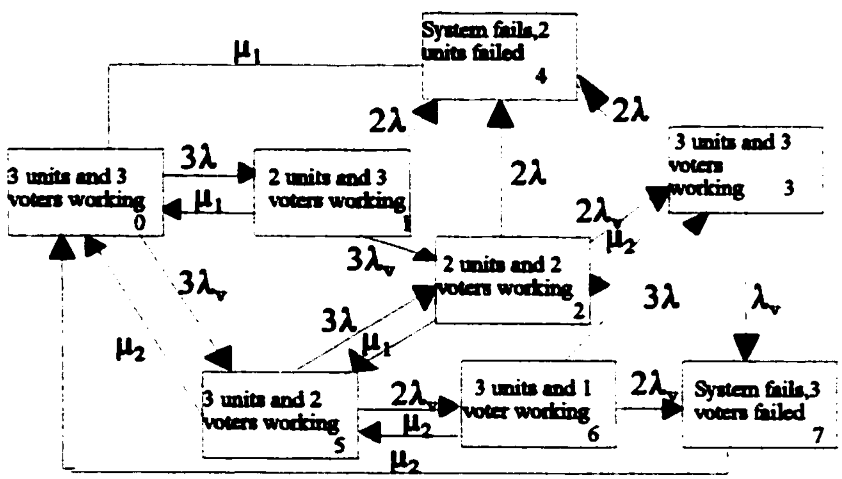


Fig.5.3.1 State space diagram of a partially failed TMR subsystem with three Voters and intermittent states in repair.

The calculations for steady-state probabilities for a Markov chain can be most easily done via balance equations. Equating the inbound and outbound flows in Figure. 5.3.1 gives a set of balanced algebraic equations that can be solved for the steady state probabilities.

$$P_0(3\lambda + 3\lambda_v) = \mu_1(P_1 + P_4) + \mu_2(P_7 + P_5) \tag{5.24}$$

$$P_1(2\lambda + 3\lambda_v + \mu_1) = P_o 3\lambda \tag{5.25}$$

$$P_2(2\lambda + 2\lambda_v + \mu_1) = P_3\mu_2 + P_5 3\lambda + P_1 3\lambda_v \tag{5.26}$$

$$P_3(2\lambda + \lambda_v + \mu_2) = P_2 2\lambda_v + P_6 3\lambda \tag{5.27}$$

$$P_4\mu_1 = 2\lambda(P_1 + P_2 + P_3) \tag{5.28}$$

$$P_5(3\lambda + 2\lambda_v + \mu_2) = P_o 3\lambda_v + P_2\mu_1 + P_6\mu_2 \tag{5.29}$$

$$P_6(3\lambda + \lambda_v + \mu_2) = P_5 2\lambda_v \tag{5.30}$$

$$P_7\mu_2 = \lambda_v(P_3 + P_6) \tag{5.31}$$

Solving the Equations (5.24)-(5.31) in terms of P_o results in:

$$P_o = P_o \tag{5.32}$$

$$P_1 = \left[\frac{3\lambda}{2\lambda + 3\lambda_v + \mu_1}\right]P_o \tag{5.33}$$

$$P_2 = \left[\frac{\mu_2}{2\lambda + 2\lambda_v + \mu_1}\right]P_3 + \left[\frac{3\lambda}{2\lambda + 2\lambda_v + \mu_1}\right]P_5 \tag{5.34}$$

$$P_3 = \left[\frac{2\lambda_v}{2\lambda + \lambda_v + \mu_2}\right]P_2 + \left[\frac{3\lambda}{2\lambda + \lambda_v + \mu_2}\right]P_6 \tag{5.35}$$

$$P_4 = \left[\frac{2\lambda}{\mu_1}\right][P_1 + P_2 + P_3] \tag{5.36}$$

$$P_5 = \left[\frac{3\lambda_v}{3\lambda + 2\lambda_v + \mu_2}\right]P_o + \left[\frac{\mu_1}{3\lambda + 2\lambda_v + \mu_2}\right]P_2 + \left[\frac{\mu_2}{3\lambda + 2\lambda_v + \mu_2}\right]P_6 \tag{5.37}$$

$$P_6 = \left[\frac{2\lambda_v}{3\lambda + \lambda_v + \mu_2}\right]P_5 \tag{5.38}$$

$$P_7 = \left[\frac{\lambda_v}{\mu_2}\right][P_3 + P_6] \tag{5.39}$$

Using the fact that $P_o+P_1+P_2+P_3+P_4+P_5+P_6+P_7=1$

$$P_o = \left[1/(1+\frac{3\lambda}{D_{20}} + B_{88} + B_{89} + B_{90} + B_{91} + B_{92} + B_{93}\right] \tag{5.40}$$

The constants where B_{88} to B_{93} are defined in the appendix B. The symbol D_{20} is defined in the appendix C.

By substituting equation 5.40 into equations 5.39, 5.38, 5.37, 5.36, 5.35, 5.34, and 5.33 the steady state probabilities P_1, P_2, P_3, P_4, P_5, P_6, and P_7 can be determined.

The system steady availability is

$$AV = P_o + P_1 + P_2 + P_3 + P_5 + P_6 \tag{5.41}$$

The plots of steady state probabilities and availabilities are shown in Figures. 5.3.2 and 5.3.3 for the specified values of the model parameters.

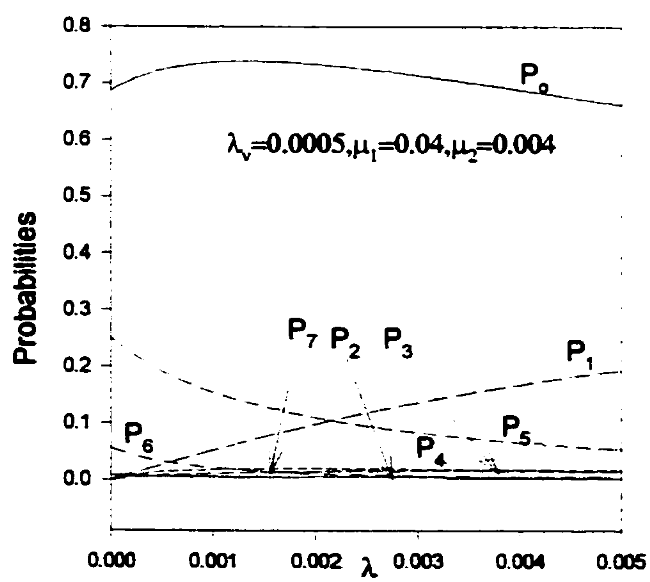


Fig.5.3.2 Steady state probablities of a partially failedTMR subsystem having three Voters with failed and intermittent states in repair.

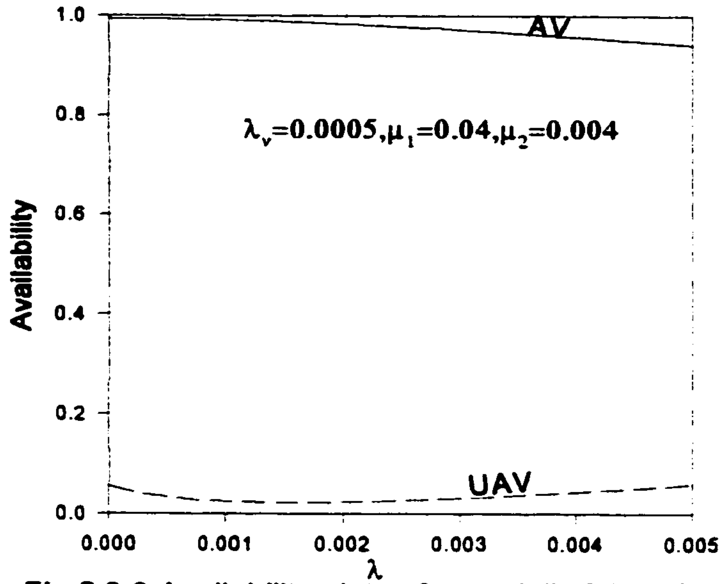


Fig.5.3.3 Availability plots of a partially failed TMR subsystem having three Voters with failed intermittent states in repair.

In the above sections 5.1, 5.2 and 5.3 we have seen the availability figures for one, two and three voters. In the Figure.5.3. 4 the comparison of the three voters is shown and the improvement in availability can be seen. The Table 5.1 shows the numerical values.

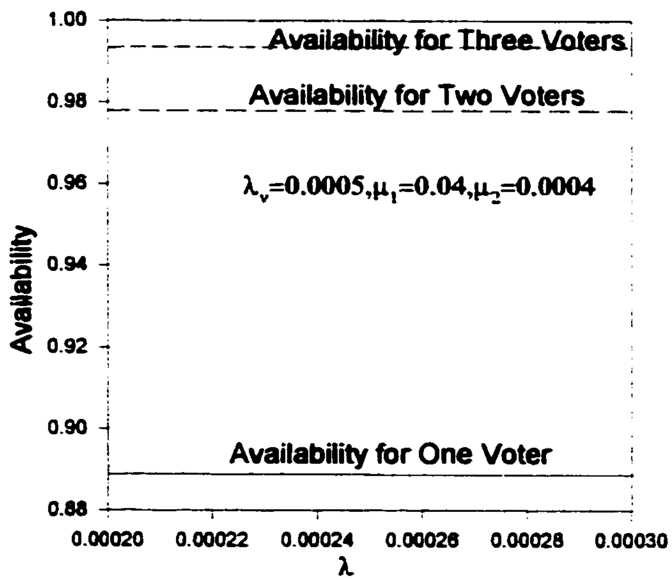


Fig.5.4.1 Steady state availability plots for partially failed TMR subsystem with increase in voters.

Table.5.1 A numerical comparison of the steady state availabilities for partially failed TMR subsystems with one, two and three redundant voters in series.

