IMPROVING SYSTEM RELIABILITY UNDER BUDGETARY CONSTRAINT

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY
IN
PRODUCTION ENGINEERING

SUBMITTED BY

GAURAV SHARMA
ROLL NO. -2K16/PIE/06

Under the supervision of

Dr. Girish Kumar
(ASSOCIATE PROFESSOR)

DEPARTMENT OF MECHANICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042
JULY, 2018.

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE’S DECLARATION

I, GAURAV SHARMA, Roll No. 2K16/PIE/06 student of M.Tech (Prodution Engineering), hereby declare that the project Dissertation titled “Improving System Reliability Under Budgetary Constraint” which is submitted by me to the Department of Mechanical Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi **GAURAV SHARMA**

Date:

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the project Dissertation titled “Improving System Reliability Under Budgetary Constraint” which is submitted by GAURAV SHARMA, Roll No. 2K16/PIE/06 Department of Mechanical Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date:

Dr. GIRISH KUMAR
ASSOCIATE PROFESSOR
Department of Mechanical, Production and Industrial Engineering
Delhi Technology University,
DELHI-110042

ACKNOWLEDGEMENT

Any accomplishment is a result of positivity of thoughts and efforts. It is important here to appreciate contribution, encouragement and support from persons who stood as ‘Light House’ throughout the voyage.

I wish to express my sincere gratitude for my project supervisor and mentor, Dr. Girish Kumar, Associate Prof., Department of Mechanical Engineering, Delhi Technological University (Delhi). It was a golden opportunity to work under his kind supervision. His scholastic guidance and sagacious suggestions helped me to complete the project.

I wish to thank Dr. Vipin, Head, Department of Mechanical Engineering, Delhi Technological University (Delhi), for constantly motivating and providing able guidance.

I am also thankful to the staff of Department of Mechanical Engineering for their continual support and cooperation.

Finally, but importantly, I would like to express my heartfelt thanks to my beloved family and friends who have endured the long working hours and whose motivation kept me going.

GAURAV SHARMA
(Roll No. 2K16/PIE/06)
M.Tech. (Production Engineering)
Delhi Technological University, Delhi, India

ABSTRACT

In this thesis a system is designed to improve its reliability under given budgetary constraint. Models are developed for the system to enhance the reliability by adding standby redundancies and incorporative appropriate maintenance. All the feasible states and, failure and repair transitions are identified to develop the system model. Due to the limitation of the Markov model the failure and repair rates are taken as constant. The sets of ordinary differential equations are obtained for the change of probability of being in respective system states with respect to time in each model. The system of rate equations is solved using Runge – Kutta method in MatLab. The system reliability assessment is based on the sum of probabilities of all working states. Dynamic programming is used for optimizing the reliability under cost limitation.

These results are helpful to design the highly reliable systems for various industrial applications like thermal power plant, machineries used in medical field, aviation industries etc.

Keywords : Reliability improvement, Redundancies, Maintenance, Dynamic programming, Markov.

TABLE OF CONTENT

Content Page no.

LIST OF FIGURES

LIST OF TABLES

LIST OF NOMENCLATURE

$\lambda1$= Failure rate of condenser pump

$\lambda2$ = Failure rate of high pressure compressor

$\lambda3$= Failure rate of low pressure compressor

$\mu1$= Repair rate of condenser pump

$\mu2$= Repair rate of high pressure compressor

$\mu3$= Repair rate of low pressure compressor

$R(t)$= Reliability function

$P(t)$= Probability of particular state at time t

O= Operating state of a component

S= Standby state of a component

F = Failed state of a component

Σ= Summation

n= Stage

s= State

X= Decision variable

LIST OF ABBREVIATIONS

MTTF- Mean time to repair

MTTR- Mean time to failure

ETA- Event tree analysis

FMEA- Failure mode and effects analysis

FMECA- Failure mode, effects and criticality analysis

FTA- Fault tree analysis

MA- Markov analysis

PNA- Petri net analysis

RBD- Reliability block diagram.

CHAPTER 1

INTRODUCTION

1.1 Overview

In today’s world, various industries like automobile, aviation, manufacturing, power sector, refineries and many more are running continuously without any break down/down time effectively. In such industries break down for a minute could be reason for big loss. Failure of any component can lead to stop the whole system. Especially in aviation industries we can’t afford these situations, as it might be a reason for the death of passengers. To overcome such conditions the concept of reliability comes into existence. Reliability is defined as

“The ability of an item to perform a required function, under given environmental and operational conditions and for a stated period of time”

Reliability is a broadly utilized idea. Its main concern is to focuses on the ability of an item to perform its functions. Mathematically, assuming that a component is performing its intended functions at zero time, reliability can be explained as probability that item will continue to perform its intended functions without any failure for a specified time lapse under given conditions. Redundancy is a common approach to improve the reliability of a system. Adding redundancy to a system increases the cost and complexity of design of system. However, if the cost related to the failure of a system is very high then redundancy may be a very good option. Redundancies can be categorized as active or passive (standby) redundancy.

- Active Redundancy: when all the redundant components works simultaneously and share the load with the main component is known as active redundant system.
- Passive Redundancy: When the standby component/redundant is works only after the failure of the redundant is known as passive redundant system.

1.2 Current Status of Work

Reliability modest beginning was in 1816. During the world war as army required more reliable arms so that they could fight for longer time. Since that time continuous improvement are going on new reliability techniques and now industries are focusing on more and more reliability part so that their product run for longer time.

As of current scenario reliability is the one of the important part for designing and development of any product/system. In aviation industries, automobile industries, power plants etc. these kind of industries required more reliable system/components as failure of anything can lead to major loss of industries in terms of money, good will, designing defects etc.

1.3 Organization of Report

In the chapter one of this report discussed about the basic definition of reliability, terminologies used and the current status of work. In Chapter number two literature survey has done. In chapter three discussed about the reliability tools and techniques. These reliability tools are highly recommended in the design system for its improvement. In chapter four discussed the system on which this thesis is based, development of models, cost models and optimization tools and techniques used. In chapter five mathematical modeling of system, transition diagram of models developed by Markov modeling and differential equation generated by transition diagrams based on Markov models are presented. In the chapter number six conclusion and future scope has discussed.

CHAPTER 2

LITERATURE REVIEW

This chapter deals with literature review mainly covering reliability improvement, optimization techniques, research gap.

2.1 Reliability improvement

Postnikov Ivan, et al.[2017] suggest a methodology to determine optimal reliability parameters (failure and repair rates) of heat supply system's components, which provide the required level of reliability to the heat supply system. The methodological approach consists in the economically rational distribution of the total effect of reliability improvement among the system components, which is calculated using the average of reliability parameters of the components. The task to ensure structural reliability, is one of the key reliability task within a more general problem of optimal synthesis of heat supply systems and is urgent for both the systems under design and the existing systems having insufficient reliability.

Goo Bongeun, et al. [2017] told that in system design, the initial design stage is being increasingly emphasized because it significantly influences the successive system/product development and production stages. However, for larger and more complex systems, it is very difficult to accurately predict system reliability in the initial design stage. Various design methodologies have been proposed to resolve this issue, but maintaining reliability while exploring design alternatives is yet to be achieved. Therefore, they propose a methodology for conceptual design considering reliability issues that may arise in the successive detailed design stages. The methodology integrates the independency of axiomatic design and the hierarchical structure of failure mode effects, and criticality analysis (FMECA), which is a technique widely used to analyze system reliability. To verify effectiveness of this method in the reliability improvement of the design process, we applied the proposed methodology to a liquefied natural gas (LNG) supply system.

M.H. Khoshgoftar Manesha, et al, [2018] proposed research method for determination of reliability and availability of complex cogeneration systems by improving the approximated Markov chain method. There are two procedures to solve reliability problems analytical techniques and stochastic techniques. Each has own advantages and disadvantages. One of the important analytical techniques for repairable systems is the Markov chain method. This method uses state space method to consider all states that may occur. To use this method for complex systems, the model of the system must be simplified. For this purpose, many states are removed from the space state having very less chance to occur. In this way, although the probabilities of the states are calculated, these probabilities are often not accurate. In the present work, a new approach is proposed that considers both the simplified system and the calculation of the probability of each state accurately. The new method can calculate the probabilities by taking into account minimum states. Site utility systems are used to illustrate the procedure for applying this method. Site utilities have several repairable components (e.g. gas turbine, steam turbine, , HRSG, deaerator, boiler). So, this system can generate a large and complex state space for which it is difficult to calculate the probabilities. The new procedure can reduce the number of states and aggregates of the exploded state space due to the high number of components in the system. The results show that the new procedure can predict state probabilities with high accuracy.

W. Zhou, et al, [2017], research about an improved equivalent component approach to evaluate the system reliability of series systems. An analytical expression is derived to evaluate the unit normal vector, in the context of the first-order reliability method, associated with the equivalent component. This improves the computational efficiency for determining the correlation coefficients between the system and equivalent components in the equivalent component approach. An adaptive combining process is also proposed, whereby the two components with the highest correlation coefficient are combined at each combining step. The accuracy and efficiency of the improved equivalent component approach are demonstrated for series systems with equally and unequally correlated components. Finally, the improved equivalent component approach is illustrated through the system reliability analysis of a pressurized steel pipeline segment containing ten active corrosion defects.

Kai Hou, et. al.,[2016] tells about a continuous time Markov chain (CTMC) based sequential analytical approach for composite generation and transmission systems reliability assessment. The basic idea is to construct a CTMC model for the composite system. Based on this model, sequential analyses are performed. Various kinds of reliability indices can be obtained, including expectation, variance, frequency, duration and probability distribution. In order to reduce the dimension of the state space, traditional CTMC modeling approach is modified by merging all high order contingencies into a single state, which can be calculated by Monte Carlo simulation (MCS). Then a state mergence technique is developed to integrate all normal states to further reduce the dimension of the CTMC model. Moreover, a time discretization method is presented for the CTMC model calculation. Case studies are performed on the RBTS and a modified IEEE 300-bus test system. The results indicate that sequential reliability assessment can be performed by the proposed approach. Comparing with the traditional sequential Monte Carlo simulation method, the proposed method is more efficient, especially in small scale or very reliable power systems.

Mohammad Feizabadi, et. al.,[2017] develops a new model for reliability optimization of series-parallel systems with non-homogeneous components. In previously presented models for reliability optimization of series-parallel systems, there is a restricting assumption based on which all components of a subsystem must be homogeneous. This constraint limits system design engineer in selecting components and prevents achieving higher levels of reliability goal. In this paper, a new model is proposed for reliability optimization of series-parallel systems, which makes possible the use of non-homogeneous components in each sub-systems. As a result of this flexibility, the process of supplying system components will be easier. To solve the proposed model, since the redundancy allocation problem (RAP) belongs to the NP-hard class of optimization problems, a genetic algorithm (GA) is developed. The computational results of the designed GA are indicative of high performance of the proposed model in increasing system reliability and decreasing costs.

2.2 Tools and Techniques

Qianru Ge, et. al., [2018] develop an optimization model to determine the reliability design of critical components in a serial system. The system is under a service contract, and a penalty cost has to be paid by the OEM when the total system down time exceeds a predetermined level, which complicates the evaluation of the expected cost under a given reliability design. Furthermore, in the design phase for each critical component, all possible designs are subject to uncertain component failure rates. Considering the computational intractability of evaluating the system performance, they develop approximate evaluation methods which take the system uncertainty into account. Numerical results show that the method which includes randomness in the number of failures, failure rates and repair times leads to efficient and accurate evaluations and to close-to-optimal design decisions when used in an enumeration procedure for the optimization problem. They also show that ignoring these three types of uncertainty may result in bad design decisions.

Yan Lia, et. al., [2017] focus on the development of reliability measures for a repairable multi-state system which operates under dynamic regimes under the discrete-time hypothesis. The switching process of regimes is governed by a Markov chain, and the functioning process of the system also follows another Markov chain with different transition probability matrices under different regimes. In terms of two chains or more as above, a new Markov chain is constructed to depict the evolution process of the dynamic system. For the regime consideration, some novel reliability indices are essential and firstly introduced in this paper. By means of hierarchical partitions for the new state space, Ion-Channel modeling theory and discrete-time Markov chain, the traditional and novel reliability and availability functions for the system under random regimes are easily obtained with the closed form solutions, such as two types of system reliabilities, two types of system point availabilities, two types of system multiple-point availabilities and the associated system multi-interval availabilities and so on.

Ta-Cheng Chen, et. al., [2005] considers the series–parallel redundant reliability problems in which both the multiple components choices of each subsystem and the redundancy levels of every selected components are to be decided simultaneously so as to increase the system reliability. The reliability design optimization problem has been studied in the literature for decades, usually using heuristic optimization approaches or mathematical programming. The difficulties encountered for both methodologies are the number of constraints and the difficulty of satisfying the constraints. A penalty-guided immune algorithms-based approach is presented for solving such integer nonlinear constraints of redundant reliability design problem. The results obtained by using immune algorithms-based approach are compared with the results obtained from 33 test problems from the literature that dominate the previously mentioned solution techniques. As reported, solutions obtained by the proposed method are better than or as well as the previously best-known solutions of the problems.

