

The Application of Bayesian Networks in System Reliability

by

Duan Zhou

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2014 by the
Graduate Supervisory Committee:

Rong Pan, Chair
Daniel McCarville
Muhong Zhang

ARIZONA STATE UNIVERSITY

December 2014

ABSTRACT

In this paper, a literature review is presented on the application of Bayesian networks applied in system reliability analysis. It is shown that Bayesian networks have become a popular modeling framework for system reliability analysis due to the benefits that Bayesian networks have the capability and flexibility to model complex systems, update the probability according to evidences and give a straightforward and compact graphical representation. Research on approaches for Bayesian network learning and inference are summarized. Two groups of models with multistate nodes were developed for scenarios from constant to continuous time to apply and contrast Bayesian networks with classical fault tree method. The expanded model discretized the continuous variables and provided failure related probability distribution over time.

TABLE OF CONTENTS

	Page
LIST OF TABLES	iv
LIST OF FIGURES	v
CHAPTER	
1 INTRODUCTION	1
1.1 Background	1
1.2 Thesis Structure	2
2 LITERATURE REVIEW	3
2.1 Bayesian Networks Introduction.....	3
2.2 Relevant Research.....	7
2.2.1 Bayesian Networks Learning	8
2.2.2 Bayesian Networks Inference.....	10
2.3 Advantage and Limitation.....	11
2.4 Summary	12
3 CASE STUDY	14
3.1 System Introduction	14
3.2 Scenario I – Discrete Failure Distribution	17
3.2.1 Fault Tree Analysis Approach.....	17
3.2.2 Bayesian Network Approach	20

CHAPTER	Page
3.2.3 Conclusion for Scenario I.....	27
3.3 Scenario II – Continuous Failure Time.....	28
3.3.1 Continuous Time Fault Tree Analysis Approach.....	28
3.3.2 Dynamic Bayesian Network Approach.....	31
3.3.3 Conclusion for Scenario II	36
4 CONCLUSION AND RECOMMENDATION.....	37
4.1 Conclusion	37
4.2 Limitations	38
4.3 Recommendation	39
REFERENCES	40
APPENDIX	
I MATLAB CODE AND RESULTS FOR SCENARIO I	44
II MATLAB CODE AND RESULTS FOR SCENARIO II.....	49

LIST OF TABLES

Table	Page
3.1 Parameter Definition and Failure Rate	21
3.2 MPDs and CPDs for the nodes in BN	24
3.3 Conditional Probability Distributions of Root Nodes	27
3.4 Conditional Probability Distributions of Leaf Node T.....	27
3.5 Failure Parameters and Rates for each component in the System	31
3.6 Updated CPDs for Filter A, B, C, D	34

LIST OF FIGURES

Figure	Page
2.1 Bayesian Network Example.....	7
3.1 Water Filter System Work Flow Chart	18
3.2 Fault Tree Structure for the Water Filter System.....	20
3.3 Bayesian Network Structure	23
3.4 Generated Matlab Bayesian Networks Structure	26
3.5 Result of the Bidirectional Inference	29
3.6 Water Filter Component Failure Probability	32
3.7 System Failure Rate over Time.....	33
3.8 System Failure Probabilities Distribution.....	34
3.9 Conditional Failure Probabilities with $T=1$	35
3.10 Conditional Failure Probabilities with $X_i=1$	36
3.11 Conditional Probability: System Works Slowly Given Water Storage Failed	37
3.12 Conditional System Failure Rate given $A=1$ and $W=1$	3

CHAPTER 1 INTRODUCTION

1.1 Background

Reliability is an important quality characteristic that customers require from the manufacturer of products. Different from quality control, reliability is more about the quality in the future, while quality control is concerned more about the current quality of the product. In general, reliability is the ability that a product will operate or a service will be provided properly for a specific time under some specific conditions without failure. It can be measured as a probability [1]. Systems reliability plays a significant role in many areas, especially for some industries requiring high quality and safety assurance. For example, aerospace exploration, nuclear and chemical plants, and military systems must have extraordinary reliability. 21st century is the century for aerospace. Some manned or unmanned aerospace projects, such as communications satellite, manned space orbiting laboratory and spacecraft, are all high-tech products involving costly investments. They have an increasing requirement for good system reliability. One of the painful lesson about system reliability is the Chernobyl nuclear power plant accident in the Soviet Union in 1986. This accident has caused great damage to people's lives and badly influenced the environment for a very long time [2]. Also, failures for complex military systems may lead to a disastrous war. In just a six month period between 1979 and 1980, there were three false alarms given by the US army's malfunctioning early warning system that almost caused a war. Even though reliability originated from the military field, it has been widely applied in many industries and yields enormous economic benefits.

Traditional approaches for system reliability analysis, for example, fault tree analysis, usually does not consider the frequency of failures and the correlation for different system components. The analysis and calculation under the assumption that the failures for components or subsystems are independent can be subject to large errors.

1.2 Thesis Structure

The structure of the paper is as follows.

In this paper, a brief introduction of Bayesian networks methodology is given in the beginning of Chapter 2. Also, a literature review and a salute to some pioneers in this area are summarized in the second part of Chapter 2. In the very end of Chapter 2, some current research concerns and remaining problems are also stated. A case study is presented in Chapter 3. This case study will show how the Bayesian networks method is applied in system reliability inference and assessment. At first, a water filter system model is built in the introduction of Chapter 3, then two scenarios regarding discrete and continuous failure probability distribution are introduced, and two fault tree based Bayesian networks are constructed for each scenario. Also, the states of events are expanded from binary to multiply in the Bayesian networks. Some meaningful conclusions for each scenario are drawn during the system reliability analysis from the result of the calculation. Chapter 3 finishes with BNs models that are coded in Matlab, and the widely used BNs Toolbox for Matlab is introduced. Conclusion for the thesis and recommendations for future research are summarized in Chapter 4.

CHAPTER 2 LITERATURE REVIEW

2.1 Bayesian Networks Introduction

Bayes' theorem is the fundamental of Bayesian inference and Bayesian networks. Bayes' theorem was initially developed in the 1760s, which updates probabilities based on new information. Some statisticians, including Pierre-Simon LaPlace, developed Bayes' formula as a systematic statistical inference and decision-making method [3]. Pearl proposed the Bayesian networks in 1988, which is the current methodology [4]. The so-called Bayesian Analysis has a distinctive method that uses the prior statistics information in the statistics and succeeds in many practical applications.

Bayes' Theorem:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

Where $P(A)$ is the probability of event A happening, $P(B)$ is the probability of event B happening, $P(B|A)$ is the conditional probability of event B, given the probability of a given event A, and $P(A|B)$ is the conditional probability of event A happening given event B happening. For event B, $P(B)$ is the prior probability and $P(B|A)$ is the posterior probability, which are in terms of some evidences. In conclusion, Bayesian Analysis describes the relationship between the prior probability and posterior probability by using Bayes' Theorem.

Bolstad suggest a general form as:

$$P(B_i|A) = \frac{P(A \cap B_i)}{P(A)} = \frac{P(A|B_i)P(B_i)}{\sum_{j=1}^n P(A|B_j)P(B_j)}$$

where $P(A)$ and $P(B) \geq 0$ and $P(B_i)$ consists of mutually exclusive events within the universe S [5].

Bayesian Network (BN) is a probabilistic graphical model that represents the relationships among a set of random variables and their conditional dependencies via a directed acyclic graph (DAG) [6]. BNs contain two parts. The first part is the directed acyclic graph (DAG). In the DAG, every node represents a random variable. Edges represent conditional dependencies. For example, if an edge is from node A to node B , then A is B 's parent variable. The second part is the Conditional Probability Distribution (CPD) at each node. If the variables are discrete, this can be represented as a table (CPT), which lists the probability that the child node takes on each of its different values for each combination of values of its parents.

In order to define a Bayesian network, both the graph structure and the probability parameters must be defined. The following is a widely used example adapted from Russell and Norvig's article [7]. In this example, all the nodes have only two states, T and F . The Bayesian network structure and CPDs are shown in Figure 2.1.

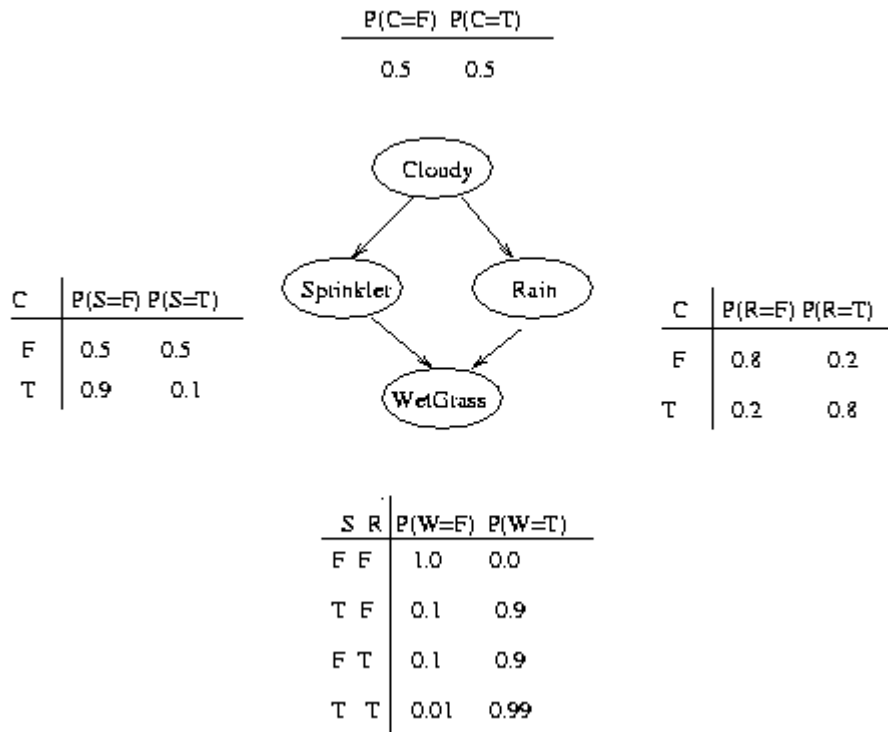


Figure 2.1 Bayesian Network Example

This structure represents the relationship between Cloudy, Sprinkler, Rain and Wet Grass. According to the conditional probability distributions in Figure 2.1, when the weather is cloudy, the probability of raining is 0.8. Also, given the evidence that there is no sprinkler but it is raining, then the probability of glass is wet is 0.9.

It is important to point out that there are strong conditional independences within the Bayesian network. According to the chain rule, when conditional independence is not considered, the joint probability of all the nodes in the graph above is:

$$P(C, S, R, W) = P(C)P(S|C)P(R|S, C)P(W|C, S, R)$$

By using conditional independence relationships [8], the above equation can be simplified as:

$$P(C, S, R, W) = P(C)P(S|C)P(R|C)P(W|S, R)$$

For a generalized model with n components $X_1, X_2, \dots, X_i, \dots, X_n$, when the parent node $pa(X_i)$ of the node X_i is given, X_i is conditionally independent from other nodes except the child nodes of X_i . According to the conditional independence, the joint probability is

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | pa(X_i))$$

In this equation, $pa(X_i)$ is the set of all the parent nodes of X_i .

While if there is no conditional independence between the nodes, the joint probability can be presented as

$$\begin{aligned} P(X_1, X_2, \dots, X_n) &= P(X_1)P(X_2|X_1)P(X_3|X_1, X_2) \dots P(X_n|X_1, X_2, \dots, X_{n-1}) \\ &= \prod_{i=1}^n P(X_i|X_1, X_2, \dots, X_{i-1}) \end{aligned}$$

The number of independent parameters needed in the above chain rule is $2^n - 1$; when assuming that there are m parameters in the parent set $pa(X_i)$, the maximum number of independent parameters for Bayesian networks joint probability is $n2^m$. In most normal systems, m is far smaller than n , hence the computing complexity of Bayesian networks is dramatically reduced.