λ	A_{v1}*	A_{v2}*	A_{v3}*
0.0000	0.8889	0.9778	0.9930
0.0010	0.8863	0.9761	0.9913
0.0020	0.8796	0.9699	0.9833
0.0030	0.8701	0.9603	0.9715
0.0040	0.8586	0.9482	0.9573
0.0050	0.8458	0.9344	0.9415

* A_{v1} is the availability of a partially failed TMR subsystem with one voter in series.

A_{v2} is the availability of a partially failed TMR subsystem with two voters in series.

A_{v3} is the availability of a partially failed TMR subsystem with three voters in series.

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

In this study the repairable and non-repairable TMR system with redundant voters in series reliability and availability analyses were presented. Generalized expressions for system reliability, system availability and system mean time to failure with constant failure and repair rates were developed.

In the case of repairable and non-repairable TMR systems with redundant Voters in series, the study yielded the following results:

- The reliability analysis showed an improvement as the voters increased from zero to three, but any further increase in voters there was no improvement in reliability.
- The system mean time to failure increase as the number of voters are increased from one to four.
- The system mean time to failure for repairable systems is much higher than for non-repairable systems.
- The availability analysis shows an improvement in steady state availability as the voters are increased from one to four.
- The availability of partially/fully failed systems is higher than that for fully failed systems.

6.2 Recommendations for Further Study

The reliability and availability analysis performed in this study assumed that both the failure rates and repair rates are constant. An incorporation of common-cause failures in the TMR systems which have been discussed would give us a better understanding of the environmental aspects and human errors that cause fatal errors even when systems are operating normally. The models presented in the study were purely qualitative and should be used for quantitative analysis of TMR redundant systems by incorporating occurrence of common-cause failures and human errors. Also, under conditions when item and system failure and repair rates are not constant.

Bibliography

1 . Abbott, R. J., Resourceful Systems for Fault Tolerance, Reliability and Safety, Journal of Computing Surveys, Vol-22,1990, pp. 35-68

2. Avizienis, A., Short history of Fault tolerant computing, Informationstechnik - It, Computer, Systeme, Anwedungen, Vol.30, 1988, pp.162-168.

3. Avizienis, A., Laprice, J. C., Recovery Block Reliability Analysis with Failure Clustering, Proceedings of the Dependable Computing for Critical Applications, 1989, pp 75-103.

4. Adrian, E.C., Ambuj, G., Monte Carlo Simulation of Computer System, Availability/ Reliability Models, Proceedings of the17th Fault Tolerant Computing Symposium, 1987, pp.230- 235.

5. Alan, P. W., Jon, G. E., A Comparison of Predicted MTBF's to Field and Test Data, Proceedings of the Annual Reliability and Maintainability Symposium,1994, pp 153-156.

6. Ammann, P. E., Knight, J. C., Data Diversity , An Approach to Software Fault Tolerance, Digest of papers-Fault tolerant Computing Symposium , 1987, pp. 122-126.

7. Anderson, T., Barrett., Peter, A., Software Fault Tolerance, An Evaluation. IEEE Transactions on Software Engineering, Vol-11,1995, pp.1502-1510.

8. Anderson, T., Moulding, M . R., Evaluating Software Fault Tolerance in a Real-time System, Proceedings of the Second European Dependable Computing Symposium, 1982, pp. 327-328.

9. Anderson, T., Fault Tolerant Control Software, IEE Colloquium on Software for the Control of Gas Turbines , n 1986/21, 1-2.3.

10. Andrzej, H., Lecture notes in Computer Science, Proceedings of the Second European Dependable Computing Conference, Taormina, Italy, Springer-Verlag New York 1996,

11. Alper, K.C., Dave, E.E., An Experimental Evaluation of Software Redundancy as a Strategy for Improving Reliability, IEEE Transactions on Software Engineering, Vol.17, 1991, pp. 692-702.

12. Anthony, H . P., Lawrence, S., Some metrics from a formal development ,IEE Computing and Control Division Colloquium on practical application of formal methods, IEE Colloquium(Digest), 1995, p. 6/1-6/4.

13. Arevalo, S., Castellanos, J., Barco, L. J., Fault Tolerant Server on Mach, Journal Microprocessing and Microprogramming , Vol. 38, 1993, pp.793-800.

14. Arlat, J.,Kanoun, K., Laprie, J. C., Dependability Evaluation of Software Fault Tolerance Digest of Papers , Proceedings of the Fault Tolerant Computing Symposium, 1988, pp.142-147.

15. Arthur, E. L., Martin, L. S., Bounding and Approximating Markov Models, Proceedings of the Annual Reliability and Maintainability Symposium , 1990, pp. 254-258.

16. Augus, J. E., James, L .E., Combined Hardware/Software Reliability Models, Proceedings of the Annual Reliability and Maintainability Symposium ,1982, pp. 176-181.

17. Avizienis, A., The N-Version Approach to Fault tolerant, IEEE Transactions on Software Engineering , Vol.11 ,1985, pp.1491-1501.

18. Balasoma, S., Mura , I ., Synchronization Delay in Hardware Fault tolerance Techniques, Proceedings of the IEEE International Computer Performance and Dependability Symposium , 1996, pp.240-249.

19 . Ball, M ., Hardie, F., Effects and Detection of Intermittent Failures in Digital Systems, Proceedings of the Fall Joint computer Conference,1969, pp. 329-335.

20. Bastani, F. B., Yen, I, L., Chen, I, R., Class of Inherently Fault Tolerant Distributed programs , IEEE Transactions on Software Engineering, Vol. 14, 1988, pp. 1432-1442.

21. Belli, F. P., An approach to the Reliability Optimization of Software with Redundancy, IEEE Transactions on Software Engineering , Vol.17, 1991, pp 301-312.

22. Bengt, O., Maintaining Permanent And Temporary Faults in a Communications System, Proceedings of the Reliability and Quality Control Conference, 1987, pp.50-53.

23. Bhandari, I., Halliday, M. J., Chaar, J., Chillarege, R., Jones, K., Atkinson, J. S., Leopri-Costello, C., Jasper, P.Y., Tarver, E.D., Lewis, C. C.,Yonezawa, M ., In process Improvements Through Defect Data Interpretation, IBM Systems Journal, Vol.33,1994, pp. 182-214.

24. Bharsava, B., Hua, C., Cost Analysis of Recovery Block Scheme and its Implementation Issues, International Journal of Computer and Information Sciences, Vol.10,1981, pp. 359-382.

25. Bruce, F. T., Eve, A.T., Introduction to Mathematica, Cambridge, United Kingdom, Cambridge University Press, 1999.

26. Bondavalli, A., Di Giandomenico, F., Xu, J., Cost- Effective and Flexible Scheme for Software Fault Tolerance, Computer Systems Science and Engineering ,Vol.8,1994, pp. 234- 244.

27. Bondavalli, A., Simoncini, L., Structured Software Fault Tolerance with BSM, Proceedings of the Third Workshop on Future Trends of Distributed Computing Systems,1992, pp.278-286.

28. Boyd, M. A., Converting Fault Trees to Markov Chains for Reliability Prediction , Masters Degree Thesis, Dept of Computer Science, Duke University , Durham, North Carolina,1986.

29. Breuer, M. A., Testing for Intermittent Faults in Digital Circuits , IEEE Transactions on Computers., Vol.23, 1973, pp.241-245.

30. Bruce, M., Executable Assertion Development of The Distributed Parallel Environment, Proceedings of the 12th Annual International Computer Software and Application Conference ,1988, pp 284-291.

31. Burdick, G. R.,Fussell, J, B., Rasmuson, D, M .,Wilson, J, R., Phased Mission Analysis, IEEE Transactions on Reliability , Vol.26, 1977, pp. 43-49.

32. Butler, R. W.,White, A, L., Semi-Markov Unreliability Range Evaluator(SURE): Program and Mathematics , Technical Report No. TM-87593, NASA Langley Research Center , Virginia,1988.

33. Campbell, R. H., Randell, B., Error recovery in Asynchronous Systems, IEEE

Transactions on Software , Vol.12, 1986, pp. 811-826.

34. Carpenter, G. F., Tyrell, A. M ., Software Fault Tolerance in Concurrent Systems Programs, Journal of Microprocessing and Microprogramming, Vol.32, 1991, pp.373-380.

35. Charles, E. W., John, L. B., Reliability Characteristics of a Markov System with a Mission of Random Duration, IEEE Transactions on Reliability, Vol.34,1985, pp.393-396.

36. Chen, S. , Mills, S., A Binary Markov Process Model for Random Testing, IEEE Transactions on Software , Vol.22, 1996, pp. 218-223.

37. Chen, C., Cherkassky, H. V., Task Allocation and Reallocation for Fault Tolerance in Multicomputer systems, IEEE Transactions on Aerospace and Electronic Systems , Vol. 30, 1994, pp. 1094-1104.

38. Cha, S. D., Recovery Block Model and its Analysis, Proceeding of the 5th International Federation of Automatic Control, Workshop ,1986, pp.21- 26.

39. Chillarege, R ., Siewiorek, D.P., Experimental Evaluation of Computer systems Reliability, IEEE Transactions on Reliability, Vol .39, 1990, pp.506-510.

40. Cheng, J., Methods of Improving the Generality and Fault Tolerance of Software with Dbase III, Journal of Xi'An Petroleum Institute , Vol.10,1995, pp.4.

41. Chiueh, T., A Storage Architecture for the fast Checkpointing ,Journal of Information Science and Engineering , Vol. 9, 1993, pp.61-80.

42. Clematis, A., Gianuzzi,V., Software Fault Tolerance in Concurrent Ada Systems ,

Journal of Microprocessors and Microprogramming, Vol.32, 1991, pp.365-372.

43. Corless, R.M., "Essential Maple: An Introduction for Scientific Programmer's", Springer Verlag Inc., New York, 1995.

44. Clematis, A., Dodero, G., Gianuzzi, V., Hierarchical Program Structure for Concurrent Fault Tolerant Software, Journal of Microprocessors and Microprogramming , Vol.37, 1993, pp.179-182.

45. Clematis, A., Dodero, G., Gianuzzi, V., Designing Fault Tolerant Software with a Recovery Meta Program, Proceedings of the 9th Annual International Phoenix Conference on Computers and Communications ,1990, pp.864-865.