Peter X. Liua, et. al.,[2003] presents a novel continuous-state system model for optimal design of parallel–series systems when both cost and reliability are considered. The advantage of a continuous-state system model is that it represents realities more accurately than discrete-state system models. However, using conventional optimization algorithms to solve the optimal design problem for continuous-state systems becomes very complex. Under general cases, it is impossible to obtain an explicit expression of the objective function to be optimized. In their paper, they propose a neural network (NN) approach to approximate the objective function. Once the approximate optimization model is obtained with the NN approach, the subsequent optimization methods and procedures are the same and straightforward. A 2-stage example is explained to compare the analytical approach with the proposed NN approach. A complicated 4-stage example is solved to illustrate that it is easy to use the NN approach while it is too difficult to solve the problem analytically.

Chia-Ling Huang [2015] proposes a new swarm intelligence method known as the Particle-based Simplified Swarm Optimization (PSSO) algorithm while undertaking a modification of the Updating Mechanism(UM), called N-UM and R-UM and simultaneously applying an Orthogonal Array Test (OA) to solve reliability–redundancy allocation problems (RRAPs) successfully. One difficulty of RRAP is the need to maximize system reliability in cases where the number of redundant components and the reliability of corresponding components in each subsystem are simultaneously decided with nonlinear constraints. In his paper, four RRAP benchmarks are used to display the applicability of the proposed PSSO that advances the strengths of both PSO and SSO to enable optimizing the RRAP that belongs to mixed-integer nonlinear programming. When the computational results are compared with those of previously developed algorithms in existing literature, the findings indicate that the proposed PSSO is highly competitive and performs well.

A.A. Najafi, et. al., [2013] tell the redundancy allocation problem is one of the main branches of reliability optimization problems. Traditionally, the redundancy allocation model has focused mainly on maximizing system reliability at a predetermined time. Hence, they develop a more realistic model, such that the mean time to failure of a system is maximized. To overcome the structural complexity of the model, the Monte Carlo simulation method is applied. Two metaheuristics, Simulated Annealing (SA) and Genetic Algorithm (GA), are proposed to solve the problem.

Some more research papers have been also studied to identify the more clarity about the redundancy allocation, Markov models, Dynamic programming and finding research gap.

Table 2.1 Researchers and their contribution

S. No.	Author’s Name	Title
1	Amirhossain Chambari, et. al. [2012]	A bi-objective model to optimize reliability and cost of system with a choice of redundancy strategies

Table 2.1 Researchers and their contribution (Contd.)

S. No.	Author’s Name	Title
2	C.S. Sung, et. al., [1999]	Reliability optimization of a series system with multiple-choice and budget constraints
3	Yun-Chia Liang, et. al., [2006]	Redundancy allocation of series-parallel systems using a variable neighborhood search algorithm
4	Kelly M. Sullivan, et. al., [2017]	An integrated approach to redundancy allocation and test planning for reliability growth
5	R. Tavakkoli-Moghaddama, et. al., [2007]	Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm
6	Mohamed Ouzineb, et. al., [2007]	Tabu search for the redundancy allocation problem of homogenous series–parallel multi-state systems
7	Heungseob Kima, et. al., [2016]	Reliability–redundancy allocation problem considering optimal redundancy strategy using parallel genetic algorithm
8	L.Y. Xu, et. al., [2011]	Reliability and failure pressure prediction of various grades of pipeline steel in the presence of corrosion defects and pre-strain
9	Ling-ling Lia, et. al., [2018]	Reliability measure model for electromechanical products under multiple types of uncertainties
10	Hyung-Geun Kwag, et. al., [2014]	Reliability modeling of demand response considering uncertainty of customer behavior
11	Mostafa Abouei Ardakan, et. al., [2014]	Optimizing bi-objective redundancy allocation problem with a mixed redundancy strategy
12	Jozef Baláka, et. al., [2017]	Modeling of transition of system with standby redundancy into failed state
13	Anatoly Lisnianski, [2007]	Extended block diagram method for a multi-state system reliability assessment

Table 2.1 Researchers and their contribution (Contd.)

S. No.	Author's Name	Title
14	Anatoly Lisnianski, et. al., [2011]	A multi-state Markov model for a short-term reliability analysis of a power generating unit
15	Devesh K. Jha, et. al., [2015]	Depth Estimation in Markov Models of Time-series Data via Spectral Analysis
16	Amin Ziaei-Nia, et. al., [2018]	Dynamic Cost Optimization Method of Concrete Mix Design
17	Pradyumna Kumar Tripathy, et. al. [2015]	A dynamic programming approach for layout optimization of interconnection networks

2.3 Research gap

From the above literature review it has been found that till now focus of the work was on redundancy allocation. There is lack of literature on cost consideration while improving system reliability. Moreover, mainly Monte Carlo techniques was used for optimization and a very few researcher used Dynamic programming.

2.4 Objective

Based on the literature review on the section 2.1 and 2.2 on the literature review and research gap find in section 2.3, the following gaps are formulated

- To develop Markov models with passive redundancies and maintenance.
- To optimize the system reliability under budgetary constraint.

2.5 Problem Statement

The system reliability modelling with redundancies and maintenance using Markovian approach and optimization under cost limitation through Dynamic programming.

CHAPTER 3

RELIABILITY

In this chapter, various reliability tools and techniques used in design phase will be discussed. Modeling of a various mechanical systems in reliability is done by these reliability tools.

3.1 Reliability Basics

The basic function of reliability is the elimination of failure modes, and the proper management of resources to minimize the frequency of uncontrollable failures. It can deliver substantial financial as well as other benefits to a company. Reliability is an important consideration in the engineering design process. The analysis of reliability guides practicing engineers in selecting an appropriate design strategy and in improving performance of the system. It has always been considered as a useful tool for design of systems, risk analysis, production availability studies.

"Reliability is defined to be the probability that a component or system will perform a required function for a stated period of time when used under stated operating conditions"[CHARLES E. EBELING]

Mathematically reliability can be defined as take a continuous random variable T to be the time to failure of system i.e. $T \geq 0$.

Then reliability can be expressed as

$$R(t) = \Pr\{T \geq t\}$$

Where $R(t) \geq 0$, $R(0) = 1$, and $R(t) = 0$ (for infinite time interval)

Or we can say

$$F(t) = 1 - R(t) = \Pr.\{T \prec t\}$$

Where $F(0) = 0$, $F(t) = 1$ (for infinite time interval), and F(t) is probability of failure occurrence before time t.

Assuming R(t) as reliability function and F(t) as cumulative distribution function of failure distribution. We also can conclude as a probability distribution function as it describes the shape of failure distribution.

$$f(t) = \frac{dF(t)}{dt} = -\frac{dR(t)}{dt} \quad f(t) = \frac{dF(t)}{dt} = -\frac{dR(t)}{dt}$$

3.2 Reliability Improvement

Within reliability field, many models and methods are used in design phase of a component/subsystem/system, such as failure models and system analysis methods and model. These methods are classified into the following categories with regard to their main purpose:

- Part Count Approach
- Event tree analysis(ETA)
- Stress strength analysis
- Failure mode and effects analysis(FMEA)
- Functional Analysis
- Reliability block diagram
- Markov modeling
- Petri net analysis
- Part de-rating and selection
- Fault tree analysis(FTA)

3.2.1 Part Count Approach

The Part Count is the easiest and the most despairing form of approach where it is assumed that every component failure can leads to fail system. This approach first of all the components noted down along with their probability of failure and after this individual probability of components are added. The sum obtained gives an upper bound on the probability of system failure.

The part count technique for a particular system provides a very pessimistic estimate of probability of system failure. It is not quantifiable, but conservative because even if critical component exists it appears redundantly so that even a single failure is not disastrous for the system. More often, the component can depart from its normal working mode in many ways and this failure mode, not generally, will have some pernicious effect on the operation of the system. If the appropriate failure modes of the system operation are not known then the summation of failure probabilities for all the possible failure modes is done.

Advantages -

- Very helpful during the designing of initial phase of any system to analyze reliability.

3.2.2 Event Tree Analysis

ETA is a widely used technique to analyze the risk and reliability of a system. It is one of the most common methods to analyze the progression of failure. It is a diagram of logic tree which starts from the initial event and gives the proper time details of every event propagation for the outcome of intended function. The ETA is a natural part of most risk analyses but they can be used as a design tool to demonstrate the effectiveness of protective systems in a plant. In quantitative ET this method can be used independently or, is often combine with fault tree analysis. ET and FT are known as complement to each other. ET can also be used for human reliability assessment.

The major benefit of an event tree is the possibility to evaluate consequences of an event, and thus provide for possible mitigation of a highly probable, but unfavorable consequence. The event tree analysis is thus beneficial when performed as a complement to fault tree analysis. An event tree analysis can also be used as a tool in the fault mode analysis.

3.2.3 Stress-Strength Analysis

Stress-Strength analysis is a method to determine the capability of a component or an item to withstand electrical, mechanical, environmental, or other stresses that might be a cause of their failure, where reliability is the probabilistic measure of assurance of the component performance. This analysis determines the physical effect of stresses on a component, as well as the mechanical or physical ability of the component. Probability of component failure is directly proportional to the applied stresses. The specific relationship of stresses versus component strength determines component reliability.

The stress-strength analysis is primarily used in the determination of reliability or equivalent failure rate of mechanical components. It is also used in physics of failure to determine likelihood of occurrence of a specific failure mode due to a specific individual cause in a component. Evaluation of stress against strength and resultant reliability of parts depends upon the evaluation of the second moments, the mean values and variances of the expected stress and strength random variables. This evaluation is often simplified to one stress variable compared to the strength of the component. In general terms, the strength and stress shall be represented by the performance function or the state function, which is a representative of a multitude of design variables including capabilities and stresses. The positive value of this function represents the safe state while negative value represents the failure state.

Advantages of stress strength analysis -

- Stress-strength analysis is that it can provide accurate representation of component reliability as a function of the expected failure mechanisms.
- It includes variability of design as well as variability of expected applied stresses, and their mutual correlation.
- This technique provides a more realistic insight into effects of multiple stresses and is more representative of the physics of component failure, as many factors – environmental and mechanical – can be considered, including their mutual interaction.

Disadvantages of stress strength analysis -

- In the case of multiple stresses, and especially when there is an interaction or correlation between two or more stresses present, the mathematics of problem solving can become very involved, requiring professional mathematical computer tools.
- Another disadvantage is possible wrong assumption concerning distribution of one or more random variables, which, in turn, can lead to erroneous conclusions.

3.2.4 Failure Mode and Effects Analysis (FMEA)

This technique was developed in 1950 to study the failures in military systems, latterly it find its major applications in aerospace industries. FMEA is an approach which is used to identify how a system or process can fail and estimating risk associated with specific causes. It also tells about the actions that should be taken to minimize risk and evaluating design validation plan (design FMEA) or current control plan (process FMEA). It is generally the first step for the analysis of reliability. For a specified system, starting condition or any fault is note down and after that criticize the impact of fault or starting conditions on system, an inductive system analysis is carried out. It begins from failure initiators and essential event initiators, and afterward determines the subsequent framework impacts of a given initiator. A set of conceivable causes are investigated for their impact. There are a few systems giving rules to this technique, for example, more established military standard.

Advantages of FMEA –

- An FMEA offers a systematic review of all components, assemblies and subsystems if possible, in order to identify failure modes and the causes and effects of such failures. It connects single failures with their effects and identifies the causes of those failures.

- The output of an FMEA is input to other reliability analyses such as Fault Tree, Event Tree, Reliability Block Diagram, etc.

Limitations of FMEA –

- The analysis is limited to single failures and is time-consuming.

3.2.5 Functional Analysis

Functional Analysis is a qualitative method and an important step in a system reliability analysis. In order to identify all potential failures, the analyst has to understand the various functions of the system, each functional block in the system and the performance criteria related to all those functions. The objectives of a Functional Analysis are to:

1. Identify all the functions of the system
2. Identify and classify the functions required in different operational modes
3. Provide hierarchical decomposition of the system functions
4. Describe how each function is realized
5. Identify interrelationships between functions
6. Identify interfaces with other systems and with the environment

Advantages of Functional Analysis –

- Functional Analysis provides an understanding of the systems functionality, interconnection between functions, and a base for further reliability.

Limitations of Functional Analysis –

- Wrong assumptions can lead to erroneous conclusions.

3.2.6 Reliability Block Diagram (RBD)

A Reliability Block Diagram is a success- oriented network describing the function of the system. RBD is an inductive model wherein a system is divided into blocks that represents distinct elements such as components or subsystems. These elemental blocks are then combined according to system-success pathways as shown in Figure 3.1. RBDs are generally used to represent active elements in a system, in a manner that allows an exhaustive search for and identification of all pathways for success. Dependencies among elements can be explicitly addressed.

Initially developed top-level RBDs can be successively decomposed until the desired level of detail is obtained. Alternately, series components representing system trains in detailed RBDs can be logically combined, either directly or through the use of Fault Trees, into a supercomponent that is then linked to other super-components to form a summary model of a system. Such a representation can sometimes result in a more transparent analysis. Separate blocks representing each system are structurally combined to represent both potential flow paths through the system.

The model is solved by enumerating the different success paths through the system and then using the rules of Boolean algebra to continue the blocks into an overall representation of system success. When an element is represented by a block it usually means that the element is functioning. Each element has also a probabilistic model of performance, such as Weibull, If the system has more than one function, each function must be considered individual.