2.2 Relevant Research

Bayesian networks are widely used in many fields. There are a lot of practical applications of Bayesian networks. Along with the speedy development of the computer science, especially in machine learning and data mining fields, the Bayesian network has been a hot topic for many years [9].

Regarding the system reliability reassessment and inference, Mahadevan used Bayesian network concept to structure system reliability assessment, incorporated multiple failure sequences and correlations between component failures in the Bayesian network model and validated the reliability assessment approach [10]. Doguc applied a K2 algorithm which constructs the Bayesian network model based on historical data [11]. Langseth published several articles about the application of Bayesian networks in system reliability [9], the advantage and limitations of BNs [12], and some approximate inference for continuous time hybrid BNs [13]. Neil presented an iterative algorithm that efficiently combines dynamic discretization with robust propagation algorithms on junction trees, but it is still hard to choose the discrete time intervals [14].

BNs can be used to build decision-theoretic troubleshooting models. It is always applied as an extension of the fault tree. As it has the capability to include multiple failure sequences and components, and combine both forward and backward information, some multilevel system reliability problems can be better solved in the BNs. The conditional probability for any interested nodes or node sets provides a targeted analysis on system

reliability and gives suggestions for system failure detection and maintenance. Also, Bayesian networks are good for dealing with missing data and uncertainty.

Current research on Bayesian networks are mainly concerned about two aspects:

Bayesian networks learning and Bayesian networks inference.

2.2.1 Bayesian Networks Learning

When we want to build a BN, we rely on two sources of information: input from domain experts and statistical data [9]. Both the graph structure and the probability parameters are necessary to define a Bayesian network model. Even though some experts can help to define the objectives and variables for a BN, the subjective suggestions may not be accurate in some times. Expert's experience combined with historical data will make a model with better analytical and predicting ability.

Bayesian networks learning can be defined as two parts: Bayesian network structure learning and BNs parameters learning. Most of the learning related research articles are related to machine learning and data mining fields. For example, in 2009, Doguc used a K2 Bayesian network construction algorithm to build a Bayesian networks model [11]. K2 algorithm is a popular and efficient data mining association rule method. It develops the Bayesian networks structure by a search-and-score approach. Some other search-and-score approaches included are [8]:

- Exhaustive search, which enumerates all the possible DAGs. It is not efficient and limited to the size of Bayesian networks. It is not realistic to enumerate all possible DAGs when the number of nodes is larger than 5;

- K2 algorithm proposed by Cooper and Herskovits [15] is a greedy search algorithm used to score Bayesian networks structures. The basic idea is at the beginning, each of the nodes has no parents, and then it searches for the parents set which has the maximum association with it. It stops adding parent when no addition single parent can increase the score, resulting in a full Bayesian network is constructed;
- Markov Chain Monte Carlo (MCMC). This algorithm is used to search the space of all DAGs. The idea is to create statistical distributions computationally by suitable random sampling methods. It will consider all nearest neighbors. Monte Carlo generally refers to methods based on random number generation, so the distribution of sampling data need to be defined. In practice, the MCMC has been found to take a long time to generate if the graph has more than 10 nodes;
- Hill-Climbing algorithm starts from a specific node, then scoring for all the nearest neighbors of this node and move to the nearest neighbor with highest score [16].

There are also constraint-based methods introduced by Pearl (1991) [17], Spirtes, Glymour, and Scheines (1993) [18]. They are not as popular as the search-and-score approaches, but can still represent the whole Markov equivalence class and construct the Bayesian networks.

The parameter learning for Bayesian networks is primarily about the learning of the Conditional Probability Table (CPT). It still focus on the traditional statistics and Bayesian statistics [19]. The maximum likelihood estimates of the parameters are easily leaned when the dataset is complete. When there are missing values in the dataset,

usually an EM (Expectation Maximization) algorithm is used to find the maximum likelihood [20].

2.2.2 Bayesian Networks Inference

Over the last decade, several commercially available tools have been developed for calculating with BNs, such as HUGIN, WinBUGS, and BayesiaLab. These tools are able to use variable elimination [21], junction tree algorithm [22] and Gibbs sampling inference [23] to do Bayesian network inference. Understanding and describing the algorithms is beyond the scope of this paper. However, it is important to understand that the limitations of the current research on Bayesian network inference is the computational complexity of the inference.

From most of the applications of Bayesian networks, the variables in the BNs are all discrete. The application of Bayesian networks is limited when only consider discrete nodes. For system with continuous variables, the main approach is to discretize them. Some recent researches about the dynamic BNs and hybrid BNs are helpful to solve this problem. For example, Langseth explored four approaches to inference in hybrid BNs: discretization, mixtures of truncated exponentials (MTEs), variational methods, and Markov chain Monte Carlo (MCMC) for some specific types of inferences [13]. David M. and Martin N combined the modeling capabilities of BNs with the dynamic discretization inference algorithm. It provides a general technique for dynamic system reliability analysis [24].

The Bayesian networks inference on continuous, hybrid and multi-level systems is still a topic under development.

2.3 Advantage and Limitation

Compared with traditional system reliability analysis methods, Bayesian networks can present the uncertainty and correlation of variables, and make up some shortages of the traditional methods when doing system reliability reasoning under uncertainties.

For example, fault tree analysis is a widely used method to analyze safety and reliability of complex systems. Fault tree analysis is directly related with the classical probability theory. To perform a fault tree analysis, simple Boolean relationships among various events are built with a tree structure, where the ends of its branches represent components with known failure rates. Fault tree analysis is usually applied to systems with certain fault causes and clear fault logic relationships. All of the events in a fault tree have only binary states such as failed and operating. Relationships among events fall mostly into two groups: AND and OR, which are the gates in the fault tree [25]. But in reality, many events have more than two states. The traditional fault tree cannot represents events other than two states. Also, a fault tree requires certain causal relationship between upper events and lower events. In a real case, often the causality among events is not certain and correlativity must be learned. Conditional probability is a more proper way to describe the relationship.

Bayesian network can make up these shortage very well. BNs can be easily transformed from a fault tree model. The nodes in Bayesian networks which correspond to the events

and gates in a fault tree can have more than two states, or the failure can even be continuously distributed. With conditional probabilities, there is no need to be certain of the causality between upper nodes and lower nodes; uncertainty is allowed in Bayesian networks. It is easy to represent the relationships which are not simply AND or OR in BNs with a conditional probability table. Correlation and dependence between components can be handled in BNs [9]. Also, traditional reliability analysis methods cannot update failure rates with new information added in, while Bayesian networks can update the system reliability when evidence is added during the analysis. Any joint conditional probability can be calculated and bidirectional reasoning can be used for further system reliability inference and assessment.

While there're still some limitations for Bayesian networks, in engineering practice, it is difficult to acquire all conditional probabilities $P(X_i|pa(X_i))$ to get the CPT even if domain experts' experience is involved. Building Bayesian network for a large complex system is still not easy. Also, for some hybrid Bayesian networks, the nodes are not discrete, nor do they correspond to some particular continuous distribution such as the Gauss distribution. It is difficult to make Bayesian network inference if the model is hybrid even though some approximate approaches like Markov Chain Monte Carlo can be applied.

2.4 Summary

After Pearl proposed the Bayesian networks in 1988 [4], Bayesian networks have become a popular modeling framework to build decision-theoretic troubleshooting models in system reliability analysis, many scholars tried to combine Bayesian networks

methodology with traditional system reliability methods such as fault tree analysis and Monte Carlo simulation. The intuitive representation and the modeling flexibility of BNs make it a well-suited tool for reliability analysis.

The advantages of BNs can be concluded as follows. Firstly, BBN is an indeterminate causality model. Different from other decision models; it can be viewed as a knowledge presentation and inference model that visualizes the multivariate knowledge. It shows the conditional probability between the variables, which can represent some fuzzy causality and correlativity. It performs very well when dealing with the problems having uncertainty. As it presents the relationship between each node with conditional probabilities, it can learn and predict with limited, incomplete and uncertain information. Some complex relationships can be easily represented by a conditional probability table. Secondly, it can contain some decision-making related information in the network. It is easy to use in interaction with domain experts. The network structure can help present and organize the information, and manage the information as nodes, which is very efficient. Finally, the computing complexity of BNs is smaller than some traditional approach like Markov Chain analysis. It has relatively good performance regarding speed, accuracy, complexity and generality.

The limitations of Bayesian networks mainly concentrate on the capability for inference continuous system reliability. Some discretization approaches should be conducted when doing Bayesian inference.

CHAPTER 3 CASE STUDY

In order to better demonstrate and validate the property of Bayesian networks, a case study about a water filter system is designed and introduced in this chapter. Two scenarios with different failure probability distributions are discussed. A brief methodology used in fault tree analysis is used to do an essential analysis about the system. Bayesian networks with multistate nodes are applied for the further system reliability analysis and assessment. Also, to meet the real case, an advanced model with continuous failure time is presented. An additional research on the application of Bayesian networks in the continuous reliability system is tackled. The advantages of Bayesian networks in system reliability inference are then discussed and conclusion are drawn.

3.1 System Introduction

With the rapid development of modern industry, the degree of water pollution is also increasing rapidly. Drinking healthy water is more and more difficult. Even though water pollution is well controlled in the US, some other issues like the aging of water pipes and the residual water disinfectant still cause unreliable water supply. People now have a higher awareness about water purification and its effect on the quality of live. The household water filter has been an essential appliance in many areas in the US. Not only for home use, but also more and more industries are manufacturing the water filter system to ensure people's life and health. For example, a filter based on a reverse osmosis (RO) membrane was used during the Iraq War. This system is developed by US-based Hydration Technologies Inc. (HTI), and has been proved to be essential in

preserving the lives of stranded US soldiers in Iraq [26]. According to the research of WHO, there are 780 million people do not have access to an improved water source in the world [27]. While worldwide, water sanitation and hygiene has the potential to prevent at least 9.1% of the global disease burden and 6.3% of all deaths [28].

Water filter system is important to improve the water quality and prevent potential diseases. The reliability of water filter system is closely related to people's health, but the general failures of water filters are not easy to be observed. Hence, effective approaches to detect system failure and conduct system reliability inference and prediction are of vital importance.

The ultimate purpose of the water filter system can be outlined as follows:

- Remove floating impurity in the water;
- Remove microorganism, virus and spore;
- Remove heavy metal like Pb and Cd;
- Remove the rust, residual chlorine and organic pollutants, enhance the taste of water;
- Keep beneficial minerals in the purified water;
- Ensure healthy water delivery.

A flow chart of the water filter system is introduced in Figure 3.1. Water comes from normal water supplier, then flows into a sediment filter. Usually it is a PP fibril membrane to filter some visible impurities, including sediment, worms and rust. After that is an active carbon filter cartridge, which is used to remove fine particulate, organic matter and undesirable odor caused by pesticides and chlorine. Following is the most important part, a reverse osmosis (RO) filter. The RO process uses polymeric membranes to purify water and allow only selective mass to transport through. It is the simplest and

most efficient technique to reduce salt in the seawater and brackish water [29]. The RO filter will remove the majority of pollutants, microorganisms and some injurious heavy metal. After these three filters, purified water will either directly flow through the last polish filter, which is a carbon filter to capture those chemicals not removed by the reverse osmosis membrane and enrich the taste of water, or run into a water storage first before the last process. The storage is helpful to have a larger water volume when drinking water is needed.

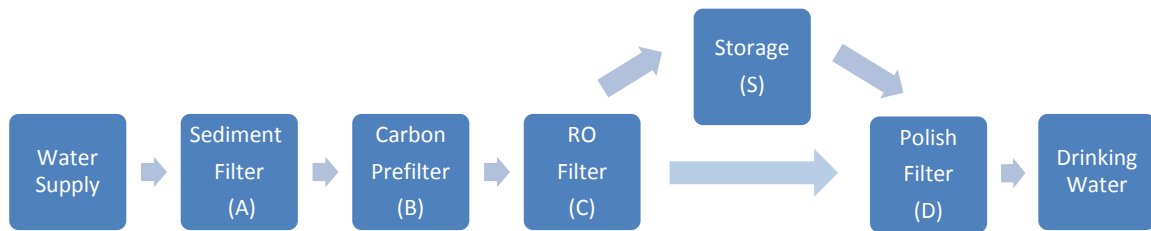


Figure 3.1 Water Filter System Work Flow Chart

Most fresh water applications like drinking water purification, military use water purification unit and wastewater purification system have the similar process as this simplified water filter system. This kind of filter plays a key role in people's daily life and has great influence on our society. It is necessary to analyze and evaluate the system reliability of the water filter system to ensure its high reliability, and consequently assure people's health.

