NTNU – Trondheim
Norwegian University of
Science and Technology

Stochastic Modelling for Condition Based Maintenance

Zehan Han

Reliability, Availability, Maintainability and Safety (RAMS)
Submission date: June 2015
Supervisor: Anne Barros, IPK

Norwegian University of Science and Technology
Department of Production and Quality Engineering

RAMS
Reliability, Availability,
Maintainability, and Safety

Stochastic Modelling for Condition Based Maintenance

Zehan Han

August 2014

MASTER THESIS

Department of Production and Quality Engineering
Norwegian University of Science and Technology

Supervisor: Professor Anne Barros

This page is intentionally left blank

“Normal people… believe that if it ain’t broke, don’t fix it. Engineers believe that if it ain’t broke, it doesn’t have enough features yet” - *Scott Adams*

Preface

This Master's thesis is carried out at Department of Production and Quality Engineering NTNU. It is a part of the 2-year master program RAMS (Reliability, Availability, Maintainability and Safety). This master's thesis is written during the spring semester of 2015. Data set for case study in master's project which is initiated in 2014, is received from SINTEF. This master's thesis is written for readers with some knowledge in RAMS, maintenance optimization, statistic theories, stochastic processes, the BOP (Blowout Preventer) system and basic Matlab programming.

Trondheim, 2015-06-14

Zehan Han

Acknowledgment

First of all, I am using this opportunity to express my warn thanks to my supervisor Professor Anne Barros for her continuously patient teaching and illuminating me throughout this Master's thesis. I am extremely thankful and indebted to her for sharing her expertise in CBM, and sincere and valuable guidance. With a very busy schedule, she can always try to find a time for me. Secondly, I would like to thank Professor Jørn Vatn, providing me knowledge throughout RAMS courses and the previous Master's project. I am grateful for his constructive criticism and friendly advice. I would also like to thank my project external guide Stig Ole Johnsen from SINTEF, who provides me with the opportunity and shares his truthful and illuminating views on a number of issues related to CBM. In addition, I take this opportunity to express gratitude to all of the Department faculty members for their help and support.

Special thanks go to my family. Words cannot express how grateful I am to my mother and father for all of the sacrifices that you've made on my behalf. I would also like to thank all of my friends who support me in writing. At the end I would like to express appreciation to my beloved fiancée Qingqing who is always my support in the moments when there is no one to answer my queries.

Z.H.

Summary and Conclusions

This Master's thesis covers almost all aspects of Condition Based Maintenance (CBM). All objectives in Chapter 1 are met. The thesis is mainly comprised of three parts. First part introduces the world of CBM to readers. This part presents data acquisition, data processing and databases, which are the foundation to CBM. Then it highlights models which are divided into physics based models, data-driven models and hybrid models, for diagnostic and prognostic use. Three promising diagnostic and prognostic models are specified, i.e., Markov model, Artificial neural networks and the time-dependent proportional hazard model. Afterwards, CBM main steps are presented in Figure 2.2. This figure is made based on a large quantity of literature review and can function as an index when readers are querying CBM data, diagnostic and prognostic models and steps. It can also give readers a whole picture of CBM. Next, introductions of Prognostic and Health Management (PHM), CBM industry applications and CBM state of the art are followed. Specific challenges, phenomena and questions are summarized.

Second part presents a Matlab toolbox made by the writer. This toolbox estimates components' Remaining Useful Life (RUL) with a standard deviation, the probability to survive till the next maintenance time and Probability of Failure on Demand (PFD) based on numerous simulations. The stochastic processes behind are the (continuous time) Markov model, the Brownian motion process and the Gamma process. Users can choose among them in the toolbox. Specially, MTTF and the steady-state-probability can be achieved by using the Markov model. This toolbox is used often in the next part when data is analyzed. The writer makes the graphic interface of this toolbox easy for people to use. All instructions are given. All code is also attached, from the whole toolbox code to a tiny simulation step with detailed explanations. This toolbox makes it possible for the people with little knowledge in statistics and maintenance to make their own maintenance plans. This toolbox can be download on-line.

The third part of this Master's thesis uses 7 statistic models and 3 stochastic processes to model the degradation process of the elastomeric annular body from the annular preventer of a BOP system. To make these models, many relevant papers and books are studied. In this Master's

thesis, these models are not just "theories" or "formulas". Instead, for each model, the writer gives a vivid example by analyzing the data with all Matlab code and detailed explanations following. The writer believe by doing this readers can have a deeper understanding of each model. They may use one of these models for their own data in the future. To make it easier for readers to follow, the difficulty of these models is increasing one after one. The complex model can give a more precise estimation of the lifetime with more influence factors being taken into consideration. As to the structure of the third part, firstly, much literature about BOP is read. A brief description about BOP systems is followed. Virtual failure data is simulated based on a trustful BOP reliability report. The exponential model is firstly used to give a preliminary understanding of the data. Afterwards, the Weibull model, the log-logistic model and the log-normal model are used. All these models use Maximum likelihood Estimation (MLE). Minitab is the analysing software used here. Then, the Brownian motion process is introduced to model the degradation process. Next, the covariates are introduced (e.g., the temperature). The Weibull regression model is elaborated followed by Proportional hazard model (Cox regression model) and Arrhenius model. These three are very promising models used in CBM. Brownian motion is used again to model the degradation. However, this time, the covariates are taken into account. It leads to the change of the path of the Brownian motion process each time when covariates are changing. It is more complex but more realistic. This is the final step to model the degradation in a component level. To model the degradation in a system level, two extra models are included. That is the Markov model and the Brownian motion process for a koon system with covariates. They are shown in the same chapter. Finally, relevant maintenance plans are made based on the result of "RUL" and "the probability to survive till the next maintenance interval.

Contents

Chapter 1

Introduction

1.1 Background

Condition Based Maintenance (CBM) is a maintenance program that recommends maintenance decisions based on the information collected through condition monitoring (Jardine et al., 2006). By implementing CBM, the under-maintenance situation and over-maintenance situation can be reduced, resulting in the reduction in downtime, spare parts inventory, maintenance cost and risk hazards.

A report from International Atomic Energy Agency tells a successful CBM application for the nuclear plants. The use of CBM enables intrusive maintenance operations to be replaced by diagnostic. By such way, the stipulated inspection interval could be prolonged and total uptime could be ensured. For example, for electric servo-motors, the original interval for complete maintenance is 10 years. By performing diagnostic of leak tightness enables this interval to be raised to 15 years, the availability deviations are highlighted (of about 5 to 7%) and downtime due to incorrect operation is reduced by one third. Finally, given the number of valves (6000) in the EDF fleet, the savings were about 300000 euros /year (I.A.E.A, 2007).

More and more similar stories can be found, telling CBM is a correct tool to both enhance the reliability and reduce the maintenance cost. Recently, CBM attracts many interests, both in industries and researches. Figure 2.2 shows the main steps in CBM. literature shows that many types of data and models could be used in CBM. From the literature survey, one clear phenomenon is that the theoretical models of CBM are well developed. Numerous advanced mathematical models are proposed so far. Usually these models are validated by laboratory experiments only and the results indicate corresponding methods are satisfying. With the help of modern computers, these sophisticated mathematical models can be easily applied.

A key indicator for equipment health condition in CBM is called Remaining Useful Life (RUL). RUL is the useful life left on an asset at a particular time of operation. Its estimation is central to condition based maintenance and prognostic and health management (Si et al., 2011). RUL is usually assessed through various sources (i.e., performance data and condition monitoring data). Various methods on assessing RUL have been shown on many publications.

Despite the huge achievements found from literature within the CBM concept, there are many remaining problems. Some problems are listed below:

1. Are there any universal rule to choose type of data and models used in the rotating machinery and the non-rotating machinery?

2. It is believed that theory/models of CBM have reached their bottlenecks and application of CBM in industry has just started. There is a huge gap between theory and application. How can future work narrow the gap? That is, the industry should give more data to the academic world. In return, the academic world should support the industry by supplying more promising models.

3. How to combine both performance data (e.g., the speed) and condition monitoring data (e.g., the vibration) to make maintenance decisions?

4. Methods for fault diagnostic are numerous. As to prognostic, due to its huge uncertainty, there is high risk in making the maintenance decisions based on the assessment of RUL. Usually validations for RUL are undertaken in laboratory only. How can further work improve the accuracy of RUL assessment?

5. What are the most promising models for prognostic shown in the literature?

1.1.1 Literature Survey

Literature survey for this Master's thesis is comprised of three parts. First part is to find the state of the art of CBM, emphasising on models and applications. Large quantity of literature survey has been performed on CBM and relevant topics. Some representative review papers and suggested papers for readers with interests in CBM are also shown. Techniques within the Prognostic and Health Management (PHM) field are also reviewed. Second part is to view literature concerning Blowout Preventer (BOP). It covers BOP test methods, BOP components descriptions and BOP monitoring. In addition, literature involving elastomer condition monitoring is surveyed since these elastomers are critical to annular preventers in a BOP system. The third part is to study some books and papers regarding to stochastic processes.

There are many publications on models used in CBM. For this report, the publications from acknowledged scientific journals are selected. These publications are ranked according to cited numbers, reputation of author/university, published date and relevance. Only the publications with high rank are reviewed and cited in this article. Many representative models are discussed in this master's thesis.

This paragraph lists some inspiring review papers regarding to CBM. Jardine et al. (2006) is a great review paper for CBM. It visualizes a very clear structure about CBM including many data types and models. However, most papers cited are around year 2000. Considering researches in the CBM area developing rapidly, and many heuristic papers being published in the last decade, especially for prognostic models, more recently published review papers are also summarised. Heng et al. (2009) synthesises and places methods for rotating machinery prognostics from publications in the past few years. For each method, merits and weaknesses are listed. Si et al. (2011) reviews the recent modeling developments for estimating the RUL. The review is centred on statistical data driven approaches which rely only on available past observed data and statistic models. Lee et al. (2014) provides a comprehensive review of the PHM field (see Section 2.4), followed by an introduction of a systematic PHM design methodology, and methods of converting data to prognostics information.

Suggestion papers for readers with interests in CBM are (Jardine et al., 2006) and (Heng et al., 2009). Suggestion articles for readers with interests in BOP systems are (Kenny, 2004) and (Holand, 1999). A suggested book for stochastic processes is (Ross, 2014). Research centers focusing on CBM can be found in Appendix B.6. Key words to search for CBM papers can be found in Appendix B.7.

1.2 Objectives

The main objectives of this Master's thesis are

1. Introduce CBM to readers. All aspects of CBM, including data/models/the state of the art and industry applications, should be covered.
2. Make a toolbox to help make maintenance plans. It should cover some models with detailed explanations.
3. Use some models to analyze the data in a component level and in a system level with and without the covariates.

1.3 Limitations

This Master's thesis doesn't make any repetitions on formulas and methods which are already elaborated in RAMS courses. The approaches discussed in this Master's thesis for CBM are not covered in RAMS main courses. The writer believe it is a more meaningful job to explore "new field" than repeating the methods which are well taught during the lectures.

Only simulated data and data from Minitab database is used in Chapter 4 to demonstrate the models. In the future, more data from the industry should be used.

1.4 Approach

The first objective is achieved by performing a large quantity of literature review. It consumes a lot of time. The second objective is achieved by both literature survey and Matlab on-line courses learning. Stochastic processes knowledge is gained through a course from NTNU and the supervision provided by Professor Anne Barros. The third objective is achieved by learning the lifetime data analysis and the supervision from Anne.

1.5 Structure of the report

The rest of the report is organized as follows. Chapter 2 elaborates CBM data, CBM models, the state of the art of CBM and CBM industry applications. Chapter 3 presents a self-made toolbox to help make maintenance plans. Chapter 4 first describes BOP systems. Then the data is analyzed by different models with the Matlab code and explanations. Chapter 5 summarizes the thesis and gives recommendations for the further work.

1.6 A summary of previous work

The previous master's project discusses spare parts optimization by using stochastic processes. Vivid examples are given to demonstrate these approaches. Two cost equations are given as well. In addition, gearbox data sets from "Aker Solutions" are analyzed. These data sets are successfully analyzed to assess gearbox health conditions. The model behind is explained. Many efforts are made to analyze these data sets and to write Matlab code. All necessary explanations for Matlab code are also supplied. The result shows that for the first 20-30 minutes, the friction coefficient will increase dramatically. After that, it begins to drop and remains steady for the rest of the operation time. In other words, the most vulnerable time for the gearbox to fail is the first 20-30 minutes when it starts to work.

Chapter 2

The world of condition based maintenance: CBM data, models, steps and the state of the art

This Chapter gives an overview of CBM to readers. It covers almost all aspects of CBM.

☞ **Condition Based Maintenance (CBM)**: It is a maintenance program that recommends maintenance decisions based on the information collected through condition monitoring (Jardine et al., 2006). By implementing CBM, the under-maintenance situation and over-maintenance situation can be reduced, resulting in the reduction in downtime, spare parts inventory, maintenance cost and risk hazards.

☞ **Diagnostic**: Diagnostic is the process of identifying health states, detecting and isolating each failure mode.

☞ **Prognostic**: prognostic is the process of estimating the Remaining Useful Life (RUL), in some cases, being able to continuously make trustful predictions.

2.1 CBM data

This section firstly discusses data acquisition and data processing for CBM. Use of historical data is discussed, and OREDA database is mentioned. Most methods for analyzing data in CBM are covered in this section.

2.1.1 Data acquisition and processing

Data acquisition is the first step to implement CBM for the machine fault diagnostic and prognostic. Useful data contains event data (e.g., breakdown, oil change, preventive maintenance), which is stored usually manually, and condition monitoring data (e.g., vibration, temperature, pressure and oil analysis data), which is stored either manually or automatically. Afterwards useful information from the raw data needs to be extracted. This procedure is called “feature extraction”. For example, from the vibration raw data, time-domain analysis can extract “Root Mean Square (RMS)” and “wavelet transform” can extract “wavelet coefficient” . These extracted features are further used for diagnostic and prognostic. For example, RMS, kurtosis and wavelet coefficient can be used as the inputs in Artificial Neural Network (ANN) or Hidden Semi Markov Model (HSMM) for diagnostic and prognostic.

The algorithms and tools for data processing are dependent on the type of data acquired. For example, oil analysis data and temperature belong to value type data. Vibration data and acoustic data belong to waveform type data. For waveform type data, time-domain analysis, frequency-domain analysis and time-frequency analysis are mostly used methods to extract the features. A time-domain graph shows how a signal changes with time. Popular approaches are time synchronous averaging and time series analysis. In the time-domain analysis, extracted features are usually estimated mean, estimated variance, estimated skewness, estimated kurtosis, root mean square, etc. A frequency-domain graph shows how much of the signal lies within each given frequency band over a range of frequencies. Most widely used analysis is spectrum analysis by means of fast Fourier transform. Here, a spectrum is a collection of sine waves that, when combined properly, produce the time-domain signal under examination. However, machinery fault occurs commonly with non-stationary waveform. Thus time-frequency analysis has been introduced to handle with this problem. Short-time Fourier transform or spectrogram and Wigner-Ville distribution are mostly used. The results suggest that acoustic signals like vibration signals are very effective for the early detection of faults. Infrequently vibration signals are easily affected by mechanical resonance. The acoustic emissions signals show more stable performance. Wavelet transform is another approach to deal with time-frequency analysis.

Value type data are simpler than waveform data. When the number of variables is large, the correlation structure makes analysis complex. Principal component analysis (PCA) and independent component analysis (ICA) are two useful multivariate analysis techniques. Regression analysis is also used for analyzing value type data.

2.1.2 OREDA, data used in SCADA, NASA database

The foundation of CBM is data. The quality of data will eventually influence the accuracy of machine health diagnostic and prognostic. Take measuring vibration data for example. There is a term called “resolution” which is determined by the number of data points measured from accelerometer. It also means that the more data points are captured, the more detailed waveform/spectrum are. A low resolution waveform data set may lead to wrong diagnostic. Following subsections present some databases.

The OREDA handbook collects generic data, containing failure rate, failure number for specific failure mode, and is a limited extract from the OREDA database. See (www.oreda.com). The OREDA database also contains failure time, downtime and more detailed information. Fundamental maintenance optimization can be made by this trend with the target minimizing the cost per time unit. The prediction made based on that cannot be accurate. OREDA gives historical data (failure data), which is less enough for diagnostic and prognostic. For accurate diagnostic/prognostic, real time data is most preferred (i.e., OREDA database could also collect condition monitoring data, in order to help validate diagnostic and prognostic models. However this may be impossible to obtain in general through OREDA project, but it could be gathered from specific companies or through relevant project such as the PROASENSE project. See (www.proasense.eu).

In some wind turbines, SCADA system (supervisory control and data acquisition) acquires and stores real-time process data (at 5-min to 10-min interval). It enables maintenance personnel to consistently have access to timely and accurate indications. For each wind turbine, 60 parameters have been grouped into 4 categories: wind parameters, energy conversion parameters, vibration parameters and temperature parameters and recorded. E.g., turbine status information or signals such as temperatures, currents and pressures can be used to give a sign of the turbine condition. Some papers showed that changes in signal behavior can be detected weeks in advance.

Another example is NASA Prognostics Data Repository which focuses exclusively on prognostic data sets. This data set can be used for the development of prognostic algorithms. Most of them are time series data from some normal state to a failure state. See (www.ti.arc.nasa.gov).

2.2 Models for diagnostic and prognostic

This section tries to summarize recent researches and applications with regard to diagnostic and prognostic models in CBM. It discusses physics-based model, some data-driven models and hybrid models. The common questions for choosing models are:

- What is the underline failure mode/reasons corresponding to the condition characteristics observed?
- How to perform prognostic?

2.2.1 Physics based model

Physics based models are mathematical models to describe the physical phenomena of the system for a certain failure mechanism (e.g., degradation). If built in a correct way, they can greatly improve the accuracy of prognostics (e.g., remaining useful life). However, it is difficult to build authentic physics based models for complex systems. Physics-based model is powerful both in diagnostic and prognostic. In physics-based model approaches, the component and its degradation are represented by a set of mathematical laws. Each physics-based model is unique for its corresponding system. To build physics-based models, sometimes, interruption of operation is needed. For complex system, it is difficult to build physics-based models. Thus, for most industry applications, physics-based models are not commonly used. However, if these models are built in a correct way, it can provide accurate estimation. Moreover, with the improving understanding of degradation of system (e.g., crack propagation and spall growth), the accuracy of physics-based models can be improved. Another merit of physics-based models is that they require less data comparing to data-driven models.

2.2.2 Data-driven models

Data-driven models are more commonly used in CBM. They do not need physical knowledge. they utilizes collected monitoring data to estimate the health state of equipment.