46. Clematis, A., Dodero, G., Gianuzzi, V., A Design Tool for Fault Tolerant Software, Proceedings of the International Conference on Computing System Software Engineering ,1990, pp.130-137.

47. Cuyvers, R., Lauwereins, R., Peperstrate, J., Software Environment for Uniform Hardware and Software Fault Tolerant Multi-processing, Computers, Computer Architecture and Microprocessors in Education Modeling and Simulation, Proceedings of the Annual Pittsburgh Conference , Vol. 21, 1990, pp.1305-1310.

48. Cuyvers, R., Lauwereins, R., Peperstrate, J., A Kernel for Multilevel Fault Tolerant Multiprocessing , Proceedings of the IEEE SouthEaston Conference, Vol. 1, 1991, pp. 248-252.

49. Deb, A. K., Goel, A. L., Model for Execution Time Behavior of a Recovery Block, Proceedings IEEE Computer Society's International Computer Software and Applications Conference,1986, pp. 497-502.

50. Deb .A. K., Stochastic Modeling for execution Time and Reliability of Fault Tolerant Programs Using Recovery Block and N-version Schemes, Dept of Computer Science, Ph.D. Thesis, Syracuse University, Syracuse, 1988.

51. DeMillo, R. A., Statistical Measures of Software Reliability , Technical Report No NTISADA100662, National Technical Information Service, Springfield, Virginia ,1980.

52. Don, O. K., Kua, H. K., The Impact of Maintenance Activities on Network Reliability Levels, Proceedings of the Reliability and Quality Control Conference, 1988, pp. 40-44.

53. Doyle, S. A., Dungan., Patterson-Hine , A. A., Combinatorial approach to Modeling Imperfect Coverage, IEEE Transactions on Reliability, Vol.40, 1991, pp.53-58.

54. Dugan, J.B., Lyu, M. R., System-Level Reliability and Sensitivity Analysis for Three Fault-Tolerant System Architectures, Proceedings of the 4th IFIP International Working Conference on Dependable Computing for Critical Applications, 1994, pp. 295-307.

55. Dugan, G .B., Bavsuo , S ., Boyd , M ., Dynamic Fault Tree Models for Fault Tolerant Computer Systems, IEEE Transactions on Reliability , Vol.41, 1992, pp.363-376..

56. Dugan, J.B., Bavuso, S., Boyd, M.,Fault trees and Markov models for reliability Analysis of Fault Tolerant Systems , Journal of Reliability Engineering and System Safety , Vol.39,1993, pp. 291-307.

57. Dugan, J. B., Simple Models of Fault Tolerant Software , Proceedings of the Annual Reliability and Maintainability Symposium,1993, pp. 354- 358.

58. Dugan, J.B., Correlated Hardware Failures in Redundant Systems, Proceedings of the 2nd IFIP Working Conference Dependable Computing for Critical Applications, Tucson, Arizona ,1991,pp.157-174.

59. Dugan, J. B., On Measurement and Modeling of Computer Systems Dependability, A Dialogue among Experts, IEEE Transactions on Reliability , Vol.39,1990, pp.506-510.

60. Dugan, J.B., Trivedi, K. S., Coverage Modeling for Dependability Analysis of Fault Tolerance, IEEE Transactions on Computers, Vol.38,1989, pp.775.

61. Dugan, G .B., Automated Analysis of Phased Mission Reliability, IEEE Transactions on Reliability, Vol.40, 1991, pp.45-52.

62. Dugan, J.B., Veerarahavan, M., Boyd, M., Mittal, N., Bounded Approximate Rcliability Models for Fault Tolerant Distributed Systems, Proceedings of the 8th Symposium on Reliable Distributed Systems, 1989, pp.137-147.

63. Echtle, K., Lovric, T., Hardware and Software Fault Tolerance using Fail Silent Virtual Duplex Systems ,Proceedings of the Conference on Fault Tolerant Parallel and Distributed Systems, 1995, pp. 10-17.

64. Faulkner, T., Bartlett, C. W., Small, M., Hardware Design Faults : A Classification and some Measurements, ICL Technical Journal, Vol. 6 ,1982, pp.218-228.

65. Ferens, D. V., Computer Software Fault Tolerance, IEEE Proceedings of the National Aerospace and Electronics Conference, 1987, pp.845-849.

66. Fevzi, B., Piotr, J., Reliability Modeling of the Fault Tolerant Program, Proceedings of the Reliability and Quality Control Conference, 1989, pp. 53-56.

67. Fuhrman, C. P., Chutani, S., Nussbaumer, H. J., Hardware/Software Fault Tolerance with Multiple Task Redundancy, Proceedings IEEE Symposium on Computers and Communications , 1995, pp.171-177.

68. Furuyana, T., Yoshio, Arai., Iio, K., Fault Generation Model and Mental Stress

Effective Analysis, Journal of Systems and Software, Vol. 26, 1994, pp.31-42.

69.Garth, A. G., Redundant Disk Arrays , Reliable , Parallel Secondary Storage, Phd Thesis, Dept of Computer Science, University of California, Berkeley,1988.

70. Gary, M.. K., Clifford, G. Lau., Foundations of Dependable Computing , Models and Frameworks for Dependable Systems, Prentice Hall, Englewood Cliffs, New Jeresey,1994.

71. Geist ,R., Trivedi ,K.,Reliability Estimation of Fault Tolerant Systems :Tools and Techniques, IEEE Transactions on Computers, Vol.40, 1990, pp.52-61.

72. Geoghegan, S. J., Avresky, D., Design , Verification, and Validation of Self Checking Software Components, Proceedings of the International Phoenix Conference on Computers and Communications, 1996, pp. 420-426.

73. George, D. Alan., Hawkes, W. Lois., Microprocessor Based Parallel Architecture for Reliable Digital Signal Processing Systems, CRC Press, London,1992.

74. George, F.C., Evolution of Fault Tolerant Computing at AT&T , Proceedings of the One-day Symposium of Evolution of Fault Tolerant Computing,1986, pp.27-37.

75. Giandomenico, F. D., Strigini, L., Adjudicators for Diverse-Redundant Components, Proceedings of the 9th Symposium on Reliable Distributed Systems,1990, pp. 114-123.

76. Gilles, M ., Michel, B., Nadine P., Bruno R., Lessons from FTM, An Experiment in the Design and Implementation of a Low-Cost Fault-Tolerant System, IEEE Transactions on Reliability, Vol. 45, 1996, pp.332-339.

77. Gilol, W. K, Gueth. R., Concept and Realization of a High Performance Data Type Architecture, International Journal of Computer and Information Sciences, Vol.11,1982, pp. 25-54.

78. Goel, A.. L., Okumunto, K., Time-Dependent Error Detection Rate Model for Software and other Performances Measures , IEEE Transactions on Reliability, Vol.28, 1979, pp.206-211.

79. Gopal, G., Griffeth, N .D., Software Fault Tolerance in Telecommunication Systems, Journal of Operating Systems Review, Vol. 25, 1991, pp.112-116.

80.Goseva, P. K.,Grnarov, A., N-Version Programming with Majority Voting Decision Dependability Modelling and Evaluation, Journal of Microprocessing and Microprogramming ,Vol.38, 1993, pp.811-818.

81.Goseva, P. K ., Grnarov, A., Perform Ability and Reliability Modeling of N Version Fault Tolerant Software in Real Time Systems, New Frontiers of Information Tech, Proceedings of the Euromicro Conference, 1997, pp.532-539.

82. Goseva, P. K., Grnarov, A., Performability Modelling of N Version Programming Technique, Proceedings of the International Symposium on Software Reliability Engineering,1995, pp.209-218.

83. Goyal, A ., Lavenburg, S. S., Trivedi, K. S., Probabilistic Modeling of Computer System, Availability, Journal of Annals of Operations Research,1987, pp.285-306.

84. Goyal, A.,Carter, W.C., Ede, S. S., Lavenberg, S. S.,Trivedi, K. S., The System Availability Estimator, Proceedings of the IEEE Symposium on Fault Tolerant Computing, 1986, pp.84-89.

85. Gray, J., A Census of Tandem System Availability between 1985 and 1990, IEEE Transactions on Reliability, Vol.39,1990, pp.409-418.

86. Gudae, D. D., Lee, C., Transactions on Fault Tolerant Power Controller, Proceedings

of the International Society Energy Conversion Engineering Conference, Vol. 3,1988, pp. 441-444.

87. Hansen, J.P ., Siewiorek, D.P., Models for Time Coalescence in Event Logs, Proceedings of the 22nd International Symposium on Fault - Tolerant Computing, 1992 , pp.221-227.

88. Hassanein, H. A., Edward, J. M., Weighted Coverage In Fault-Tolerant Systems, Proceedings of the Annual Reliability and Maintainability Symposium, 1987, pp. 187-191.

89. Hecht, H., Fiorentino, E., Reliability Assessment of Spacecraft Electronics, Proceedings of the Annual Reliability and Maintainability Symposium,1987, pp.341-346.

90. Heidmann , K. D., Deterministic Reliability Modeling of Dynamic Redundancy, IEEE Transactions on Reliability, Vol 41,1992, pp.567-571.

91. Heimann, D., Mittal, N.,Trivedi, K., Availability and Reliability Modeling for Computer Systems , Advances in Computers, Vol.31,1990, pp.175-233.

92. Hitt, E. F., Real time Fault Tolerant Software in Distributed Avionics Systems Architectures using Digital Data Buses, Proceedings of the International Society Energy Conversion Engineering Conference, 1986, pp. 523-528.

93. Hivaicka, J., Kotteck, E., Fault Model for TTL circuits , Journal on Digital Processors, Vol. 19, 1980, pp.169-180.

94. Hobley, P., Jesty, P., Recovery Block Scheme for a VAX11/750 running UNIX, Software Practice and Experience , Vol. 18, 1988, pp.561-581.