In this case study, a Bayesian networks methodology combined with fault tree analysis is used in two scenarios for system reliability analysis and assessment. The first scenario

considers a constant discrete failure rate for all the components in this water filter system; the second scenario introduces a continuous failure distribution for the four water filters.

It is important to declare that it is a reduced model and some components and variables were removed, but it can still present the deployment and exemplification of the Bayesian networks methodology.

3.2 Scenario I – Discrete Failure Distribution

In this scenario, all the failure rate of the components in the water filter system are assigned constantly. A reduced fault tree model and Bayesian network model with multiply states are built to better facilitate the implementation and understanding. A more complex model will be discussed in scenario II.

3.2.1 Fault Tree Analysis Approach

Taking the water filter system failure as the top event T, analyzing from the top to down and step by step, we can get the following fault tree shown in Figure 3.2.

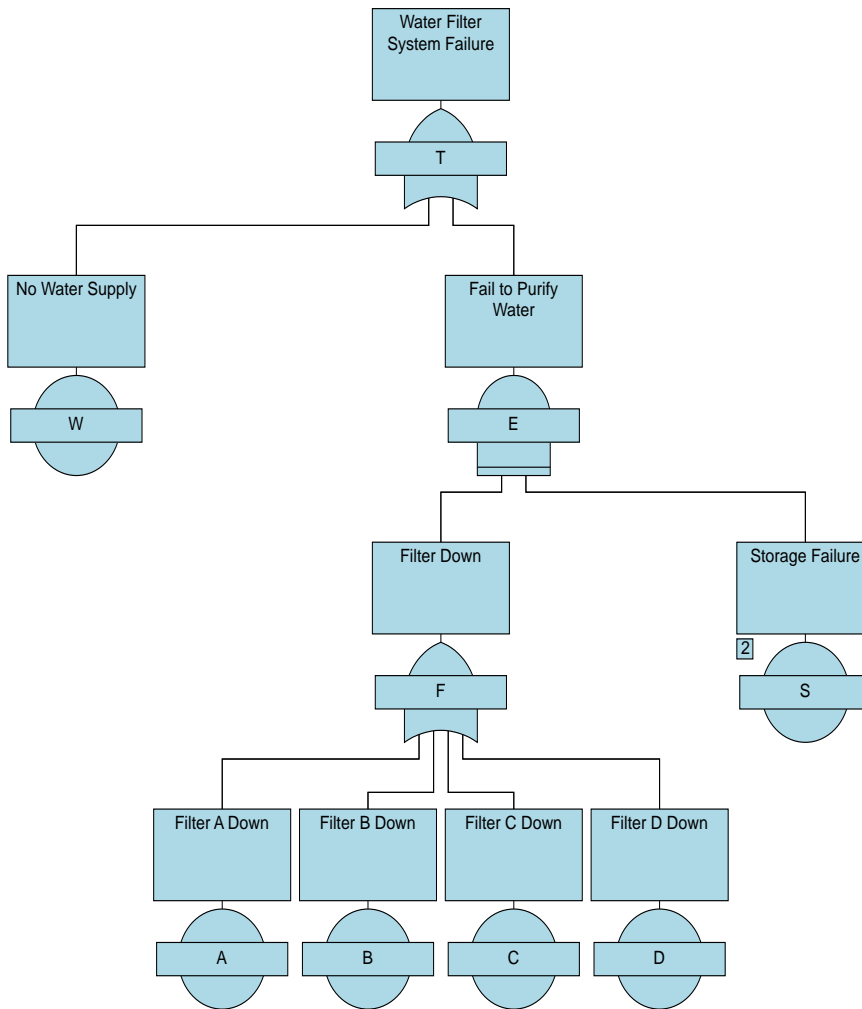





Figure 3.2 Fault Tree Structure for the Water Filter System

In which,  and  are OR gates;  is a priority AND gate.

The top event T with an OR gate connects with event W, which presents the water supply failure, and event E, which presents water purification failure. It is important to notice that event E has a priority AND gate (PAND), which connect E with event F (Filters

Down) and event S (Water Storage Fails). It means that event E will happen only when event F happens. If water storage fails but all the filters are still functional, event E won't happen as water can directly flow through the last filter and come out as drinking water. Event F has an OR gate which means all the filters should be functional to ensure the system availability. Any one of the four filters down can lead to event F happen.

Parameter definition for this fault tree and component failure rate λ is shown in the following Table 3.1.

Event Symbol	Event Description	Failure Rate λ
A	Sediment filter fails	0.5
B	Carbon pre-filter fails	0.25
C	RO filter fails	0.15
D	Polish filter fails	0.1
S	Water storage fails	0.15
W	No water supply	0.002

Table 3.1 Parameter Definition and Failure Rate

Some assumptions are made for this fault tree model. The failures of each components are independent. There is no correlation between the paralleled events, hence all the probabilities of upper level event failure can be calculated very directly according to the function of the gates. Also, as we are not considering a scheduled maintenance plan for the water filter system at this time, all the components are not replaced or repaired during the analysis. The failure rates keeps unchanged as we assumed.

With the above assumptions, determining the unavailability of complex events is very straightforward.

Firstly, we need to know how to inference the probability of event E happens. As event E has a priority AND gate which means only when event F happens, event E will happen, then $P(E) = P(F) = 1 - P(\bar{A} \cap \bar{B} \cap \bar{C} \cap \bar{D})$. Hence,

$$\begin{aligned}
P(T) &= 1 - P(\bar{W} \cap \bar{E}) = 1 - P(\bar{W})P(\bar{E}) \\
&= 1 - (1 - P(W))(1 - P(E)) \\
&= 1 - (1 - P(W)) \left(1 - \left(1 - P(\bar{A} \cap \bar{B} \cap \bar{C} \cap \bar{D}) \right) \right) \\
&= 1 - (1 - P(W))P(\bar{A})P(\bar{B})P(\bar{C})P(\bar{D}) \\
&= 1 - (1 - P(W))(1 - P(A))(1 - P(B))(1 - P(C))(1 - P(D)) \\
&= 1 - (1 - \lambda_W)(1 - \lambda_A)(1 - \lambda_B)(1 - \lambda_C)(1 - \lambda_D)
\end{aligned}$$

Then the failure rate of the water filter system is 0.7137 according the above equation.

3.2.2 Bayesian Network Approach

With the fault tree analysis, not much information can be inferred. Even though the failure rate of the top event can be calculated and some events with more critical influence on the system reliability can be identified from minimum cut sets and the calculation of structural importance, probability importance and pivotal importance degrees, it is still not good enough for make better failure detection and prediction.

Bayesian networks can make a bidirectional inference by given relational information, and generate the conditional probabilities between the input nodes and output nodes.

Also, system with multistate nodes are very common in the real case. Binary failure distribution limits the analysis of system reliability. Bayesian network can perform the functions of fault tree analysis and avoid the limitation of FTA.

According to the fault tree structure, we can transform the topological structure of the fault tree to the network structure of Bayesian networks. The transformed Bayesian networks model is shown in Figure 3.3.

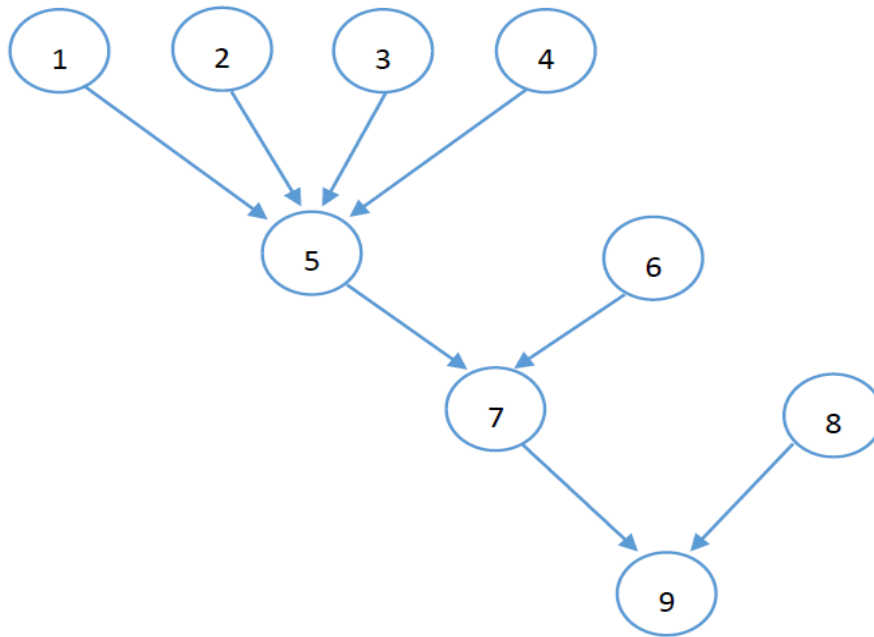


Figure 3.3 Bayesian Network Structure

There are four layers in this Bayesian network structure. The relationship between each node and its corresponding failure event are described as below:

- First layer: nodes 1, 2, 3, 4 correspond to bottom events A, B, C, D, respectively;
- Second layer: nodes 5 and 6 correspond respectively to event F and bottom event S;
- Third layer: node 7 and 8 correspond respectively with event E and bottom event W;
- Forth layer: node 9 correspond to top event T.

One of the limitation of fault tree analysis is that all the events can only have binary states. By using Bayesian networks, we can assume that components have more than two

states. To expand the water filter system more close to real case, we assume that the system failure has three states: {failed, operating, slow}. For each root nodes, there is a Marginal Probability Distribution (MPD). This will present all the possible states of the node and their probabilities. For every other node in the Bayesian network, a Conditional Probability Distribution (CPD) is used to describe its probability distribution given the states of the parent nodes. Then the MPDs and CPDs for the nodes in the water filter system Bayesian network model are inferred from the fault tree as in the following Table 3.2.

A	P(A)	B	P(B)	C	P(C)	D	P(D)	S	P(S)	W	P(W)
1	0.5	1	0.25	1	0.15	1	0.1	1	0.15	1	0.002
2	0.5	2	0.75	2	0.85	2	0.9	2	0.85	0	0.998

F	S	P(E=1 F,S)	P(E=2 F,S)	P(E=3 F,S)	A	B	C	D	P(F=1 A,B,C,D)
1	1	1	0	0	1	1	1	1	1
2	1	0	0	1	1	1	1	2	1
1	2	1	0	0	1	1	2	1	1
2	2	0	0	0	1	2	1	1	1
					1	2	1	2	1
					1	2	2	1	1
					1	2	2	2	1
					2	1	1	1	1
					2	1	1	2	1
					2	1	2	1	1
					2	2	1	1	1
					2	2	1	2	1
					2	2	2	1	1
					2	2	2	2	0

E	W	P(T=1 E,W)	P(T=2 E,W)	P(T=0 E,W)
1	1	1	0	0
2	1	1	0	0
3	1	1	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1

Table 3.2 MPDs and CPDs for the nodes in BN

When a node is equal to 1, then it means this node fails; when it is equal to 2, then this part of the system is still functional; when it is equal to 3, then it means water flows

slowly in the system. It is necessary to mention that the CPDs for node 5, 7, and 9 represent the relationships between components. It is a better approach to explain some complex relationships which are not easily described with fault tree gates.