Statistical process control, cluster analysis, expert system and fuzzy logic

Statistical process control and cluster analysis are for diagnostic use. statistical process control is to measure deviation of the current signal from normal condition. If the deviation exceeds the control limits, machine is considered in an abnormal state. Cluster analysis is a statistical classification approach. It groups signals in such a way that signals within same cluster have similar characteristics or features. Signals in different clusters have large differences. Simple cluster analysis can be realized in SPSS, Minitab and Excel. Expert system is a computer system using expert knowledge extracted by human to solve problem. In the area of diagnostic, expert system contains rule-based reasoning, case-based reasoning, model-based reasoning and negative reasoning. In modern application expert system is usually combined with artificial neural network for machine diagnostic to enhance accuracy. Yang et al. (2004) presents an approach combining expert system with an neural network for fault diagnostic. Fuzzy logic theory is developed to measure uncertainty. It is also applied for machine diagnostic. Expert system and fuzzy logic usually integrated with each other or artificial neural network to achieve more accurate diagnostic.

Following paragraphs elaborate Markov model, Artificial Neural Network (ANN) and Proportional Hazard Model (PHM). Unlike aforementioned models, these models can be built for prognostic.

ANN-based methodology and Bayesian Network

Artificial Neural Network (ANN) based methodology have been considered to be a very promising way to assess equipment health condition. The ANN mimics animal's central nervous system. It consists input layer, hidden layer, output layer, and one or several nodes within each layer. The training process is that the ANN learns unknown functions altering its weights to fit the observed input and output. One famous training algorithm is called back propagation. The multi-layer feed forward networks trained using back propagation algorithm is the most widely used method in machinery fault diagnostic. Other neural networks models are "cascade correlation neural network", "recurrent neural network", "counter propagation neural network" and "racial basis function neural network". These neural networks are trained usually by experimental data and simulated data with seeded failure mode. it is called supervised learning. While unsupervised neural networks learn by itself with new coming information. For example, self-organizing map and back propagation neural network, these types of neural network is popularly used for prognostic. Huang et al. (2007) is a good example in which vibration data is used.

For diagnostic, Li et al. (2000) discusses several rolling bearing vibration features in both time and frequency domain, and these features are used for bearing fault diagnostics. Temperature sensors, vertical/horizontal mounted vibration sensors and motor current sensors are combined to diagnose faults in bearing. Simulated data is used to train ANN and the actual data is used to validate the model with result showing around 90% accuracy. The simulation tool used is called MotorSim which is a MATLAB-SIMULINK based program. Rafiee et al. (2007) uses a multi-layer perception neural network (feed forward neutron network) to diagnose gears and bearings faults of a typical gearbox system. All vibration signals are collected using the accelerometer which is mounted on the outer surface of the bearing. Standard deviation of wavelet packet coefficient is the feature extracted to train the ANN. The results show that a multi-layer perception network has close to 100% perfect accuracy to identify gear failures and bearing defects. However, gear failures and bearing defects are not specified in his paper. Samanta and Al-Balushi (2003) presents a procedure for a rolling element bearing fault diagnostic through ANN. Time-domain vibration signals of the rotating machinery with normal and defective bearings have been used as inputs.

For prognostic, Tian et al. (2010) uses failure and suspension data, including age, vibration data, to predict RUL. Output is the percentage of life. The proposed approach is validated by real-world vibration monitoring data collected from pump bearings. Normally, recurrent neural network is applied when dealing with time series prediction. Wang et al. (2004) uses an extended recurrent neural network to predict the health condition of gearbox based on the vibration data. The result shows that the average normalized prediction error is less than 5%, which is more than satisfactory. Mahamad et al. (2010) proposes a feed forward neural network with Levenberg Marquardt of training algorithm to achieve RUL of bearing failures. Root mean square and kurtosis from present and previous points are inputs, and normalized life percentage is the output.

This paragraph briefly introduces Bayesian networks (BNs). BNs is the key computer technology in artificial intelligence to deal with probability. It uses nodes to describe the variable and arcs to connect those variables. Bayesian networks combine prior knowledge and updated information to provide full representations of probability distributions over their variables (e.g., the newly updated wavelet transform combines prior knowledge to show the change of belief on health condition of the rotating machinery). The following four types of software are used when dealing with BNs: BayesiaLab, GeNIe, Hugin and Netica. Within BNs, the most commonly used tool to model the machine health state is Dynamic Bayesian network (DBN). It is a dynamic graphical model to predict future process based on the observed data and underlying correlation as time passes. Medjaher et al. (2012) proposes a Mixture of Gaussians Hidden Markov Models (MoG-HMMs), represented by Dynamic Bayesian Networks (DBNs) to model the bearing degradation and predict RUL.

Remark: ANN and BNs are powerful tools for diagnostic and prognostic used in CBM. Normally methods used in waveform data analysis are: wavelet transform, bispectrum analysis, etc. Feature extracted are: wavelet packet coefficient, standard deviation of wavelet packet coefficient, bispectral patterns, vibration root mean square, kurtosis, etc.

Hidden Markov model

Hidden Markov Model (HMM) is the simplest kind of dynamic (also called “temporal”) Bayesian network, which is particularly useful to estimate the machine health condition for both diagnostic and prognostic (Bunks et al., 2000). HMM is built based on data and extract the specific feature in data to predict the health condition of machine.

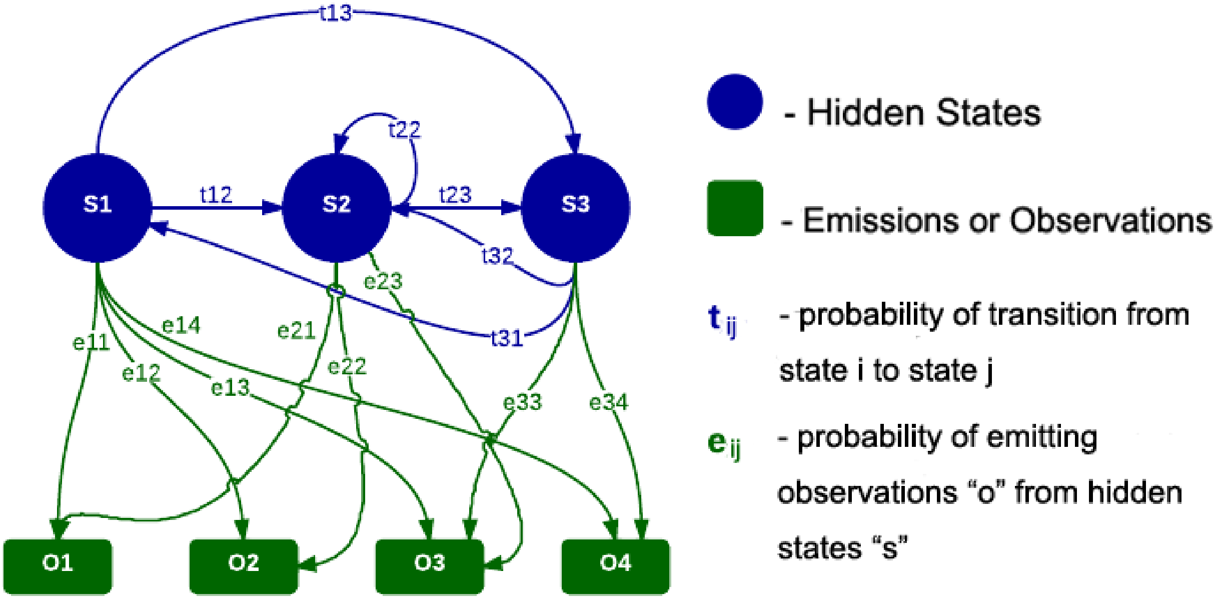

t13
t22
S1
t12
S2
t23
S3
t32
e23
e11
e14
e21
e22
e12
e13
t31
e33
e34
O1
O2
O3
O4
● - Hidden States
🟩 - Emissions or Observations
t_{ij} - probability of transition from state i to state j
e_{ij} - probability of emitting observations “o” from hidden states “s”

Figure 2.1: Transition diagram of a Hidden Markov Model

The Figure 2.1 is an example of Hidden Markov Model (HMM). O is called the sequence of emission which can be observed (e.g., the vibration RMS). However, the sequence of states, which is S_i in Figure 2.1, is usually invisible. That is the reason to use word “hidden”. It means that HMM states are hidden from direct observations. One reason to use HMM in CBM is that the underlying mechanism (fault classes) can be treated as a hidden process (S_i) and can be inferred through system emission (O_i). One goal of analysis of the hidden Markov mode is to seek to recover the sequence of states (S_i) from the observed data (O_i). t_{ij} is the state transition probability, which illustrates the law of transition between states. e_{ij} is the emission probability showing the probability of certain observation under each state. The HMM is trained first and used to predict the sequence of states, Posterior State Probabilities, etc. Another reason to choose HMM in CBM is that there have been many publications on how to estimate the parameters of a HMM, which provide the required theoretical foundation. Moreover, HMM functions can be found in MATLAB statistic toolbox (e.g., "hmmviterbi" is to calculate the most probable state path for a hidden Markov model). These functions can be directly applied in MATLAB, which makes realization of HMM in practice easier.

Bunks et al. (2000) first points out that HMM could be applied in the area of prognostic in machining processes. The idea comes from speech processing and handwriting word recognition which are made based on HMMs. A HMM based on helicopter gearbox data is proposed, in which vibration data under different operation conditions (e.g., torque level) are used. The result shows that HMMs have a strong potential for constructing algorithms for CBM. Li et al. (2005) uses HMM to classify feature vectors, which are obtained by the FFT, wavelet transform, and bi-spectrum, to diagnose faults in rotating machinery. Zhang et al. (2005) presents an integrated fault diagnostic and prognostic approach for bearing in CBM. Firstly the principal signal features are extracted by Principal Component Analysis (PCA). Secondly, HMM are used to simulate a component degradation process. Finally, on-line RUL prediction for a component is realized. Experiments are conducted showing the effectiveness of this scheme. Baruah and Chinnam (2005) presents a HMM using sensor signals to identify the health state of machine and estimate RUL. A test on a vertical drilling machine validates the proposed methods. However, HMM has its inherent limitation: a failure threshold needs to be defined and duration of states is exponentially distributed. Thus, a Hidden Semi Markov Model (HSMM) is introduced to make up for HMM's deficiencies. HSMM is a HMM with temporal structures. HSMM does not follow the Markov chain assumption that the duration of states is exponentially distributed. So it is a more powerful method to deal with real problems. Dong and He (2007) proposes a HSMM which shares a same structure of HMM except the probabilities of being a changes in hidden states depend on the time. A statistical modeling methodology for performing both diagnostic and prognostic is proposed. The target failure position is the bearing in the hydraulic pumps. Oil containing dust is used to define the health states of pump. They are also hidden states. Totally four hidden states are defined. Vibration data is processed by using wavelet packet (i.e., to extract features). Wavelet coefficients is obtained by the wavelet packet decomposition. These inputs are actually “emission” or "observation" in HSMM. They can be used to train and test the model. Two outcomes are: duration of each states and RUL of bearing. The result also shows that HSMM-based diagnostic has a much better performance than HMM-based diagnostic. Peng and Dong (2011) presents an age-dependent HSMM to assess equipment health states. That is to say the probability of making change to a less healthy states will increase with age. Therefore, failure rate is dependent on both equipment age and equipment conditions. Finally, RUL is predicted by combining both hazard rate and health-state transition matrix. For both HMM and HSMM, challenge is to relate actual physical degradation and states changing.

Time-dependent proportional hazard model

Time-dependent proportional hazard model is a popular model used in CBM. The failure rate can be assessed by knowing machine condition variables and current time. Lin et al. (2004) extracts useful condition covariates from raw vibration signals and develop optimal maintenance policies for the gearboxes by using proportional hazard model. Axial vibration data is captured by accelerometer. Wavelet is used to decompose gear motion error signal and to extract the residual error for gear fault detection. ARL algorithm is used for processing vibration signal. A software called EXAKT is used to analyze the data and provides the tool to build the proportional hazard model . Raw data acquired are: changes in load, input shaft rate, amplitude, phase, wavelet and event data (e.g., installation, failure time). The targeted failure mode is gear tooth failure, and the corresponding failure mode is shaft failure. After data processing, covariates obtained from raw data are: gear motion error signal , residual error signal, fault growth parameter(FGP), revised fault growth parameter (FGP1), etc. These covariates can be derived from raw data by using algorithms suggested. The Weibull proportional hazard model is used in this case.

The time-dependent proportional hazard model has a failure rate form:

$$h(t) = \frac{\beta}{\eta}(\frac{t}{\eta})^{\beta-1} \exp(\gamma_1 Z_1(t) + ...\gamma_p Z_1(t)) \tag{2.1}$$

where η is the scale parameter. β is the shape parameter. $Z_1(t)$,..., $Z_1(t)$ are covariates at time t, representing the extracted features from condition monitoring data. γ_1, ..., γ_p are the weights (coefficients). Time-dependent proportional hazard model depends on gearbox geometry, i.e.,η, β, $\gamma_1, \ldots, \gamma_p$ will change if the mechanical configuration of gearbox is changed. In his paper, only one covariate is used. Maintenance decision is based on the average costs per unit time. A threshold level, where the "minimal cost " can be obtained, is defined. If the hazard rate is exceeding this limit, gearbox needs to be replaced immediately. Otherwise, no maintenance activity is needed. Sun et al. (2006) proposes proportional covariate model to deal the situation where failure data are sparse or zero. Hazard rate is estimated by using accelerated life tests and condition monitoring data. Lin et al. (2006) firstly applies principal component analysis to reduce the number of covariates (measurements), and then uses time-dependent proportional hazards model for CBM optimization.

2.2.3 Hybrid models

Hybrid models combine two or more models together to get more accurate outcome. Loutas et al. (2011) presents a model combined with vibration, acoustic emission and oil debris to monitor the progressive wear in gears. Health gears are tested until they are seriously damaged in the lab. On-line monitoring for vibration, acoustic emission and oil debris monitoring (ODM) are performed. Features are extracted from the acquired waveforms and data fusion is accomplished. i.e., to choose the most representative extracted features. Principal component analysis (PCA) and independent component analysis (ICA) is applied. At the end, health monitoring scheme for gearboxes is proposed. The result showed that the combination of vibration, acoustic emission and oil debris monitoring data gives a more reliable outcome for the machine health indication.

2.3 CBM main steps

CBM main steps are as follows. Raw data acquisition → data processing (extracting features and diagnostic) → diagnostic (patten recognition) → prognostic→ maintenance optimization. Figure 2.2 shows CBM steps. The figure is made based on a large quantity of the literature survey and should cover almost all types of models in the literature. Chapter 4 elaborates many promising diagnostic and prognostic models from Figure 2.2. In addition, for promising models, many representative papers are cited in this Master's thesis.

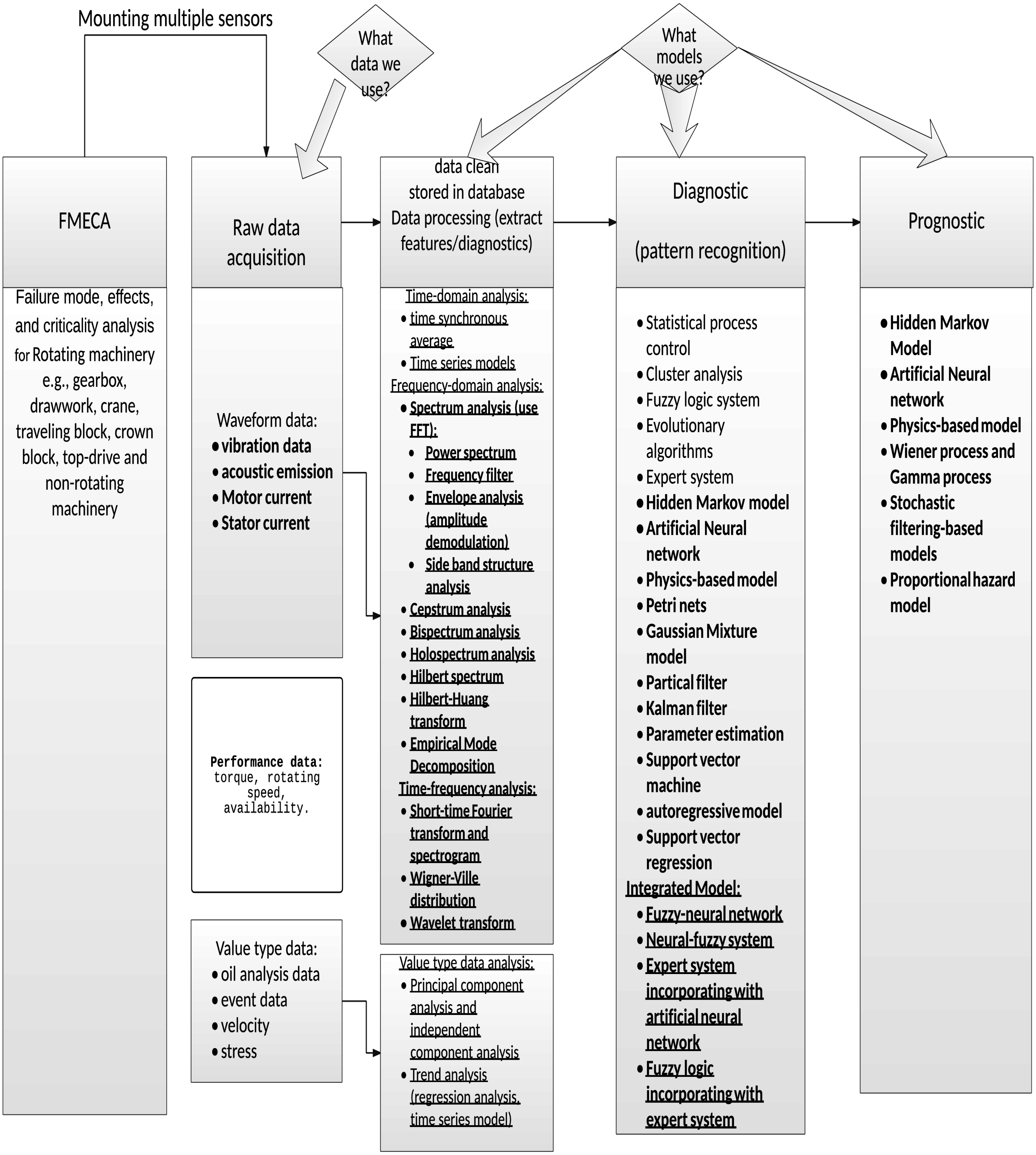

Mounting multiple sensors
What data we use?
What models we use?
FMECA
Failure mode, effects, and criticality analysis for Rotating machinery e.g., gearbox, drawwork, crane, traveling block, crown block, top-drive and non-rotating machinery
Raw data acquisition
Waveform data:
• vibration data
• acoustic emission
• Motor current
• Stator current
Performance data: torque, rotating speed, availability.
Value type data:
• oil analysis data
• event data
• velocity
• stress
data clean
stored in database
Data processing (extract features/diagnostics)
Time-domain analysis:
• time synchronous average
• Time series models
Frequency-domain analysis:
• Spectrum analysis (use FFT):
• Power spectrum
• Frequency filter
• Envelope analysis (amplitude demodulation)
• Side band structure analysis
• Cepstrum analysis
• Bispectrum analysis
• Holospectrum analysis
• Hilbert spectrum
• Hilbert-Huang transform
• Empirical Mode Decomposition
Time-frequency analysis:
• Short-time Fourier transform and spectrogram
• Wigner-Ville distribution
• Wavelet transform
Value type data analysis:
• Principal component analysis and independent component analysis
• Trend analysis (regression analysis, time series model)
Diagnostic
(pattern recognition)
• Statistical process control
• Cluster analysis
• Fuzzy logic system
• Evolutionary algorithms
• Expert system
• Hidden Markov model
• Artificial Neural network
• Physics-based model
• Petri nets
• Gaussian Mixture model
• Partical filter
• Kalman filter
• Parameter estimation
• Support vector machine
• autoregressive model
• Support vector regression
Integrated Model:
• Fuzzy-neural network
• Neural-fuzzy system
• Expert system incorporating with artificial neural network
• Fuzzy logic incorporating with expert system
Prognostic
• Hidden Markov Model
• Artificial Neural network
• Physics-based model
• Wiener process and Gamma process
• Stochastic filtering-based models
• Proportional hazard model

Figure 2.2: CBM main steps

2.4 Prognostic and health management

To find more accurate prognostic models, a recently developed concept called Prognostic and Health Management (PHM) is described here. CBM and PHM have many similarities. The type of sensors and prognostic models used in PHM are also used in CBM. Moreover, more and more research efforts have moved to PHM.