95. Holding, D. J., Hill, M .R., Carpenter, G. F., Design of Distributed , Software Fault Tolerant, Real time Systems Incorporating Decision Mechanisms, Journal of Microprocessor and Microprogramming, Vol. 24,1988, pp. 801-806.

96. Holding, D. J., Software Fault Tolerance, Proceedings of the IEE Colloquium on Fault Tolerant Techniques, London,1990, pp.6/1-6/9.

97. Hseuh, M .C., Iyer, R., Trivedi, K. S., Performability Modeling Based on Real Data , A case study , IEEE Transactions on Computers , Vol. 37,1988, pp.478-484.

98. Huang, Y., Kintala, C. M. R., Bernstein, L., Wang, Y., Components for Software Fault Tolerance and Rejuvenation, AT&T Technical Journal, Vol.75, 1996, pp.29-37.

99. Huang, Y., Kintala, C., Software Implemented Fault Tolerance Technologies and Experience, Proceedings of the Fault Tolerant Computing Symposium ,1993, pp. 2-9.

100.Ibe, O. C., Wein ,A.S., Availability of Systems with Partially Observable Failures , IEEE Transactions on Reliability , Vol. 41, 1992, pp.92-96.

101.Iyer, R. K., Faults, Symptoms , and Software Fault Tolerance in the Tandem Guardian90 Operating System, Proceedings of the Fault Tolerant Computing Symposium, 1993, pp. 20- 29.

102.Iyer, R. K., Rosetti, D. J., Hseuh, M .C., Measurement and Modeling of Computer Reliability as Affected by Systems Activity, Journal of ACM Transactions on Computer Systems, 1986, pp. 214-237.

103.Iyer , R . K., Young ,L.T., Iyer , P. V. K., Automatic Recognition of Intermittent Failures: An Experimental Study of Field Data, IEEE Transactions on Computers, Vol.39,1990, pp.525-537.

104.Jagannathan, R., Ascroft, E. A ., Fault Tolerant Aspects of the Education and Model Architecture ,Proceedings of the International Society Energy Conversion Engineering Conference, 1986, pp. 515-522.

105.James, L. E., Angus, J. E.,Bowen, J. B., McDaniel, J., Combined Hardware/Software Reliability Models, Reliability Analysis Center, Griffins Air Force Base, Rome , New York, 1982.

106.Jaynarayan, H . L., A Byzantine Resilient Fault Tolerant Computer for Nuclear Power Plant Applications, Proceedings of the16th Fault Tolerant Computing Symposium, 1986, pp.338-343.

107.Jie, X., Randell ,B., Romanovsky, A., Rubira , C., Stroud, R. J., Wu, Z., Fault Tolerance in Concurrent Object Oriented Software through Coordinated Error Recovery, Proceedings of the Fault Tolerant Computing Symposium ,1995, pp. 499-508.

108.Johnson, B.W., Design and Analysis of Fault Tolerant Digital Systems, Addison-Wesley, Reading, Massachusetts, 1989.

109.Jurgen, D., On the Modeling of Workload Dependent Memory Faults, Proceedings of the 20th Symposium on Fault Tolerant Computing ,1990, pp. 348-354.

110.Kamal, S., Page , V.C ., Intermittent Faults , A model and a Detection Procedure , IEEE Transactions on Computers , Vol.24, 1974, pp. 241-245.

111. Kanoun ,K., Kaaniche, M., Beounes ,C., Laprie, J.-C., Arlat, J., Definition and Analysis of Hardware and Software Fault Tolerant Architectures , IEEE Transactions on Computers, Vol. 23, 1990, pp. 39-51.

112. Kanoun ,K., Kaaniche, M., Beounes ,C., Laprie, J. C., Arlat, J., Reliability Growth of Fault Tolerant Software, IEEE Transactions on Reliability, Vol. 42, 1993, pp. 223- 246.

113. Kanoun, K., De Bastos, M ., A Method for Software Reliability Analysis and

Prediction , An Application to the Tropico-R Switching System, IEEE Transactions on Software , Vol.17, 1991, pp.334-344.

114. Kant, K., Software Fault Tolerance in Real time Systems, Journal of Information Sciences, Vol. 42, 1987, pp. 255-282.

115. Kenney, G.G., Estimating Defects in Commercial Software during Operational Use, IEEE Transactions on Reliability, Vol. 42, 1993, pp.107-115.

116. Kim, H. C., Nair, V. S. S., Application Layer Software Fault Tolerance for Distributed Object Oriented Systems , Proceedings of the IEEE Computer Society's International Computer Software and Applications Conference ,1995, pp. 199-204.

117. Kim, H . C., Nair, V. S. S., Software Fault Tolerance for Distributed Object Based Computing, Journal of Systems and Software , Vol.39,1997 , pp.103-117.

118. Kim, K. H., Welch, H .O., Distributed Execution of Recovery Blocks: An Approach for Uniform Treatment for Hardware and Software Faults in Real-Time Applications, IEEE Transactions on Computers , Vol.38, 1989, pp.626-636.

119. Kline, M. B., Software and Hardware R&M, What are the Differences ?, Proceedings of the Annual Reliability and Maintainability Symposium , 1980, pp.179-185.

120. Koren, I., Kohavi, Z., Diagnosis of Intermittent Faults in Combinatorial Networks , IEEE Transactions on Computers , Vol 27, 1977, pp.1154-1158.

121. Knight, C., Ammann, J., Design Fault Tolerance, Journal of Reliability Engineering and System Safety , Vol 32,1991, pp.25-49.

122. Kreutzfeld, R. J., Nesse, R. E., Methodology for Cost Effective Software Fault Tolerance for Mission- Critical Systems, IEEE Aerospace and Electronic Systems

Magazine, Vol.12, 1997, pp.25-30.

123. Kreutzfeld , R. J., Nesse, R. E., Methodology for Cost Effective Software Fault Tolerance for Mission Critical Systems, AIAA/Proceedings of the IEEE Digital Avionics Systems Conference, 1996, pp.93-100.

124. Krishna, C. M., Singh, D. A., Modeling Correlated Transient Failures in Fault Tolerant Systems, Proceedings of the International Symposium on Fault Tolerant Computing,1989, pp. 374-381.

125. Kumar, A., Srivastava, A., Singru, A. G., Robust and Distributed Genetic Algorithm for Ordering Problems, Proceedings of the IEEE International Symposium on High Performance Distributed Computing Distributed Computing., 1996, pp. 253-262.

126. Kwiat, K. A ., Debany, W. H. Jr., Hariri , S., Software Fault Tolerance using Dynamically Reconfigurable FPGAs, Proceedings of the IEEE Great lakes Symposium on VLSI., 1996, pp.39- 42.

127. Lala, J. H., Alger , L. S., Hardware and Software Fault Tolerance , A Unified Architectural Approach, Digest of papers, Proceedings of the Fault Tolerant Computing Symposium., 1988, pp. 240-245.

128. Lala, J.H., Hopkins , A.L., Survival and Dispatch Probability Models for the FTMP Computer, Proceedings of the International Symposium of Fault-Tolerant Computing,1978, pp. 37-43.

129. Lala, P. K., Fault Tolerant and Fault Testable Hardware Design, Prentice Hall, Englewood Cliffs, New Jersey,1985.

130. Lala, J.H ., Harper, R. E., Architectural Principles for Safety Critical Real-time

Applications, Proceedings of the IEEE , Vol.82, 1994, pp. 25-40.

131. Lala, P. K., Missen ,I. J., Method for the Diagnoses of a Single Intermittent Fault in Combinatorial Logic Circuits , IEEE Journal Computers and Digital Techniques ,1979, pp.187- 190.

132. Laprie, J. C., Kanoun, K., X-ware Reliability and Availability Modeling , IEEE Transactions Software Engineering , Vol.1, 1992, pp.130.

133. Laprie, J. C., Arlat, J., Beounes, C., Kanoun, K., Hourtelle, C., Hardware and Software Fault Tolerance, Definition and Analysis of Architectural Solutions, Digest of papers-Fault Tolerant Computing Symposium, 1987, pp.116-121.

134. Laprie, J. C., Arlat, J., Beounes, C., Kanoun, K., Definition and Analysis of Hardware and Software Tolerant Architectures, IEEE Transactions on Computers, Vol. 40,1990, pp. 39-51.

135. Laura, L., Automated Fault Tolerance Evaluator for System Design, Simulation , and Analysis, Proceedings of the 9th Digital Avionic System Conference,1990, pp. 360-365.

136. Lee, P. N., Tamboli, A., Concurrent Correspondent Modules , a Fault Tolerant 5Ada Implementation, Proceedings of the 8th Annual International Phoenix Conference on Computers and Communication , 1989, pp. 300-304.

137. Lee, I., Iyer, R. K., Software Dependability in the Tandem Guardian system, IEEE Transactions on Software Engineering , Vol.21,1995, pp.455-467.

138. Lee, K. W., Suh, J. J.,Chun, Y. S., Reliability Analysis of the TDX-1 Switch Network Considering its Performance, Proceedings of the Reliability and Quality Control Conference ,1987, pp. 58-61.

139. Levendel, Y., Quality and Reliability Estimation of a Time Dependent Model with Controllable Testing Coverage and Repair Intensity, Proceedings of the Fourth Israel Conference on Computer Systems and Software Engineering ,1989, pp.175-181.

140. Levendel, Y., Defects and Reliability Analysis of Large Software Systems: Field Experience, Proceedings of the 19th Fault Tolerant Computing Symposium, 1989, pp.238-244.

141. Leveson, N. G., Stolzy ,J. L., Safety Analysis using Petri Nets, IEEE Transactions on Software Engineering , Vol.13, 1987, pp.386-397.

142. Li, V. O., Silvester, J. A., Performance Analysis of Networks with Unreliable Components, IEEE Transactions on Communication, Vol. 32, 1984, pp.105-110.

143. Liao, M., Guo, F., Ada Language and Software Fault Tolerance, Proceedings of the 10th IEEE Region Conference on Computer, Communication, Control and Power Engineering, 1993, pp. 395-398.