To build the water filter system Bayesian network model, a Bayes Net Toolbox for Matlab is introduced in this paper. The Bayes Net Toolbox for Matlab was written by Kevin Murphy, 1997—2002, when he was pursuing his PhD degree in University of California, Berkeley [30]. This Matlab toolkit allows users to create Bayesian networks, either manually or by learning; do some BNs inference like computing marginal distribution and joint distribution; conduct structure learning and parameter learning for the BNs by estimating from given dataset or simulation, and applying some algorithms like K2 and structural EM; it can also use some inference engines like junction tree and Monte Carlo for further research purpose. His PhD dissertation introduced some algorithms and examples which have been implemented in the Bayes Net Toolbox [31]. In this case study, the main functions used from the toolkit are manually creating the Bayesian network graph shell and imputing conditional probability distribution, and BNs inference computing. Some Matlab code was programed and can be found in the Appendix I and II.

The graph structure from Matlab is shown in Figure 3.4, which is the same as what we designed in Figure 3.3.

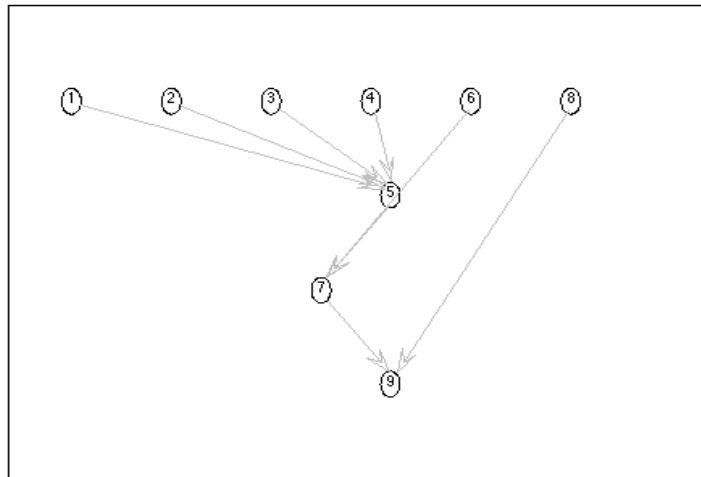


Figure 3.4 Generated Matlab Bayesian Networks Structure

Given the same failure rate as in the fault tree analysis, without additional evidence, the probability distribution of the leaf node 9 is different from what we inferred from the fault tree analysis as the system failure is no longer binary. While the probability of system failed is still the same as the fault tree result. This indicates that without evidence, using Bayesian networks can still calculate the probability distribution of leaf node, which is the top event in the fault tree analysis, and also can handle cases with multistate failure.

$$P(T = 1) = 0.7137, P(T = 2) = 0.2434, P(T = 3) = 0.0429$$

We can see without any evidence, the probability of system failed is 0.7139; the probability of system operating normally is 0.2434; the probability of system working with water flowing slowly is 0.0429.

While one of the most important advantages of Bayesian networks is when given any additional evidence, it can calculate the conditional probability between the input nodes

and the output nodes, and conduct a more accurate probability inference. The conditional probability will give valuable suggestions for the system reliability analysis, failure diagnosis and system maintenance plan. Besides, the capability of analyzing multistate failure distribution avoids a significant limitation of fault tree approach, which gives a better representation of the real system.

Now given the evidence that the water filter system fails, operates normally or works slowly, which means $T=1$, $T=2$ or $T=3$, the conditional probability for each component is calculated by using Matlab and shows in the following Table 3.3. The inference engine used in the case study is junction tree engine.

Node(X_i)	A	B	C	D	S	W
$P(X_i=1 T=1)$	0.7006	0.3503	0.2102	0.1401	0.15	0.0028
$P(X_i=1 T=2)$	0	0	0	0	0	0
$P(X_i=1 T=3)$	0	0	0	0	1	0

Table 3.3 Conditional Probability Distribution for Each Component

It is important to know how a component will influence the whole system. Now a conditional probability distribution of the leaf node T by given the evidence that each component of the water filter system is failed. The CPDs are shown in the Table 3.4 below.

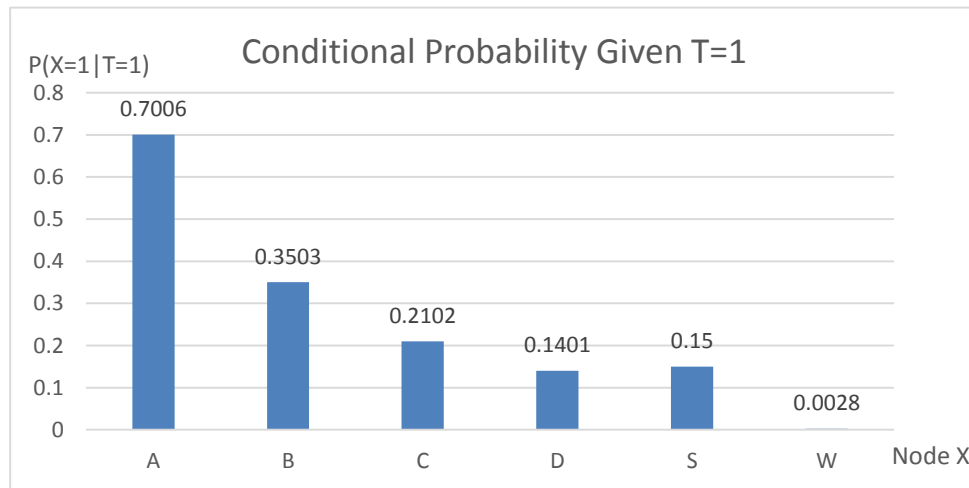
Node(X_i)	A	B	C	D	S	W
$P(T=1 X_i=1)$	1	1	1	1	0.7137	1
$P(T=2 X_i=1)$	0	0	0	0	0	0
$P(T=3 X_i=1)$	0	0	0	0	0.2863	0

Table 3.4 Conditional Probability Distributions of Leaf Node T

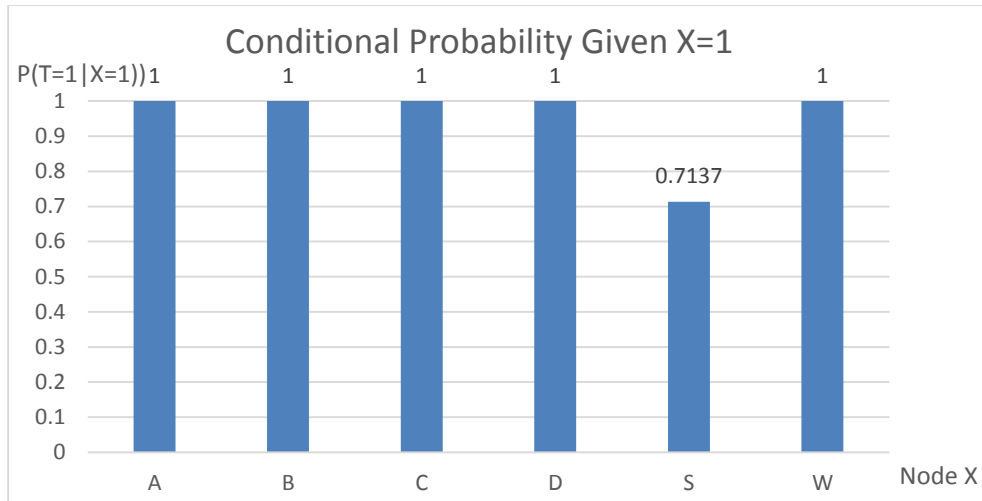
As shown in the fault tree analysis, most gates in this water filter system are OR gate; the priority AND gate indicates that the water storage part does not have great influence on

the system. The conditional probability distributions represent the facts in this system, as given the evidence that $T=1$, the CPD of node S keeps unchanged, also, given any filter node equal to 1, this water filter system will fail as filters A, B, C, D are connected in series in this system.

The following Figure 3.5 compares the influences of each component failure on the system failure. This indicates that node A, the sediment filter, has a significant influence on the system failure. Considering that filter A has the highest failure rate among all the components in this water filter system, and any failure of the filters will lead to the system failure, it is reasonable that node A has a significant CPD. To strengthen the system reliability, more reliable filters should be used to ensure a low failure rate for the filter system. Especially for filter A, a backup sediment filter need to be considered.



(a)



(b)

Figure 3.5 Result of the Bidirectional Inference

In addition, the Bayesian network inference can calculate the conditional probability with any joint evidence. For example, given that filter A works but water storage fails in the water filter system, the conditional probability distribution for leaf node T is

$$P(T = 1|A = 2, S = 1) = 0.4274;$$

$$P(T = 2|A = 2, S = 1) = 0;$$

$$P(T = 3|A = 2, S = 1) = 0.5726$$

3.2.3 Conclusion for Scenario I

In this scenario, a fault tree based Bayesian network model is built for constant failure system reliability modeling and analysis. Some advantages of Bayesian networks are presented, such as the capability to analyze nodes with multiply states; representing some complex relationship between nodes by using conditional probability table; making use

of addition evidence and conduct bidirectional inferences; driving a more flexible and obvious calculation for system reliability analysis; detecting the weak link of the system and helping make targeted maintenance suggestions. To further reflect actual circumstances accurately, an advanced model with continuous failure time is introduced in scenario II.

3.3 Scenario II – Continuous Failure Time

In the real case, a lot of systems contain components have a continuous failure time, especially for those products which will fatigue over time. In the water filter system introduced in scenario I, the failure probabilities of the filters will increase over time in the real case. It is necessary to evaluate the system with continuous failure time for all the water filters. Also, the system reliability along with time will be obviously influenced by the changing of failure probabilities for the four filters. This scenario will present the implement of Bayesian networks and show how BNs can be applied in some real cases like scenario II.

3.3.1 Continuous Time Fault Tree Analysis Approach

There are some approaches for continuous time system reliability assessment. Most of the recent researches are focus on the discretization of continuous features. One of the majority methodologies is Markov Chain analysis. While using the Markov model, the complexity of calculation increases dramatically when the number of nodes, the states for each node and the relationship complexity between each node increase. To reduce the

computing complexity and reassess the system reliability more straightforwardly, a continuous time fault tree analysis is applied in this scenario.

The system process flow chart is still the same as in Figure 3.1. The structure of fault tree analysis is also the same as in Figure 3.2 in scenario I. In this scenario, failure probabilities for the four filters are all functions related to the working time t . In the real case, as time goes by, there will be more and more impurities in the water filters. It is more likely to fail after working for a longer time. Assumption is made that the failure probabilities of filter A, B, C, D all have the function

$$P(X_i) = 1 - e^{-\lambda_i t}$$

In which λ_i is failures per hour, t is the system running time in hours.

The failure parameter for each component in this water filter system can be found in Table 3.5.

Event Symbol	Event Description	Failure Parameter λ_i (/hour) or Failure Rate
A	Sediment filter fails	0.00025
B	Carbon pre-filter fails	0.00015
C	RO filter fails	0.0001
D	Polish filter fails	0.00005
S	Water storage fails	0.15
W	No water supply	0.002

Table 3.5 Failure Parameters and Rates for each component in the System

In order to make the system reliability inference, discretization of time and failure functions for filters is necessary. A time interval $\Delta t = 336$ hours, which is two weeks, is

set. Also, system running time is set to 50 weeks, which means there are 25 iterations in the fault tree analysis and there are 25 different failure probabilities for each continuous node. The Matlab code to calculate the component failure rate and system failure rate for each iteration can be found in Appendix II part I.

As the failure probabilities for water supply W and water storage S remain constantly, while failure probabilities for filter A, B, C, D increase over time. A graph of the component failures for the four filters is shown in Figure 3.6.