Very similar to CBM, PHM uses machine condition information to allow early detection of impending or incipient faults, assess the health states of machine, and to predict remaining useful life (Heng et al., 2009). Due to their similarity, the scarcity of CBM applications in industry can be replenished by less scarce PHM applications. Recently, PHM has a rapid development. It is generally accepted that the PHM can use machine failure mechanism, current machine conditions to achieve a good result of faults diagnostic and prognostic. Moreover, CBM can be regarded as a supporting implementation technology to PHM. Some similar areas of research, sharing the same disciplines as PHM, are health management (SHM), integrated vehicle health management (IVHM) or engine health management (EHM). Each of these areas of research has their own specific focusing. However, the prognostic models built behind these areas of research, which are the core concept, are the same to CBM models (i.e., physics-based models, data-driven models and hybrid models). By knowing this can help us to find more literature with respect to prognostic models.

2.5 CBM industry applications

The US Department of Defense's Joint Strike Fighter program is leading the development of next generation strike weapon systems for the US Navy, Air Force, and Marine Corps. Here, the vehicle is being designed with a fully functional PHM system. Smart sensors are installed on airframe structures, engines and electronics. PHM is also applied in modern automotive (Toyota: car assembly line) and wind turbine.

NASA will develop Integrated Vehicle Health Management (IVHM) systems for the Second Generation Reusable Launch Vehicle, crew, and cargo transfer vehicles. Advanced smart sensors, diagnostic and prognostics software for sensors will be embedded in the vehicle and provide both real-time and life-cycle vehicle health information. NASA also applies PHM for faulted aerospace components. More details can be found at The Prognostic Center of Excellence (PCoE).

GE aviation developed an IVHM system. On board sensors collect aircraft health information. This information are sent and analyzed by GE aviation maintenance office. Advanced prognostic is made and correct maintenance task is performed. Finally this system increases aircraft up time and reduces maintenance cost.

Honeywell has developed some condition-based maintenance portfolios for helicopters. These portfolios include on-board or carry-on hardware, sensors, cabling, software and data management service. These sensor-driven data and analysis gathered directly from the aircraft, providing information about the engine, gearbox, shafts, fans and rotor system. A report from Honeywell shows that by applying this “Health and Usage Monitoring System”, army combat aviation brigade CBM fleet saved 101million over 22 months, avoided 2957 maintenance man-hours for 1 year, and reduced 66% mission aborts due to vibration over 5 years.

Rolls-Royce applied an improved oil-debris monitoring to monitor the Rolls-Royce Trent XWB engine, which is exclusively for the Airbus A350 XWB. This system is based on quantitative debris monitoring, which is a proven, mature technology. It offers a significant progress in fault diagnostic and prognostic.

2.6 Some condition monitoring systems

NOV has developed a condition monitoring system called eHawk whose support centers are found in Houston, Kristiansand, and Stavanger. It is a real time on-line condition monitoring system (i.e., 24/7/365 days a year). It utilizes different types of data including vibration data.

EHawk system provides faster issue response time, reduced service personnel visits to the field and enables maintenance center to remotely handle issues at a reduced cost.

GE has developed an acoustic leak detection system and a condition monitoring system called "the subsea multi-domain condition monitoring system". These two condition monitoring use acoustic emission data and electronic emission data and allow operators to monitor the integrity of their subsea installation. The acoustic leak detection system uses acoustic hydrophone technology to detect and locate subsea oil and gas leaks. The system can detect both crude oil and gas with coverage up to 500 meters.

FMC has developed a on-line Condition and Performance Monitoring system supported by an historical database. This system enables early detection and diagnostic,and makes proactive and condition-based maintenance possible (i.e., Recognizing and understanding equipment condition based on performance). In 2012, FMC project team successfully implemented the world's first subsea CPM system at the Gjøa field.

Cameron announced a subsea BOP condition monitoring system in 2014. This system synthesize both real-time data and historical data to provide diagnostic.

National Instruments has also developed many condition monitoring systems which use vibration data, lube oil data, strain, acoustic data and temperature data.

Chapter 3

A self-made Matlab toolbox to model degradation by simulating three types of stochastic processes

This chapter elaborates simulations for degradation models by using the Markov model, Brownian motion (the Wiener process) and the Gamma process. The theory, the Matlab code with detailed explanations and the logic diagrams are presented. For a better illustration, a Matlab toolbox is developed. Users can fill in the value for each parameter and get a RUL (Remaining Useful Life) with a standard deviation, a probability to survive till the next maintenance interval and the PFD (Probability of Failure on Demand). The results are calculated based on the current degradation level (i.e., condition monitoring under each inspection), inherent mathematical laws and the number of simulations. The code files for this Matlab toolbox can be download from "https://www.dropbox.com/sh/lxwnvfj5inq44wc/AAAJqbJ0sCpz8d5NN-wDgzyJa?dl=0" and are also attached in the Appendix B. The code for simulations of the Markov model, Brownian motion and the Gamma process can be found in this chapter. This toolbox is also used often in Chapter 4 when those stochastic models are introduced, showing the toolbox can help to make maintenance decisions.

3.1 The continuous-time Markov model

In the probability theory, the Markov model is a stochastic model that assumes the Markov property (Ross, 2014). That is to say the future states of the process depends only upon the present state and does not depend on the history. This chapter discusses the continuous-time Markov model whose transition rates are defined as "frequency", e.g., 0.005 per year. The previous Master project has discussed the discrete-time Markov models in which transition rates are defined as "probability", e.g., 0.02. However, The former one has much wider applications. For simplicity, The Markov model in this chapter actually refers to the continuous-time Markov model.

For degradation models, conventional reliability models like FTA (Fault Tree Analysis) is insufficient. Because FTA is mainly a static model which is used to find the top event probability. It can not model dynamic processes like the degradation propagation. Unlike FTA, the Markov model is a strong tool to deal with dynamic problems and can model both the maintenance strategy and the degradation propagation. Thus it is reasonable to use the Markov model for CBM. However, it has some limitations. Firstly, all possible states and the transition rates need to be defined. For a complex system, it is very likely to omit some states. Secondly, transition rates are difficult to determine due to various reasons, e.g., lack of the data. Thirdly, the sojourn time has to be exponentially distributed. Ross (2014) is a good textbook for the theory of the Markov model. Beyond that, Rausand and Høyland (2004) illustrates methods to the find steady state probabilities, MTTF, PFD, PFH by using the Markov model. Readers can find these methods elaborately illustrated in the textbooks with all necessary equations, e.g., Chapman-Kolmogorov Equation, Kolmogorov's back/forward equations. There is no necessity to make more repetitions here.

When it comes to the simulation. There is a merit. It breaks the limitation of the Markov model that the sojourn time has to be exponentially distributed. The randomly simulated sojourn time can be chosen as any form of distributed value, e.g., normal distribution, Weibull distributed. One can get numerical solutions instead of analytical solutions. The main ideas behind the simulation are following. For a certain state, there are several transition rates with different values.

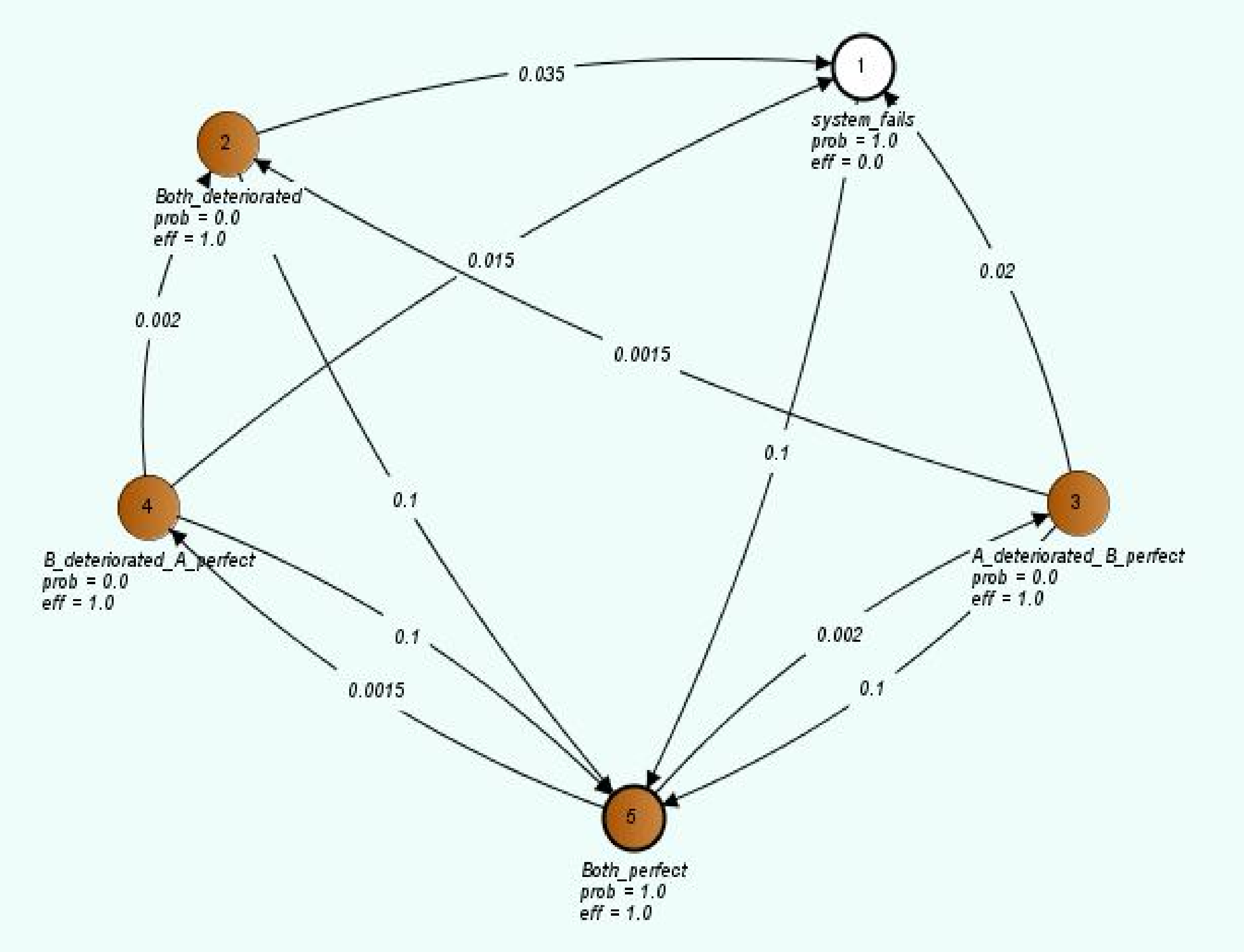


Figure 3.1: A 2oo2 system with three degradation levels

For each transition rate, Matlab generates a random number with λ = the transition rate. As mentioned before, this number can be simulated by any forms of distribution. Take exponential distribution as an example, the virtual sojourn time for each transition rate is simulated by the Matlab code "exprnd(λ)". The "real" sojourn time is chosen as the minimum value of these simulated "virtual" sojourn time. For example, Figure 3.1 is a Markov diagram for a 2oo2 system. Each component has three degradation level, i.e., perfect, deteriorated and failure. State 5 means both are perfect, while state 4 and state 3 mean that only component B is deteriorated and only component A is deteriorated, respectively. State 2 means both are deteriorated but system is still functioning. Since system will not fail until one of the components fail. State 1 refers to at least one component fails and is a "unavailable state". The "failure rates" for component A among perfect state, deteriorated state and failure state are 0.002 per hour and 0.02 per hour.

For component B, they are 0.0015 per hour and 0.015 per hour. The assumption for the maintenance strategy is that this system is inspected periodically. During each condition monitoring, a maintenance task is executed immediately if the system enters state 4,3,2 and 1. The repair rate is 0.1 per hour regardless of the degradation level and the number of the deteriorated components. The transition rates between states are shown in Figure 3.1. There are two transition rates from the state 2, i.e., $\lambda_{25} = 0.1$ and $\lambda_{21} = 0.035$. Two exponentially distributed random numbers are generated by Matlab as, e.g., 9.3 by λ_{25} and 34.3 by λ_{21}. The minimum value is 9.3. Thus the next step which the process goes to is state 5 and the sojourn time is 9.3. This value is recorded and it is called "one step". Each simulation runs 25000 steps and the results, e.g., the RUL with standard deviation, the steady state probabilities and PFD are estimated based on the above principles. Figure 3.2 is a logic diagram for Markov chain simulations. As shown in red, the input contains the transition matrix, the initial state, the number of simulations and the maintenance interval. As shown in green, there are two types of output. It depends on whether the transition matrix has absorbing states or not.

The Matlab code for the Markov chain simulations with detailed explanations is below.

```
%This script is used to simulate a countinuous time Markov chain. It finds
%steady-state probabilities or RUL and PFD.
%The following variables can be changed to suit the different situation.
ll = [-0.001 0.001 0; 0 -0.1 0.1; 0.1 0 -0.1];
% Write your transition matrix above.
lambdas=1./ll;
numStates = length(lambdas); % Here number of states are specified.
numSimulations=20000; % Choose the number of simulation
numSteps = 25000;% Here the chain length is specified.
States = zeros(1,numSteps);
%States(1) = randi(numStates)%;if the initial state is unknow
States(1)=1; % if the initial state is 1
tao=1000; %Define your maintenance interval here to find PFD

% Changing the following variables are not recommended.
warn=ones(1,numStates)*inf;
% If an absorbing state exists, this "warn" vector helps to find RUL
MTTF_vector=zeros(1,numSimulations);
```

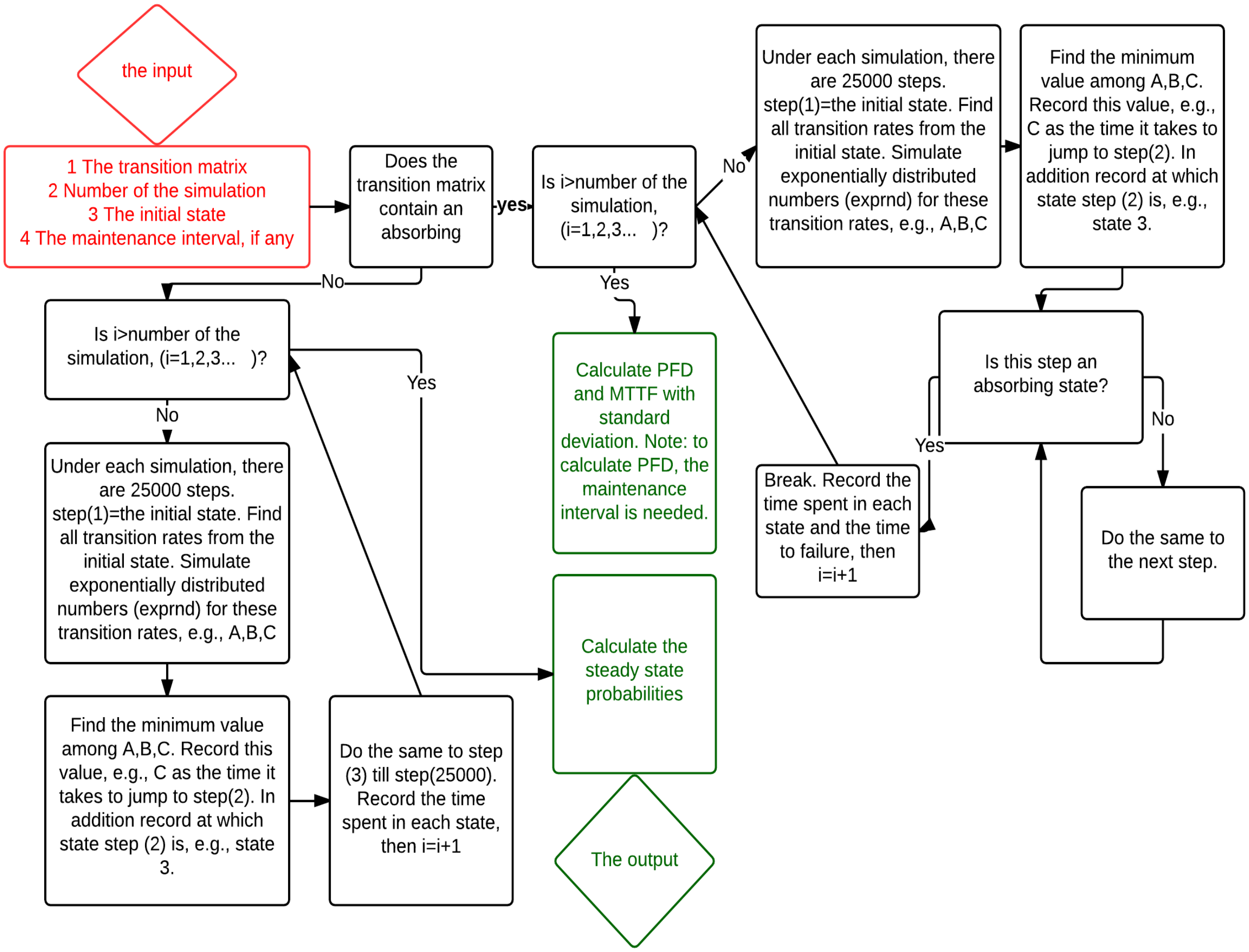

the input
1 The transition matrix
2 Number of the simulation
3 The initial state
4 The maintenance interval, if any
Does the transition matrix contain an absorbing
yes
No
Is i>number of the simulation, (i=1,2,3...)?
No
Yes
Under each simulation, there are 25000 steps. step(1)=the initial state. Find all transition rates from the initial state. Simulate exponentially distributed numbers (exprnd) for these transition rates, e.g., A,B,C
Find the minimum value among A,B,C. Record this value, e.g., C as the time it takes to jump to step(2). In addition record at which state step (2) is, e.g., state 3.
Do the same to step (3) till step(25000). Record the time spent in each state, then i=i+1
Is i>number of the simulation, (i=1,2,3...)?
No
Yes
Calculate PFD and MTTF with standard deviation. Note: to calculate PFD, the maintenance interval is needed.
Calculate the steady state probabilities
The output
Under each simulation, there are 25000 steps. step(1)=the initial state. Find all transition rates from the initial state. Simulate exponentially distributed numbers (exprnd) for these transition rates, e.g., A,B,C
Find the minimum value among A,B,C. Record this value, e.g., C as the time it takes to jump to step(2). In addition record at which state step (2) is, e.g., state 3.
Is this step an absorbing state?
Yes
No
Break. Record the time spent in each state and the time to failure, then i=i+1
Do the same to the next step.