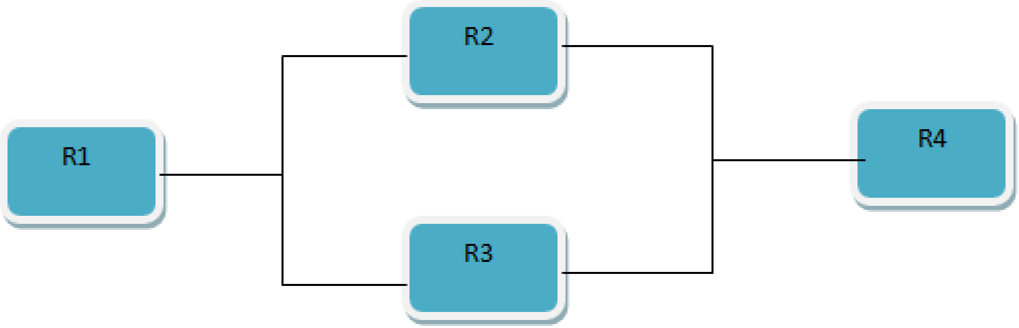

Fig. 3.1: Reliability block diagram

Advantages of using RBD –

- Often constructed almost directly from the system functional diagram; this has the further advantage of reducing constructional errors and/or systematic depiction of functional paths relevant to system reliability.
- Deals with most types of system configuration including parallel, redundant, standby and alternative functional paths.
- Capable of complete analysis of variations and trade-offs with regard to changes in system performance parameters.
- Provides (in the two-state application) for fairly easy manipulation of functional or nonfunctional paths to give minimal logical models.
- Capable of sensitivity analysis to indicate the items dominantly contributing to overall system reliability.
- Capable of setting up models for the evaluation of overall system reliability and availability in probabilistic terms.

- Results in compact and concise diagrams for a total system.

Limitations using RBD -

- Does not, in itself, provide for a specific fault analysis, i.e. the cause-effect(s) paths or the effect-cause(s) paths are not specifically highlighted.
- Requires a probabilistic model of performance for each element in the diagram.
- it will not show spurious or unintended outputs unless the analyst takes deliberate steps to this end.
- It is primarily directed towards success analysis and does not deal effectively with complex repair and maintenance strategies or general availability analysis.
- It is in general limited to non-repairable systems.
- The analysis is limited to single failures and is time-consuming.

3.2.7 Markov Modeling

Markov modeling is a probabilistic method that allows the statistical dependence of the failure or repair characteristics of individual components to be adapted to the state of the system. Hence, Markov modeling can capture the effects of both order dependent component failures and changing transition rates resulting from stress or other factors. Due to this reason, Markov analysis is very useful for dependability evaluation of functionally complex system structures and complex repair and maintenance strategies.

The proper field of application of this technique is when the transition failure or repair rates depend on the system state or vary with load, stress level, system structure such as in stand-by, maintenance policy or other factors. In particular, the system structure and the maintenance policy induce dependencies that cannot be captured by other, less computationally intensive techniques. The size of a Markov model in terms of the number of states and transitions, grows exponentially with the number of components in the system. For a system with many components, the solution of a system using a Markov model may be infeasible, even if the model is truncated. However, if the system level can be divided into independent modules, and the modules solved separately, then the separate results can be combined to achieve a complete analysis.

Advantages of using Markov model -

- It provides a flexible probabilistic model for analyzing system behavior.

- It is adaptable to complex redundant configurations, complex maintenance policies, complex fault-error handling models including intermittent faults, fault latency, reconfiguration, and the degraded modes of operation and common cause failures.
- It provides probabilistic solutions for modules to be plugged into other models such as block diagrams and fault trees.
- It allows accurate modeling of the event sequences with a specific pattern or order of occurrence.

Limitations of using Markov model -

- As the number of system components increases, there is an exponential growth in the number of states resulting in laborious analysis.
- The model can be difficult for users to construct and verify, and requires specific software for the analysis.
- The numerical solution step is available only with constant transition rates.
- Specific measures, such as MTTF and MTTR, are not immediately obtained from the standard solution of the Markov model, but require direct attention.

3.2.8 Petri Nets (PN)

Petri nets (PN) are a graphical tool for the representation and analysis of complex logical interactions between components of a system. Typical complex interactions that are naturally included in the Petri net language are concurrency, conflict, synchronization, mutual exclusion and resource limitation. An example of a generic Petri Net diagram is shown in the Figure.

A condition is valid in a given situation if the corresponding place is marked, i.e. contains at least one token. The dynamics of the system are represented by means of the movement of the tokens in the graph. A transition is enabled if its input places contain at least one token. An enabled transition may fire, and the transition firing removes one token from each input place and puts one token into each output place. The distribution of the tokens into the places is called marking. Starting from an initial marking, the application of the enabling and firing rules produces all the reachable markings called the reachability set. The reachability set provides all the states that the system can reach from an initial state.

Standard Petri nets do not carry the notion of time. However, many extensions have appeared in which timing is superimposed onto the Petri net. If a constant firing rate is assigned to each transition, the dynamics of the Petri nets can be analyzed by means of a continuous Markov time chain whose state space is isomorphic with the reachability set of the corresponding petri net.

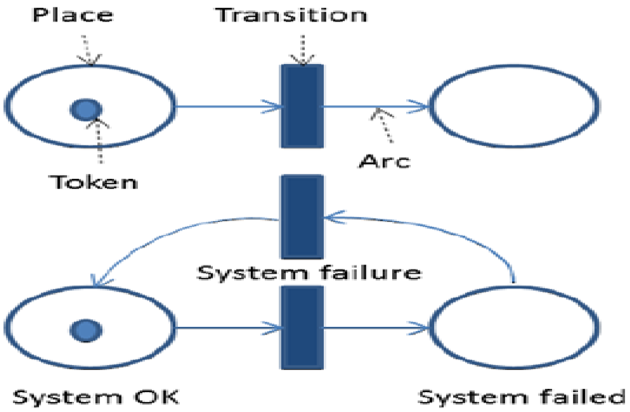

Fig. 3.2: Example of a generic Petri net diagram

PN graph is composed of three elements:

- Places (drawn as circle) that represent the conditions in which the system can be found.
- Transitions (drawn by bars) that represent the events that may change conditions in to another one.
- Arc (drawn as arrow) that connect places to transitions and transition to places and represent the logical admissible connections between conditions and events.

The key element of the Petri net analysis is a description of the system structure and its dynamic behavior in terms of primitive elements like places, transitions, arcs and tokens, of the Petri net language; this step requires the use of ad hoc software tools:

a) Structural qualitative analysis
b) Quantitative analysis: if constant firing rates are assigned to the Petri net transitions the quantitative analysis can be performed via the numerical solution of the corresponding Markov model, otherwise simulation is the only viable technique.

The Petri net can be utilized as a high level language to generate Markov models, and several tools in performance dependability analysis are based on this methodology. Petri nets provide also a natural environment for simulation. The use of Petri nets is recommended when complex logical interactions need to be taken into account mainly concurrency, conflict, synchronization, mutual exclusion, resource limitation. Moreover, PN are usually an easier and more natural language to describe a Markov model.

Advantages of using PN -

- Petri nets are suitable for representing complex interactions among hardware or software modules that are not easily modeled by other techniques.
- Petri Nets are a viable way of generating Markov models. In general, the description of the system by means of a Petri net requires far fewer elements than the corresponding Markov representation.
- The Markov model is generated automatically from the Petri net representation and the complexity of the analytical solution procedure is hidden to the modeler who interacts only at the Petri net level.
- In addition, the PN allow a qualitative structural analysis based only on the property of the graph. This structural analysis is, in general, less costly than the generation of the Markov model, and provides information useful to validate the consistency of the model.

Since the quantitative analysis is based on the generation and solution of the corresponding Markov model, most of the limitations are shared with the Markov analysis. The PN methodology requires the use of software.

3.2.9 Parts Derating and Selection

Derating can be defined as to limiting electrical, thermal and mechanical stresses on devices to levels below their specified capabilities in order to improve the reliability. For a reliable system the major contributing factors must be a conservative design approach incorporating part derating. The maximum permissible stress considered as the maximum levels in circuit applications. Parts are selected, taking into account two criteria; a part’s reliability and its ability to withstand the expected environmental and operational stresses when used in a system. Each component type, whether electronic (active or passive) or mechanical, must be evaluated to ensure that its temperature rating, construction, and other specific attributes like mechanical or other, are adequate for the intended environments.

Derating a part means subjecting it to reduced operational and environmental stresses, the goal being to reduce its failure probability to within the period of time required for proper product operation. When comparing the rated component strength to the expected stress, it is important to allow for a margin, which may be calculated based on the cumulative or fatigue stress and the component strength, or based on other engineering analysis criteria and methods. This margin allows the desired part reliability to be achieved regarding the particular fault modes and the respective causes.

The benefit of the part selection and derating practices is the achievement of the product's desired reliability. The only limitation is when there is no information on part reliability in any of the available databases or from the part manufacturer. In such a case, limitation extends to the part derating, when the derating guidelines involve reliability guidelines.

3.2.10 Fault Tree Analysis (FTA)

Fault Tree Analysis (FTA) is one of the most important logic and probabilistic techniques used in system reliability and safety assessment. FTA can be simply described as an analytical technique, whereby an undesired state of the system is specified usually a state that is critical from reliability standpoint, and the system is then analyzed in the context of its environment and operation to find all realistic ways in which the undesired event can occur.

The FT itself is a graphic model of the various parallel and sequential combinations of faults that will result in the occurrence of the predefined undesired event. A variety of elements are available for building a fault tree such as gates and events.

The faults can be events that are associated with component hardware failures, human errors, software errors, or any other pertinent events which can lead to the undesired event. A FT shows the logical interrelationships of basic events that lead to the undesired event, the top event of the FT. A fault tree is tailored to its top event that corresponds to some particular system failure mode, and the fault tree thus includes only those faults that contribute to this top event. Moreover, these faults are not exhaustive, they cover only the faults that are assessed to be realistic by the analyst.

Intrinsic to a fault tree is the concept that an outcome is a binary event i.e., either success or failure. A fault tree is composed of a complex of entities known as “gates” that serve to permit or inhibit the passage of fault logic up the tree. The gates show the relationships of events needed for the occurrence of a “higher” event. The “higher” event is the output of the gate; the “lower” events are the “inputs” to the gate. The gate symbol denotes the type of relationship of the input events required for the output event.

The qualitative evaluations basically transform the FT logic into logically equivalent forms that provide more focused information. The principal qualitative results that are obtained are the minimal cut sets (MCSs) of the top event. A cut set is a combination of basic events that can cause the top event. A minimal cut set (MCS) is the smallest combination of basic events that result in the top event. The basic events are the bottom events of the fault tree. Hence, the minimal cut sets relate the top event directly to the basic event causes. The set of MCSs for the top event represent all the ways that the basic events can cause the top event. A more descriptive name for a minimal cut set may be “minimal failure set.” Top event frequencies, failure or occurrence rates, and availabilities can also be calculated. These characteristics are particularly applicable if the top event is a system failure. This method is used in System Safety Analysis as well as in System Reliability Analysis. The FT can include basic events of Common Cause. The quantification of those events is made according to Common Cause Failure methods..

Some of the advantages of using FTA are:

- Can be started in early stages of a design and further developed in detail concurrently with design development. Identifies and records systematically the logical fault paths from a specific effect, back to the prime causes by using Boolean algebra.
- Allows easy conversion of logical models into corresponding probability measures.
- Assists in decision-making as a base and support tool due to variety of information obtained by a FTA.

Some of the disadvantages to using FTA are:

- FTA is not able to represent time or sequence dependency of events correctly.
- FTA has limitations with respect to reconfiguration or state-dependent behavior of systems.

These limitations can compensate for by combining FTA with Markov models, where Markov models are taken as basic events in fault trees.

3.3 Role of Redundancy

Redundancy is basically added to the system for providing the backup support after the failure of any component. Higher the no. of redundancies higher will be the reliability of the system. But due to more no. of redundancies the overall price/cost of the system may increase, which might be not a good solution instead of which a fresh new system may be cost cheaper. So to overcome this problem cost factor also added and different models has been created for getting the optimize no. of redundancies should be added to the system. There are two types of redundancies i.e. repairable and non-repairable. In my models I used repairable redundancies.

CHAPTER 4

METHODOLOGY

In this chapter, reliability model is developed for a system, in which three components are working as a series system. Standby redundancy and repairs are employed to achieve the higher system reliability goals.

4.1 System Description

In this project a system having three components connected in series has taken. Some standby redundancies have been added and all possible combinations have been made for finding out the reliability of the system at each possible combinations.

4.2 Development of Model

To achieve high reliability targets, standby redundancy is used for the component have highest failure rate to lowest failure rate component. In standby redundancy, component is connected in parallel. In case of failure of one component in parallel configuration second component start working. System will fail if all components fails in particular configuration. So here eight models will be developed by adding redundancy to different component to achieve maximum reliability.