144. Liestman, A. L., Campbell, R. H., Fault Tolerant Scheduling problem, IEEE Transactions on Software Engineering , Vol.11,1986, pp.1089-1095.

145. Long, D. D. E.,Carroll, J. L.,Park, C. J., A Study of the Reliability of Internet sites, Proceedings of the10th Symposium on Reliable Distributed Systems, 1991, pp. 177-186.

146. Lyu, M. R., He, Y., Improving the N-version Programming Process through the Evolution of Design Paradigm, IEEE Transactions on Reliability, Vol.42, 1993, pp.179-189.

147. Lyu, M. R., Yu, J., Keramidas, E., Dalal, S., Analyzer for Reducing Module Operational Risk, Proceedings of the 25th International Symposium on Fault Tolerant Computing System,1995, pp137-142.

148. Makam, S. V., Avizienius, A., Aries 81, A Reliability and Life Cycle Evaluation Tool for Fault Tolerant Systems, Proceedings of the IEEE Fault Tolerant Computing Systems,1982, pp. 267-274.

149. Malaiya, Y. K., Su, H .Y . S., A Survey of Methods for Intermittent Fault Analysis, Proceedings of the International Conference of Computing and Information,1979, pp. 577-584.

150. Malaiya, Y. K., Von, M ., Anneliese, S., An Examination of Fault Exposure Ratio. IEEE Transactions on Software, Vol.19, 1993, pp.1087-1094.

151. Malaiya, Y. K., Linear Correlated Intermittent failures, IEEE Transactions on Reliability, Vol.31,1981, pp.211-215.

152. Marcelo, L., Bernard, C., A Reliable Fail-Safe System, IEEE Transactions on Computers, Vol. 47,1998, pp.236-241.

153. Markham, K. C., Milliken, R, A., Software Fault Tolerance for a Flight Control System, Proceedings of the First International Conference on the use of Programmable Electronic Systems in Safety, No.314,1989, pp.18-22.

154. McAllister, D. F., Sun ,C. E., Vouk, M . A., Reliability of Voting in Fault Tolerant Software Systems for Small Output Spaces, IEEE Transactions on Reliability, Vol.39,1990, pp.524-534.

155. McCluskey, E. J., Wakerly, F. J., A Circuit for Detecting and Analyzing Temporary Failures, Proceedings of the Computer Conference ,1981, pp. 317-321.

156. Mendiratta, V. B., Assessing the Reliability Impacts of Software Fault Tolerance Mechanisms, Proceedings of the International Symposium on Software Reliability

Engineering,1996, pp.256-265.

157. Meyer, J. F., Schlichting, R. D., Limits of Parallelism in Fault-Tolerant Multi processors, Proceedings of the Dependable Computing for Critical Applications , 1992, pp 135-154.

158. Meyer, J. F.,Schlichting, R. D., Impact of Correlated Failures on Dependability in a VAX cluster System ,Proceedings of the Dependable Computing for Critical Applications-2, Vol.6, 1991, pp.175-194.

159. Micheal, O. F., Risk Assessment of Computer Controlled Systems , IEEE Transactions on Software Engineering, Vol .11, 1985, pp.125-129.

160. Moulding, M. R., Architecture to Support Software Fault Tolerance and an Evaluation of its Performance in a Naval Command and Control Application, IEE Colloquium on Performance Measurement and Prediction, 1986/28, 1-1.4.

161. Moulding, M. R., Techniques for Achieving Software Fault Tolerance, IEE Colloquium on High Integrity Systems-Theory and Practice , 1986/112, 1-2.11.

162. Mirko ,V., Daniel, M.., Reliability Evaluation and Optimization of Redundant Dynamic Systems, IEEE Transactions on Reliability, Vol.34,1985, pp.171-174.

163. Murugesan, S.,Dependable Software through Fault Tolerance, Proceedings of the International Conference TENCON (Trends in Electronic Conference, 1989, pp.391-398.

164. Myron, H., Jeffrey, A ., Herbert, H.,Kim, K. H., A Distributed Fault Tolerant A Architecture for Nuclear Reactor and other Critical Process Control, Proceedings of the Fault Tolerant Computing Symposium,1991, pp.462-469.

165. Nancy, G. L., Janice, L. S., Safety Analysis of Ada Programs Using Fault Trees,

IEEE Transactions on Reliability, Vol,32, 1983, pp.479-488.

166. Nakagawa, T., Osaki, S., Markov Renewal Processes with some Non-Regeneration Points and their Applications to Reliability Theory, Microelectronics and Reliability, Vol-15,1976, pp. 633- 636.

167. Neches, P. M., Hardware Support for Advanced Data Management Systems, IEEE Transactions on Computers, Vol.17,1984, pp. 29-40.

168. Nelson, V. P., Fault Tolerant Computing , Fundamental Concepts, IEEE Transactions on Computers, Vol. 39 , 1990, pp. 29-45.

169. Nikolos, D.,Vergos, H. T., On the yield of VLSI Processors with On-Chip CPU cache, Proceedings of the Second European Dependable Computing Conference, 1996, pp.214-230.

170. Nitin H. V., Dhiraj, K. P., Fault Tolerant Design Strategies for High Reliability and Safety, IEEE Transactions on Computers , Vol. 42, 1993, pp.1195-1205.

171. Noushin, A., Oded, B., Michal, C., Optimal Design Of Large Software Systems Using N-Version Programming, IEEE Transactions on Reliability , Vol 43, 1994, pp.344-350.

172. Ozaki, B. M., Fernandez, E. B., Implementation of Software Fault Tolerance in the Intel 80286 Processor, Proceedings of the IEEE SouthEastcon Conference,1988, pp. 75-80.

173. Ozaki, B. M., Fernandez, E. B., Gudes, E., Software Fault Tolerance in Architectures with Hierarchical Protection Levels, IEEE Microcomputer, Vol.8,1988, pp. 30-43.

174. Park, E. K., Software fault tolerance in real time embedded systems ,Proceedings of the17th Annual ACM Comput Science Conference, 1989, pp. 431.

175. Pecht, M., Editor., Product Reliability Maintainability and Sportability Handbook, CRC Press Inc, Boca Raton, Florida, 1995.

176. Pedar, A., Sharma, V. V. S., Phased Mission Analysis for Evaluating the Effectiveness of Aerospace Computing System, IEEE Transactions on Reliability, Vol.30, 1981, pp.429-437.

177. Pham, H., Upadhyaya, S., Reliability Analysis of a Class of Fault Tolerant Systems, IEEE Transactions on Reliability, Vol. 38,1989, pp.333-337.

178. Pradhan ,D. K., Ed., Fault-Tolerant Computing Theory and Techniques, Vol .I and II, Prentice Hall , Inc Englewood Cliffs, New Jersey, 1986.

179. Pullum., Laura, L., New Adjudicator for Fault Tolerant Software Applications Correctly Resulting in Multiple Solutions, Proceedings of the IEEE/AIAA 12th Digital Avionics System Conference , 1993, pp.147-152.

180. Purtilo, J. M., Jalote, P., System for Supporting Multi Language Versions for Software Fault Tolerance, Proceedings of the Fault Tolerant Computing Symposium, 1990, pp. 268- 290.

181. Raghunandan, K., Coakley, F. P., Evans, B.G., Fault Tolerance of On-Board Digital Signal Processing Circuits, International Journal of Satellite Communications, Vol. 9,1991, pp.399- 413.

182. Robert, G., Mark, S., Micheal, B., Ultrahigh Reliability Estimates for Systems Exhibiting Globally Time-Dependent Failure Processes, Proceedings of the 19th Fault Tolerant Computing Symposium ,1989, pp.152-158.

183. Robert, G., Kishor, T., An Analytic Treatment of the Reliability and Performance of

Mirrored Disk Subsystems, Proceedings of the Fault Tolerant Computing Symposium, 1993, pp.442-450.

184. Ronald, R., Modeling and Testing a Critical Fault-Tolerant Multi-Process System, Proceedings of the 25th Fault Tolerant Computing Symposium, 1995, pp.516-521.

185. Saglietti, F., Strategies for the Achievement and Assessment of Software Fault Tolerance, International Federation of Automatic Control Symposia Series- Proceedings of the Triennial World Congress, Vol. 4, 1990, pp.303-308.

186. Sahner ,R.A., Reliability Modeling Using Sharpe, IEEE Transactions on Reliability, Vol.36,1987, pp.186-193.

187. Sarma, V. V. S., Survey of Software Dependability, Sadhana , Vol. 11,1987, pp.23- 48.

188. Savir, J., Optimal Random Testing of Single Intermittent Failures in Combinatorial Circuits, Proceedings of the International Symposium of Fault Tolerant Computing,1977, pp. 180-185.

189. Scott ,R. K., Gault, J. W., McAllister, D. F., Fault Tolerant Software Reliability Modeling ,IEEE Transactions on Software Engineering, Vol-13,1987, pp. 582-592.

190. Shakil, A., Reliabilty of k-out-of -n: Systems with Imperfect Fault-Coverage, IEEE Transactions on Reliability, Vol.43, 1994, pp.101-106.

191. Shatz, S. M., Superprocesses: A Distributed Program Configuration Concept Supporting Hardware Fault Tolerance, Journal of Computer Systems Science and Engineering, Vol. 3, 1988, pp. 3-12.

192. Sheth, A., Fault Tolerance in a very Large Database System, A Strawman Analysis, Proceedings of the International Conference on Distributed Computing Systems, 1989,

pp. 227-236.

193. Shimeall, T. J., Leveson, N. G., An Empirical Comparison of Software Fault Tolerance and Fault Elimination, IEEE Transactions on Software Engineering, Vol.17, 1991, pp.173-182.

194. Sullivan, D.,Gregory, F., Masson, G. M.,Using Certification Trails to Achieve Fault Tolerance, Proceedings of the Fault Tolerant Computing Symposium , 1990, pp. 423- 431.

195. Shimeall, T. J., Leveson, N. G., Empirical Comparison of Software Fault Tolerance and Fault Elimination, Proceedings of the Second Workshop Software Test Verification Anal.,1988, pp.180-187.