Figure 3.2: Logic diagram of Markov model simulations

```
Time = zeros(1,numSteps); %To record time spent in each simulation
 sojourn_time= zeros(1,numStates);% Find "virtual" sojourn time
 rate = zeros(1,numStates);% Find transition rate for each state.
Count = zeros(1,numStates);
 absorb=sum(ismember(lambdas,warn,'rows')); %This value is to check if there
% is one or more obsorbing states. "obsorb=0" means that there is no
% absorbing state. Otherwise, it means that there is at least one
% absorbing state.

%Case one: no obsorbing states. In this case, we will get the steady state
% probabilities since there is no obsorbing state.
 if absorb==0
     for i = 1:numSteps-1
```

```
    rate = lambdas(States(i),:);
    for n=1:numStates
    sojourn_time(n) = exprnd(rate(n)); % MAIN PRINCIPLE, to generate a
    % random exp distributed number
    end
    m = min(sojourn_time(sojourn_time>0)); % Choose the minimum
    % From state i to i, the transition rate is
    % negative. in such case, function "exprnd" will not give the value.
    % Thus it will not influence the "min" function.
    index = find(sojourn_time == m);% find the next state
    States(i+1) = index;% find what is the next step of process
    Count(States(i)) = Count(States(i))+m; %To sum each sojourn time
    % for the each state
    Time(i+1) = Time(i) + m; % To record the time
    end
plot(Time,States,'x');
prop = Count/Time(end); % To find the steady state probabilities
disp(['The steady state probabilities are' num2str(prop)]);

% Case two: there is at least one obsorbing state. In this case, we will
% get RUL with standard deviation and PFD
else
for j=1:numSimulations
    for i = 1:numSteps-1
    rate = lambdas(States(i),:);
    for n=1:numStates
    sojourn_time(n) = exprnd(rate(n));
    end
      if isequal(sojourn_time,warn)==0
      m = min(sojourn_time(sojourn_time>0));
      % From state i to i, the transition rate is
      % negative. in such case, function "exprnd" will not give the value.
      % Thus it will not influence the "min" function.
      index = find(sojourn_time == m);% To find the next state
      States(i+1) = index;
      Count(States(i)) = Count(States(i))+m; %To sum each sojourn time
```

```
        % for the each state
        Time(i+1) = Time(i) + m; % To record the time
        else
        MTTF_vector(j)=Time(i);
        break;
        end
    end
end
MTTF=sum(MTTF_vector)/numSimulations;
failuretime=(tao-MTTF_vector)./tao;
failuretime(find(failuretime<0))=0;
pfd=sum(failuretime)/numSimulations;
disp('An absorbing state is found!');
disp(['MTTF is ' num2str(MTTF) 'PFD is ' num2str(pfd)]);
end
```

Take a 2oo2 system for an example. As shown in Figure 3.1, each component has three degradation level, i.e., perfect, deteriorated and failure. The state 5 means both are perfect, while state 4, 3 and 2 mean that the system is still functioning with deteriorated performance. The state 1 is a failure state. By using the simulation, the steady state probabilities are found to be $0.0053319, 0.00037, 0.016018, 0.012522, 0.96576$ for the state 1,2,3,4,5, respectively. If the transition rate, i.e., the repair rate, from 1 to 5 is removed, and maintenance interval is set as 1000, the simulation gives MTTF=1913(hours) and PFD=0.215.

3.2 Brownian motion (The Wiener process) with drift

Let us consider to simulate a Markov chain with infinite states with $P_{i,i+1} = P_{i,i-1} = \frac{1}{2}, i = 0, \pm1, \pm2, \ldots$. It is actually a symmetric random walk with taking smaller and smaller time intervals. This process is called Brownian motion. The Brownian motion process, sometimes called the Wiener process, is one of the most useful stochastic processes in applied probability theory (Ross, 2014). Ross (2014) also gives a definition of Brownian motion:

A stochastic process $X(t), t \geq 0$ is said to be a Brownian motion process if

- $X(0) = 0$;

- $X(t), t \geq 0$ has stationary and independent increments;
- for every $t > 0, X(t)$ is normally distributed with mean 0 and variance $\sigma^2 t$.

Note that Brownian motion without introducing "Drift" value is not suitable to model degradation propagation. It has two main reasons. First, under a certain period of time, the degradation level is either unchanged or increased. It shall never decrease, i.e., the trend is monotonically increasing. However, Brownian motion is non-monotonic. In addition Brownian motion is normally distributed with mean 0. It means that after a period of time, the expectation remains unchanged and the process path has neither an increasing trend nor an decreasing trend. Second, the process does not necessarily start from $X(0) = 0$. The value of $X(0)$ is only based on the current degradation level. Thus to fit the degradation model, two changes are made.

- $X(0) =$ current degradation level;
- for every $t > 0, X(t)$ is normally distributed with mean μt and variance $\sigma^2 t$.

where μ is called the drift coefficient and σ^2 is variance parameter. Ross (2014) gives another equivalent definition

$$X(t) = \sigma B(t) + \mu t$$

where $B(t)$ is standard Brownian motion whose density function is

$$f(x) = \frac{1}{\sqrt{2\pi t}} \exp(-x^2/2t)$$

Even though Brownian motion with drift is still non-monotonic, it can have a monotonic trend over a period of time. For any positive value μ, the path will fluctuate around μt. These fluctuations can be considered as noise. The expectation after time t is μt. See Figure 3.4. By introducing the drift value μ, Brownian motion can model degradation propagation. In this Master thesis, "Brownian motion with drift" is the only type of Brownian motion introduced. Another type of Brownian motion is "geometric Brownian motion" which is also suitable to model the degradation propagation.

The main ideas behind this simulation are following. Firstly a small time interval as a time unit t is defined, e.g., one second. An increment after this time unit t is normally distributed with mean μt and variance $\sigma^2 t$ where μt and variance σ^2 are determined by the expert judgement and the previous data. It is like the methods used to determine the failure rates. Secondly, these increments are recorded as the degradation level at each time. Under each simulation, the process runs until the time meets a predefined limit, e.g.,a year. Finally, the results, e.g., the RUL and the probability to survive till the next maintenance interval can be estimated after a large number of simulations. Figure 3.3 is a logic diagram for Brownian motion simulations. As shown in red, the input contains a drift value, a sigma, the number of simulations, a current degradation level, a failure level and the maintenance interval. As shown in green, the output is RUL with standard deviation and the probability to survive till the next maintenance interval. The Matlab code for Brownian motion with detailed explanations is below.

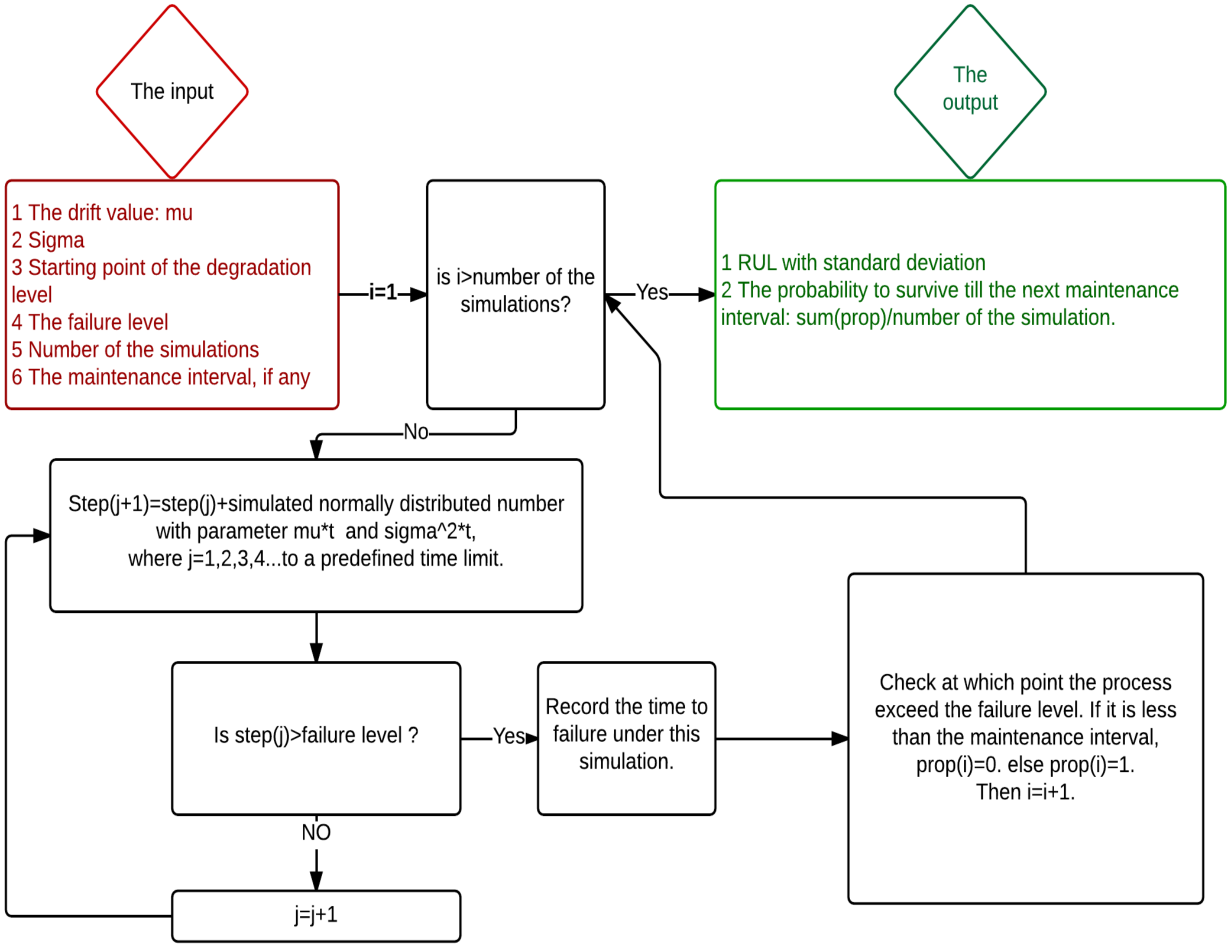

The input
The output
1 The drift value: mu
2 Sigma
3 Starting point of the degradation level
4 The failure level
5 Number of the simulations
6 The maintenance interval, if any
i=1
is i>number of the simulations?
Yes
1 RUL with standard deviation
2 The probability to survive till the next maintenance interval: sum(prop)/number of the simulation.
No
Step(j+1)=step(j)+simulated normally distributed number with parameter mu*t and sigma^2*t, where j=1,2,3,4...to a predefined time limit.
Is step(j)>failure level ?
Yes
Record the time to failure under this simulation.
Check at which point the process exceed the failure level. If it is less than the maintenance interval, prop(i)=0. else prop(i)=1. Then i=i+1.
NO
j=j+1

Figure 3.3: Logic diagram of Brownian motion simulations

```
% Simulate a Brownian Motion process with Drift. To estimate the RUL and the
% probability to survive till the next maintenance interval.
mu=0.2; % Define your degradation paramters here.
sigma=1; % These two values are defined by the previous data and the expert judgement.
t=1; % Define the time unit.
NumSteps=1000; % Define the number of steps under each simulation.
Time=1:NumSteps;
D=zeros(1,NumSteps); % Before a simulation, the degradation level is set to be 0
D(1)=55;% Write your starting point of the degradation level under an inspection.
Fail_level=69; % Write your failure level above which the system fails.
NumSimulations=25; % Write the number of simulations.
MTTF=NumSteps*ones(1,NumSimulations); %Default MTTF
maintenance_interval=37; % Write your maintenance interval.It needs to be integer
prop=zeros(1,NumSimulations);% Default probabilities of surviving till the next
% maintenance interval

for j=1:NumSimulations
for i=1:NumSteps-1
   D(i+1)=D(i)+normrnd(mu*t,sqrt(t)*sigma);% MAIN PRINCIPLE, to generate a
    % random normally distributed number
end
if D(NumSteps)>Fail_level % if the degradation level exceed the failure level
                          % after each simulation
MTTF(j)=find(D>Fail_level,1,'first');
% or use: MTTF(j)=find(D==min(D(D>=Fail_level))); but a bit slow.
% disp(['It fails in ' num2str(MTTF) 'seconds']);
else
% disp(['It will not fail in ' num2str(NumSteps*t) 'seconds' ]);
end
if D(maintenance_interval)>Fail_level
    % If the degradation level exceed the failure level at the next
    % maintenance interval
    prop(j)=0; % The probability to survive is set to be 0.
else
    prop(j)=1; % Otherwise, it is 1.
end
```

```
plot(Time,D);
hold on;
end
prop_to_survicetilnextmaintenanceinterval=sum(prop)/NumSimulations;
RUL=sum(MTTF)/NumSimulations;
variance=(sum((MTTF-RUL).^2))/NumSimulations;
standard_deviation=sqrt(variance);
disp([RUL standard_deviation prop_to_survicetilnextmaintenanceinterval]);
```

For example, $\mu = 0.2$, $\sigma = 1$, the current degradation level under an inspection is 55, the failure level is 69 and the maintenance interval is 37. The simulation gives the result of the RUL, the standard deviation of RUL and the probability to survive till the next maintenance interval, $74.5320, 41.8636, 0.8680$ respectively. Figure 3.4 is an example of 25 simulations under aforementioned criteria.

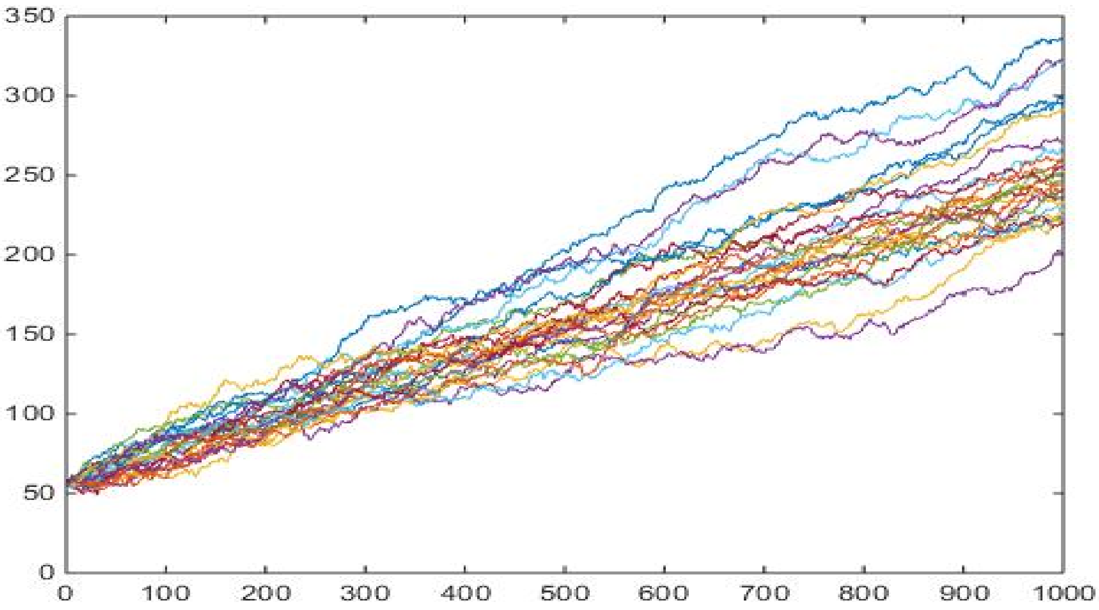


Figure 3.4: Degradation modelled by Brownian motion

3.3 The Gamma process

Section 3.2 discusses Brownian motion where the increment after time t is normally distributed. If this increment after time t is Gamma distributed, the process is called the Gamma process. That is to say $X(t) \equiv \text{Gamma}(k,\theta)$ where k is the shape parameter and θ is the scale parameter. The probability density function of $X(t)$ using the shape-scale parametrization is

$$X(t) = \frac{x^{k-1}\exp\frac{-x}{\theta}}{\theta^k\Gamma(k)} \quad \text{for} \quad k,\theta > 0$$

Here $\Gamma(k)$ is the gamma function evaluated at k.

The main idea behind the simulation of the Gamma process is similar to the one behind the Brownian motion process. Firstly a small time interval as a time unit t is defined, e.g., one second. An increment after this time unit t is Gamma distributed with shape parameter A and scale parameter B. This random number can be generated by the function "gamrnd(A,B)" in Matlab. The shape parameter A and the scale parameter B are determined by the expert judgement and the previous data sets. Secondly, these increments are recorded as the degradation level at each time. Unlike Brownian motion, the Gamma process always has a non-negative value of the increment. Finally, the results, e.g., the RUL and the probability to survive till the next maintenance interval can be estimated after a large number of simulations. Figure 3.5 is a logic diagram for Gamma processes simulations. As shown in red, the input contains a shape parameter, a scale parameter, the number of simulations, a current degradation level, a failure level and the maintenance interval. As shown in green, the output is RUL with standard deviation and the probability to survive till the next maintenance interval.