4.2.1 Assumptions

The assumptions used in developing the models are:

- Every unit has multiple performance levels, which are active, standby and failed. This assumption is accepted as most industrial systems have multiple performance and conditions level.
- Two or more failure in system simultaneously will not be considered because the probability of occurrence of these states are negligible.
- The standby unit is cold standby. The assumption is acceptable because the system in standby will not wear out and failed.
- It is assumed that switching will always be successful.
- All failure rates and repair rates are constant over time and statistically independent. This assumption is acceptable due to the degradation of the system units is modeled as a multi-

state failure. In addition, this assumption can facilitate the mathematical modeling without losing the generality.

- The repaired unit has the priority to be sent back to the standby state. This assumption is based on the scenario that operating units will normally not be stopped.
- The repaired part performance is as good as a new part.
- There is no time lag between the component failure and repairing.
- Redundancies are of the same capacity and same time as the active units.
- Failure and repair rates follow exponential distribution.

4.2.2 Deriving space state models

4.2.2.1 Model 1

In this model three components are connected in series, as shown in Fig. 4.1, each component have two states, operating and failure. There is no redundancy added to this model.

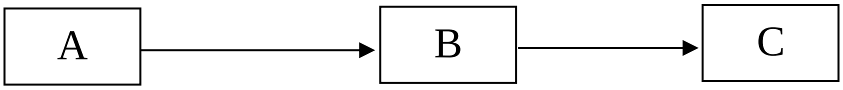

Fig. 4.1: Reliability Block Diagram of Model 1

From this model total 8 states can be created as shown in Table 4.1, out of which only 4 states are feasible and from these four only 1 state is in operating condition.

Table. 4.1: System Space State of Model 1

STATE	A	B	C	SYSTEM	TRANSITION FROM	TRANSITION TO
1	O	O	O	Operating		2,3,5
2	O	O	F	Failed	1	
3	O	F	O	Failed	1	
4	O	F	F	Not Feasible		
5	F	O	O	Failed	1	
6	F	O	F	Not Feasible		
7	F	F	O	Not Feasible		
8	F	F	F	Not Feasible		

Probabilistic modal of this system is developed with the help of above table in which failure and repair rate of all components are represented. Mathematical equations are generated with the help of all these states of components. Each mathematical equation represents the transition of components from operating state to failed state. Failure rate and repair rate are taken as constant. Every transition state depends upon its just previous state and it is independent of past history.

4.2.2.1 Model 2

In this model three components are connected in series, component C has one redundancy while component A and B has no redundancy as shown in below Figure 4.2. Here component C have three states operating, standby and failed.

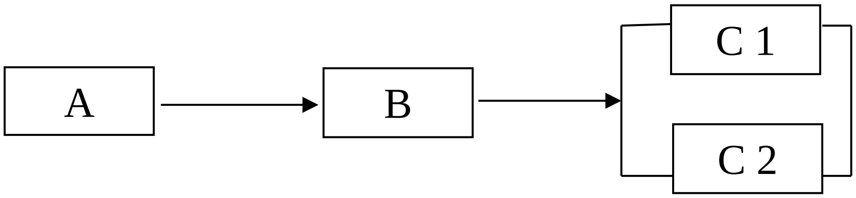

Fig. 4.2: Reliability Block Diagram of Model 2

From this model total 2×2×3×3 = 36 states has been derived out of which only 13 are feasible as shown in below Table 4.2.

Table 4.2: System Space State of Model 2

STATE	A	B	C1	C2	SYSTEM	TRANS. FROM	TRANS. TO
1	O	O	O	S	Operating	4,6,10	2
2	O	O	O	F	Operating	1,5,7,11	3
3	O	O	S	O	Operating	2,8,12	4
4	O	O	F	O	Operating	3,5,9,13	1
5	O	O	F	F	failed		2,4
6	O	F	O	S	failed		1
7	O	F	O	F	failed		2
8	O	F	S	O	failed		3
9	O	F	F	O	failed		4
10	F	O	O	S	failed		1
11	F	O	O	F	failed		2
12	F	O	S	O	failed		3
13	F	O	F	O	failed		4

4.2.2.3 Model 4

In this model component B and component C have one redundancy each as shown Fig.4.3.

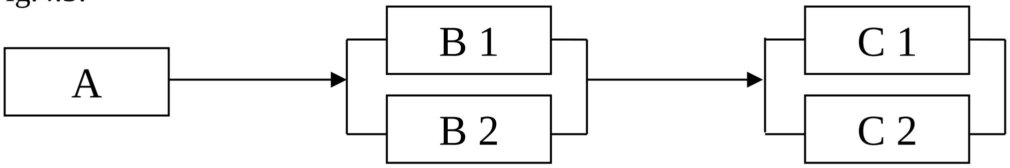

Fig. 4.3: Reliability Block Diagram of Model 4

In this model total 2×3×3×3×3 = 162 has been derived out of which only 40 states are feasible as shown in below table and transition of one state to another state is also shown in Table 4.3.

Table 4.3: System Space State of Model 4

STATE	A	B1	B2	C1	C2	SYSTEM	TRANS. TO	TRANS. FROM
1	O	O	F	O	F	Operating	2,4,6,16,25	5,20
2	O	O	F	O	S	Operating	3,7,17,26	1,21
3	O	O	F	F	O	Operating	4,5,8,18,27	2,22
4	O	O	F	F	F	Failed		1,3
5	O	O	F	S	O	Operating	1,10,19,28	3,24
6	O	O	S	O	F	Operating	7,9,11,29	1,10
7	O	O	S	O	S	Operating	8,12,30	2,6
8	O	O	S	F	O	Operating	9,10,13,31	3,7
9	O	O	S	F	F	Failed		6,8
10	O	O	S	S	O	Operating	6,15,32	5,8
11	O	F	O	O	F	Operating	12,14,16,20,33	6,15
12	O	F	O	O	S	Operating	13,17,21,34	7,11
13	O	F	O	F	O	Operating	14,15,18,22,35	8,12
14	O	F	O	F	F	Failed		11,13
15	O	F	O	S	O	Operating	11,19,24,36	10,13
16	O	F	F	O	F	Failed		1,11
17	O	F	F	O	S	Failed		2,12
18	O	F	F	F	O	Failed		3,13
19	O	F	F	S	O	Failed		5,15
20	O	S	O	O	F	Operating	1,21,23,37	11,24
21	O	S	O	O	S	Operating	2,22,38	12,20
22	O	S	O	F	O	Operating	3,23,24,39	13,21
23	O	S	O	F	F	Failed		20,22

Table 4.3: System Space State of Model 4(continued)

STATE	A	B1	B2	C1	C2	SYSTEM	TRANS. TO	TRANS. FROM
24	O	S	O	S	O	Operating	5,20,40	15,22
25	F	O	F	O	F	Failed		1
26	F	O	F	O	S	Failed		2
27	F	O	F	F	O	Failed		3
28	F	O	F	S	O	Failed		5
29	F	O	S	O	F	Failed		6
30	F	O	S	O	S	Failed		7
31	F	O	S	F	O	Failed		8
32	F	O	S	S	O	Failed		10
33	F	F	O	O	F	Failed		11
34	F	F	O	O	S	Failed		12
35	F	F	O	F	O	Failed		13
36	F	F	O	S	O	Failed		15
37	F	S	O	O	F	Failed		20
38	F	S	O	O	S	Failed		21
39	F	S	O	F	O	Failed		22
40	F	S	O	S	O	Failed		24

4.2.2.4 Model 8

In this model each component has one redundancy added as shown in Fig. 4.4. Each component has three states i.e. operating, standby and failed.

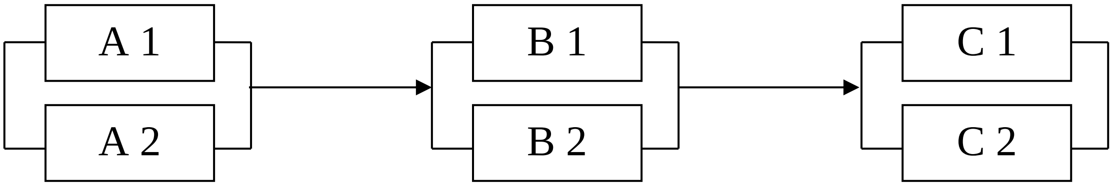

Fig. 4.4: Reliability Block Diagram of Model 8

In this model total 3×3×3×3×3×3 = 729 states has been derived out of which only 112 are feasible. In the below table 4.4 all the feasible states and their transition has shown in appendix A1.

4.2.3 Transition diagrams

The transition diagrams show the how a state transform into another states by birth process i.e. failure of component and to death process i.e. repairing of component to reach previous state.

4.2.3.1 Model 1

In this diagram each node represent the state of the system and arrow shows the transition as shown in Fig.4.5.

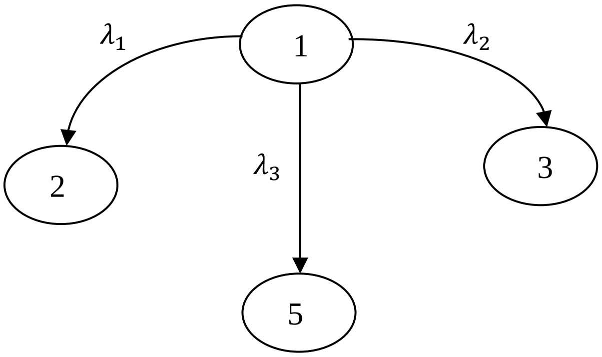

Fig. 4.5: Transition diagram of Model 1

4.2.3.2 Model 2

In this diagram each node represent the state of the system and arrow shows the transition as shown in Fig.4.6.

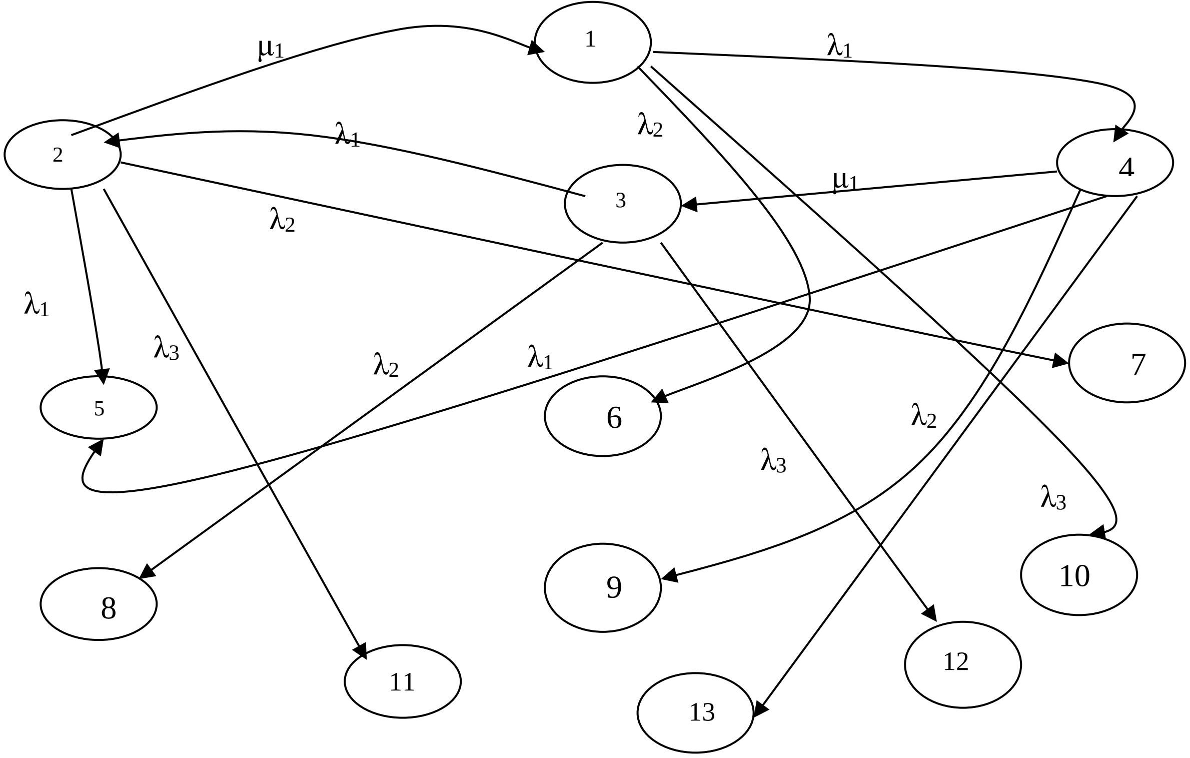

Fig. 4.6: Transition diagram of Model 2

The transition diagrams for model 3,4,5,6,7 and 8 are not incorporated in text due to space limitations.

4.2.4 Markov based mathematic work

The differential equations are developed using Markov birth-death process. In birth process, there is one step change in the probability function in forward direction due to failures of the components. While due to repairs of the components, there is one backward change in the probability function like death process. The reliability analysis related with a discrete state continuous time model, called a Markov process. The other assumptions of Markov models are:

- At any given time the system is either in operating or standby or in failed state.
- The state of the system changes as time progresses.
- The transition of the system from one state to the other takes place instantaneously. It means time span between two states is assumed to be zero.
- The failure and repair rates are constant (follows exponential distribution).