196. Shokri, E. H., Tso, K. S., Ada95 Object Oriented and Real time Support for Development of Software Fault Tolerance Reusable Components, Proceedings of the Workshop on Object Oriented Real time Dependable Systems, 1996, pp.93-100.

197. Shokri, E. H ., Tso, K . S. Dziegiel, R., Development of Software Fault Tolerant Applications with Ada95 Object Oriented Support, Proceedings of the IEEE National Aerospace and Electronics Conference , Vol.2,1996, pp. 519-526.

198. Shooman, M. L., Laemmel, A. E.,Simplification of Markov Models by State Merging, Proceedings of the Annual Reliability and Maintainability Symposium,1987, pp.159-163.

199. Shunji, O., Toshihiko, N., Reliability Evaluation of Some Fault Tolerant Computer Architectures, Springer-Verlag, New York, 1980.

200. Siegrist, K., Reliability of Systems with Markov Transfer of Control II, IEEE Transactions on Software Engineering, Vol.14, 1988, pp. 1478-1480.

201. Siegrist, K., Reliability of Systems with Markov Transfer of Control, IEEE Transactions on Software Engineering , Vol.14, 1988, pp.1049-1053.

202. Siewiorek, D.P., Swarz R.S., The Theory and Practice of Reliable Systems Design, Digital Press, Bedford, Massachusetts, 1982.

203. Simon, D., Hourtolle, C., Bionde, H., Bernelas, J., Duverneuil, P., Gallet, S., Vielcanet,P., A Software Fault Tolerance, Proceedings of the Fault Tolerant Computing Symposium, 1990, pp. 28- 35.

204. Smotherman, M . K., Zemoudeh, K ., A Nonhomogeneous Markov Model for Phased Mission Reliability Analysis, IEEE Transactions on Reliability, Vol. 38,1989, pp. 585.

205. Stark, G.E., Dependability of Integrated Hardware/ Software Systems, IEEE Transactions on Reliability, Vol.36,1987, pp.440-444.

206. Stifler, J. I., Robust Detection of Intermittent Faults , Proceedings of the International Symposium on Fault Tolerant Computing,1980, pp. 216-218.

207. Su, S. Y. H., Koren , I ., Malaiya , K. Y., A Continuous Parameter Markov Model and Detection Procedures for Intermittent faults, IEEE Transactions on Computers., Vol.28, 1978, pp. 567-569.

208. Sumita.,Ushio., Masuda., Yasushi., Analysis of Software Availability/Reliability under the Influence of Hardware Failures, IEEE Transactions on Software Engineering, Vol.12, 1986, pp .32-41.

209. Suri, G., Huang, Y., Wang, Y. M.., Fuchs, W. K., Kintala , Chandra., Implementation and Performance Measurement of the Progressive Retry Technique, Proceedings of the International Conference on Computer Performance and Dependability Symposium,

1995, pp. 41- 48.

210. Tamir, Y., Gafni, E., A Software-Based Hardware Fault Tolerance Scheme For Multi Computers, Proceedings of the International Conference on Parallel Processing, 1987, pp. 117-120.

211. Tang, D., Iyer , R.K., Analysis and Modeling of Correlated Failures in Microcomputer Systems, IEEE Transactions on Computers, Vol. 42, 1993, pp.62-75.

212. Tang,D., Iyer ,R.K., Subramani, S., Failure Analysis and Modelling of a VAXcluster System, Proceedings of the 20th International Symposium on Fault Tolerant Computing, 1990, pp 244-251.

213. Tasar, O., Tasar, V., A Study of Intermittent Faults in Digital Computers , Proceedings AFIPS Conference, 1977, pp. 807-811.

214. Taylor, D., Morsan, J., David, E. B., James, P., Redundancy in Data Structures in Improving Software fault tolerance ,Redundancy in Data structures, IEEE Transactions on Software Engineering, Vol-6, 1980, pp. 585-594 & pp.595-602.

215. Tian, J., Integrating Time Domain and Input Domain Analyses of Software Reliability using Tree Based Models, IEEE Transactions on Software Engineering, Vol.21, 1995, pp. 945-958.

216. Tian, J., Integrating Time Domain and Input Domain Analyses of Software Reliability using Tree Based Models, IEEE Transactions on Software Engineering, Vol.20, 1994, pp 740-747.

217. Tomek., L. A., Muppala, J. K., Trivedi, K. S., Modeling Correlation in Software Recovery Blocks, IEEE Transactions on Software Engineering , Vol. 19, 1993, pp. 1071-1080.

218. Tryell, A. M ., Increasing Software Reliability of Distributed Systems with Occam, Proceedings on Computers and Applications, 1987, pp. 249-254.

219. Tso, K. S., Hecht, M., Marzwell, N, I., Fault Tolerant Robotic System for Critical Applications , Proceedings of the International Conference on Robotics and Automation, Vol.3,1993, pp.691-696.

220.Tyrrell, A. M ., Sillitoe, I . P. W., Evaluation of Fault Tolerant Software Structures for Parallel Systems in Industrial Control, Proceedings of the IEE Conference on Control, Vol.1,1991, pp.393- 398.

221. Tyrrell, A . M .,Carpenter, G. F.,Specification and Design of Atomic Actions for Fault Tolerant Concurrent Software, Journal of Microprocessing and Microprogramming, Vol.35,1992, pp.363- 368.

222. Vaidya, N. H. S., Krishna, C. M., Trade-offs in Developing Fault Tolerant Software IEE Proceedings-E Computers and Digital Techniques, Vol. 140, 1993, pp.320-326.

223. Voas, J. M., Dynamic Test Ability Analysis for Assessing Fault Tolerance, Journal of High Integrity Systems, Vol. 1, 1994, pp.171-178.

224. Vouk, M. A., Helsabeck, M. L., Tai, K. C., McAllister, D. F., On Testing of Functionally Equivalent Components of Fault Tolerant Software, Proceedings of the IEEE Computer's Society's International Computer Software and Applications Conference, 1986, pp. 414-419.

225. Wakefield, G. S., Dziegiel, R. Jr., Pullum, L. L., Centurion Software Fault Tolerance Design and Analysis Tool, Proceedings of the Annual Conference on Computer Assurance, 1994, pp. 300-304.

226. Walter, A. B., Jai, M., Disk Array Storage System Reliability, Proceedings of the Fault

Tolerant Computing Symposium, 1993, pp.432-441.

227. Wang, S. Y., Meyer, S. E., Saxena, D. R., Lai, M . Lee., Applications of Software Fault Tolerance Testing Methodology ,Proceedings of the IEEE Global Telecommunications Conference, Vol.1,1996, pp. 670-674.

228. Weber, G. G., Performability of Computer Systems, Proceedings of the Reliability and Quality Control Conference, 1987, pp.156-159.

229. Wein ,A. S., Sathaye, A., Validating Complex Computer System Available Models, IEEE Transactions on Reliability, Vol 39, 1990, pp.468-479.

230. Wensley, J. H., A Review of Techniques for Achieving High Reliability Hardware and Software, Proceedings of the International Conference and Exhibition, 1982, pp.1023-1031.

231. Wild, C., Automating Software Fault Tolerance, Journal of Spacecrafts and Rockets, Vol. 24, 1987, pp. 86-89.

232. Wojcieh, M., Fault Tolerance Issues in Local Area Networks, Proceedings of the Reliability and Quality Control Conference, 1989, pp. 61-64.

233. Wu, J., Wang, Y., Fernandez, E. B., Uniform Approach to Software and Hardware Fault Tolerance, Journal of Systems and Software, Vol. 26,1994, pp.117-127.

234. Wu, J., Software Fault Tolerance using Hierarchical N-Version Programming, Proceedings of the IEEE SouthEastcon Conference, 1991, pp. 243-247.

235. Wu, J., Wang, Y., Fernandez, E. B., Uniform Approach to Software and Hardware Fault Tolerance, Journal of Systems and Software, Vol.26, 1994, pp.117-127.

236. Xu, J., The t(n-1)- Diagnosability and its Applications to Fault Tolerance, Proceedings

of the Fault Tolerant Computing Symposium,1986, pp.496-503.

237. Xu, J., Randell, B., Software Fault Tolerance, t/(n minus 1)- Variant Programming, IEEE Transactions on Reliability, Vol. 46, 1997, pp.60-68.

238. Xu, J., Randell, B., Rubira, C. C. M.. F., Stroud, R. J., Toward an Object Oriented Approach to Software Fault Tolerance, Proceedings of the Conference on Fault Tolerant Parallel and Distributed Systems, 1995, pp. 226-233.

239. Yennum, H., Pankaj, J., Availability Analysis of the Primary Site Approach for Fault Tolerance, Proceedings of the 8th Symposium on Reliable Distributed Systems,1989, pp. 130-136.

240. Zeil ,S. J., Perturbation Techniques for Detecting Domain Errors, IEEE Transactions on Software , Vol. 5, 1989, pp. 737-746.

241. Zhuang, W. J., Xie, M., Design and Analysis of some Fault Tolerance Configurations based on a Multiple Principle ,Journal of Systems and Software, Vol.25, 1994, pp.101-108.