The Matlab code for The Gamma process simulations with detailed explanations is below.

```
% gamma process simulation
  t=1;% Define a time unit
  A=1.1; B=10; % Define two parameters
  numSteps=20;
  y=zeros(1,numSteps);
  y(1)=55; % Write your starting point of the degradation level under an inspection.
  time=1:numSteps;
```

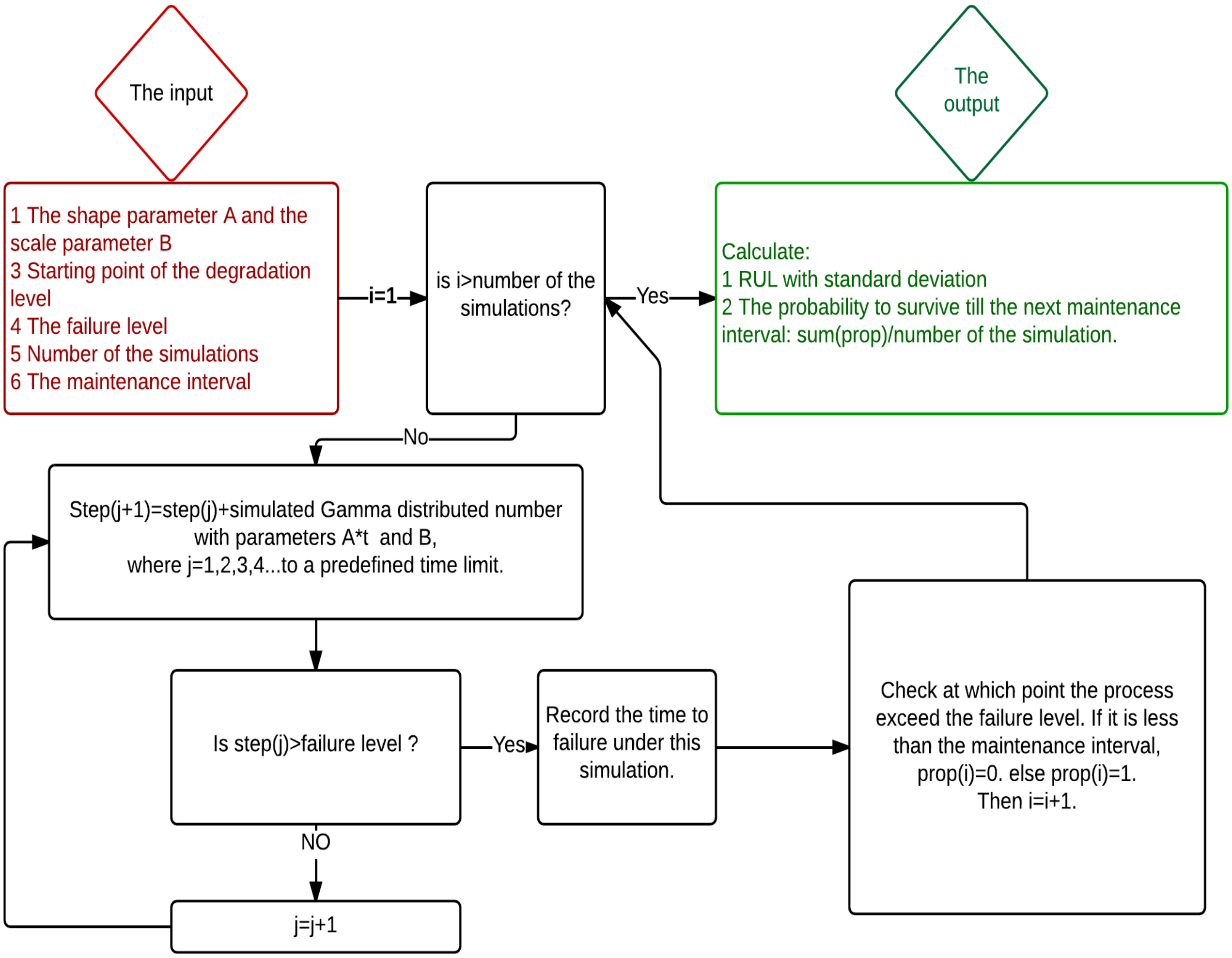

The input
1 The shape parameter A and the scale parameter B
3 Starting point of the degradation level
4 The failure level
5 Number of the simulations
6 The maintenance interval
i=1
is i>number of the simulations?
Yes
The output
Calculate:
1 RUL with standard deviation
2 The probability to survive till the next maintenance interval: sum(prop)/number of the simulation.
No
Step(j+1)=step(j)+simulated Gamma distributed number with parameters A*t and B,
where j=1,2,3,4...to a predefined time limit.
Is step(j)>failure level ?
Yes
Record the time to failure under this simulation.
Check at which point the process exceed the failure level. If it is less than the maintenance interval, prop(i)=0. else prop(i)=1.
Then i=i+1.
NO
j=j+1

Figure 3.5: Logic diagram of Gamma processes simulations

```
 numSimulation=150; % Write the number of simulations.
 Failure_level=270; % Write your failure level above which the system fails.
 maintenance_in=20;  % Write your maintenance interval.It needs to be integer
MTTF=ones(1,numSimulation).*numSteps*t; % Default MTTF
 prop=ones(1,numSimulation); % Default probabilities of surviving till the next
% maintenance interval
 for j=1:numSimulation

 for i=1:numSteps-1
     y(i+1)=y(i)+gamrnd(A*t,B);
   % MAIN PRINCIPLE, to generate a random Gamma distributed number
 end
 if y(numSteps)>Failure_level
```

```
first_time=find(y>Failure_level,1,'first');
 MTTF(j)=first_time*t;
if first_time*t < maintenance_in
     prop(j)=0;
else
end
else
end
plot(time,y);
hold on;
end
Mttf=sum(MTTF)/numSimulation;
variance=sum((MTTF-Mttf).^2)/numSimulation;
sd=sqrt(variance);
survive=sum(prop)/numSimulation;
disp([Mttf sd survive]);
```

For example, A = 1.1, B = 10, the current degradation level under the inspection is 55, the failure level is 270 and the maintenance interval is 20. The simulation gives the result of the RUL, the standard deviation of RUL and the probability to survive till the next maintenance interval $19.0267, 1.861, 0.6933$ respectively. Figure 3.6 is an example of 50 simulations under aforementioned criteria.

3.4 The toolbox instruction

The toolbox combines aforementioned three degradation models. In the pop-up menu located in the upper left corner, users can choose the type of the degradation model. To use Markov simulation, first type in all parameters. See Figure 3.7 Secondly, click the "simulation button". Finally all results are shown on the same dashboard. Figure 3.7 is an example of using this self-made toolbox to simulate Markov chains by inserting the same parameter in Section 3.1.

If the repair rate from the failure state is removed, there will be an absorbing state in the Markov diagram. The corresponding row in the transition matrix is replaced with all elements 0. In such case, the user needs to fill in the value of maintenance interval in the up right corner to estimate

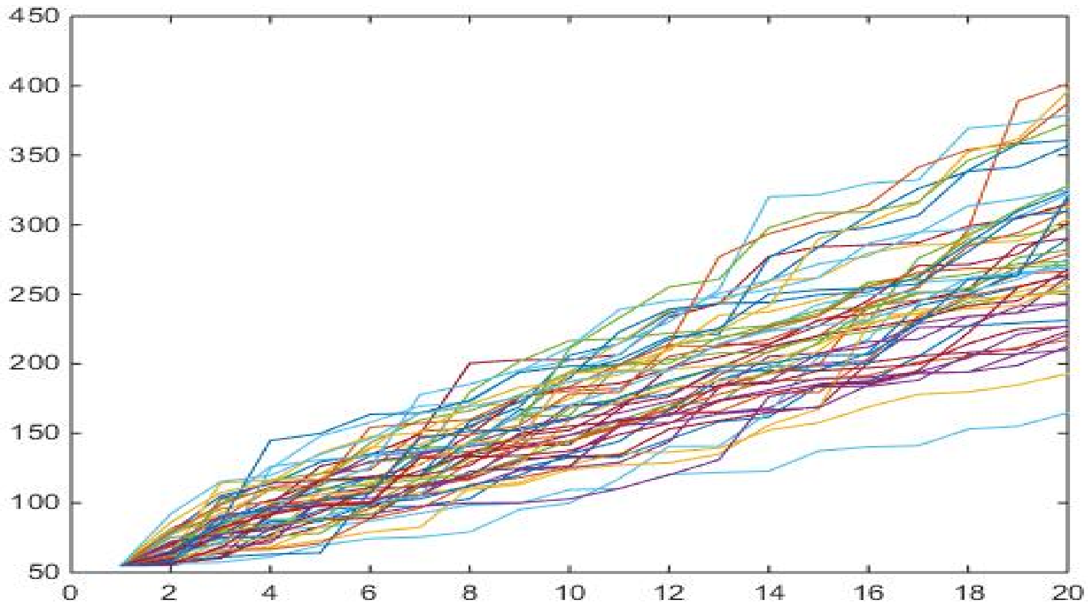
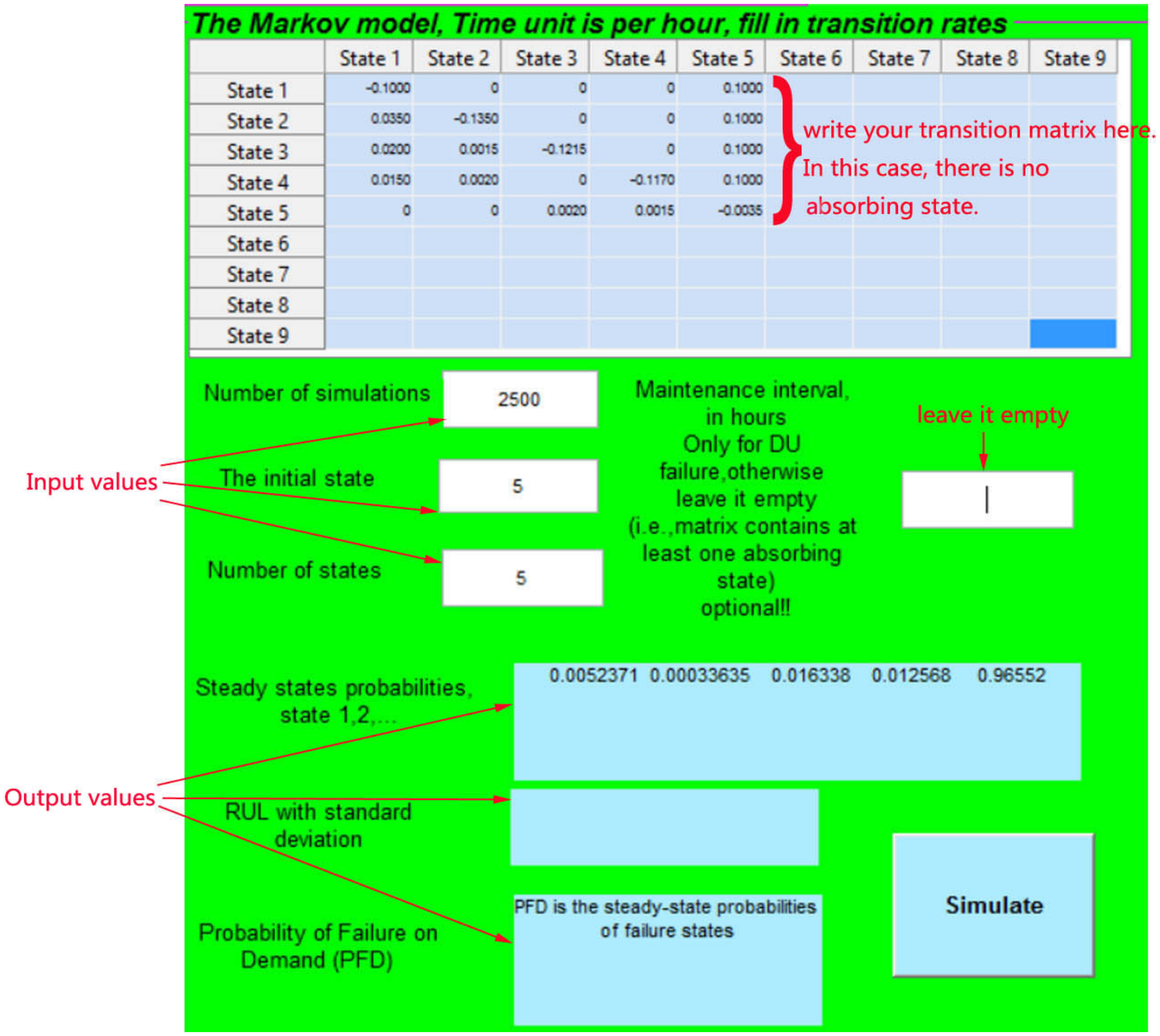


Figure 3.6: Degradation modelled by Gamma processes

the PFD. The RUL with standard deviation is also estimated. See Figure 3.8.



The Markov model, Time unit is per hour, fill in transition rates

	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9
State 1	-0.1000	0	0	0	0.1000				
State 2	0.0350	-0.1350	0	0	0.1000				
State 3	0.0200	0.0015	-0.1215	0	0.1000				
State 4	0.0150	0.0020	0	-0.1170	0.1000				
State 5	0	0	0.0020	0.0015	-0.0035				
State 6									
State 7									
State 8									
State 9									

write your transition matrix here. In this case, there is no absorbing state.

Number of simulations: 2500

The initial state: 5

Number of states: 5

Input values

Maintenance interval, in hours Only for DU failure,otherwise leave it empty (i.e.,matrix contains at least one absorbing state) optional!!

leave it empty

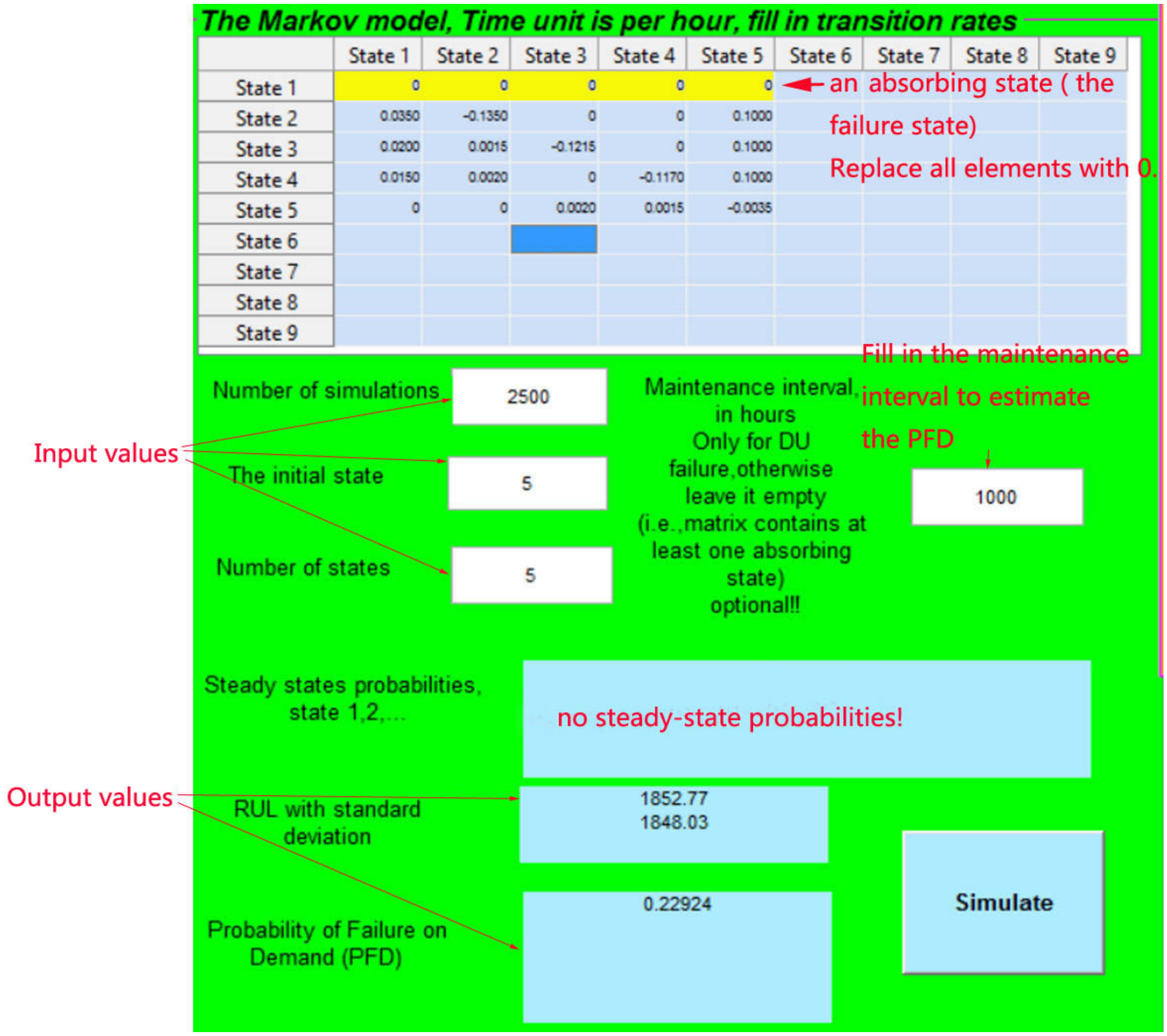
Steady states probabilities, state 1,2,...: 0.0052371 0.00033635 0.016338 0.012568 0.96552

RUL with standard deviation:

Probability of Failure on Demand (PFD): PFD is the steady-state probabilities of failure states

Output values

Figure 3.7: Markov model with no absorbing state



The Markov model, Time unit is per hour, fill in transition rates

	State 1	State 2	State 3	State 4	State 5	State 6	State 7	State 8	State 9
State 1	0	0	0	0	0				
State 2	0.0350	-0.1350	0	0	0.1000				
State 3	0.0200	0.0015	-0.1215	0	0.1000				
State 4	0.0150	0.0020	0	-0.1170	0.1000				
State 5	0	0	0.0020	0.0015	-0.0035				
State 6									
State 7									
State 8									
State 9									

an absorbing state (the failure state)
Replace all elements with 0.