4.2.4.1 Markov model 1

The ordinary differential equations have been developed from the state transition possibilities with the help of Markov process.

$$\frac{d}{dt} P_1(t) = -(\lambda_1 + \lambda_2 + \lambda_3) P_1(t)$$

$$\frac{d}{dt} P_2(t) = \lambda_1 P_1(t)$$

$$\frac{d}{dt} P_3(t) = \lambda_2 P_1(t)$$

$$\frac{d}{dt} P_5(t) = \lambda_3 P_1(t)$$

Where, initial condition at time t = 0; $P_1(t) = 1$ and $P_2(t), P_3(t) \ldots\ldots P_4(t) = 0$ and at any time ‘t’ $\sum_{n=1}^{n=4} Pn\ (t) = 1$.

4.2.4.2 Markov model 2

For model 2 thirteen differential equations has derived.

$$\frac{d}{dt} P_1(t) = -(\lambda_1 + \lambda_2 + \lambda_3) P_1(t) + \mu_{11} P_2(t)$$

$$\frac{d}{dt} P_2(t) = -(\lambda_1 + \lambda_2 + \lambda_3) P_2(t) + \lambda_{11} P_3(t) - \mu_{11} P_2(t)$$

$$\frac{d}{dt}P_3(t) = -(\lambda_2+\lambda_3)P_3(t) - \lambda_{11}P_3(t) + \mu_1 P_4(t)$$

$$\frac{d}{dt}P_4(t) = -(\lambda_2+\lambda_3)P_4(t) - \lambda_{11}P_4(t) + \lambda_1 P_1(t) - \mu_1 P_4(t)$$

$$\frac{d}{dt}P_5(t) = \lambda_1 P_2(t) + \lambda_{11} P_4(t)$$

$$\frac{d}{dt}P_6(t) = \lambda_2 P_1(t)$$

$$\frac{d}{dt}P_7(t) = \lambda_2 P_2(t)$$

$$\frac{d}{dt}P_8(t) = \lambda_2 P_3(t)$$

$$\frac{d}{dt}P_9(t) = \lambda_2 P_4(t)$$

$$\frac{d}{dt}P_{10}(t) = \lambda_3 P_1(t)$$

$$\frac{d}{dt}P_{11}(t) = \lambda_3 P_2(t)$$

$$\frac{d}{dt}P_{12}(t) = \lambda_3 P_3(t)$$

$$\frac{d}{dt}P_{13}(t) = \lambda_3 P_4(t)$$

With initial condition at time t = 0; $P_1(t) = 1$ and $P_2(t), P_3(t)$ …….. $P_{13}(t) = 0$ And at any time 't' $\sum_{n=1}^{n=13} Pn\,(t) = 1$.

The mathematical models for model 3,4,5,6,7 and 8 are not incorporative in text due to space limitations.

4.2.5 Develop MatLab Code

After all these process these ordinary differential equations are written in the MatLab in the form of codes and solve these with the use of "ode45" function. After successfully running of codes in MatLab it gives the probabilities of each state of each model at a given failure rate and repair rate within a given time period.

4.2.5.1 Model 1

This model have only four feasible states.

```
function dydt = model1(t,y,lemda1,lemda2,lemda3,mu1,mu2,mu3)
%tspan=[0,1000,2000];
%y0=[1;0;0;0];
dydt= [(-lemda1-lemda2-lemda3)*y(1);
     lemda1*y(1);
```

```
    lemda2*y(1);
    lemda3*y(1);];
```

4.2.5.2 Model 2

```
function dydt = model2(t,y,lemda1,lemda2,lemda3,mu1,mu2,mu3)
%tspan=[0,1000,2000];
%y0=[1;0;0;0;0;0;0;0;0;0;0;0;0];
dydt = [-(lemda1+lemda2+lemda3)*y(1)+mu1*y(2);
      -(lemda1+lemda2+lemda3)*y(2)+lemda1*y(3)-mu1*y(2);
      -(lemda1+lemda2+lemda3)*y(3)+mu1*y(4);
      -(lemda1+lemda2+lemda3)*y(4)+lemda1*y(1)-mu1*y(4);
      lemda1*y(2)+lemda1*y(4);
      lemda2*y(1);
      lemda2*y(2);
      lemda2*y(3);
      lemda2*y(4);
      lemda3*y(1);
      lemda3*y(2);
      lemda3*y(3);
      lemda3*y(4);];
```

MatLab codes for model-4 and model-8 has shown in appendix A2 and A3 respectively. While codes of models 3,5,6 and 7 are inappropriate due to space limitations.

4.3 Cost Consideration

Each component have different cost. For my project to make it more realistic I also considered the cost concession factor i.e. if I take 2 or more units of same component how much concession I will get which make my system more economical and reliable. For this lots of market has to be done for getting maximum concession.

4.4 Optimization

For optimization of reliability with the cost factor, Dynamic programming with backward recursive function is used.

4.4.1 Dynamic programming

Dynamic programming is an optimizing technique which divides the main problem into sub problem as the no. of variables, solve the sub problem and the after integrating the solution get the solution of main problem. The following terminology used in dynamic programming.

- Stage (i)- Each sub-problem of original problem is known as i.
- Alternative, m(i) - For a given stage i, each choice is known as alternative.
- State variable, x(i)- Possible value of a resources within its permitted range at a given stage i is known as state variable
- Recursive function- a function which links the measure of performance of interest of current stage with the cumulative measure of performance of previous stage.

$$f_i(x_i) = \max[R(m_i)]$$

CHAPTER 5

CASE STUDY

In this chapter discussed about the actual system based on this project. Reliability model of refrigeration system of cold chain are developed in which three component are working in series system. These components i.e. low pressure compressor, high pressure compressor and condenser pump are very critical for any of the refrigeration system as failure of any component leads to shut down of whole system. Standby redundancies are employed as a means to achieve the higher reliability within budget.

5.1 Description of Cold Chain Refrigeration System

Cold chain refrigeration system is made up of many units/components out of which low pressure compressor, high pressure compressor and condenser pump is critical components as failure of anyone lead to shutdown of the system. Following layout is showing the all components used in refrigeration system and showing the flow of refrigerant as shown in Fig. 5.1.

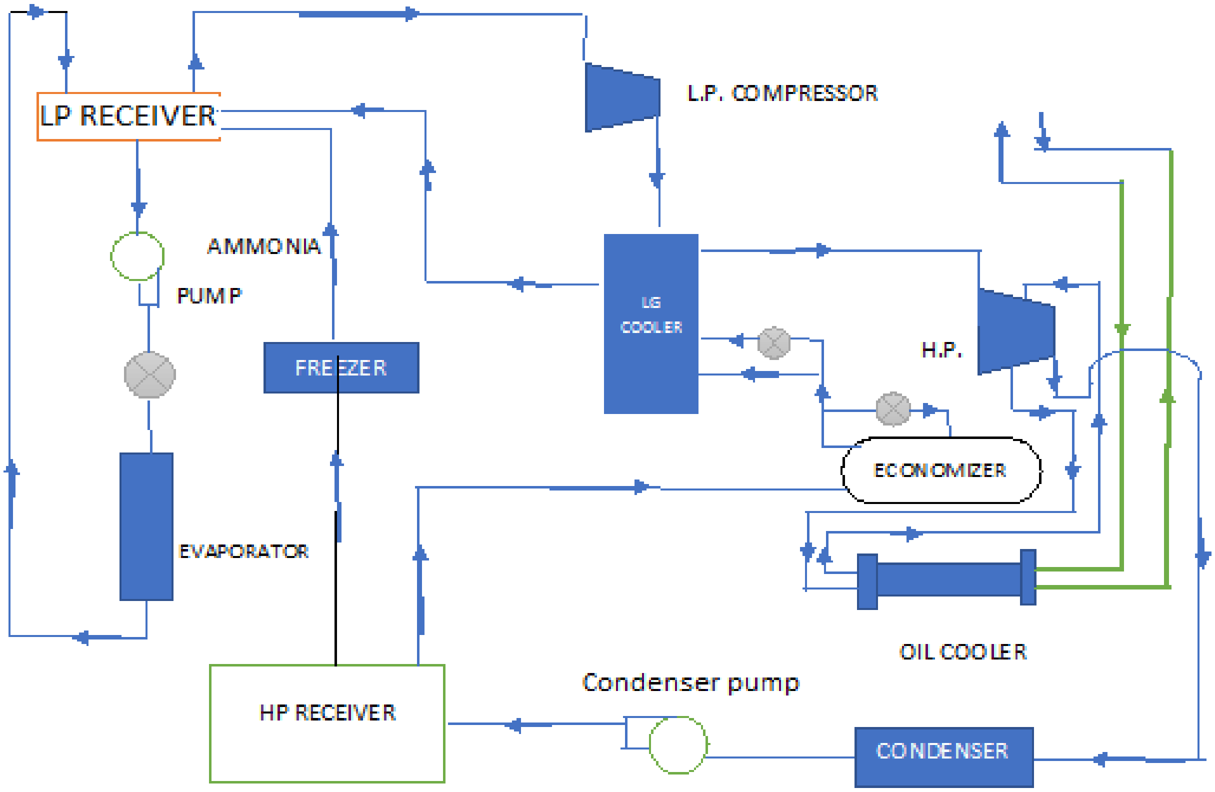

Fig. 5.1: Flow diagram of refrigeration system

5.1.1 Low pressure compressor

The compressor receives low pressure gas from the evaporator and converts it to high pressure gas.

5.1.2 High pressure compressor

This compressor further increases the pressure of gas up to a required pressure.

5.1.3 Condenser pump

A condensate pump is used to pump the condensate liquid refrigerant produced in refrigeration, system. It is electrically powered centrifugal pumps.

5.2 Assumptions

Following assumptions have taken into consideration during the development of models

- Every component nit has multiple states, which are active, standby and failed.
- Two or more failure in system simultaneously will not be considered because the probability of occurrence of these states is negligible.
- The standby component is cold standby. The assumption is acceptable because the component in standby will not wear out and failed.
- It is assumed that switching will always be successful and considered as perfect.
- All failure and repair rates are constant over time and statistically independent. This assumption facilitates the mathematical modeling without losing the generality.
- The repaired component has the priority to be sent back to the standby state. This assumption is based on the scenario that operating units will normally not be stopped.
- A repaired component is considered as good as new.
- Sufficient repair facilities are provided, i.e. no waiting time to start the repairs.
- Standby units are of the same nature and capacity as the active units.
- Failure and repair rates follow exponential distribution.

5.3 Deriving System State Space Model

For the system each component have three possibilities of state i.e. operating, standby and failed. λ and μ represent the failure rate and repair rate of component respectively. $\lambda1$ & $\mu1$ represent the failure rate and repair rate of condenser pump, $\lambda2$ & $\mu2$ represent the failure rate and repair rate of high pressure compressor and $\lambda3$ and $\mu3$ represent failure rate and repair rate of low pressure compressor as shown in table 5.1. All the calculation of reliability/probability of the system has done for 2000 hours.

Table 5.1; Failure rate(λ) and repair rate(μ) of each components

Component	Failure Rate (λ)	Repair Rate (μ)
LP Compressor	λ3 = 1/30000	μ3 = 1/249
HP Compressor	λ2 = 1/25000	μ2 = 1/150
Condenser Pump	λ1 = 1/20000	μ1 = 1/42

5.3.1 Model 1

It is the basic series model in which all three components are connected in series as shown in Fig 5.2.

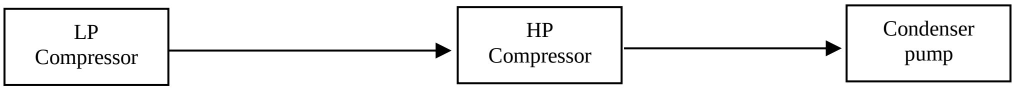

Fig. 5.2: Reliability Block Diagram of Model 1

As discussed earlier in previous chapter this state have four feasible states out of which three states are absorbing states so total reliability of this model for 2000 hours is probability of only working state.

$$R(1) = P_1 = 0.7814$$

5.3.2 Model 2

The combination of this model has shown in Fig. 5.3.

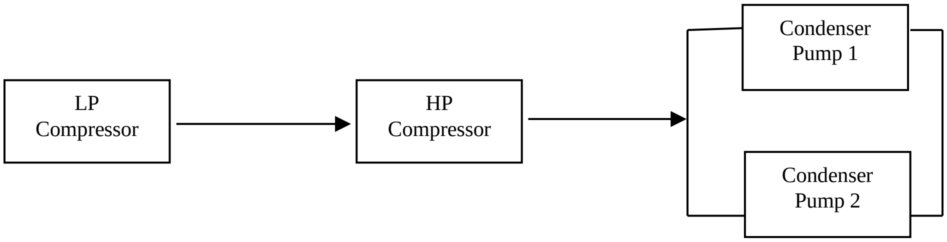

Fig.5.3: Reliability Block Diagram of Model 2

In this model total 13 feasible states out of which only four are operating so total reliability of the system for 2000 hours is sum of probability of working states i.e.

$$R(2) = P_1 + P_2 + P_3 + P_4 = 0.8634$$

5.3.3 Model 3

The combination of this model has shown in Fig 5.4.