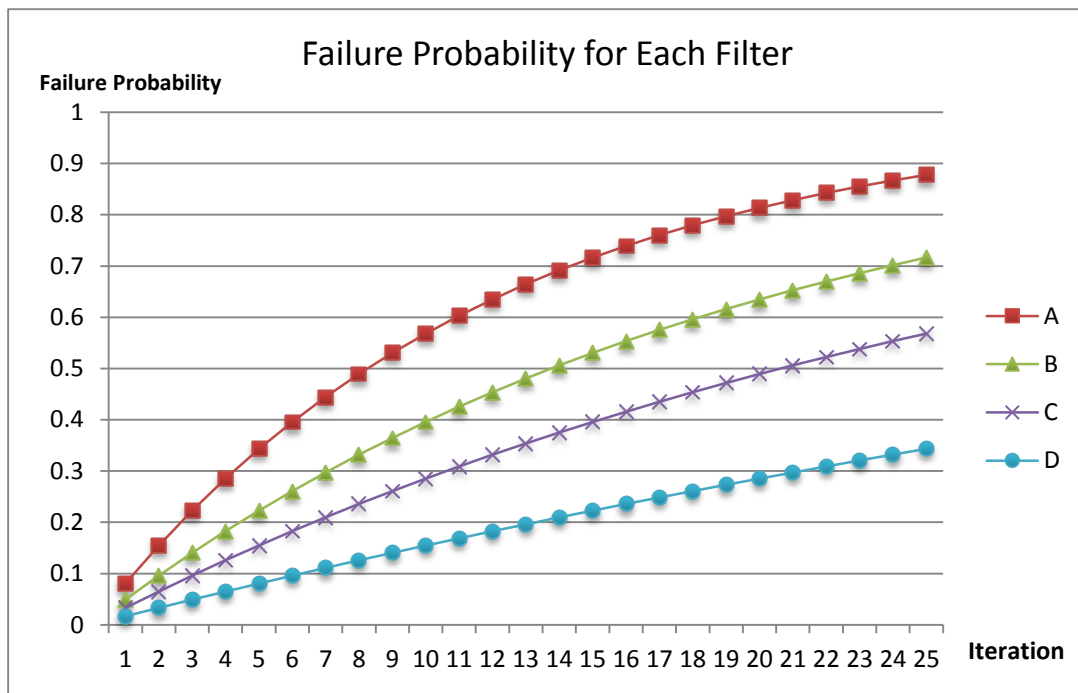


Figure 3.6 Water Filter Component Failure Probability

Failure probability for the water filter system changes over time is shown in Figure 3.7.

System running time is still 50 weeks.

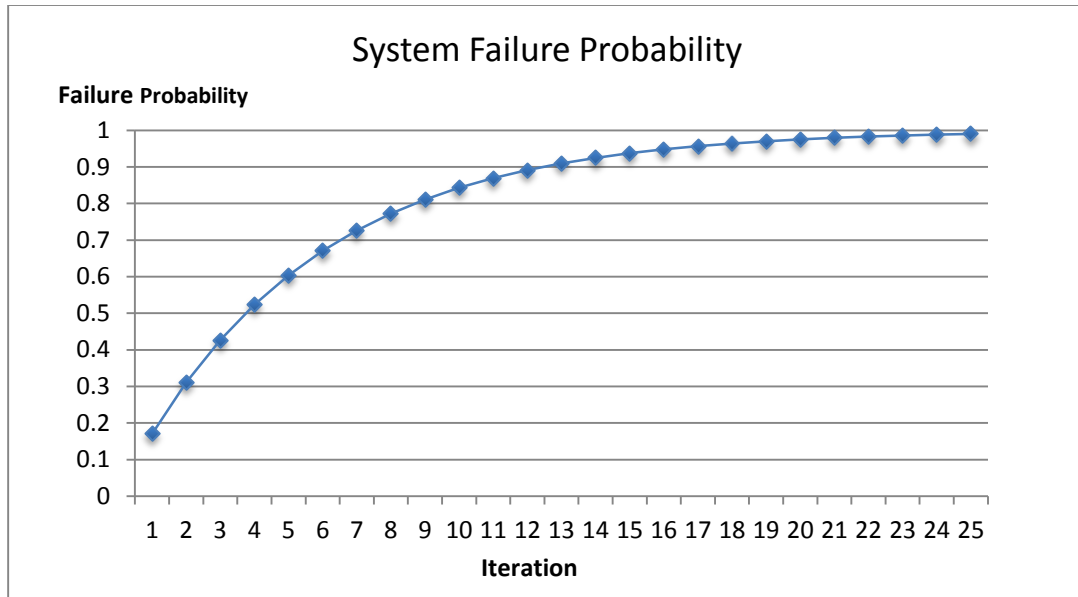


Figure 3.7 System Failure Rate over Time

The continuous time fault tree can present the change of component failures and system failures over time. It is a very good way to solve the computing complexity problem involved in traditional Markov Chain analysis model. A fault tree based Bayesian network model can make a better inference about the system reliability and give more useful facts for reliability related purposes. A dynamic Bayesian network model is conducted in the following sub-section 3.3.2.

3.3.2 Dynamic Bayesian Network Approach

According to the hybrid model introduced in section 3.3.1, the structure of the dynamic Bayesian network model keeps unchanged. It still has the same structure as shown in Figure 3.3. By using the Bayes Net Toolbox for Matlab, a Matlab program is coded for the BN in scenario II. This can be found in Appendix II part II.

The CPDs for node E, F, T and MPDs for node S and W remain the same as shown in Table 3.2. The MPDs for node A, B, C, D are updated in the following Table 3.6.

A	P(A)	B	P(B)	C	P(C)	D	P(D)
1	$1 - e^{-\lambda_A t}$	1	$1 - e^{-\lambda_B t}$	1	$1 - e^{-\lambda_C t}$	1	$1 - e^{-\lambda_D t}$
0	$e^{-\lambda_A t}$	0	$e^{-\lambda_B t}$	0	$e^{-\lambda_C t}$	0	$e^{-\lambda_D t}$

Table 3.6 Updated CPDs for Filter A, B, C, D

In which $\lambda_A = 0.00025$; $\lambda_B = 0.00015$; $\lambda_C = 0.0001$; $\lambda_D = 0.00005$.

Without any additional evidence, the failure rate distributions for the four filters are still the same as shown in Figure 3.6. As we expanded node T to three states, the failure probability distribution of the system is shown in Figure 3.8. Some further inference is made by adding evidence to this model. A bidirectional inference will be introduced for this dynamic Bayesian network model.

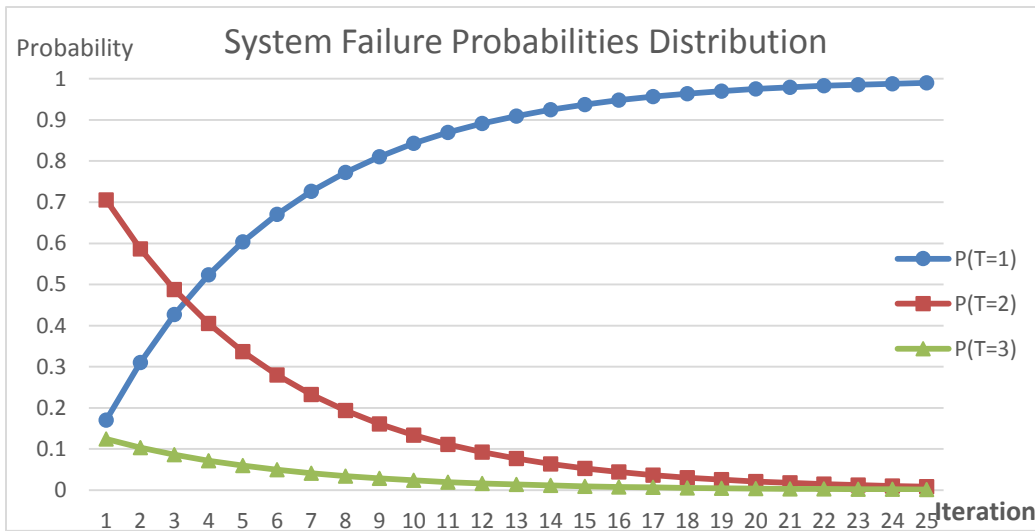


Figure 3.8 System Failure Probabilities Distribution

Figure 3.9 is the conditional failure probability distribution over time given that system is failed. In this figure, when system is failed, the failure probabilities of filter A, B, C, D are increasing with time increasing; the failure rate of water supply slightly decreases over the system running time; the failure rate of storage remains the same. It shows that for the water filter system, the reliabilities of four filters have a significant weight contributing to the reliability of the whole system.

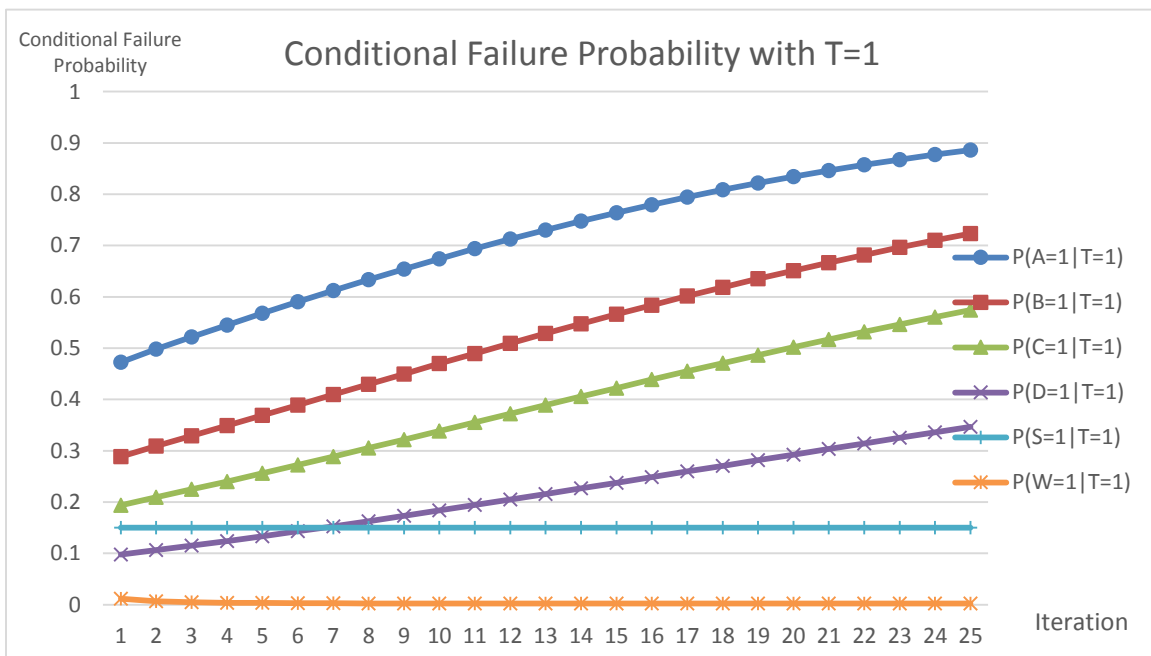


Figure 3.9 Conditional Failure Probabilities with T=1

Another inference from the root nodes to leaf node T is also conducted in the Bayesian network model. The conditional probabilities for the system failure given each node's failure is shown in Figure 3.10. The conditional failure probability $P(T = 1|X_i = 1)$ shows that any failure of component A, B, C, D, W will cause the system failure, while $P(T = 1|S = 1)$ is exactly the same as $P(T = 1)$ which without any evidence. This

result is corresponding to the fault tree structure as the OR gate determines a low failure tolerance capacity, and the priority AND gate indicates that water storage will not influence the probability that system fails.