Number of simulations: 2500

The initial state: 5

Number of states: 5

Maintenance interval, in hours Only for DU failure,otherwise leave it empty (i.e.,matrix contains at least one absorbing state) optional!!: 1000

Fill in the maintenance interval to estimate the PFD

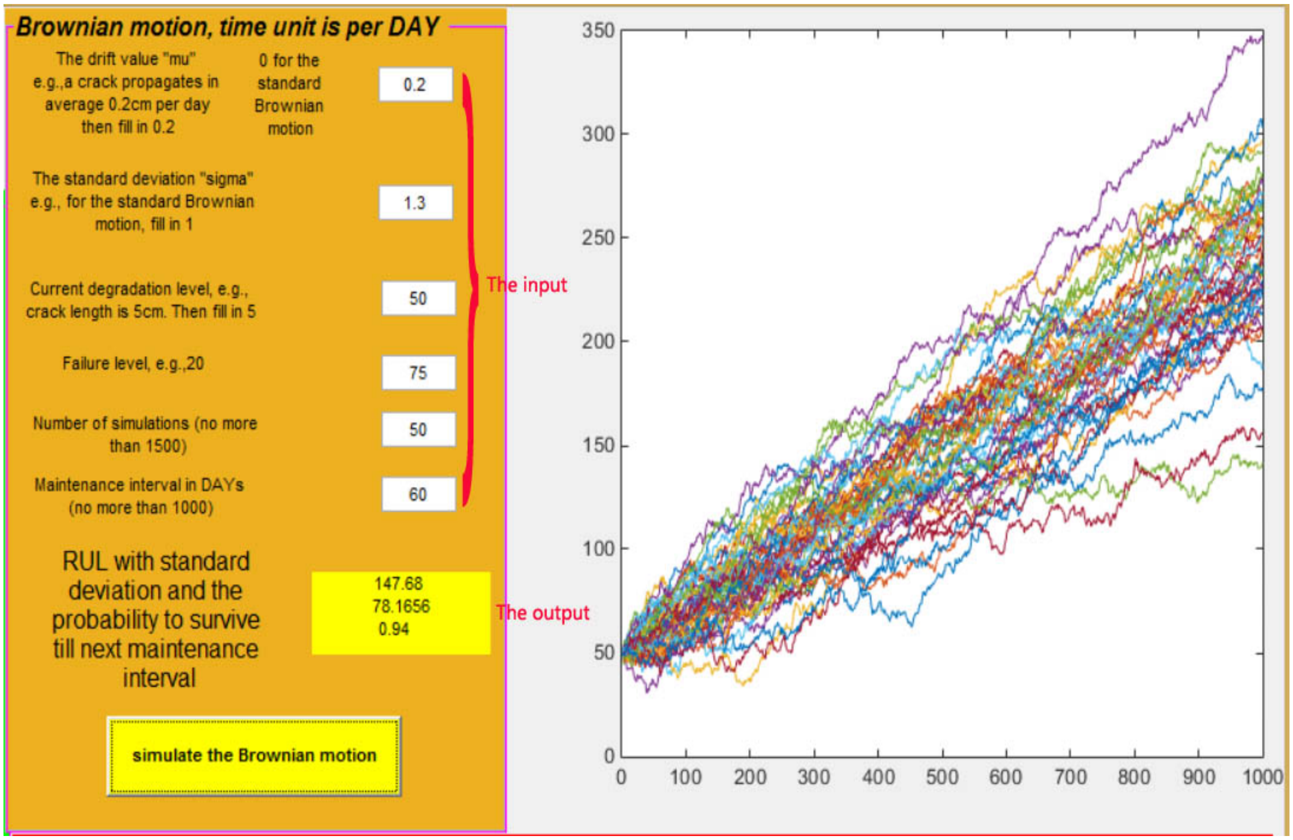
Steady states probabilities, state 1,2,...: no steady-state probabilities!

RUL with standard deviation: 1852.77
1848.03

Probability of Failure on Demand (PFD): 0.22924

Figure 3.8: Markov model with an absorbing state

Compared to the dashboard for the Markov model, the dashboard for Brownian motion and the Gamma process is more clear and easier to use. Fill in all input values and click the "simulation button". All results are shown on the same dashboard. The plot for the simulations is on the right corner. The Y-axis refers to the degradation level. The X-axis refers to the time. See Figure 3.9 for Brownian motion and Figure 3.10 for the Gamma process.



Brownian motion, time unit is per DAY

The drift value "mu" e.g.,a crack propagates in average 0.2cm per day then fill in 0.2 — 0 for the standard Brownian motion: 0.2

The standard deviation "sigma" e.g., for the standard Brownian motion, fill in 1: 1.3

Current degradation level, e.g., crack length is 5cm. Then fill in 5: 50

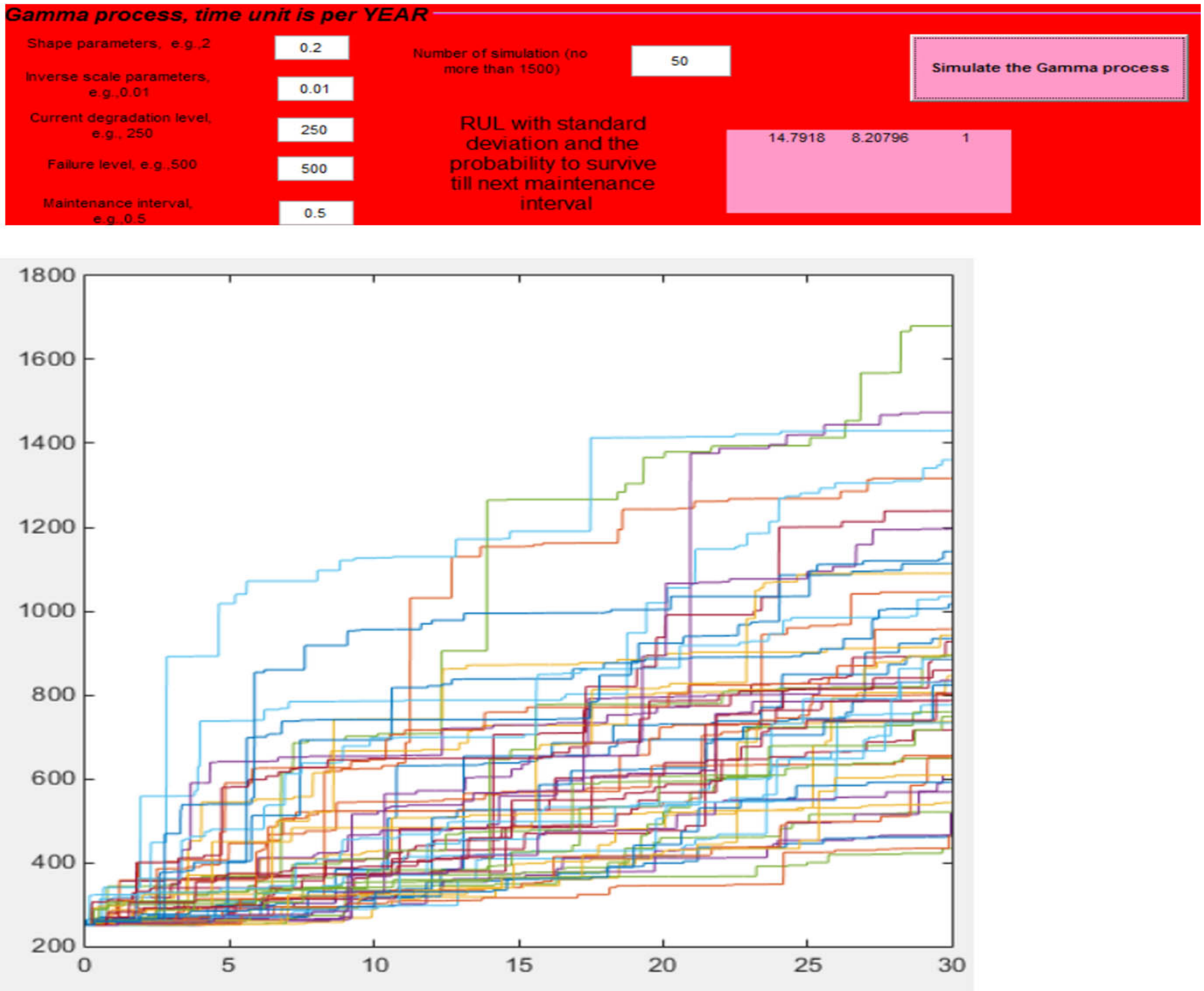
Failure level, e.g.,20: 75

Number of simulations (no more than 1500): 50

Maintenance interval in DAYs (no more than 1000): 60

RUL with standard deviation and the probability to survive till next maintenance interval: 147.68
78.1656
0.94

Figure 3.9: Toolbox for the Brownian motion


Gamma process, time unit is per YEAR

Shape parameters, e.g.,2 0.2

Inverse scale parameters, e.g.,0.01 0.01

Current degradation level, e.g., 250 250

Failure level, e.g.,500 500

Maintenance interval, e.g.,0.5 0.5

Number of simulation (no more than 1500) 50

RUL with standard deviation and the probability to survive till next maintenance interval 14.7918 8.20796 1

Figure 3.10: Toolbox for the Gamma process

3.5 Summary

This chapter discusses the Markov model, the Brownian motion process and the Gamma process and uses these models to simulate degradation propagation. In addition, a Matlab toolbox is developed. Users without any stochastic processes knowledge can still use this toolbox to make good maintenance strategies. The toolbox comprises two files. One is the code "Three stochastic models for CBM Version1.m". Another one is the figure "Three stochastic models for CBM Version1.fig". Both can be download from "`https://www.dropbox.com/sh/lxwnvfj5inq44wc/AAAJqbJ0sCpz8d5NN-wDgzyJa?dl=0`". To use the toolbox in Matlab, both files need to be placed in the Matlab folder. Double click the code and press the "RUN" button which could be found in Matlab EDITOR panel. Choose a desired model and fill in all parameters. Press the "simulate" button, and the results can be found in the same panel. It usually takes less than 13 seconds to find the results. For a large number of simulations, more time is expected. The code and the figure for the toolbox are also enclosed in Appendix B.

Chapter 4

Stochastic modelling for the BOP system condition based maintenance

BOP means Blow Out Preventer. It is pressure control equipment used as a final safety barrier by controlling the formation pressure and fluid encountered in the well. If a well control situation occurs, e.g., the formation fluid entering the well uncontrolled, the well can be closed in at the BOP. A BOP is installed on the top of the wellhead, and it consists of three main subsystems, which are Lower Marine Riser Package (LMRP), Lower Blowout Prevent (LBOP) and the control system. LBOP is also called the BOP stack. These three subsystems are designed to work together in a drilling operation.

Figure 4.1 is the Deepwater Horizon BOP stack. This configuration has two annular preventers, two shear rams and three pipe rams. This configuration is used as the basis for the analysis in this report. Many variations to this stack configuration exit. These variations are called stack classes. According to API standard 53, a subsea BOP shall be class 5 or greater, which implies the BOP consisting of at least one annular preventer, two pipe rams and two sets of shear rams. By searching the API standard 53, the Deepwater Horizon BOP stack is found to be class 7. The higher number reflects the higher redundancy in the BOP system. However, a BOP system with high redundancy has some weaknesses. It may increase the complexity, the chance of common cause failures and the total BOP system maintenance downtime. Thus increasing the redundancy may not be the best solution. However, by introducing the BOP system condition mon-

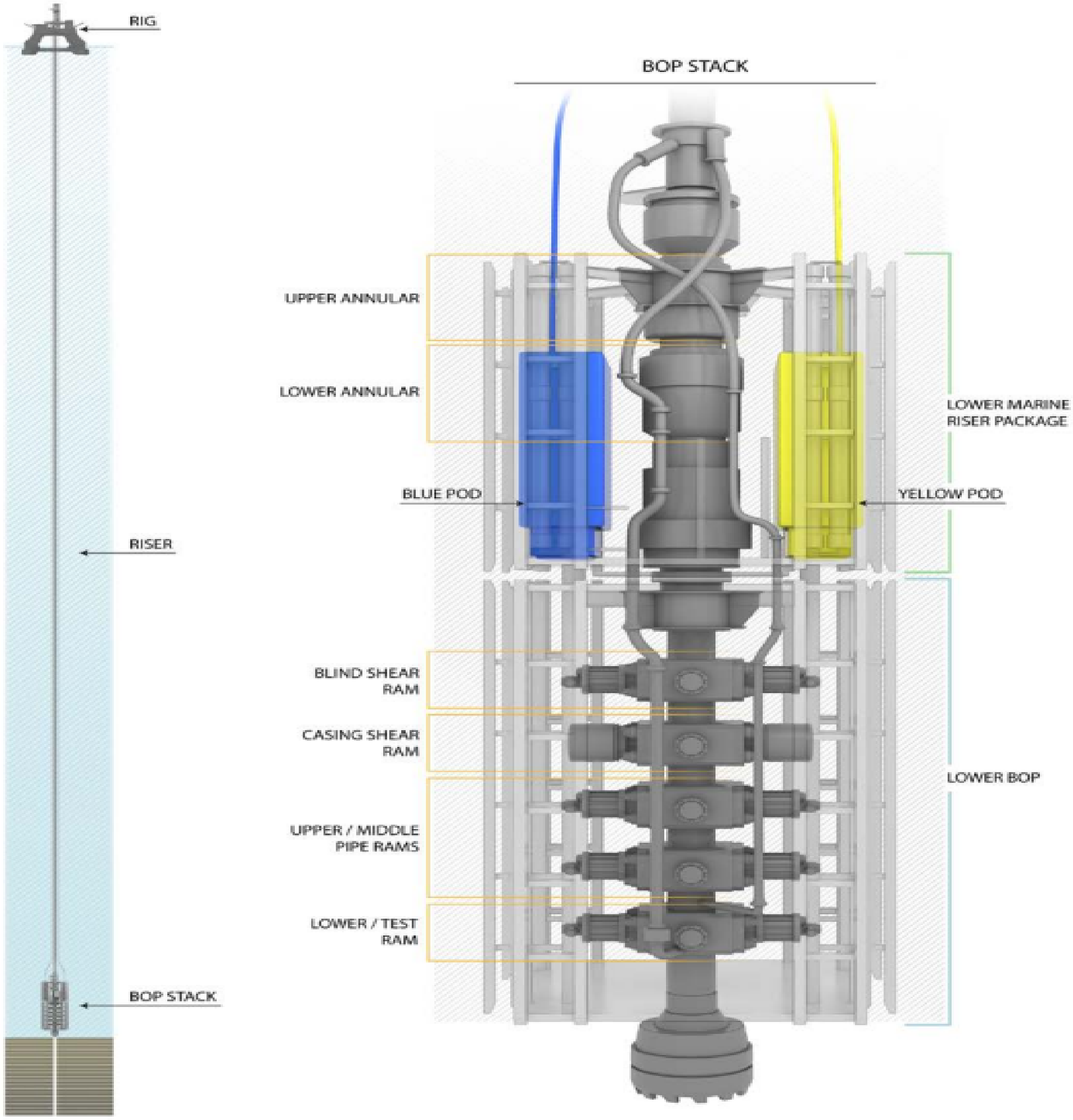


Figure 4.1: Deepwater Horizon BOP stack. (U.S. Chemical Safety And Hazard Investigation Board, 2010), page 18

itoring can, in some extent, tell exact the health state of some subsystems and give real-time well operation information. It can raise the alarm about incipient failures, and consequently it can enhance the reliability of the BOP system. In addition, it can cut many unnecessary maintenance tasks without compromising on reliability and safety.

The cost of BOP downtime can led to significant expense to the drilling contractor. The total BOP downtime cost for one of the drilling contractors was around $80 million for year 2012 (Kenny, 2004). So there is a huge benefit in having a BOP condition monitoring system and perform CBM.

4.1 BOP system description

4.1.1 Lower Marine Riser Package

The major components of the Lower Marine Riser Package (LMRP) in Figure 4.1 from top to bottom are:

Flex Joint

Though the flex joint is not labelled in Figure 4.1, it is very easy to be found. It locates between LMRP and riser adapter. It is the cylinder-shape part above upper annular in Figure 4.1. The main function of the flex joint is to allow the riser adapter and riser system to pivot on top of the LMRP up to 10 degrees off center to reduce the bending moments on the BOP stack and the well head (Kenny, 2004).

Annular Preventers

Two annular preventers are labelled in Figure 4.1, i.e., the upper annular and the lower annular. The annular has a donut shaped rubber seal called elastomeric packing element reinforced with steel ribs or inserts (Kenny, 2004). Annular preventers are actually rubber sealing elements and can seal any size of the drilling string, including an open hole to close the wellbore. Because it is forming a 360 degree seal (not 2 half moons like the rams). When a kick is detected, the annular preventer is the first element to seal against the well pressure. It withstands much lower pressure compared to the pipe ram, and it is the reason that the annular preventer is located on the top part of the stack.

Control Pods

Control pods control the supply of hydraulic fluid for the operation of LMRP and LBOP. There are two redundant control pods used in the BOP, known as the blue pod and the yellow pod. See Figure 4.1.

Subsea Gate Valve

It is at the bottom of LMRP (between the lower annular and LBOP in Figure 4.1. While running the BOP stack, the subsea gate valve is used to test riser connections (Kenny, 2004).

Choke and Kill Lines

They are the two "pipes" through the whole BOP stacks. They are depicted in Figure 4.1 (without labelled). The choke line is used to circulate the fluid from the well and the kill line is used to pump fluid into the well when the rams are closed (Kenny, 2004).

4.1.2 Lower Blowout Preventer

The major components in LBOP are rams. Four types of rams from top to bottom in Figure 4.1 are:

Blind Shear Ram

Blind shear ram is designed to shear the drilling string (if any) and seal the wellbore. It contains rubber sealing elements and can seal the well when the well does not contain a pipe. It is the only ram in the BOP stack has the capability in both sealing off a well and shearing the drill pipe. Though the ability for a blind shear ram of sealing off an open hole is lower compared to the annular preventer or the pipe ram. Blind shear rams are used as the last resort to regain the control of the well. However, if there is a drill pipe in the wellbore, the price of actuating the blind shear ram is very high, both in terms of downtime and drill pipe destruction. If the blind shear ram fails to act as the last barrier, it may lead to the total loss of the well control, or even a blowout. Some investigation reports of Deepwater Horizon accident suggest that the drill pipe buckled due to high pressure. The blind shear ram failed to cut the drill pipe completely. Other suggest that the drill pipe tool joint was passing the blind shear ram, and the blind shear ram can not shear the tool joint since it is not designed to do so. In either way, the blind shear ram failed to shear the drill pipe. Unfortunately, a blow out occurred subsequently. Thus, ensuring that the blind shear ram can function on the demand is very important.

Casing Shear Ram

Casing shear ram is designed to shear the casing or bigger sized drill pipe. it can not seal off the wellbore since it does not contain rubber sealing elements. A casing shear ram has a higher capacity in shearing compared to a blind shear ram.

Pipe Rams

Pipe Rams are designed to seal around the outer diameter of the drill pipe to control the pressure from a wellbore. There are two types of pipe rams. Standard pipe rams are to seal on a specified drill pipe outer diameter. Variable bore rams are more flexible and can seal multiple size of pipes with some loss in pressure capacity. The Upper and the middle pipe ram in Figure 4.1 have a same function as aforementioned. In addition, the middle pipe ram can be used to strip through to hang off drill pipes (Kenny, 2004). Pipe ram does not contain rubber sealing elements.

Test Ram

At the bottom of the LBOP, there is a test ram. Test rams are upside down pipe rams which permit the BOP to hold pressure from the top of the rig. The test ram can be closed and the annular and all the rams can be pressure tested against the drill string without exposing the BOP to well pressures. Using test rams at the bottom of the stack can lead to significant time and cost savings (Kenny, 2004).