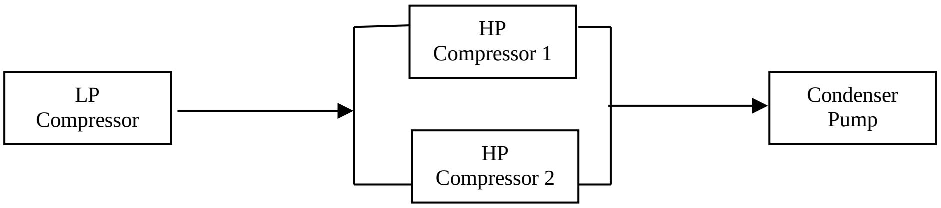

Fig. 5.4: Reliability Block Diagram of Model 3

Total reliability of this model is sum of probability of working states i.e.

$$R(3) = P_1 + P_2 + P_3 + P_4 = 0.8461$$

5.3.4 Model 4

The combination of this model has shown in Fig 5.5.

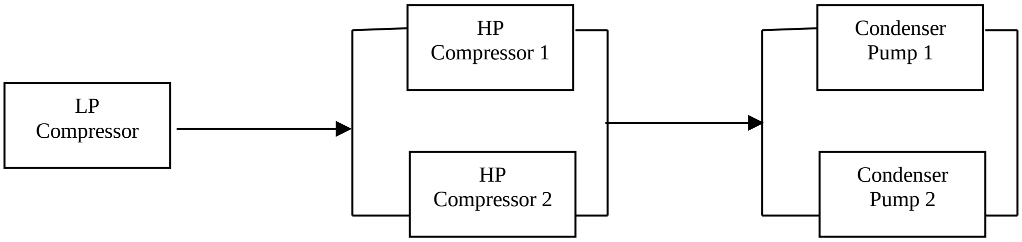

Fig. 5.5: Reliability Block Diagram of Model 4

Total reliability of this model is sum of probability of working states i.e.

$$R(4)=P_1 + P_2 + P_3 + P_5 + P_6 + P_7 + P_8 + P_{10} + P_{11} + P_{12} + P_{13} + P_{15} + P_{20} + P_{21} + P_{22} + P_{24} = 0.9274$$

5.3.5 Model 5

The combination of this model has shown in Fig 5.6.

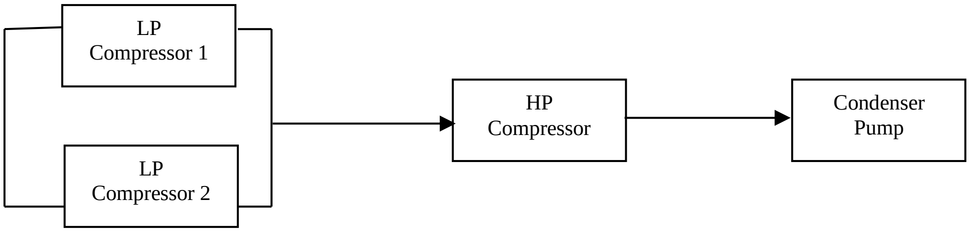

Fig. 5.6: Reliability Block Diagram of Model 5

Total reliability of this model is sum of probability of working states i.e.

$$R(5) = P_1 + P_2 + P_3 + P_4 = 0.8349$$

5.3.6 Model 6

The combination of this model has shown in Fig. 5.7.

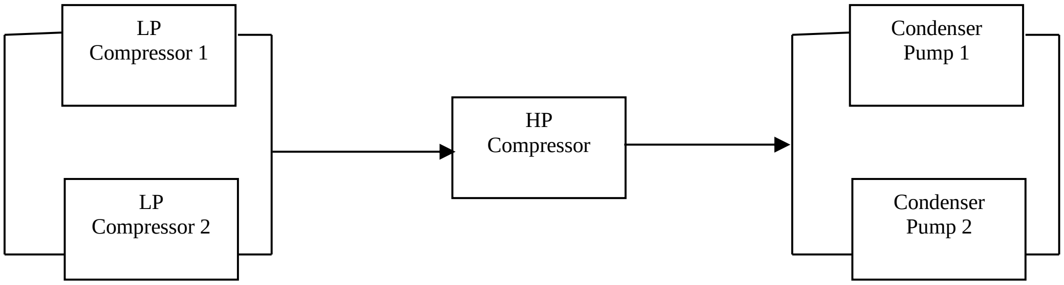

Fig. 5.7: Reliability Block Diagram of Model 6

Reliability of this model is sum of probability of working states i.e.

$R(6)=P_1 + P_2 + P_3 + P_5 + P_6 + P_7 + P_8 + P_{10} + P_{11} + P_{12} + P_{13} + P_{15} + P_{20} + P_{21} + P_{22} + P_{24} = 0.9130$

5.3.7 Model 7

The combination of this model has shown in Fig. 5.8.

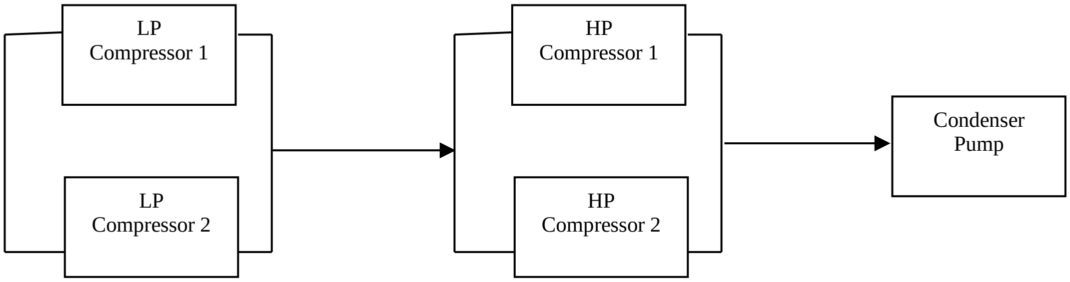

Fig 5.8: Reliability Block Diagram of Model 7

Reliability of this model is sum of probability of working states i.e.

$R(7) = P_1 + P_2 + P_3 + P_5 + P_6 + P_7 + P_8 + P_{10} + P_{11} + P_{12} + P_{13} + P_{15} + P_{20} + P_{21} + P_{22} + P_{24} = 0.8913$

5.3.8 Model 8

The combination of this model has shown in Fig. 5.9.

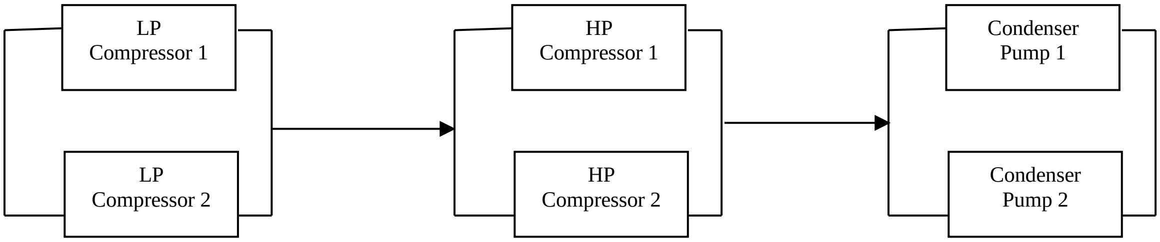

Fig. 5.9: Reliability Block Diagram of Model 8

Reliability of this model is sum of probability of working states i.e.

$R(8) = P_1 + P_2 + P_3 + P_4 + P_5 + P_6 + P_7 + P_8 + P_9 + P_{10} + P_{11} + P_{12} + P_{13} + P_{14} + P_{15} + P_{16} + P_{17} + P_{18} + P_{19} + P_{20} + P_{21} + P_{22} + P_{23} + P_{24} + P_{25} + P_{26} + P_{27} + P_{28} + P_{29} + P_{30} + P_{31} + P_{32} + P_{33} + P_{34} + P_{35} + P_{36} + P_{37} + P_{38} + P_{39} + P_{40} + P_{41} + P_{42} + P_{43} + P_{44} + P_{45} + P_{46} + P_{47} + P_{48} + P_{49} + P_{50} + P_{51} + P_{52} + P_{53} + P_{54} + P_{55} + P_{56} + P_{57} + P_{58} + P_{59} + P_{60} + P_{61} + P_{62} + P_{63} + P_{64} = 0.9994$

5.3.9 Model 9

This model is have only components of C connected parallel, this is required in dynamic programming as shown in Fig.5.10.

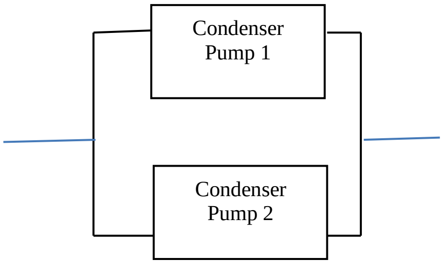

Fig. 5.10: Reliability Block Diagram of Model 9

Table 5.2: System Space State of Model 9

STATE	C1	C2	SYSTEM STATE	TRANSITION TO	TRANSITION FROM
1	O	S	OPERATING	4	2
2	O	F	OPERATING	5,1	3
3	S	O	OPERATING	2	4
4	F	O	OPERATING	3,5	1
5	F	F	ABORBING		2,4

$$R(9) = P_1 + P_2 + P_3 + P_4 = 0.9998$$

Other models are not incorporated in text due to space limitations.

5.4 Optimization of system

Our main concern of optimization is to find the best combination/model with maximum reliability at given budget i.e. 2000000/-. Price of each component are given below in the Table 5.3. To make this problem more realistic cost concession also has taken into consideration when buy same components more than one unit.

Table 5.3: Cost of components

S. No.	Component	Cost/unit(if buy one unit)	Cost/unit(if buy two unit)
1	LP Compressor	700000/-	630000/-
2	HP Compressor	500000/-	450000/-
3	Condenser Pump	100000/-	92000/-

By using backward recursive approach of Dynamic programming optimization has done.

Total available budget = 2000000/-

Terminology used

- Stage (n to go) – Component
- State (s)- amount of money available
- Decision variables (X)- How many units of each component
- Criterion of effectiveness (f(s,X))- Maximize Reliability

Table 5.4 : N=1, One more stage to go

S	X(3)	f1*(s)	X*(3)
100000-183999	1	0.9048	1
184999-800000	2	0.9998	2

Table 5.5, N=2, Two more stage to go

S	X(2)		f2*(s)	X*(2)
	1	2		
600000-683999	0.8465	-	0.8465	1
683000-999999	0.9229	-	0.9229	1
1000000-1083999	0.9229	0.9044	0.9229	1
1084000-1300000	0.9229	0.9994	0.9994	2

Table 5.6 : N=3, Three more stage to go

S	X1		f3*(s)	X*(1)
	1	2		
1300000-1383999	0.7814		0.7814	1
1384000-1783999	0.8634		0.8634	1
1784000-1859999	0.9274		0.9274	1
1860000-1943999	0.9274	0.8349	0.9274	1
1944000-2000000	0.9274	0.9130	0.9274	1

CHAPTER 6

CONCLUSION AND SCOPE FOR FUTURE WORK

In this chapter, conclusion and scope for future work is presented.

6.1 Conclusion

In this thesis, a methodology is presented for improving the system reliability within budgetary constraint. To improve the reliability of system under budgetary constraint, first reliability evaluation of repairable system are performed which is based on Markov approach. Secondly optimization done with cost factor using Dynamic programming. Constant failure and repair rates are considered for the different components in the system. Standby redundancy has been used to increase the reliability of system. Dynamic programming has given the optimized result for the model-6 with reliability 0.9130, with the total cost of 1784000/- which is under the budget available i.e. 2000000/-. Model-6 has one redundancy of HP Compressor and one redundancy of Condenser pump. The use of dynamic programming for optimization of reliability under cost constraint for higher number of redundancy reduces the complexity of calculation.

The proposed analysis is useful to design a system with improved reliability and to achieve higher reliability goals under budgetary constraint.

6.2 Future Scope

This section presents a brief on potential future directions.

- An exponential distribution is assumed for failure and repair time due to the limitation of Markov approach. This assumption can be relaxed and appropriate non-exponential distribution such as Weibull for failure time and Lognormal for repair time can be considered for more realistic analysis.
- Software can be developed based on the proposed methodology to design a system with reliability goal.
- Warm standby can be considered in place of cold standby redundancy, which is more realistic in nature.
- Imperfect switching can also be considered in system modeling.

REFERENCES

1. Postnikov, I., Stennikov, V., Mednikova, E., & Penkovskii, A. (2017). Methodology for optimization of component reliability of heat supply systems. Applied Energy.
2. Goo, B., Lee, J., Seo, S., Chang, D., & Chung, H. (2017). Design of reliability critical system using axiomatic design with FMECA. International Journal of Naval Architecture and Ocean Engineering.
3. Manesh, M. K., Rad, M. P., & Rosen, M. A. (2018). New procedure for determination of availability and reliability of complex cogeneration systems by improving the approximated Markov method. Applied Thermal Engineering, 138, 62-71.
4. Gong, C., & Zhou, W. (2017). Improvement of equivalent component approach for reliability analyses of series systems. Structural Safety, 68, 65-72.
5. Hou, K., Jia, H., Xu, X., Liu, Z., & Jiang, Y. (2016). A continuous time Markov chain based sequential analytical approach for composite power system reliability assessment. IEEE Transactions on Power Systems, 31(1), 738-748.
6. Feizabadi, M., & Jahromi, A. E. (2017). A new model for reliability optimization of series-parallel systems with non-homogeneous components. Reliability Engineering & System Safety, 157, 101-112.
7. Ge, Q., Peng, H., van Houtum, G. J., & Adan, I. (2018). Reliability optimization for series systems under uncertain component failure rates in the design phase. International Journal of Production Economics, 196, 163-175.
8. Li, Y., Cui, L., & Lin, C. (2017). Modeling and analysis for multi-state systems with discrete-time Markov regime-switching. Reliability Engineering & System Safety, 166, 41-49.
9. Chen, T. C., & You, P. S. (2005). Immune algorithms-based approach for redundant reliability problems with multiple component choices. Computers in Industry, 56(2), 195-205.
10. Liu, P. X., Zuo, M. J., & Meng, M. Q. H. (2003). Using neural network function approximation for optimal design of continuous-state parallel–series systems. Computers & Operations Research, 30(3), 339-352.