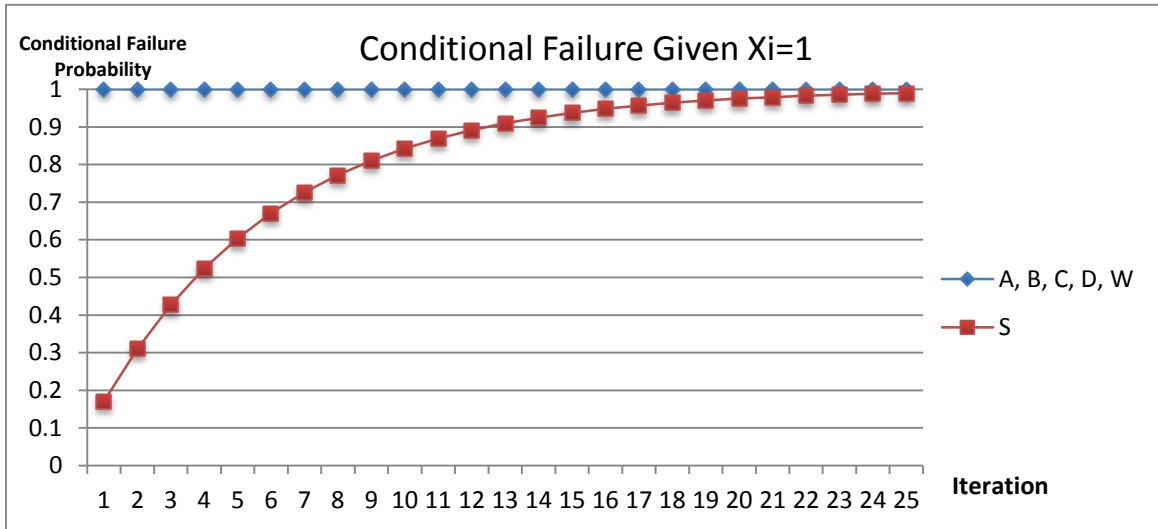


Figure 3.10 Conditional Failure Probabilities with Xi=1

While as there are three states for node T and water storage is an influential component to system working slowly, the conditional probability system working with slow water flow given that water storage is failed, $P(T = 3|S = 1)$, is shown in Figure 3.11. It indicates that as time goes by, this water filter system is more unlikely to work with slow water flow. Combined with Figure 3.10, it can be explained that the system is more likely to fail with time other than work slowly.

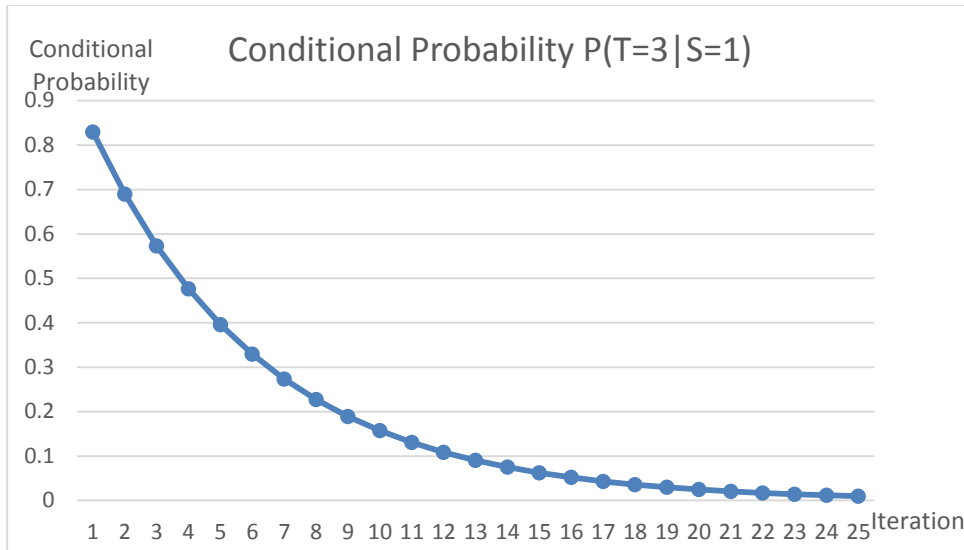


Figure 3.11 Conditional Probability: System Works Slowly Given Water Storage Failed

Also, the system failure rate given a joint evidence can be calculated from the dynamic Bayesian network. For the same example in scenario I, given water filter A operating but water storage failed in the system, the conditional probability $P(T = 1 | A = 2, S = 1)$ is shown in Figure 3.12.

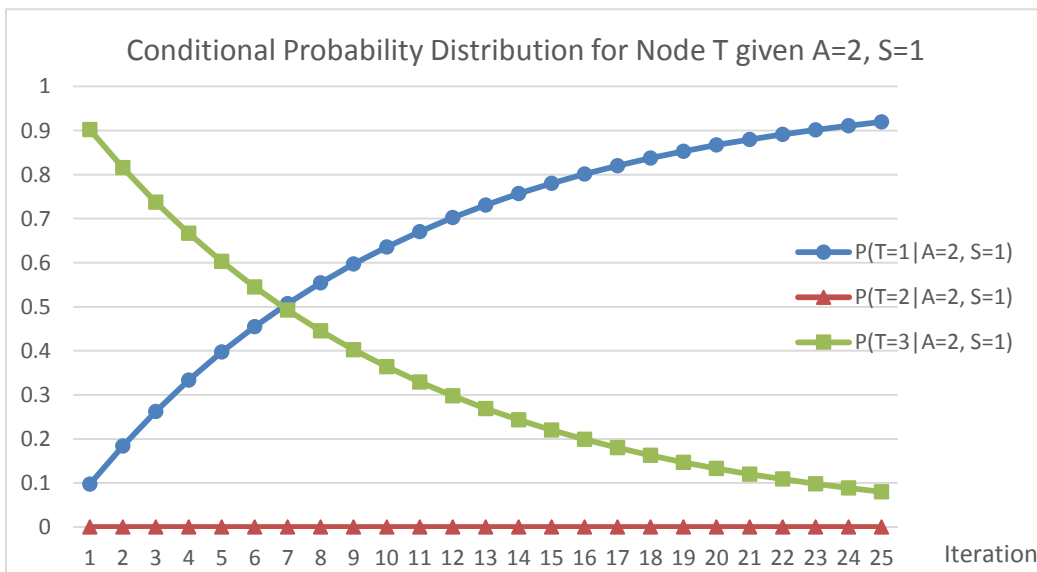


Figure 3.12 Conditional System Failure Rate given A=1 and W=1

3.3.3 Conclusion for Scenario II

In this scenario, a fault tree based dynamic Bayesian network is applied for the water filter system modeling and assessment. Bayesian networks is not only a powerful methodology for reasoning under uncertainty and making better inference by taking the advantage of using more information, but also, by discretizing some variables, Bayesian networks can be very straightforward to reassess continuous and dynamic system reliability. Besides, the result of bidirectional inference is of great importance for system failure detection and prediction, also useful for guiding a targeted maintenance plan. By using Bayesian networks, system reliability can be improved when the weak link of system is identified and improved.

CHAPTER 4 CONCLUSION AND RECOMMENDATION

4.1 Conclusion

This research is placed on the application of Bayesian networks in system reliability inference and assessment. An introduction of the Bayesian networks methodology, its development and application in system reliability field are reviewed. Some advantages and limitations of Bayesian network and available improving areas are marked during the literature review. The case study in this thesis developed a water filter system, which considers both discrete and continuous scenarios and system with multiply states. Two fault tree based Bayesian networks were applied in the case study to better present the application of Bayesian networks in system reliability analysis.

Some highlights in this thesis:

- Reviewed the development of Bayesian networks. Approaches for Bayesian networks structure learning, parameters learning were concluded. Problems and limitation associated with Bayesian network inference were summarized. Research Status of Bayesian networks was introduced. Compared Bayesian networks with traditional system reliability analysis methods and concluded its advantages and limitations.
- Designed a water filter system with both discrete and continuous failure probability distribution. A fault tree analysis and a further Bayesian network approach were presented. Scenario I expanded the system with binary states to a multistate system. In addition, by discretizing the continuous failure probability distribution in scenario II, a more proper model which mirrors more reality was introduced and satisfying results were generated from this model. The bidirectional inference for conditional

probabilities with given evidence presents the advantage of Bayesian. By adding evidence in Bayesian network model, the failure rate of components or the system can be updated. A more accurate and targeted analysis can be conducted to ensure the reliability of system and give suggestions for failure detection and system maintenance.

- A BNs Toolbox for Matlab was applied in the case study. In order to analyze the system with continuous failure distribution, a Matlab program was coded to deal with multistate nodes and discretize the continuous variables. Similar systems with continuous time associated components can be easily assessed by adjusting the model in the case study.

4.2 Limitations

In this thesis, the review of Bayesian networks and the case study which BNs applied indicate that Bayesian networks still have some limitations. Even though building the structure of Bayesian networks is not a very complex work, the quantitative part of BNs is difficult to define. BNs need the support of many prior probabilities and conditional probabilities. To collect these probabilities is a huge work and easy to be influenced by subjective opinions. Also, it is hard to get the prior knowledge and transform the information to probabilities. Sometimes, domain experts are needed when building BNs and the work is hard to quantify.

Also, learning BN structure and model parameters can be NP-hard (non-deterministic polynomial). The Bayesian networks learning is enormously limited by the size and complexity of systems. It can take a lot of time to complete the inference, especially when there are many undirected loops in the BNs.

Except that, BNs cannot deal with the continuous variables very well, especially for systems having continuous failure mode. BNs are efficient for calculating the distribution with discrete variables. When using the BNs, scholars always need to do some discretization of continuous variables. Even though those systems can be discretized to

multistate dynamic systems, it may not get a good estimation. Bayesian network inference for large, safety critical and complex hybrid system is still under development.

Regarding the case study, even though there are some expanded research in Scenario I and Scenario II, it is still a very simplified model. The real-world system reliability problems cannot be fitted as the model is narrowly defined.

4.3 Recommendation

A lot of research about the Bayesian networks combined with fault tree analysis, Markov Chain Monte Carlo, K2 algorithm and EM algorithm has been done. As the Bayesian network learning is highly related to some machine learning and data mining techniques, while those techniques are not widely applied for Bayesian networks with system reliability assessment purpose, there is a potential possibility that some other algorithms, such as neural networks and support vector machine, are helpful for Bayesian networks structure and parameters learning. Further research on continuous Bayesian network reliability modeling is necessary to build more realistic models and make more accurate and approximative inference.

A more complete model regarding the system in the case study need to be considered. An additional research issue to be tackled is the number of states for components in a real system can be more than just several. As the number of states increases, the complexity of conditional probability table will increase dramatically. It is getting difficult to use Bayesian Networks in system reliability analysis. Also, only consider discrete states is not appropriate. A continuous failure state combined with gradually degenerative failure mode is a possible research direction to integrate the model.

REFERENCES

- [1] Elsayed, Elsayed A. *Reliability engineering*. Wiley Publishing, 2012.
- [2] Onisawa, Takehisa. "An approach to human reliability in man-machine systems using error possibility." *Fuzzy Sets and Systems* 27.2 (1988): 87-103.
- [3] Grover, Jeff. *Strategic economic decision-making: Using Bayesian belief networks to solve complex problems*. Vol. 9. Springer, 2012.
- [4] Pearl, Judea. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.
- [5] Bolstad, William M. *Introduction to Bayesian statistics*. John Wiley & Sons, 2007.
- [6] Heckerman, David. *A tutorial on learning with Bayesian networks*. Springer Netherlands, 1998.
- [7] S. Russell, P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, 1995.
- [8] Murphy, Kevin. "An introduction to graphical models." *A Brief Introduction to Graphical Models and Bayesian Networks* 10 (2001).
- [9] Langseth, Helge, and Luigi Portinale. "Bayesian networks in reliability." *Reliability Engineering & System Safety* 92.1 (2007): 92-108.

- [10] Mahadevan, Sankaran, Ruoxue Zhang, and Natasha Smith. "Bayesian networks for system reliability reassessment." *Structural Safety* 23.3 (2001): 231-251.
- [11] Doguc, Ozge, and Jose Emmanuel Ramirez-Marquez. "A generic method for estimating system reliability using Bayesian networks." *Reliability Engineering & System Safety* 94.2 (2009): 542-550.
- [12] Langseth, Helge. "Bayesian networks in reliability: The good, the bad, and the ugly." *Advances in Mathematical Modeling for Reliability* (2008): 1.
- [13] Langseth, Helge, et al. "Inference in hybrid Bayesian networks." *Reliability Engineering & System Safety* 94.10 (2009): 1499-1509.
- [14] Neil, Martin, Manesh Tailor, and David Marquez. "Inference in hybrid Bayesian networks using dynamic discretization." *Statistics and Computing* 17.3 (2007): 219-233.
- [15] Cooper, Gregory F., and Edward Herskovits. "A Bayesian method for the induction of probabilistic networks from data." *Machine learning* 9.4 (1992): 309-347.
- [16] Chickering, David Maxwell. "Optimal structure identification with greedy search." *The Journal of Machine Learning Research* 3 (2003): 507-554.
- [17] Dechter, Rina, Itay Meiri, and Judea Pearl. "Temporal constraint networks." *Artificial intelligence* 49.1 (1991): 61-95.
- [18] Spirtes, Peter, Clark N. Glymour, and Richard Scheines. *Causation, prediction, and search*. Vol. 81. MIT press, 2000.