4.1.3 Summary

For a short summary, the function of different rams and annular preventers are:

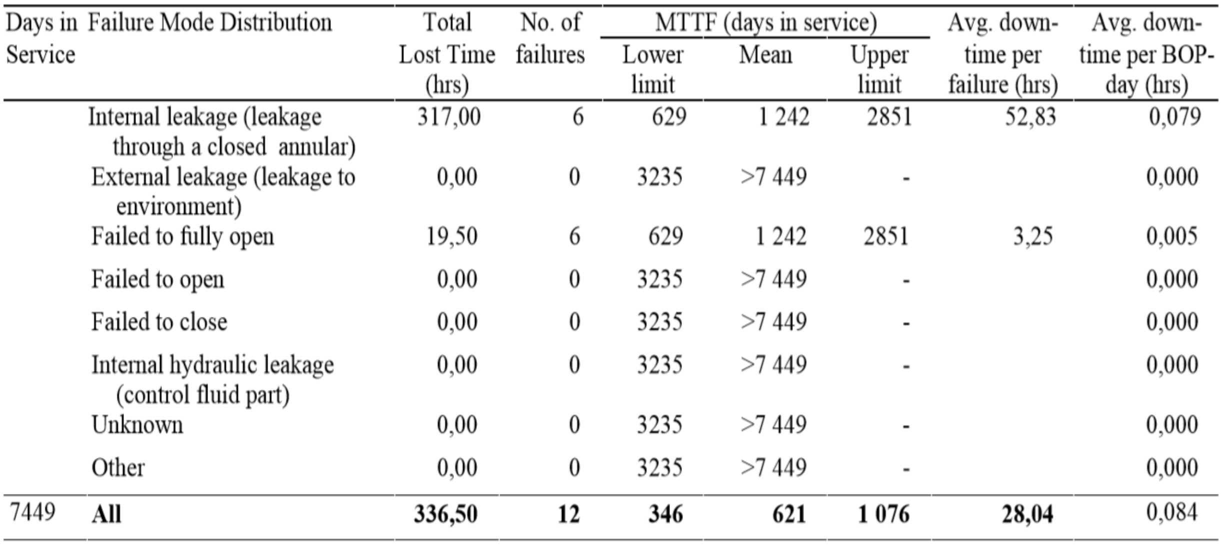
- Blind Shear Rams are designed to shear the pipe and seal the well during a well control situation.
- Casing Shear Rams are designed to shear the casing.
- Pipe Rams are designed to seal around the drill pipe.
- Variable Bore Pipe Rams can seal around various size of pipes.
- The annular preventers are designed to seal the wellbore by closing around the drill pipe.

Kenny (2004) gives several primary functions of a BOP system:

- Regulate and monitor wellbore pressure.
- Shut in the well (e.g. seal the void, annulus, between drillpipe and casing).
- “Kill” the well (prevent the flow of formation fluid, influx, from the reservoir into the wellbore).
- Seal the wellhead (close off the wellbore).
- Sever the casing or drill pipe (in case of emergencies).

4.2 Annular preventer: the data source, inferences and assumptions

Following sections chooses the annular preventer to analyze. An annular preventer is the preliminary element that is activated in a BOP system. When a kick is detected, in many cases, the well pressure is low, and the drill pipe is in the well. The casing is usually not passing the BOP system. This scenario is regarded as the only situation in this Master’s thesis, and it is the most common scenario which can be observed during a daily drilling operation. For this scenario, two annular preventers are activated at first. Afterwards, the pipe rams can be activated if it is necessary. In total, the whole BOP stack acts as a 1oo4 system, i.e., two annular preventers and two pipe rams. See Figure 4.1. In terms of downtime and drill pipe destruction, for this scenario, cutting drill pipe is not an option. Thus neither the blind shear ram nor the casing shear ram is taken into consideration.



Days in Service	Failure Mode Distribution	Total Lost Time (hrs)	No. of failures	MTTF (days in service) Lower limit	Mean	Upper limit	Avg. down-time per failure (hrs)	Avg. down-time per BOP-day (hrs)
	Internal leakage (leakage through a closed annular)	317,00	6	629	1 242	2851	52,83	0,079
	External leakage (leakage to environment)	0,00	0	3235	>7 449	-		0,000
	Failed to fully open	19,50	6	629	1 242	2851	3,25	0,005
	Failed to open	0,00	0	3235	>7 449	-		0,000
	Failed to close	0,00	0	3235	>7 449	-		0,000
	Internal hydraulic leakage (control fluid part)	0,00	0	3235	>7 449	-		0,000
	Unknown	0,00	0	3235	>7 449	-		0,000
	Other	0,00	0	3235	>7 449	-		0,000
7449	**All**	**336,50**	**12**	**346**	**621**	**1 076**	**28,04**	0,084

Figure 4.2: Annular preventer failure modes and associated number of failures. (Holand, 1999), page 50

An annular preventers has at least five types of failure modes. Among them, "internal leakage", "external leakage" and "fail to close" are dangerous undetected failure (DU failure). "Fail to open" and "fail to fully open" are safe failure. Table 4.2 shows an overview of annular preventer failure modes, including corresponding number of failures, MTTF, average downtime, etc. As can be seen from Table 4.2, 6 out 12 annular preventers failure is "internal leakage", which also accounts for 94% of total lost time (317hours/336.5hours). Thus DU failure "internal leakage" is considered as the most critical failure mode. If "internal leakage" can be successfully modelled, or a condition based maintenance plan is made, it can significantly reduce the downtime due to the annular preventers failure and improve the reliability of the whole BOP stack.

Figure 4.3 is a cutaway view of an annular preventer. The shadow part labelled with 107 is the rubber sealing elements, called elastomeric annular body (hereinafter referred to as “the elastomer”). Number 109 refers to the metal inserted in which is like ribs. When a sealing demand occurs, the metal inserted pushes the elastomeric annular body to have a 360 degree seal. The elastomer can seal multiple size of drill pipes or an open hole (at least for some annular preventers). Since the elastomer is made of rubber, it is particularly sensitive to the time of exposure and to high temperatures. As is known to all, BOP subsystems are often exposed in a High Pressure High Temperature (HP/HT) condition. Thus the aging rate of the elastomer accelerates. When the elastomer ages, it loses its ability to return to original condition. Exposing in a high temperature condition for a long time can increase an elastomer's hardness considerably, significantly affecting its ability to seal. It may result that more force is needed to seal, or internal leakage after sealing.

Though the oxidizing process of the elastomer can be accelerated by various things, high temperature is considered to be the most significant factor. Following sections discusses models to model the elastomer degradation process. For example, if the temperature in a wellbore remains unchanged, the elastomer deterioration process can be modelled by a Brownian motion process. Let $X(t)$ define the deterioration level at time t. For every $t > 0, X(t)$ is normally distributed with mean μt and variance $\sigma^2 t$.

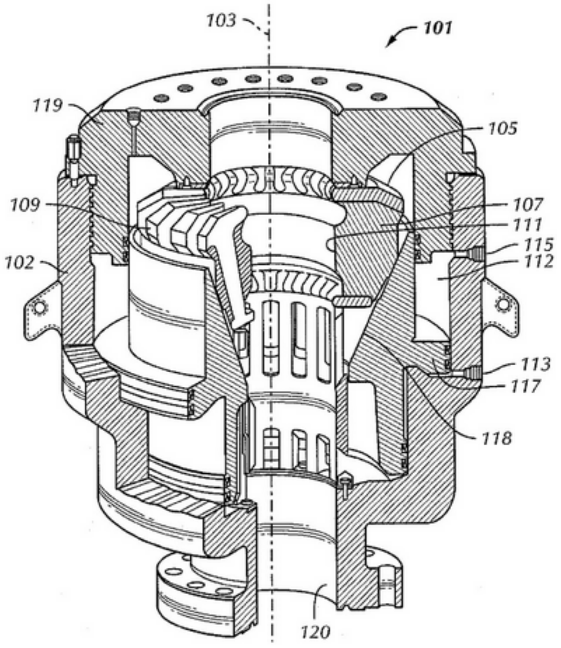


Figure 4.3: Cutaway view of an annular preventer

Holand (1999) gives MTTF for the failure mode "internal leakage" of the annular preventer 1242 hours and 90% confidence interval (629,2851). Holand (1999) also assumes the number of failures during a specific time period is modelled by a homogeneous Poisson process with failure rate λ. This λ is estimated by:

$$\hat{\lambda} = \frac{\text{Number of failure}}{\text{accumulated operating time}} = \frac{n}{s}$$

Though it is not explicitly indicated in his report which method he chooses to estimate the failure rate. Clearly, the method is Maximum Likelihood (MLE) with the exponential distribution. As can be found in his report, the number of failures of the annular preventer internal leakage is 6. The number of observed annualr preventers is 26. Unfortunately, Holand (1999) does not give the value of each annular preventer survival time or failure time. To find which type of censoring his report chooses, the 90% confidence interval from Figure 4.2 is examined. If the experiment has a set number of subjects or items and stops the experiment at a predetermined time, i.e., type I censoring, the 90% confidence interval is:

$$629 = \frac{2\sum_{i=1}^{n} x_i}{\chi^2(0.05, 2r+2)} \leq \frac{1}{\hat{\lambda}} = \hat{\theta} \leq \frac{2\sum_{i=1}^{n} x_i}{\chi^2(0.05, 2r)} = 2851$$

If the experiment has a set number of subjects or items and stops the experiment when a predetermined number are observed to have failed, i.e., type II censoring, the 90% confidence interval is:

$$527 = \frac{2\sum_{i=1}^{n} x_i}{\chi^2(0.05, 2r)} \leq \frac{1}{\hat{\lambda}} = \hat{\theta} \leq \frac{2\sum_{i=1}^{n} x_i}{\chi^2(0.05, 2r)} = 2851$$

The only difference is the degree of freedom of χ^2 distribution on the left. r is the failure number "6". The accumulated operation time $\sum_{i=1}^{n} x_i$ is 1242 × 6 = 7452 days. In addition, the value of χ^2 distribution can be found in the table. After comparing the confidence interval calculated above and the one in Figure 4.2, it is clear that the experiment is type I censored, i.e., the time is truncated at some predetermined value. All these findings can help to simulate failure times.

Five assumptions to make the following models work:

1. Temperature varies from time to time. It is very difficult and impractical to test all critical elastomers with every drilling fluid combination at all the possible temperature ranges. We assume the temperature is the only influence factor to accelerate the degradation rate. In addition, the average temperature is used.

2. It is assumed that the failure mode "internal leakage" in an annular preventer is 100% caused by a deteriorated elastomer.

3. Specific tests (such as sealing characteristics tests for ram preventers or drift tests for annular preventers) can be used to re-qualify elastomers prior to a drill stem test or during between-well maintenance. We assume after each test, we know exactly the degradation level of the elastomer.

4. under each test (usually every 14 days), the probability to reveal a failure during each test, given such failure has occurred, is a random variable between 0 and 1. We assume this number is 1.

5. A failure could be found either during a test or under a demand. Then it is recorded as a failure time. The time to perform the preventive maintenance will be regarded as right censored, if a preventive maintenance should perform before the failure.

4.3 Analyze failure data without introducing covariates

The following Matlab code simulates 30 failure data for an annular preventer with the failure mode "internal leakage". According to (Holand, 1999), the failure time is assumed to be exponentially distributed with $\theta = 1242$. The inference of the confidence interval from Figure 4.2 shows the experiment is right censored. Here, the truncated time is assumed to be 350 days.

```
% To Simulate the lifetime for 30 Annular preventer.
% failure mode: internal leakage. The experiment stops after 350 days.
lifetime=exprnd(1242,1,30); censored=zeros(1,30);
censored(find(lifetime>350))=1; lifetime(find(lifetime>350))=350;
```

22 out of 30 simulated failure data for the annular preventer is right censored at time 350. The rest is the exact failure time. They are 308.227, 90.953, 315.435, 250.074, 285.195, 259.295, 78.169, 164.508. Value 1 means the exact failure time while value 0 means right censored. See Appendix B.3.

4.3.1 Exponential model

This is the starting point of analyzing this data set. Under this model, the lifetime is assumed to be exponentially distributed. If a failure is observed at time t_i, the contribution to the likelihood function is $(\frac{1}{\theta}e^{-\frac{t_i}{\theta}})$. If a right censored value, i.e., 350, is observed, the contribution to the likelihood function is $1-e^{-\frac{350}{\theta}}$. Thus the likelihood function for exponential model is:

$$L(\theta) = \prod_{\text{failure}} (\frac{1}{\theta}e^{-\frac{t_i}{\theta}}) \prod_{\text{right censored}} (1-e^{-\frac{350}{\theta}})$$

Minitab gives the Maximum value of log-likelihood function "-64.596", at which $\hat{\theta}$ is 1181.48. Figure 4.4 is the probability plot for this exponential model with the 90% confidence interval . Eight points refer to 8 observed failure time. It is an "OK" fit since only one point is outside the 90% confidence interval. The way Holand (1999) calculating the 90% confidence interval is different from the one used in Minitab. Minitab calcuates 90% confidence interval as $\hat{\theta}e^{\pm 1.645\frac{\widehat{SD(\hat{\theta})}}{\hat{\theta}}}$, where $\Phi(1.644854) = 0.95, \Phi(-1.644854) = 0.05$. $\widehat{SD(\hat{\theta})} = \frac{\hat{\theta}}{\sqrt{r}} = \frac{1181}{\sqrt{6}}$ is the standard error. Lower standard error means better estimation.

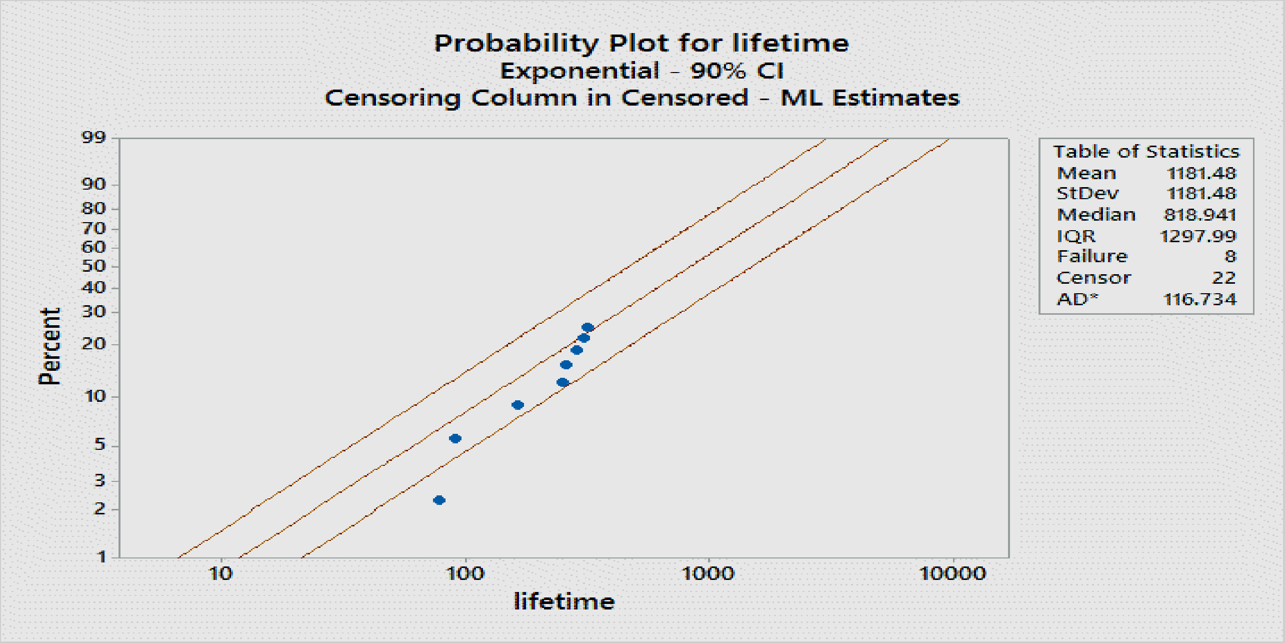


Table of Statistics	
Mean	1181.48
StDev	1181.48
Median	818.941
IQR	1297.99
Failure	8
Censor	22
AD*	116.734

Figure 4.4: Probability plot for the exponential model

4.3.2 Weibull model, log-logistic model and log-normal model

Other distributions may fit the data better. Here three types of models are analysed, i.e., the Weibull model, the log-logistic model and the log-normal model. These three distributions be-

long to the log-location-scale family. The lifetime for a annular preventer with failure mode "internal leakage" T has a log-location-scale family of distributions if lnT has a location-scale family. Lawless (2002) gives lnT = $\mu+\sigma U$, where
if ln T ∼ Normal (μ,σ), then T is log-normal ∼ (μ,σ), i.e., lnT= $\mu+\sigma U$ and U ∼ Normal $(0,1)$.
if lnT∼logistic(μ,σ), then T is log-logistic (μ,σ), i.e., lnT= $\mu+\sigma U$ and U ∼ logistic $(0,1)$.
if T ∼Weibull(θ,α), i.e., lnT= $\ln\theta+\frac{1}{\alpha}$ and U ∼ Gumbel(0, 1), then lnT∼ Gumbel $(\ln\theta, \frac{1}{\alpha})$.

Thus the cumulative distribution function for log-location-scale family is

$$F_T(t) = P(T \le t) = P(\ln T \le \ln t) = P(U \le \frac{\ln t-\mu}{\sigma}) = \Phi(\frac{\ln t-\mu}{\sigma}).$$

And the probability density function for log-location-scale family is

$$f_T(t) = \phi(\frac{\ln t-\mu}{\sigma})\frac{1}{\sigma t}.$$

Thus the likelihood function for all log-location-scale family is

$$L(\mu,\sigma) = \prod_{i:\delta_i=1}\phi(\frac{\ln t_i-\mu}{\sigma})\frac{1}{\sigma t}\prod_{i:\delta_i=0}(1-\Phi(\frac{\ln t_i-\mu}{\sigma})). \tag{4.1}$$

where $\delta_i = 1$ means the failure time is observed while $\delta_i = 0$ means it is right censored. Figure 4.5 is the probability plot for four models. It is used to determine which model is the best one to fit the data. From the value of Anderson-Darling, the Weibull model seems slightly better (with the lowest value except the exponential model). Besides, the eight points are close to the red line.

Therefore Weibull model is chosen for the further analysis. After inserting the failure data to Equation 4.1, maximum value of the likelihood function can be found. Minitab gives the maximum value of Log-Likelihood "-63.283", at which the shape parameter α and the scale parameter θ are estimated as 1.84315, 600.802 respectively. Figure 4.6 is the probability plot for the Weibull model. It is a good fit since all points are within 90% confidence interval.