APPENDIX A

$A=k_2+k_3+k_4$

$A_1=k_3k_4+k_2k_4+k_3k_2$

$A_2=k_3k_2k_4$

$A_3=k_1k_2k_3k_4$

$A_4=k_6+k_5+k_2+k_3+k_4$

$A_5=k_3k_4+k_2k_5+k_4k_6+k_4k_2+k_3k_2+k_3k_5+k_6k_5+k_3k_6+k_6k_2+ k_4k_5$

$A_6=k_4k_2k_5+ k_4k_6k_5+k_4k_6k_2+k_3k_4k_2+k_3k_6k_2+k_3k_6k_5+k_2k_6k_5+k_3k_2k_5+k_3k_4k_5+k_3k_4k_6$

$A_7= k_3k_4k_2k_5+k_3k_4k_6k_5+k_4k_2k_6k_5+k_3k_2k_6k_5+k_3k_4k_6k_2$

$A_8=k_4k_2k_1k_6k_5+k_3k_2k_1k_6k_5+k_3k_4k_1k_6k_2+ k_3k_4k_2k_6k_5+k_3k_4k_1k_6k_5$

$A_9=k_1k_6k_5k_2k_3k_4$

$A_{10}=k_1+k_2+k_3+k_4$

$A_{11}= -3\mu_1\lambda-2\mu_2\lambda_v+k_2k_1+k_3k_1+k_3k_4+k_2k_4+k_3k_2+k_1k_4$

$A_{12}= -3\mu_1\lambda k_4-3\mu_1\lambda k_3-2\lambda_v k_3\mu_2-2\lambda_v k_2\mu_2+k_3k_1k_4+k_2k_1k_4+k_3k_1k_2+k_3k_2k_4$

$A_{13}= -3\lambda\mu_1k_4k_1-2\mu_2\lambda_v k_3k_2+k_1k_2k_3k_4$

$A_{14}=k_6+k_5+k_1+k_2+k_3+k_4$

$A_{15}=k_3k_4+k_3k_1+k_2k_5+k_6k_1+k_4k_6+k_2k_1+k_4k_2+k_3k_2+k_3k_5+k_6k_5+k_3k_6+k_4k_1+k_6k_2+$

$k_4k_5-3\lambda\mu_1+k_1k_5-8\mu_2\lambda_v$

$A_{16}=k_4k_2k_1-5\lambda v\mu 2k_1+k_6k_2k_1+k_4k_2k_5+k_3k_2k_1+ k_4k_6k_5+k_4k_6k_2+k_3k_4k_2+k_3k_6k_2+k_3k_6k_5+$

$k_4k_1k_5+k_4k_1k_6-8\lambda_v k_4\mu_2+k_2k_1k_5+k_1k_6k_5+ k_2k_6k_5+k_3k_1k_5+k_3k_1k_6+k_3k_2k_5+k_3k_4k_5+$

$k_3k_4k_6+k_3k_4k_1-5dk_3\mu_2-3\lambda_v\mu_2k_5-6\lambda_v k_6\mu_2-5\lambda_v k_2\mu_2-3\lambda\mu_1(k_5+k_4+k_3+k_6)$

$A_{17}=15\mu_2{}^2\lambda_v{}^2-3\lambda\mu_1\mu_2\lambda_v-3\lambda k_3c_1k_5-3\lambda k_3c_1k_6-3\lambda\mu_1k_6k_5-3\lambda k_3\mu_1k_5-3ak_4\mu_1k_6-3ak_3k_4\mu_1+$

$k_4k_2k_1k_5-2\lambda_vk_3k_1\mu_2-6\lambda_vk_4\mu_2k_6-3\lambda_vk_3k_6c_2-3\lambda_v\mu_2k_6k_2-3\lambda_vk_4c_2k_5-5\lambda_v\mu_2k_3k_2-3\lambda_v\mu_2k_1k_5-$

$3\lambda_v\mu_2k_1k_6-2\lambda_vk_2k_1\mu_2-3\lambda_v\mu_2k_6k_5+k_4k_3k_2k_1-5\lambda_vk_3k_4\mu_2-5\lambda_vk_4\mu_2k_1+k_3k_4k_2k_5+k_3k_2k_1k_5+$

$k_3k_4k_6k_5+k_3k_1k_6k_5+k_4k_2k_6k_5+k_2k_1k_6k_5+k_3k_1k_6k_2+k_3k_2k_6k_5+k_3k_4k_6k_2+k_4k_1k_6k_2+$

$k_4k_1k_6k_5+k_3k_4k_1k_5-5\lambda_vk_4k_2\mu_2+k_3k_4k_1k_6$

$A_{18}=k_4k_2k_1k_6k_5+k_3k_2k_1k_6k_5+k_3k_4k_1k_6k_2+k_3k_4k_2k_6k_5-3\lambda\lambda_vk_4\mu_1\mu_2-3\lambda k_4\mu_1k_6k_5-$

$3\lambda k_3\mu_1k_6k_5-3ak_3k_4\mu_1k_6-3\lambda k_3k_4\mu_1k_5+9\lambda^2\mu_2{}^2k_6+15\lambda_v{}^2k_4\mu_2{}^2-2\lambda_vk_3k_4k_1\mu_2-$

$3\lambda vk_4\mu_2k_1k_5-3\lambda vk_4\mu_2k_1k_6-3\lambda vk_4\mu_2k_6k_2-2\lambda_v\mu_2k_1k_3k_2-3\lambda_v\mu_2k_1k_6k_5-3\lambda_vk_3k_2k_6\mu_2-$

$5\lambda_vk_4\mu_2k_3k_2-3\lambda_vk_3k_4k_6\mu_2-2\lambda_vk_4k_2k_1\mu_2-3\lambda_vk_4\mu_2k_6k_5+k_3k_4k_2k_1k_5+6\lambda\lambda_vk_3\mu_1\mu_2-$

$9\lambda\lambda_v\mu_2k_6k_1+k_3k_4k_1k_6k_5+6\lambda_v{}^2\mu_2{}^2k_1$

$A_{19}=-3\lambda_vk_3k_4k_2k_6\mu_2-2\lambda_vk_4\mu_2k_1k_3k_2-k_3k_4k_2k_1k_6k_5-3\lambda k_3k_4\mu_1k_5k_6+6\lambda\lambda_vk_3k_4\mu_1\mu_2$

APPENDIX B

$B=3\lambda\lambda_v+\lambda_v k_2$

$B_1=k_4+k_3$

$B_2=k_3k_4$

$B_3=6\lambda\lambda_v(k_4+k_2)$

$B_4=k_4+k_3+4\lambda_v$

$B_5=k_3k_4+2\lambda_v(k_2+k_4)$

$B_6=k_3+k_2$

$B_7=k_3k_2$

$B_8=k_2+k_3+6\lambda$

$B_9=k_3k_2+3\lambda(k_4+k_2)$

$B_{10}=k_4+k_6+k_3$

$B_{11}=k_4k_6+k_4k_3+k_3k_6$

$B_{12}=k_4k_3k_6$

$B_{13}=k_3+k_4+k_5+k_6$

$B_{14}=k_4k_3+k_3k_5+k_3k_6+k_4k_5+k_4k_6+k_6k_5$

$B_{15}=k_4k_3k_5+k_4k_3k_6+k_3k_6k_5+k_4k_6k_5$

$B_{16}=k_4k_3k_6k_5$

$B_{17}=18\lambda\lambda_vk_6+9\lambda_v\lambda_vk_2+18\lambda\lambda_vk_4+9\lambda\lambda_vk_5$

$B_{18}=9\lambda\lambda_vk_6k_5+9\lambda_v\lambda k_6k_2+18\lambda_v\lambda k_4k_6+9\lambda_v\lambda k_4k_2+9\lambda_v\lambda k_4k_5$

$B_{19}=9\lambda\lambda_vk_4k_6k_5+9\lambda\lambda_vk_4k_6k_2$

$B_{20}=18\lambda\lambda_v^2+36\lambda_v^2\lambda k_2+36\lambda\lambda_v^2k_6+18\lambda\lambda_v^2k_3$

$B_{21}=18\lambda\lambda_v^2k_6k_5+18\lambda\lambda_v^2k_3k_2+18\lambda\lambda_v^2k_6k_2$

$B_{22}=k_2+k_3+k_4$

$B_{23}=k_3k_2+k_4k_2+k_4k_3$

$B_{24}=k_4k_3k_2$

$B_{25}=k_2+k_3+k_4$

$B_{26}=k_3k_2+k_4k_2+k_4k_3$

$B_{27}=k_4k_3k_2$

$B_{28}=9\lambda+k_4+k_3+k_2$

$B_{29}=6\lambda k_2+3\lambda k_3+3\lambda k_5+6\lambda_vk_6+k_4k_3+k_4k_2+k_3k_2$

$B_{30}=3\lambda k_6k_5+3\lambda k_3k_2+3\lambda_vk_6k_2+k_4k_3k_2$

$B_{31}=k_4+k_3+k_6+k_5+6\lambda_v$

$B_{32}=k_6k_5+k_4k_5+k_4k_6+3\lambda_vk2+k_3k6+3\lambda_vk_5+6\lambda_vk_6+k_4k_3+18\lambda_v^2+k_3k_5+6\lambda_vk_4$

$B_{33}=12\lambda_v^2k_6+6\lambda_v^2k_5+3\lambda_vk_4k_2+k_4k_6k_5+k_4k_3k_5+k_3k_6k_5+k_4k_3k_6+3\lambda_vk_2k_6+$

$3\lambda_vk_4k_5+3\lambda_vk_6k_5+6\lambda_vk_4k_6+12\lambda_v^2k_2+6\lambda_v^2k_3$

$B_{34}=6\lambda_v^2k_6k_5+6\lambda_v^2k_2k_6+k_4k_3k_6k_5+6\lambda_v^2k_3k_2+3\lambda_vk_4k_2k_6+3\lambda_vk_4k_6k_5$

$B_{35}=3\lambda\lambda_v+\lambda_v k_2$

$B_{36}=k_4+k_3$

$B_{37}=k_3k_4$

$B_{38}=6\lambda\lambda_v(k_4+k_2)$

$B_{39}=k_4+k_3+4\lambda_v$

$B_{40}=k_3k_4+2\lambda_v(k_2+k_4+\mu_2)$

$B_{41}=k_2+k_3$

$B_{42} = k_2k_3$

$B_{43}=k_2+k_3+6\lambda$

$B_{44}=k_3k_2+3\lambda(k_4+k_2)$

$B_{45}=k_4+k_5+k_3+k_6$

$B_{46}=-2\lambda_v\mu_2+k_4k_3+k_3k_6+k_4k_5+k_4k_6+k_6k_5+k_3k_5$

$B_{47}=-2\lambda_v k_4\mu_2-2\lambda_v k_3\mu_2+k_4k_3k_5+k_4k_3k_6+k_4k_6k_5+k_3k_6k_5$