- [19] Lee, Peter M. *Bayesian statistics: an introduction*. John Wiley & Sons, 2012.
- [20] Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society. Series B (Methodological)* (1977): 1-38.
- [21] Cozman, Fabio Gagliardi. "Generalizing variable elimination in Bayesian networks." *Workshop on Prob. Reasoning in Bayesian Networks at SBIA/Iberamia*. 2000.
- [22] Zheng, Lu. *Parallel Junction Tree Algorithm on GPU*. Diss. Carnegie Mellon University, 2013.
- [23] Gelfand, Alan E., et al. "Illustration of Bayesian inference in normal data models using Gibbs sampling." *Journal of the American Statistical Association* 85.412 (1990): 972-985.
- [24] Marquez, David, Martin Neil, and Norman Fenton. "Improved reliability modeling using Bayesian networks and dynamic discretization." *Reliability Engineering & System Safety* 95.4 (2010): 412-425.
- [25] Bingham, Tom. "System Reliability through Fault Tree Analysis." *ASQ World Conference on Quality and Improvement Proceedings, Seattle, WA*. Vol. 59 No. 0
- [26] Hydration Technologies Inc, "RO keeps soldiers alive" *Membrane Technology* Vol2004, Issue 4, p1

[27] World Health Organization. "' Progress on Drinking-Water and Sanitation–2012 Update" launched on 6 March 2012." (2012).

[28] Prüss-Üstün, Annette, et al. *Safer water, better health: costs, benefits and sustainability of interventions to protect and promote health*. World Health Organization, 2008.

[29] Petersen, Robert J. "Composite reverse osmosis and nanofiltration membranes." *Journal of membrane science* 83.1 (1993): 81-150.

[30] Murphy, Kevin. "The bayes net toolbox for matlab." *Computing science and statistics* 33.2 (2001): 1024-1034.

[31] Murphy, Kevin Patrick. *Dynamic bayesian networks: representation, inference and learning*. Diss. University of California, Berkeley, 2002.

APPENDIX I

MATLAB CODE AND RESULTS FOR SCENARIO I


```

marg_T3(i,:)=marg3.T;
%add evidence i=1, calculate p(T=1|i=1)
evidence = cell(1,N);
evidence{i} = 1;
[engine, loglike] = enter_evidence(engine, evidence);
marg3= marginal_nodes(engine, T);
marg_T(i,:)=marg3.T;
end

%given A good S failed, p(T=1|A=2,W=1)
evidence = cell(1,N);
evidence{A} = 2;
evidence{S} = 1;
[engine, loglike3] = enter_evidence(engine, evidence);
m = marginal_nodes(engine, T);

prob_sys_1 % system failed
prob_sys_2 % system operating
prob_sys_3 % system slow

marg_T1 %CPD given T=1
marg_T2 %CPD given T=2
marg_T3 %CPD given T=3

marg_T %CPD given each node=1
m.T %%CPD for node T given A functional and S failed

```

Matlab Results for Scenario I

```
prob_sys_1 =  
    0.7137  
  
prob_sys_2 =  
    0.2434  
  
prob_sys_3 =  
    0.0429  
  
marg_T1 =  
    0.7006    0.2994  
    0.3503    0.6497  
    0.2102    0.7898  
    0.1401    0.8599  
         0         0  
    0.1500    0.8500  
         0         0  
    0.0028    0.9972  
  
marg_T2 =  
     0     1  
     0     1  
     0     1  
     0     1  
     0     0  
     0     1  
     0     0  
     0     1  
  
marg_T3 =  
     0     1  
     0     1  
     0     1  
     0     1  
     0     0  
     1     0  
     0     0  
     0     1  
  
marg_T =  
    1.0000     0     0  
    1.0000     0     0  
    1.0000     0     0
```

```
1.0000    0    0
  0      0    0
0.7137    0  0.2863
  0      0    0
1.0000    0    0

ans =
  0.4274
    0
  0.5726

>>
```

APPENDIX II

MATLAB CODE AND RESULTS FOR SCENARIO II

Matlab Code for the Dynamic Fault Tree Inference

```
%make the expression for the failure prob. for node A, B, C, D
lambda=[0.00025 0.00015 0.0001 0.00005];
prob_w=0.002;
for m=1:25 %count 1 for every 2 weeks
    h=336; %Time interval, 336 hours per 2 weeks
    t(m)=m*h;
    for i=1:4
        prob(i,m)=1-exp((-1)*lambda(i)*t(m));
    end;
    prob_sys(m)=1-(1-prob(1,m))*(1-prob(2,m))*(1-prob(3,m))*(1-prob(4,m))*(1-prob_w);
end;
prob %failure rate for node A, B, C, D
prob_sys %system failures for each iteration
```


Matlab Results for the Dynamic Fault Tree Inference

```
prob =  
  
Columns 1 through 9  
0.0806 0.1546 0.2228 0.2854 0.3430 0.3959 0.4446 0.4893 0.5305  
0.0492 0.0959 0.1403 0.1826 0.2228 0.2610 0.2973 0.3318 0.3647  
0.0330 0.0650 0.0959 0.1258 0.1546 0.1826 0.2096 0.2357 0.2610  
0.0167 0.0330 0.0492 0.0650 0.0806 0.0959 0.1109 0.1258 0.1403  
  
Columns 10 through 18  
0.5683 0.6031 0.6351 0.6645 0.6915 0.7163 0.7392 0.7602 0.7795  
0.3959 0.4256 0.4538 0.4807 0.5062 0.5305 0.5535 0.5755 0.5963  
0.2854 0.3090 0.3318 0.3539 0.3752 0.3959 0.4159 0.4352 0.4538  
0.1546 0.1687 0.1826 0.1962 0.2096 0.2228 0.2357 0.2484 0.2610  
  
Columns 19 through 25  
0.7973 0.8136 0.8286 0.8424 0.8551 0.8668 0.8775  
0.6162 0.6351 0.6530 0.6700 0.6863 0.7017 0.7163  
0.4719 0.4893 0.5062 0.5225 0.5383 0.5535 0.5683  
0.2733 0.2854 0.2973 0.3090 0.3205 0.3318 0.3430  
  
prob_sys =  
  
Columns 1 through 9  
0.1704 0.3104 0.4267 0.5235 0.6039 0.6707 0.7263 0.7725 0.8108  
  
Columns 10 through 18  
0.8428 0.8693 0.8913 0.9097 0.9249 0.9376 0.9481 0.9569 0.9641  
  
Columns 19 through 25  
0.9702 0.9752 0.9794 0.9829 0.9858 0.9882 0.9902  
  
>>
```



```

margS1 = marginal_nodes(engine, S);
margE1 = marginal_nodes(engine, E);
margW1 = marginal_nodes(engine, W);
marg_A1(:,m)=margA1.T;
marg_B1(:,m)=margB1.T;
marg_C1(:,m)=margC1.T;
marg_D1(:,m)=margD1.T;
marg_F1(:,m)=margF1.T;
marg_S1(:,m)=margS1.T;
marg_E1(:,m)=margE1.T;
marg_W1(:,m)=margW1.T;
%add evidence T=2
evidence = cell(1,N);
evidence{T} = 2;
[engine, loglike] = enter_evidence(engine, evidence);
margA2 = marginal_nodes(engine, A);
margB2 = marginal_nodes(engine, B);
margC2 = marginal_nodes(engine, C);
margD2 = marginal_nodes(engine, D);
margF2 = marginal_nodes(engine, F);
margS2 = marginal_nodes(engine, S);
margE2 = marginal_nodes(engine, E);
margW2 = marginal_nodes(engine, W);
marg_A2(:,m)=margA2.T;
marg_B2(:,m)=margB2.T;
marg_C2(:,m)=margC2.T;
marg_D2(:,m)=margD2.T;
marg_F2(:,m)=margF2.T;
marg_S2(:,m)=margS2.T;
marg_E2(:,m)=margE2.T;
marg_W2(:,m)=margW2.T;
%add evidence T=3
evidence = cell(1,N);
evidence{T} = 3;
[engine, loglike] = enter_evidence(engine, evidence);
margA3 = marginal_nodes(engine, A);
margB3 = marginal_nodes(engine, B);
margC3 = marginal_nodes(engine, C);
margD3 = marginal_nodes(engine, D);
margF3 = marginal_nodes(engine, F);
margS3 = marginal_nodes(engine, S);
margE3 = marginal_nodes(engine, E);
margW3 = marginal_nodes(engine, W);
marg_A3(:,m)=margA3.T;
marg_B3(:,m)=margB3.T;
marg_C3(:,m)=margC3.T;
marg_D3(:,m)=margD3.T;
marg_F3(:,m)=margF3.T;
marg_S3(:,m)=margS3.T;

```

```

    marg_E3(:,m)=margE3.T;
    marg_W3(:,m)=margW3.T;
for j=[A B C D F S E W]
    evidence = cell(1,N);
    evidence{j} = 1;
    [engine, loglike] = enter_evidence(engine, evidence);
    marg_T = marginal_nodes(engine, T);
    marg_T_j(:,m,j)=marg_T.T;
end;
    evidence = cell(1,N);
    evidence{A} = 2;
    evidence{S} = 1;
    [engine, loglike] = enter_evidence(engine, evidence);
    marg_as = marginal_nodes(engine, T);
    marg_A_S(:,m)=marg_as.T;
end;
prob_sys_1 % system failed
prob_sys_2 % system operating
prob_sys_3 % system slow
prob %failure probs for filter A, B, C, D
%CPD for each node for each iteration with T=1
marg_A1
marg_B1
marg_C1
marg_D1
marg_F1
marg_S1
marg_E1
marg_W1
%CPD for each node for each iteration with T=2
marg_A2
marg_B2
marg_C2
marg_D2
marg_F2
marg_S2
marg_E2
marg_W2
%CPD for each node for each iteration with T=3
marg_A3
marg_B3
marg_C3
marg_D3
marg_F3
marg_S3
marg_E3
marg_W3
marg_T_j %CPD for node T for each iteration given each node i=1
marg_A_S %CPD for node T for each iteration given A functional and S failed