11. Huang, C. L. (2015). A particle-based simplified swarm optimization algorithm for reliability redundancy allocation problems. Reliability Engineering & System Safety, 142, 221-230.
12. Najafi, A. A., Karimi, H., Chambari, A., & Azimi, F. (2013). Two metaheuristics for solving the reliability redundancy allocation problem to maximize mean time to failure of a series–parallel system. Scientia Iranica, 20(3), 832-838.
13. Chambari, A., Rahmati, S. H. A., & Najafi, A. A. (2012). A bi-objective model to optimize reliability and cost of system with a choice of redundancy strategies. Computers & Industrial Engineering, 63(1), 109-119.
14. Sung, C. S., & Cho, Y. K. (2000). Reliability optimization of a series system with multiple-choice and budget constraints. European Journal of Operational Research, 127(1), 159-171.
15. Liang, Y. C., & Chen, Y. C. (2007). Redundancy allocation of series-parallel systems using a variable neighborhood search algorithm. Reliability Engineering & System Safety, 92(3), 323-331.
16. Heydari, M., & Sullivan, K. M. (2018). An integrated approach to redundancy allocation and test planning for reliability growth. Computers & Operations Research, 92, 182-193.
17. Tavakkoli-Moghaddam, R., Safari, J., & Sassani, F. (2008). Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm. Reliability Engineering & System Safety, 93(4), 550-556.
18. Ouzineb, M., Nourelfath, M., & Gendreau, M. (2008). Tabu search for the redundancy allocation problem of homogenous series–parallel multi-state systems. Reliability Engineering & System Safety, 93(8), 1257-1272.
19. Kim, H., & Kim, P. (2017). Reliability–redundancy allocation problem considering optimal redundancy strategy using parallel genetic algorithm. Reliability Engineering & System Safety, 159, 153-160.
20. Xu, L. Y., & Cheng, Y. F. (2012). Reliability and failure pressure prediction of various grades of pipeline steel in the presence of corrosion defects and pre-strain. International Journal of Pressure Vessels and Piping, 89, 75-84.

21. Li, L. L., Lv, C. M., Tseng, M. L., & Sun, J. (2018). Reliability measure model for electromechanical products under multiple types of uncertainties. Applied Soft Computing, 65, 69-78.
22. Kwag, H. G., & Kim, J. O. (2014). Reliability modeling of demand response considering uncertainty of customer behavior. Applied Energy, 122, 24-33.
23. Ardakan, M. A., Hamadani, A. Z., & Alinaghian, M. (2015). Optimizing bi-objective redundancy allocation problem with a mixed redundancy strategy. ISA transactions, 55, 116-128.
24. Balák, J., & Ždánsky, P. (2017). Modelling of transition of system with standby redundancy into failed state. Procedia engineering, 192, 10-15.
25. Lisnianski, A. (2007). Extended block diagram method for a multi-state system reliability assessment. Reliability Engineering & System Safety, 92(12), 1601-1607.
26. Lisnianski, A., Elmakias, D., Laredo, D., & Haim, H. B. (2012). A multi-state Markov model for a short-term reliability analysis of a power generating unit. Reliability engineering & system safety, 98(1), 1-6.
27. Jha, D. K., Srivastav, A., Mukherjee, K., & Ray, A. (2015, July). Depth estimation in Markov models of time-series data via spectral analysis. In American Control Conference (ACC), 2015 (pp. 5812-5817). IEEE.
28. Ziaei-Nia, A., Tadayonfar, G. R., & Eskandari-Naddaf, H. (2018). Dynamic Cost Optimization Method of Concrete Mix Design. Materials Today: Proceedings, 5(2), 4669-4677.
29. Tripathy, P. K., Dash, R. K., & Tripathy, C. R. (2015). A dynamic programming approach for layout optimization of interconnection networks. Engineering Science and Technology, an International Journal, 18(3), 374-384.
30. Charles E. Ebeling, (2000). An introduction to reliability and maintainability engineering, University of Daton, Tata Magraw Hills.
31. R Panneerselvam (2009). Operations research, University of Pondicherry, Phi Learning Pvt. Ltd.

APPENDIX A1

System Space State of Model 8:

Table A1 : System Space State of Model 8

STATE	A1	A2	B1	B2	C1	C2	SYSTEM STATE
1	O	S	O	S	O	S	Operating
2	O	S	O	S	O	F	Operating
3	O	S	O	F	O	S	Operating
4	O	S	O	F	O	F	Operating
5	O	F	O	S	O	S	Operating
6	O	F	O	S	O	F	Operating
7	O	F	O	F	O	S	Operating
8	O	F	O	F	O	F	Operating
9	O	S	O	S	S	O	Operating
10	O	S	O	F	S	O	Operating
11	O	F	O	S	S	O	Operating
12	O	F	O	F	S	O	Operating
13	O	S	O	S	F	O	Operating
14	O	S	O	F	F	O	Operating
15	O	F	O	S	F	O	Operating
16	O	F	O	F	F	O	Operating
17	O	S	S	O	O	S	Operating
18	O	S	S	O	O	F	Operating
19	O	F	S	O	O	S	Operating
20	O	F	S	O	O	F	Operating
21	O	S	S	O	S	O	Operating
22	O	F	S	O	S	O	Operating
23	O	S	S	O	F	O	Operating
24	O	F	S	O	F	O	Operating
25	O	S	F	O	O	S	Operating
26	O	S	F	O	O	F	Operating

Table A1 : System Space State of Model 8(continued)

STATE	A1	A2	B1	B2	C1	C2	SYSTEM STATE
27	O	F	F	O	O	S	Operating
28	O	F	F	O	O	F	Operating
29	O	S	F	O	S	O	Operating
30	O	F	F	O	S	O	Operating
31	O	S	F	O	F	O	Operating
32	O	F	F	O	F	O	Operating
33	S	O	O	S	O	S	Operating
34	S	O	O	S	O	F	Operating
35	S	O	O	F	O	S	Operating
36	S	O	O	F	O	F	Operating
37	S	O	O	S	S	O	Operating
38	S	O	O	F	S	O	Operating
39	S	O	O	S	F	O	Operating
40	S	O	O	F	F	O	Operating
41	S	O	S	O	O	S	Operating
42	S	O	S	O	O	F	Operating
43	S	O	S	O	S	O	Operating
44	S	O	S	O	F	O	Operating
45	S	O	F	O	O	S	Operating
46	S	O	F	O	O	F	Operating
47	S	O	F	O	S	O	Operating
48	S	O	F	O	F	O	Operating
49	F	O	O	S	O	S	Operating
50	F	O	O	S	O	F	Operating
51	F	O	O	F	O	S	Operating
52	F	O	O	F	O	F	Operating
53	F	O	O	S	S	O	Operating
54	F	O	O	F	S	O	Operating

Table A1 : System Space State of Model 8(continued)

STATE	A1	A2	B1	B2	C1	C2	SYSTEM STATE
55	F	O	O	S	F	O	Operating
56	F	O	O	F	F	O	Operating
57	F	O	S	O	O	S	Operating
58	F	O	S	O	O	F	Operating
59	F	O	S	O	S	O	Operating
60	F	O	S	O	F	O	Operating
61	F	O	F	O	O	S	Operating
62	F	O	F	O	O	F	Operating
63	F	O	F	O	S	O	Operating
64	F	O	F	O	F	O	Operating
65	O	S	O	S	F	F	Failed
66	O	S	O	F	F	F	Failed
67	O	F	O	S	F	F	Failed
68	O	F	O	F	F	F	Failed
69	O	S	S	O	F	F	Failed
70	O	F	S	O	F	F	Failed
71	O	S	F	F	O	S	Failed
72	O	S	F	F	O	F	Failed
73	O	F	F	F	O	S	Failed
74	O	F	F	F	O	F	Failed
75	O	S	F	F	S	O	Failed
76	O	F	F	F	S	O	Failed
77	O	S	F	O	F	F	Failed
78	O	S	F	F	F	O	Failed
79	O	F	F	O	F	F	Failed
80	O	F	F	F	F	O	Failed
81	O	F	F	F	F	F	Failed
82	S	O	O	S	F	F	Failed

Table A1 : System Space State of Model 8(continued)

STATE	A1	A2	B1	B2	C1	C2	SYSTEM STATE
83	S	O	O	F	F	F	Failed
84	S	O	S	O	F	F	Failed
85	S	O	F	F	O	S	Failed
86	S	O	F	F	O	F	Failed
87	S	O	F	F	S	O	Failed
88	S	O	F	O	F	F	Failed
89	S	O	F	F	F	O	Failed
90	F	F	O	S	O	S	Failed
91	F	F	O	S	O	F	Failed
92	F	F	O	F	O	S	Failed
93	F	F	O	F	O	F	Failed
94	F	F	O	S	S	O	Failed
95	F	F	O	F	S	O	Failed
96	F	O	O	S	F	F	Failed
97	F	O	O	F	F	F	Failed
98	F	F	O	S	F	O	Failed
99	F	F	O	F	F	O	Failed
100	F	F	S	O	O	F	Failed
101	F	F	S	O	S	O	Failed
102	F	O	S	O	F	F	Failed
103	F	F	S	O	F	O	Failed
104	F	F	S	O	F	F	Failed
105	F	O	F	F	O	S	Failed
106	F	O	F	F	O	F	Failed
107	F	F	F	O	O	S	Failed
108	F	F	F	O	O	F	Failed
109	F	F	F	O	S	O	Failed
110	F	O	F	O	F	F	Failed
111	F	O	F	F	F	O	Failed
112	F	F	F	O	F	O	Failed

APPENDIX A2

Program for reliability analysis of model 4:

```
function dydt = model4(t,y,lemda3,lemda2,lemda1,mu3,mu2,mu1)
lemda3= 1/30000;
lemda2= 1/25000;
lemda1= 1/20000;
mu3= 1/249;
mu2= 1/150;
mu1= 1/42;
%tspan=[0,1000,2000];
%y0=[1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0];
dydt = [-(lemda3+lemda2+lemda1)*y(1)-mu1*y(1)-
mu2*y(1)+lemda1*y(5)+lemda2*y(20);
       -(lemda3+lemda2+lemda1)*y(2)-mu2*y(2)+mu1*y(1)+lemda2*y(21);
       -(lemda3+lemda2+lemda1)*y(3)-mu1*y(3)-
mu2*y(3)+lemda1*y(2)+lemda2*y(22);
       lemda1*y(1)+lemda1*y(3);
       -(lemda3+lemda2+lemda1)*y(5)-mu2*y(5)+mu1*y(3)+lemda2*y(24);
       -(lemda3+lemda2+lemda1)*y(6)-mu1*y(6)+mu2*y(1)+lemda1*y(10);
       -(lemda3+lemda2+lemda1)*y(7)+mu2*y(2)+mu1*y(6);
       -(lemda3+lemda2+lemda1)*y(8)-mu1*y(8)+mu2*y(3)+lemda1*y(7);
       lemda1*y(6)+lemda1*y(8);
       -(lemda3+lemda2+lemda1)*y(10)+mu2*y(5)+mu1*y(8);
       -(lemda3+lemda2+lemda1)*y(11)-mu1*y(11)-
mu2*y(11)+lemda1*y(15)+lemda2*y(6);
       -(lemda3+lemda2+lemda1)*y(12)-mu2*y(12)+mu1*y(11)+lemda2*y(7);
       -(lemda3+lemda2+lemda1)*y(13)-mu1*y(13)-
mu2*y(13)+lemda1*y(12)+lemda2*y(8);
       lemda1*y(11)+lemda1*y(13);
       -(lemda3+lemda2+lemda1)*y(15)-mu2*y(15)+mu1*y(13)+lemda2*y(10);
       lemda2*y(1)+lemda2*y(11);
       lemda2*y(2)+lemda2*y(12);
       lemda2*y(3)+lemda2*y(13);
       lemda2*y(5)+lemda2*y(15);
       -(lemda3+lemda2+lemda1)*y(20)-mu1*y(20)+mu2*y(11)+lemda1*y(24);
       -(lemda3+lemda2+lemda1)*y(21)+mu2*y(12)+mu1*y(20);
       -(lemda3+lemda2+lemda1)*y(22)-mu1*y(22)+mu2*y(13)+lemda1*y(21);
       lemda1*y(20)+lemda1*y(22);
       -(lemda3+lemda2+lemda1)*y(24)+mu2*y(15)+mu1*y(22);
       lemda3*y(1);
       lemda3*y(2);
       lemda3*y(3);
       lemda3*y(5);
       lemda3*y(6);
       lemda3*y(7);
       lemda3*y(8);
       lemda3*y(10);
       lemda3*y(11);
       lemda3*y(12);
       lemda3*y(13);
       lemda3*y(15);
       lemda3*y(20);
       lemda3*y(21);
       lemda3*y(22);
       lemda3*y(24);];
```