$B_{48}=-2k_4k_3\mu_2\lambda_v+k_4k_3k_6k_5$

$B_{49}=k_4+k_5+k_3+k_6$

$B_{50}=-2\mu_2\lambda_v+k_4k_3+k_3k_6+k_4k_5+k_4k_6+k_6k_5+k_3k_5$

$B_{51}=k_4k_3k_5+k_4k_3k_6+k_4k_6k_5+k_3k_6k_5$

$B_{52}=-2k_4k_3\mu_2\lambda_v+k_4k_3k_6k_5$

$B_{53}=9\lambda\lambda_v k_5+9\lambda\lambda_v k_2+18\lambda\lambda_v k_6+18\lambda_v\lambda k_4$

$B_{54}=18\lambda\lambda_v^2\mu_2+9\lambda\lambda_v k_6k_5+9\lambda\lambda_v k_6k_2+18\lambda\lambda_v k_4k_6+9\lambda\lambda_v k_4k_5+9\lambda\lambda_v k_4k_2$

$B_{55}=-18\lambda k_4\mu_2\ \lambda_v^2+9\lambda\lambda_v k_4k_6k_2+9\lambda\lambda_v k_4k_6k_5$

$B_{56}=36\lambda\lambda_v^2k_6+18\lambda\lambda_v^2k_5+36\lambda\lambda_v^2k_2+18\lambda\lambda_v^2k_3$

$B_{57}=-90\lambda\lambda_v^3\mu_2+18\lambda\lambda_v^2k_3k_2+18\lambda\lambda_v^2k_6k_5+18\lambda\lambda_v^2k_6k_2$

$B_{58}=k_2+k_4+k_3$

$B_{59}=k_4k_2+k_4k_3+k_3k_2-3\lambda_v\mu_2$

$B_{60}=-3k_4\mu_2\lambda_v+k_4k_3k_2$

$B_{61}=k_2+k_4+k_3$

$B_{62}=k_4k_3+k_3k_2+k_4k_2$

$B_{63}=k_4k_3k_2$

$B_{64}=9\lambda+k_2+k_3+k_4$

$B_{65}=3\lambda k_5+3\lambda k_3+6\lambda k_2+6\lambda k_6+k_2k_3+k_4k_3+k_4k_2$

$B_{66}=-6\mu_2\lambda_v\lambda+3\lambda k_6k_5+3\lambda k_2k_6+3\lambda k_2k_3+k_2k_4k_3$

$B_{67}=k_5+k_6+6\lambda_v+k_4+k_3$

$B_{68}=6\lambda_v k_4+k_4k_5+k_3k_5+3\lambda_v k_5-2\lambda_v\mu_2+k_3k_6+k_4k_3+6\lambda_v k_6+k_4k_6+k_6k_5+18\lambda_v^2+3\lambda_v k_2$

$B_{69}=12\lambda_v^2k_2+3\lambda_v k_4k_2-2\lambda_v k_3\mu_2-2\lambda_v k_4c_2+3\lambda_v k_6k_5+3\lambda_v k_6k_2+3\lambda_v k_4k_5+$

$6\lambda_v k_4k_6+k_4k_3k_5+k_4k_3k_6+k_3k_6k_5+k_4k_6k_5+12\lambda_v^2k_6+6\lambda_v^2(k_3+k_5-\mu_2)$

$B_{70}=-6\lambda_v^2k_4\mu_2+3\lambda_v k_4k_6k_5+3\lambda_v k_4k_2k_6-2\lambda_v k_4k_3\mu_2+k_4k_3k_6k_5+6\lambda_v^2k_2k_3-$

$12\mu_2\lambda_v^3+6\lambda_v^2k_6k_5+6\lambda_v^2k_6k_2$

$B_{71}=9\lambda\lambda_v(D_4+D_7)/(D_4D_5D_7)$

$B_{72}=2\lambda_v B_{71}/D_8+6\lambda\lambda_v B_{74}/D_6$

$B_{73}=(3\lambda/D_4+B_{71}+B_{72})2\lambda/\mu_1$

$B_{74}=6\lambda_v^2/D_7D_8$

$B_{75}=12\lambda\lambda_v/D_9D_{10}+12\lambda\lambda_v/D_{10}D_{14}$

$B_{76}=4\lambda_v B_{75}/D_{11}+36\lambda^2\lambda_v/D_{10}D_{14}D_{11}$

$B_{77}=2\lambda_v B_{76}/D_{12}+108\lambda^3\lambda_v/D_{10}D_{14}D_{11}D_{16}$

$B_{78}=(3\lambda/D_9+B_{75}+B_{76}+B_{77})2\lambda/\mu_1$

$B_{79}=(B_{77}+B_{78})\lambda_v/\mu_1$

$B_{80}=6\lambda\lambda_v/D_{17}D_{18}+B_{81}+6\lambda\lambda_v/D_{18}D_{19}$

$B_{81}=(3\lambda/D_{17}+B_{80})2\lambda/\mu_1$

$B_{82}=(2\lambda_v/D_{19}+B_{80})\ \lambda_v/\mu_2$

$B_{83}=1-2\lambda_v\mu_2\,/\,D_{23}D_{24}$

$B_{84}=2\lambda_v\mu_2/\,D_{21}D_{22}$

$B_{85}=6\lambda_v\,\lambda\mu_2\,/\,D_{21}D_{22}D_{24}$

$B_{86}=2\lambda_v\,/\,D_{21}+B_{85}3\lambda_v/B_{85}\,D_{23}$

$B_{87}=(1-2\mu_1\,B_{85}/B_{83}D_{23}-B_{84})$

$B_{88}=B_{86}\,/B_{87}$

$B_{89}=2\lambda_v\,/B_{88}D_{22}-(6\lambda\lambda_v\,/D_{22}D_{24})*(1-B_{83})(3\lambda_v/B_{83}D_{23}+B_{86}3\lambda_v/B_{83}\,D_{23})$

$B_{90}=(3\lambda_v/\,D_{20}+B_{88}+B_{89})*(2\lambda/\mu_1)$

$B_{91}=(3\lambda_v/D_{23}+\mu_1B_{88}\,/\,D_{23}+\mu_2\,B_{89}/D_{24}D_{23})$

$B_{92}=\mu_2\,B_{89}/D_{24}$

$B_{93}=\lambda_v/\mu_2(B_{90}+B_{92})$

APPENDIX C

$k_5=3\lambda+2\lambda_v$

$k_6=2\lambda+2\lambda_v$

$k_7=2\lambda+\lambda_v$

$k_8=3\lambda+\lambda_v$

$k_9=3\lambda+3\lambda_v$

$k_{10}=2\lambda+3\lambda_v$

$k_{11}=2\lambda+2\lambda_v$

$k_{12}=2\lambda+\lambda_v$

$k_{13}=3\lambda+2\lambda_v$

$k_{14}=3\lambda+\lambda_v$

$k_{19}=3\lambda+2\lambda_v$

$k_{20}=2\lambda+2\lambda_v+\mu_1$

$k_{21}=2\lambda+\lambda_v$

$k_{22}=3\lambda+\lambda_v+\mu_2$

$k_{23}=3\lambda+3\lambda_v$

$k_{24}=2\lambda+3\lambda_v+\mu_1$

$k_{25}=2\lambda+2\lambda_v$

$k_{26}=2\lambda+\lambda_v$

$k_{27}=3\lambda+2\lambda_v+\mu_2$

$k_{28}=3\lambda+\lambda_v+\mu_2$

$M=-36A_{10}A_{11}A_{12}-288A_{11}A_{13}+108A_{12}^2+8A_{11}^3+12(18A_{10}^2A_{12}^2A_{13}-3A_{10}^2A_{11}^2A_{12}^2+57A_{10}A_{12}A_{13}^2-$

$432A_{11}A_{12}^2A_{13}-432A_{10}^2A_{11}A_{13}^2+12A_{10}^2A_{11}^3A_{13}-768A_{12}^3+12A_{10}^3A_{12}^3+384A_{11}^2A_{13}^2-$

$48A_{11}^4A_{13}+12A_{11}^3A_{12}^2+81A_{10}^4A_{13}^2-54A_{10}A_{11}A_{12}^3+240A_{10}A_{11}^2A_{12}A_{13}+81A_{13}^4-$

$54A_{10}^3A_{11}A_{12}A_{13})^{1/2}$

$L=(3A_{10}M^{1/3}-8A_{11}M^{1/3}+2M^{2/3}-24A_{10}A_{12}+96A_{13}+8A_{11}^{\ 2})/M^{1/3}$

$D_1=2\lambda+2\lambda_v$

$D_2=2\lambda+\lambda_v$

$D_3=3\lambda+\lambda_v$

$D_4=2\lambda+3\lambda_v$

$D_5=2\lambda+2\lambda_v$

$D_6=2\lambda+\lambda_v$

$D_7=3\lambda+2\lambda_v$

$D_8=3\lambda+\lambda_v$

$D_9=2\lambda+4\lambda_v$

$D_{10}=2\lambda+3\lambda_v$

$D_{11}=2\lambda+2\lambda_v$

$D_{12}=2\lambda+\lambda_v$

$D_{13}=3\lambda+3\lambda_v$

$D_{14}=3\lambda+2\lambda_v$

$D_{15}=3\lambda+\lambda_v$

$D_{16}=2\lambda+\lambda_v+\mu_1$

$D_{17}=2\lambda+2\lambda_v+\mu_1$

$D_{18}=2\lambda+\lambda_v+\mu_1$

$D_{19}=3\lambda+\lambda_v+\mu_2$

$D_{20}=2\lambda+3\lambda_v+\mu_1$

$D_{21} = 2\lambda + 2\lambda_v + \mu_1$

$D_{22} = 2\lambda + \lambda_v + \mu_2$

$D_{23} = 3\lambda + 2\lambda_v + \mu_2$

$D_{24} = 3\lambda + \lambda_v + \mu_2$