```

Matlab Results for the Dynamic Bayesian Network Inference

prob_sys_1 =									
Columns 1 through 9									
0.1704	0.3104	0.4267	0.5235	0.6039	0.6707	0.7263	0.7725	0.8108	
Columns 10 through 18									
0.8428	0.8693	0.8913	0.9097	0.9249	0.9376	0.9481	0.9569	0.9641	
Columns 19 through 25									
0.9702	0.9752	0.9794	0.9829	0.9858	0.9882	0.9902			
prob_sys_2 =									
Columns 1 through 9									
0.7052	0.5862	0.4873	0.4051	0.3367	0.2799	0.2327	0.1934	0.1608	
Columns 10 through 18									
0.1337	0.1111	0.0924	0.0768	0.0638	0.0530	0.0441	0.0367	0.0305	
Columns 19 through 25									
0.0253	0.0211	0.0175	0.0146	0.0121	0.0101	0.0084			
prob_sys_3 =									
Columns 1 through 9									
0.1244	0.1034	0.0860	0.0715	0.0594	0.0494	0.0411	0.0341	0.0284	
Columns 10 through 18									
0.0236	0.0196	0.0163	0.0135	0.0113	0.0094	0.0078	0.0065	0.0054	
Columns 19 through 25									
0.0045	0.0037	0.0031	0.0026	0.0021	0.0018	0.0015			
prob =									
Columns 1 through 9									
0.0806	0.1546	0.2228	0.2854	0.3430	0.3959	0.4446	0.4893	0.5305	
0.0492	0.0959	0.1403	0.1826	0.2228	0.2610	0.2973	0.3318	0.3647	
0.0330	0.0650	0.0959	0.1258	0.1546	0.1826	0.2096	0.2357	0.2610	
0.0167	0.0330	0.0492	0.0650	0.0806	0.0959	0.1109	0.1258	0.1403	
Columns 10 through 18									
0.5683	0.6031	0.6351	0.6645	0.6915	0.7163	0.7392	0.7602	0.7795	
0.3959	0.4256	0.4538	0.4807	0.5062	0.5305	0.5535	0.5755	0.5963	
0.2854	0.3090	0.3318	0.3539	0.3752	0.3959	0.4159	0.4352	0.4538	
0.1546	0.1687	0.1826	0.1962	0.2096	0.2228	0.2357	0.2484	0.2610	
Columns 19 through 25									
0.7973	0.8136	0.8286	0.8424	0.8551	0.8668	0.8775			
0.6162	0.6351	0.6530	0.6700	0.6863	0.7017	0.7163			
0.4719	0.4893	0.5062	0.5225	0.5383	0.5535	0.5683			
0.2733	0.2854	0.2973	0.3090	0.3205	0.3318	0.3430			
marg_A1 =									
Columns 1 through 9									

0.4728	0.4983	0.5220	0.5452	0.5679	0.5903	0.6121	0.6335	0.6542
0.5272	0.5017	0.4780	0.4548	0.4321	0.4097	0.3879	0.3665	0.3458
Columns 10 through 18								
0.6743	0.6937	0.7125	0.7304	0.7476	0.7640	0.7796	0.7945	0.8085
0.3257	0.3063	0.2875	0.2696	0.2524	0.2360	0.2204	0.2055	0.1915
Columns 19 through 25								
0.8218	0.8343	0.8461	0.8571	0.8675	0.8772	0.8863		
0.1782	0.1657	0.1539	0.1429	0.1325	0.1228	0.1137		
marg_B1 =								
Columns 1 through 9								
0.2885	0.3089	0.3288	0.3488	0.3689	0.3891	0.4093	0.4296	0.4497
0.7115	0.6911	0.6712	0.6512	0.6311	0.6109	0.5907	0.5704	0.5503
Columns 10 through 18								
0.4698	0.4896	0.5091	0.5284	0.5473	0.5658	0.5838	0.6014	0.6185
0.5302	0.5104	0.4909	0.4716	0.4527	0.4342	0.4162	0.3986	0.3815
Columns 19 through 25								
0.6351	0.6512	0.6667	0.6817	0.6962	0.7101	0.7235		
0.3649	0.3488	0.3333	0.3183	0.3038	0.2899	0.2765		
marg_C1 =								
Columns 1 through 9								
0.1939	0.2094	0.2247	0.2402	0.2561	0.2722	0.2886	0.3051	0.3218
0.8061	0.7906	0.7753	0.7598	0.7439	0.7278	0.7114	0.6949	0.6782
Columns 10 through 18								
0.3386	0.3554	0.3723	0.3890	0.4057	0.4222	0.4386	0.4548	0.4707
0.6614	0.6446	0.6277	0.6110	0.5943	0.5778	0.5614	0.5452	0.5293
Columns 19 through 25								
0.4864	0.5017	0.5168	0.5316	0.5461	0.5602	0.5739		
0.5136	0.4983	0.4832	0.4684	0.4539	0.4398	0.4261		
marg_D1 =								
Columns 1 through 9								
0.0978	0.1065	0.1152	0.1242	0.1334	0.1430	0.1528	0.1628	0.1731
0.9022	0.8935	0.8848	0.8758	0.8666	0.8570	0.8472	0.8372	0.8269
Columns 10 through 18								
0.1835	0.1941	0.2048	0.2157	0.2266	0.2376	0.2486	0.2596	0.2707
0.8165	0.8059	0.7952	0.7843	0.7734	0.7624	0.7514	0.7404	0.7293
Columns 19 through 25								
0.2817	0.2926	0.3035	0.3144	0.3251	0.3358	0.3464		
0.7183	0.7074	0.6965	0.6856	0.6749	0.6642	0.6536		
marg_F1 =								
Columns 1 through 9								
0.9902	0.9955	0.9973	0.9982	0.9987	0.9990	0.9992	0.9994	0.9995

0.0098	0.0045	0.0027	0.0018	0.0013	0.0010	0.0008	0.0006	0.0005
Columns 10 through 18								
0.9996	0.9997	0.9998	0.9998	0.9998	0.9999	0.9999	0.9999	0.9999
0.0004	0.0003	0.0002	0.0002	0.0002	0.0001	0.0001	0.0001	0.0001
Columns 19 through 25								
0.9999	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000		
0.0001	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000		
marg_S1 =								
Columns 1 through 9								
0.1500	0.1500	0.1500	0.1500	0.1500	0.1500	0.1500	0.1500	0.1500
0.8500	0.8500	0.8500	0.8500	0.8500	0.8500	0.8500	0.8500	0.8500
Columns 10 through 18								
0.1500	0.1500	0.1500	0.1500	0.1500	0.1500	0.1500	0.1500	0.1500
0.8500	0.8500	0.8500	0.8500	0.8500	0.8500	0.8500	0.8500	0.8500
Columns 19 through 25								
0.1500	0.1500	0.1500	0.1500	0.1500	0.1500	0.1500		
0.8500	0.8500	0.8500	0.8500	0.8500	0.8500	0.8500		
marg_E1 =								
Columns 1 through 9								
0.9902	0.9955	0.9973	0.9982	0.9987	0.9990	0.9992	0.9994	0.9995
0.0083	0.0038	0.0023	0.0016	0.0011	0.0008	0.0006	0.0005	0.0004
0.0015	0.0007	0.0004	0.0003	0.0002	0.0001	0.0001	0.0001	0.0001
Columns 10 through 18								
0.9996	0.9997	0.9998	0.9998	0.9998	0.9999	0.9999	0.9999	0.9999
0.0003	0.0003	0.0002	0.0002	0.0001	0.0001	0.0001	0.0001	0.0001
0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Columns 19 through 25								
0.9999	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000		
0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		
marg_W1 =								
Columns 1 through 9								
0.0117	0.0064	0.0047	0.0038	0.0033	0.0030	0.0028	0.0026	0.0025
0.9883	0.9936	0.9953	0.9962	0.9967	0.9970	0.9972	0.9974	0.9975
Columns 10 through 18								
0.0024	0.0023	0.0022	0.0022	0.0022	0.0021	0.0021	0.0021	0.0021
0.9976	0.9977	0.9978	0.9978	0.9978	0.9979	0.9979	0.9979	0.9979
Columns 19 through 25								
0.0021	0.0021	0.0020	0.0020	0.0020	0.0020	0.0020		
0.9979	0.9979	0.9980	0.9980	0.9980	0.9980	0.9980		
marg_A2 =								

Columns 1 through 16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 17 through 25	0	0	0	0	0	0	0	0	0						
	1	1	1	1	1	1	1	1	1						
marg_B2 =															
Columns 1 through 16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 17 through 25	0	0	0	0	0	0	0	0	0						
	1	1	1	1	1	1	1	1	1						
marg_C2 =															
Columns 1 through 16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 17 through 25	0	0	0	0	0	0	0	0	0						
	1	1	1	1	1	1	1	1	1						
marg_D2 =															
Columns 1 through 16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 17 through 25	0	0	0	0	0	0	0	0	0						
	1	1	1	1	1	1	1	1	1						
marg_F2 =															
Columns 1 through 16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 17 through 25	0	0	0	0	0	0	0	0	0						
	1	1	1	1	1	1	1	1	1						
marg_S2 =															
Columns 1 through 16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Columns 17 through 25	0	0	0	0	0	0	0	0	0						


```

1 1 1 1 1 1 1 1 1
marg_E2 =
Columns 1 through 16
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Columns 17 through 25
0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0

marg_W2 =
Columns 1 through 16
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Columns 17 through 25
0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1

marg_A3 =
Columns 1 through 16
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Columns 17 through 25
0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1

marg_B3 =
Columns 1 through 16
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Columns 17 through 25
0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1

marg_C3 =
Columns 1 through 16
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Columns 17 through 25
0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1

marg_D3 =

```

```

Columns 1 through 16
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Columns 17 through 25
0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1

marg_F3 =
Columns 1 through 16
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Columns 17 through 25
0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1

marg_S3 =
Columns 1 through 16
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Columns 17 through 25
1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0

marg_E3 =
Columns 1 through 16
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Columns 17 through 25
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1

marg_W3 =
Columns 1 through 16
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Columns 17 through 25
0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1

marg_T_j(:,:,1) =
Columns 1 through 9
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0 0 0

```

```

0 0 0 0 0 0 0 0 0
Columns 10 through 18
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
Columns 19 through 25
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0
0 0 0 0 0 0 0

marg_T_j(:, :, 2) =
Columns 1 through 9
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
Columns 10 through 18
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
Columns 19 through 25
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0
0 0 0 0 0 0 0

marg_T_j(:, :, 3) =
Columns 1 through 9
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
Columns 10 through 18
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
Columns 19 through 25
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0
0 0 0 0 0 0 0

marg_T_j(:, :, 4) =
Columns 1 through 9
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
Columns 10 through 18
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000

```

```

0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
Columns 19 through 25
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0
0 0 0 0 0 0 0

marg_T_j(:,5) =
Columns 1 through 9
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
Columns 10 through 18
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
Columns 19 through 25
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0
0 0 0 0 0 0 0

marg_T_j(:,6) =
Columns 1 through 9
0.1704 0.3104 0.4267 0.5235 0.6039 0.6707 0.7263 0.7725 0.8108
0 0 0 0 0 0 0 0 0
0.8296 0.6896 0.5733 0.4765 0.3961 0.3293 0.2737 0.2275 0.1892
Columns 10 through 18
0.8428 0.8693 0.8913 0.9097 0.9249 0.9376 0.9481 0.9569 0.9641
0 0 0 0 0 0 0 0 0
0.1572 0.1307 0.1087 0.0903 0.0751 0.0624 0.0519 0.0431 0.0359
Columns 19 through 25
0.9702 0.9752 0.9794 0.9829 0.9858 0.9882 0.9902
0 0 0 0 0 0 0
0.0298 0.0248 0.0206 0.0171 0.0142 0.0118 0.0098

marg_T_j(:,7) =
Columns 1 through 9
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
Columns 10 through 18
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
Columns 19 through 25

```

```

1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
  0    0    0    0    0    0    0
  0    0    0    0    0    0    0

marg_T_j(:, :, 8) =
Columns 1 through 9
 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
  0    0    0    0    0    0    0    0    0
  0    0    0    0    0    0    0    0    0
Columns 10 through 18
 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
  0    0    0    0    0    0    0    0    0
  0    0    0    0    0    0    0    0    0
Columns 19 through 25
 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
  0    0    0    0    0    0    0
  0    0    0    0    0    0    0

marg_A_S =
Columns 1 through 9
 0.0977 0.1842 0.2624 0.3332 0.3971 0.4549 0.5072 0.5544 0.5972
  0    0    0    0    0    0    0    0
 0.9023 0.8158 0.7376 0.6668 0.6029 0.5451 0.4928 0.4456 0.4028
Columns 10 through 18
 0.6358 0.6707 0.7023 0.7308 0.7566 0.7800 0.8011 0.8201 0.8374
  0    0    0    0    0    0    0    0
 0.3642 0.3293 0.2977 0.2692 0.2434 0.2200 0.1989 0.1799 0.1626
Columns 19 through 25
 0.8530 0.8671 0.8798 0.8913 0.9018 0.9112 0.9197
  0    0    0    0    0    0
 0.1470 0.1329 0.1202 0.1087 0.0982 0.0888 0.0803

>>

```