APPENDIX A3

Program for reliability analysis of model 8:

```
function dydt = model8(t,y,lemda1,lemda2,lemda3,mu1,mu2,mu3)
lemda3= 1/30000;
lemda2= 1/25000;
lemda1= 1/20000;
mu3= 1/249;
mu2= 1/150;
mu1= 1/42;
%tspan=[0,1000,2000];
%y0=[1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];
dydt = [-(lemda1+lemda2+lemda3)*y(1)+mu1*y(5)+mu2*y(3)+mu3*y(2);
-(lemda1+lemda2+lemda3)*y(2)-(mu3)*y(2)+mu1*y(6)+mu2*y(4)+lemda3*y(9);
-(lemda1+lemda2+lemda3)*y(3)-(mu2)*y(3)+mu1*y(7)+mu3*y(4)+lemda2*y(17);
-(lemda1+lemda2+lemda3)*y(4)-
(mu2+mu3)*y(4)+mu1*y(8)+lemda2*y(18)+lemda3*y(10);
-(lemda1+lemda2+lemda3)*y(5)-(mu1)*y(5)+mu2*y(7)+mu3*y(6)+lemda1*y(33);
-(lemda1+lemda2+lemda3)*y(6)-
(mu1+mu3)*y(6)+mu2*y(8)+lemda1*y(34)+lemda3*y(11);
-(lemda1+lemda2+lemda3)*y(7)-
(mu1+mu2)*y(7)+mu3*y(8)+lemda1*y(25)+lemda2*y(19);
-(lemda1+lemda2+lemda3)*y(8)-
(mu1+mu2+mu3)*y(8)+lemda1*y(36)+lemda2*y(20)+lemda3*y(12);
-(lemda1+lemda2+lemda3)*y(9)+mu3*y(13)+mu1*y(11)+mu2*y(10);
-(lemda1+lemda2+lemda3)*y(10)-(mu2)*y(10)+mu1*y(12)+mu3*y(14)+lemda2*y(21);
-(lemda1+lemda2+lemda3)*y(11)-(mu1)*y(11)+mu2*y(12)+mu3*y(15)+lemda1*y(37);
-(lemda1+lemda2+lemda3)*y(12)-
(mu1+mu2)*y(12)+mu3*y(16)+lemda1*y(38)+lemda2*y(22);
-(lemda1+lemda2+lemda3)*y(13)-(mu3)*y(13)+mu1*y(15)+mu2*y(14)+lemda3*y(1);
-(lemda1+lemda2+lemda3)*y(14)-
(mu2+mu3)*y(14)+mu1*y(16)+lemda2*y(23)+lemda3*y(3);
-(lemda1+lemda2+lemda3)*y(15)-
(mu1+mu3)*y(15)+mu2*y(16)+lemda1*y(39)+lemda3*y(5);
-(lemda1+lemda2+lemda3)*y(16)-
(mu1+mu2+mu3)*y(16)+lemda1*y(40)+lemda2*y(24)+lemda3*y(7);
-(lemda1+lemda2+lemda3)*y(17)+mu1*y(19)+mu2*y(25)+mu3*y(18);
-(lemda1+lemda2+lemda3)*y(18)-(mu3)*y(18)+mu1*y(20)+mu2*y(26)+lemda3*y(21);
-(lemda1+lemda2+lemda3)*y(19)-(mu1)*y(19)+mu2*y(27)+mu3*y(20)+lemda1*y(41);
-(lemda1+lemda2+lemda3)*y(20)-
(mu1+mu3)*y(20)+mu2*y(28)+lemda1*y(42)+lemda3*y(22);
-(lemda1+lemda2+lemda3)*y(21)+mu1*y(22)+mu2*y(29)+mu3*y(23);
-(lemda1+lemda2+lemda3)*y(22)-(mu1)*y(22)+mu2*y(30)+mu3*y(24)+lemda1*y(43);
-(lemda1+lemda2+lemda3)*y(23)-(mu3)*y(23)+mu1*y(24)+mu2*y(31)+lemda3*y(17);
-(lemda1+lemda2+lemda3)*y(24)-
(mu1+mu3)*y(24)+mu2*y(32)+lemda1*y(44)+lemda3*y(19);
-(lemda1+lemda2+lemda3)*y(25)-(mu2)*y(25)+mu1*y(27)+mu3*y(26)+lemda2*y(1);
-(lemda1+lemda2+lemda3)*y(26)-
(mu2+mu3)*y(26)+mu1*y(28)+lemda2*y(2)+lemda3*y(29);
-(lemda1+lemda2+lemda3)*y(27)-
(mu1+mu2)*y(27)+mu3*y(28)+lemda1*y(45)+lemda2*y(5);
-(lemda1+lemda2+lemda3)*y(28)-
(mu1+mu2+mu3)*y(28)+lemda1*y(46)+lemda2*y(6)+lemda3*y(30);
-(lemda1+lemda2+lemda3)*y(29)-(mu2)*y(29)+mu1*y(30)+mu3*y(31)+lemda2*y(9);
```

```
-(lemda1+lemda2+lemda3)*y(30)-
(mu1+mu2)*y(30)+mu3*y(32)+lemda1*y(47)+lemda2*y(11);
-(lemda1+lemda2+lemda3)*y(31)-
(mu2+mu3)*y(31)+mu1*y(32)+lemda2*y(13)+lemda3*y(25);
-(lemda1+lemda2+lemda3)*y(32)-
(mu1+mu2+mu3)*y(32)+lemda1*y(48)+lemda2*y(15)+lemda3*y(27);
-(lemda1+lemda2+lemda3)*y(33)+mu1*y(49)+mu2*y(35)+mu3*y(34);
-(lemda1+lemda2+lemda3)*y(34)-(mu3)*y(34)+mu1*y(50)+mu2*y(36)+lemda3*y(37);
-(lemda1+lemda2+lemda3)*y(35)-(mu2)*y(35)+mu1*y(51)+mu3*y(36)+lemda2*y(41);
-(lemda1+lemda2+lemda3)*y(36)-
(mu2+mu3)*y(36)+mu1*y(52)+lemda2*y(42)+lemda3*y(38);
-(lemda1+lemda2+lemda3)*y(37)+mu1*y(53)+mu2*y(38)+mu3*y(39);
-(lemda1+lemda2+lemda3)*y(38)-(mu2)*y(38)+mu1*y(54)+mu3*y(40)+lemda2*y(43);
-(lemda1+lemda2+lemda3)*y(39)-(mu3)*y(39)+mu1*y(55)+mu2*y(40)+lemda3*y(33);
-(lemda1+lemda2+lemda3)*y(40)-
(mu2+mu3)*y(40)+mu1*y(56)+lemda2*y(44)+lemda3*y(35);
-(lemda1+lemda2+lemda3)*y(41)+mu1*y(57)+mu2*y(45)+mu3*y(42);
-(lemda1+lemda2+lemda3)*y(42)-(mu3)*y(42)+mu1*y(58)+mu2*y(46)+lemda3*y(43);
-(lemda1+lemda2+lemda3)*y(43)+mu1*y(59)+mu2*y(47)+mu3*y(44);
-(lemda1+lemda2+lemda3)*y(44)-(mu3)*y(44)+mu1*y(60)+mu2*y(48)+lemda3*y(41);
-(lemda1+lemda2+lemda3)*y(45)-(mu2)*y(45)+mu1*y(61)+mu3*y(46)+lemda2*y(33);
-(lemda1+lemda2+lemda3)*y(46)-
(mu2+mu3)*y(46)+mu1*y(62)+lemda2*y(34)+lemda3*y(47);
-(lemda1+lemda2+lemda3)*y(47)-(mu2)*y(47)+mu1*y(63)+mu3*y(48)+lemda2*y(37);
-(lemda1+lemda2+lemda3)*y(48)-
(mu2+mu3)*y(48)+mu1*y(64)+lemda2*y(39)+lemda3*y(45);
-(lemda1+lemda2+lemda3)*y(49)-(mu1)*y(49)+mu2*y(51)+mu3*y(50)+lemda1*y(1);
-(lemda1+lemda2+lemda3)*y(50)-
(mu1+mu3)*y(50)+mu2*y(52)+lemda1*y(2)+lemda3*y(53);
-(lemda1+lemda2+lemda3)*y(51)-
(mu1+mu2)*y(51)+mu3*y(52)+lemda1*y(3)+lemda2*y(57);
-(lemda1+lemda2+lemda3)*y(52)-
(mu1+mu2+mu3)*y(52)+lemda1*y(4)+lemda2*y(58)+lemda3*y(54);
-(lemda1+lemda2+lemda3)*y(53)-(mu1)*y(53)+mu2*y(54)+mu3*y(55)+lemda1*y(9);
-(lemda1+lemda2+lemda3)*y(54)-
(mu1+mu2)*y(54)+mu3*y(56)+lemda1*y(10)+lemda2*y(59);
-(lemda1+lemda2+lemda3)*y(55)-
(mu1+mu3)*y(55)+mu2*y(56)+lemda1*y(13)+lemda3*y(49);
-(lemda1+lemda2+lemda3)*y(56)-
(mu1+mu2+mu3)*y(56)+lemda1*y(14)+lemda2*y(60)+lemda3*y(51);
-(lemda1+lemda2+lemda3)*y(57)-(mu1)*y(57)+mu2*y(61)+mu3*y(58)+lemda1*y(17);
-(lemda1+lemda2+lemda3)*y(58)-
(mu1+mu3)*y(58)+mu2*y(62)+lemda1*y(18)+lemda3*y(59);
-(lemda1+lemda2+lemda3)*y(59)-(mu1)*y(59)+mu2*y(63)+mu3*y(60)+lemda1*y(21);
-(lemda1+lemda2+lemda3)*y(60)-
(mu1+mu3)*y(60)+mu2*y(64)+lemda1*y(23)+lemda3*y(57);
-(lemda1+lemda2+lemda3)*y(61)-
(mu1+mu2)*y(61)+mu3*y(62)+lemda1*y(25)+lemda2*y(49);
-(lemda1+lemda2+lemda3)*y(62)-
(mu1+mu2+mu3)*y(62)+lemda1*y(26)+lemda2*y(50)+lemda3*y(63);
-(lemda1+lemda2+lemda3)*y(63)-
(mu1+mu2)*y(63)+mu3*y(64)+lemda1*y(29)+lemda2*y(53);
-(lemda1+lemda2+lemda3)*y(64)-
(mu1+mu2+mu3)*y(64)+lemda1*y(31)+lemda2*y(55)+lemda3*y(61);
 lemda3*y(2)+lemda3*y(13);
 lemda3*y(4)+lemda3*y(14);
 lemda3*y(6)+lemda3*y(15);
```

```
lemda3*y(8)+lemda3*y(16);
lemda3*y(18)+lemda3*y(23);
lemda3*y(20)+lemda3*y(24);
lemda2*y(3)+lemda2*y(25);
lemda2*y(4)+lemda2*y(26);
lemda2*y(7)+lemda2*y(27);
lemda2*y(8)+lemda2*y(28);
lemda2*y(10)+lemda2*y(29);
lemda2*y(12)+lemda2*y(30);
lemda3*y(26)+lemda3*y(31);
lemda2*y(14)+lemda2*y(31);
lemda3*y(28)+lemda3*y(32);
lemda2*y(16)+lemda2*y(32);
lemda3*y(34)+lemda3*y(39);
lemda3*y(36)+lemda3*y(40);
lemda3*y(42)+lemda3*y(44);
lemda2*y(35)+lemda2*y(45);
lemda2*y(36)+lemda2*y(46);
lemda2*y(38)+lemda2*y(47);
lemda3*y(46)+lemda3*y(48);
lemda2*y(40)+lemda2*y(48);
lemda1*y(5)+lemda1*y(49);
lemda1*y(6)+lemda1*y(50);
lemda1*y(7)+lemda1*y(51);
lemda1*y(8)+lemda1*y(52);
lemda1*y(11)+lemda1*y(53);
lemda1*y(12)+lemda1*y(54);
lemda3*y(50)+lemda3*y(55);
lemda3*y(52)+lemda3*y(56);
lemda1*y(15)+lemda1*y(55);
lemda1*y(16)+lemda1*y(56);
lemda1*y(19)+lemda1*y(57);
lemda1*y(20)+lemda1*y(58);
lemda1*y(22)+lemda1*y(59);
lemda3*y(58)+lemda3*y(60);
lemda1*y(24)+lemda1*y(60);
lemda2*y(51)+lemda2*y(61);
lemda2*y(52)+lemda2*y(62);
lemda1*y(27)+lemda1*y(61);
lemda1*y(28)+lemda1*y(62);
lemda2*y(54)+lemda2*y(63);
lemda1*y(30)+lemda1*y(63);
lemda3*y(62)+lemda3*y(64);
lemda2*y(56)+lemda2*y(64);
lemda1*y(32)+lemda1*y(64);];
```