A null hypothesis is raised to test if the data come from an exponential distribution. This is done

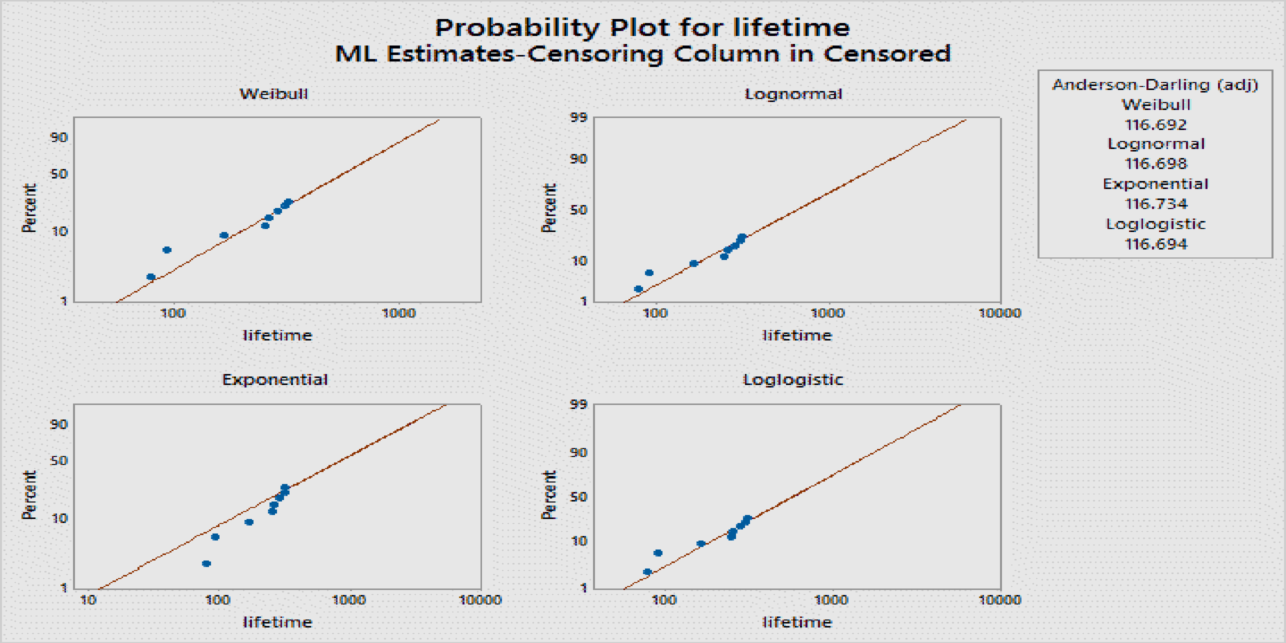


Figure 4.5: Probability plot for four models

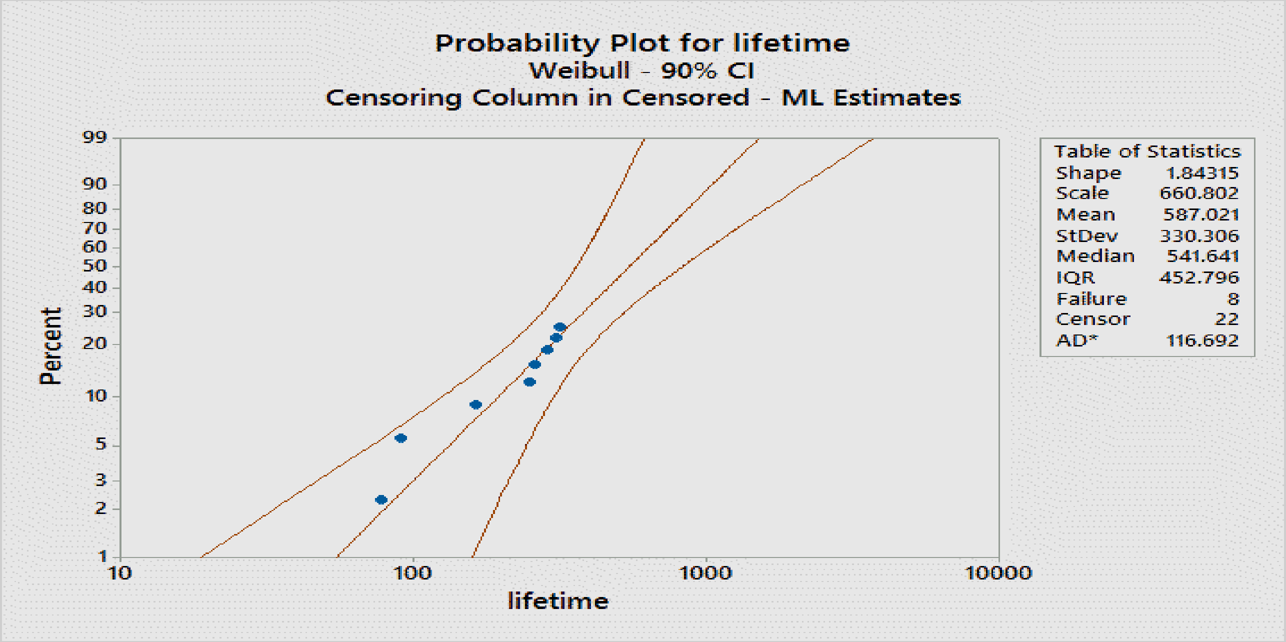


Figure 4.6: Probability plot for the Weibull model

by testing

$$H_0 : \alpha = 1 \text{ vs } H_1 : \alpha \neq 1$$

If $\alpha = 1$ is the true value, then $W(1) = 2(\hat{l}(\hat{\alpha}) - l(1)$ should be χ_1^2 distributed (the null hypothesis is true). In other word, we reject the null hypothesis if $W(1) \geq 2.71$ at 10% significant level. $\hat{l}(\hat{\alpha})$ is the maximum value of the likelihood function in the Weibull model and $l(1)$ is the maximum value of the likelihood function in the exponential model. Thus

$$W(1) = 2(\hat{l}(\hat{\alpha}) - l(1) = 2 \times (-63.283 - (-64.596)) = 2.626 < 2.71$$

So the null hypothesis is not rejected. The conclusion is hence that there is not enough evidence to conclude that the data is not exponentially distributed. Another alternative is to check the 90% confidence interval for α.

It is easy to find the the Mean residual Life (MRL) for aforementioned models. Rausand (2014) gives

$$\text{MRL}(t) = \frac{1}{R(t)} \int_t^\infty R(x)dx. \tag{4.2}$$

Take Weibull model for an example, $\text{MRL}(t) = \frac{1}{e^{-(\frac{t}{\theta})^\alpha}} \int_t^\infty e^{-(\frac{x}{\theta})^\alpha} dx$. By inserting the estimated parameters $\alpha = 1.84, \theta = 600.8$ and $t = 170$(i.e., half year), MRL is calculated as 407.5. See the following Matlab code.

```
theta=600.8; a=1.84; t=170; % from the previous analysis
fun = @(x) 1/exp(-(t/theta)^a)*exp(-(x/theta).^a);
MRL = integral(fun,t,Inf)
```

The weakness for this approach is that the influence factor, e.g., the temperature, is not taken into consideration. These models can hardly incorporate the condition of the elastomer monitored under the each inspection. Besides, the MRL is only an estimated mean value for Remaining Useful Life (RUL). The operation conditions vary. It leads to a huge uncertainty of the estimation. So these models can only give a preliminary analysis of RUL. To apply CBM, more advanced models need to be established.

4.3.3 The Browian motion process

If the deterioration process of the elastomer in an annular preventer can be modelled with Brownian motion, at each time t with observed health state of the elastomer, one can estimate RUL more precisely. To make this model work, firstly, a failure level needs to be defined. In this Master's thesis, the failure level for the elastomer is defined as number "100". It is unit-less. This number could be any positive number, e.g.,25. However, in such case, other parameters in the Brownian motion process needs to be altered to fit the historical data. Secondly, the parameters for the Brownian motion process are estimated from the historical data, i.e., from Appendix B.3. The method to estimate the parameters is same as before, i.e., maximum likelihood estimation. Thirdly, based on the health state monitored under the each inspection, RUL is estimated as the first time hitting the failure level. This value can be found by running the self-made toolbox in Chapter 3.

let $X(t_1), X(t_2)$ be the degradation levels for the elastomer at time t_1, t_2 respectively, where t_1 is the current time and t_2 is the future time. According to (Ross, 2014), $X(t_2) - X(t_1)$ is normally distributed with mean μt and variance $\sigma^2 t$, i.e., the Brownian motion process with a drift value μ. Another alternative expression is

$$X(t_2 - t_1) = \sigma B(t_2 - t_1) + \mu(t_2 - t_1).$$

where $B(t)$ is standard Brownian motion whose density function is

$$f(x) = \frac{1}{\sqrt{2\pi t}} \exp(-x^2/2t) \qquad \text{See Section 3.2.}$$

Firstly, let us consider the simplest situation where $\mu = 0$ and $\sigma = 1$. It is also called standard Brownian motion. The lifetime for an annular preventer can be modelled as the first time the process hits the failure level, i.e.,100. Let T_{FL} denote the first time the Brownian motion process hits failure level. It is interesting to find the distribution of T_{FL}. by conditioning on whether or not $T_a \leq t$, it gives

$$\begin{aligned} P(X(t) \geq FL) = P(X(t) \geq FL \mid T_{FL} \leq t) P(T_{FL} \leq t) \\ + P(X(t) \geq FL \mid T_{FL} > t) P(T_{FL} > t). \end{aligned} \tag{4.3}$$

If the first hitting time is greater than t, then X(t) must less than FL. Therefore, $P(X(t) \geq FL \mid T_{FL} > t) = 0$. If the first hitting time is less than t, then the X(t) can be either above or below the failure level with fifty-fifty chance (due to symmetry since $\mu = 0$). Hence $P(X(t) \geq FL) = P(X(t) \geq FL \mid T_{FL} \leq t) = \frac{1}{2}$. We see that

$$P(T_{FL} \leq t) = 2P(X(t) \geq FL) = \frac{2}{\sqrt{2\pi}} \int_{\alpha}^{\infty} e^{-x^2/2t} dx = F_{FL}(t). \tag{4.4}$$

Afterwards, $f_{FT}(t), R_{FT}(t)$ can be easily derived. Given lifetime data, one can have likelihood function

$$L(\mu, \sigma) = \prod_{i:\delta_i=0} (1 - \frac{2}{\sqrt{2\pi}} \int_{\alpha}^{\infty} e^{-x^2/2t} dx) \prod_{i:\delta_i=1} f_{FT}(t). \tag{4.5}$$

where $\delta_i = 1$ means the failure time is observed while $\delta_i = 0$ means it is right censored. However, to model the lifetime of the annular preventer, i.e.,$\mu > 0$, the situation is not that simple. For $\mu \neq 0$, Equation 4.4 can not be used (not symmetric any more). Consequently, Equation 4.5 is worthless. Although modelling the situation where $\mu > 0$ follows the same steps as $\mu = 0$. That is, the cumulative distribution function and the probability density function of the first hitting time should be derived firstly since it is the only way to establish the likelihood function. Secondly, the maximum likelihood estimators $\hat{\mu}, \hat{\sigma}$ are found by maximizing L(μ, σ). In practice this is done by solving the likelihood equations (two equations in two unknowns)

$$\frac{\partial L(\mu, \sigma)}{\partial \mu} = 0, \qquad \frac{\partial L(\mu, \sigma)}{\partial \sigma} = 0.$$

To simply the problem, σ is assumed to be 1 for the following analysis. Jackson et al. (2009) give the probability density function and the cumulative distribution function for the Brownian motion process first hitting time with $\mu \neq 0$ and $\sigma = 1$

$$f_{FL}(t) = \frac{FL}{\sqrt{2\pi t^3}} e^{\frac{-(FL-\mu t)^2}{2t}},$$

$$F_{FL}(t) = \Phi(\frac{-FL + \mu t}{\sqrt{t}}) + e^{2\mu FL} \Phi(\frac{-FL - \mu t}{\sqrt{t}}).$$

Hence the likelihood function is

$$L(\mu) = \prod_{i:\delta_i=0} (1-\Phi(\frac{-FL+\mu t_i}{\sqrt{t_i}}) + e^{2\mu FL}\Phi(\frac{-FL-\mu t_i}{\sqrt{t_i}})) \prod_{i:\delta_i=1} \frac{FL}{\sqrt{2\pi t_i^3}} e^{\frac{-(FL-\mu t_i)^2}{2t_i}} \quad . \tag{4.6}$$

where t_i is the lifetime from Figure 4.4 and FL=100. Inserting all numbers, maximum likelihood is found to be 7.1182×10^{-50}, where $\hat{\mu} = 0.283$. See the following Matlab code for the maximum likelihood estimation.

```
% Maximum Likelihood Estimation for the first hitting time parameters estimation
load('lifetime_master_thesis.mat', 'lifetime'); % Load the lifetime data
censored=zeros(1,length(lifetime));
censored(find(lifetime<350))=1; % 1:failure; 0: right censored
mu=[0:0.001:1]; % Define the range of value mu
likelihood=ones(1,length(mu)); % Starting point of the likelihood for each mu
FL=100; % Failure level is 100
for i=1:length(lifetime)
    if censored(i)==0 % Right censered:L=L*(1-cumulative distribution function at t_i)
       likelihood=likelihood.*(FL/sqrt(2*pi*(lifetime(i)^3))*exp((-(FL-mu*lifetime(i))
    .^2)...
           /(2*lifetime(i))))
    else % Exact failure time: L=L*probability density function at t_i
       likelihood=likelihood.*(1-normcdf((-FL+mu*lifetime(i))/sqrt(lifetime(i)))-...
           exp(2*mu*FL).*normcdf(  (-FL-mu*lifetime(i))  /sqrt(lifetime(i))))
    end
end
[Maximum_likelihood, y]=max(likelihood); % Find the maximum value
Maximum_likelihood_estimated_mu=mu(y);
```

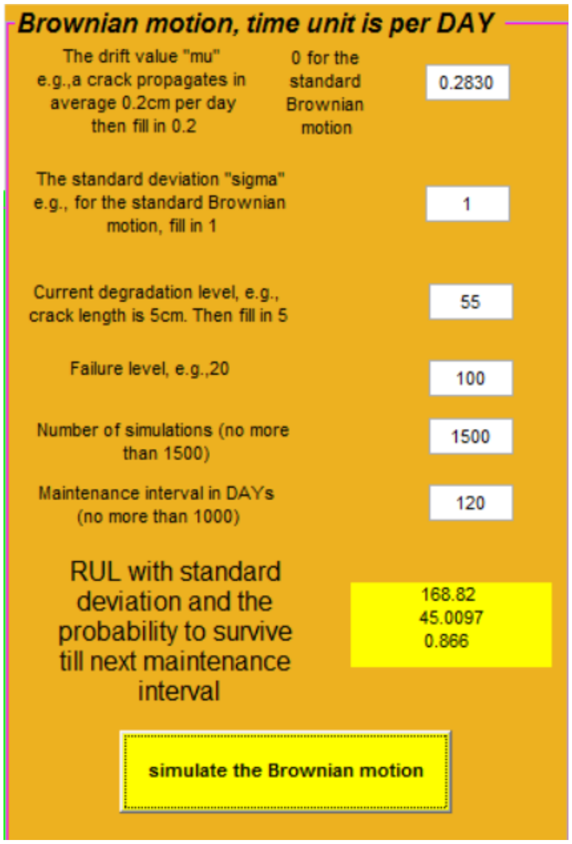
Jackson et al. (2009) gives the expected value of the first hitting time for $\mu > 0, \quad \sigma = 1$ and $X(0) = 0$, i.e., MTTF.

$$E(T_{FL}) = \frac{FL}{\mu},$$

which is $E(T_{FL}) = \frac{FL}{\hat{\mu}} = \frac{100}{0.283} = 353.36$ (days) in our case. Note that for Weibull model, it is

$$E(T) = \hat{\theta} \times \Gamma(\frac{1}{\alpha}+1) = 600.8 \times \Gamma(\frac{1}{1.84}+1) = 587(\text{ days}).$$

Minitab gives the 95% confidence interval of $E(T)$ for the Weibull model $(318.936, 1080.45)$, within which MTTF from the Brownian motion process, i.e.,353.36, can be found. It seems a good estimation since only 8 out of 30 lifetimes are the exact failure time.



Brownian motion, time unit is per DAY

The drift value "mu" e.g.,a crack propagates in average 0.2cm per day then fill in 0.2 — 0 for the standard Brownian motion — 0.2830

The standard deviation "sigma" e.g., for the standard Brownian motion, fill in 1 — 1

Current degradation level, e.g., crack length is 5cm. Then fill in 5 — 55

Failure level, e.g.,20 — 100

Number of simulations (no more than 1500) — 1500

Maintenance interval in DAYs (no more than 1000) — 120

RUL with standard deviation and the probability to survive till next maintenance interval — 168.82
45.0097
0.866

Figure 4.7: Toolbox for Brownian motion

The self-made toolbox in Section 3.2 can help to find RUL with standard deviation and the probability to survive till the next maintenance interval. For example, current inspection tells the health state of the elastomer as 55. See Assumptions 4.2. One would like to know the probability to survive 120 days, i.e., $R(120)$ and RUL. Type in number as shown in Figure 4.7 and run the simulation. RUL with standard deviation and the probability to survive 120 days are found to be 168.82, 45.0097 and 0.866 respectively. Maintenance decisions can be made based on the result from the toolbox. It is a feasible approach to model the an annular preventer elastomer degradation process when the influence factors, e.g., the temperature, are not introduced. The following sections show models incorporating the temperature, which is the main factor to influence the elastomer's lifetime.

4.4 Analyze failure data with covariates

When an elastomer in an annular preventer is used for a long period of time, it becomes hardened and loses its damping capability. This aging process results mainly from the heat, and affects not only the material property but also the useful lifetime of the elastomer (Woo et al., 2010). Nevertheless, during its lifetime, an elastomer can be submitted to different types of degradation from different exposure conditions. A failure is usually caused by the combination of these factors. Woo et al. (2010) gives some factors which could possibly affect the lifetime of an elastomer. Among them, the temperature, the humidity, the fluids (gas and vapours) and the loading pressure are the relevant factors for the annular preventer elastomer lifetime analysis. These four factors can be regarded as four covariates. The simple linear regression models can be applied. The lifetime T can be interpreted as

$$T = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + E,$$

where β_0 is the intercept, $\beta_{1,2,3,4}$ are coefficients, $X_{1,2,3,4}$ are the value of the temperature, the humidity, the fluids and the loading pressure. E is the error $\sim N(0,1)$.

For the following sections, only the temperature is chosen for the further analysis. Because it is the main factor to influence the elastomer lifetime. If the temperature inside the BOP system is recorded continuously, it can help to build more advance models and improve the accuracy of RUL estimation (for the elastomer only). In principle, the estimation can be made by extrapolating of the degree of degradation after a given time at a given temperature. Since the BOP data is confidential and hard to obtain, this Master's thesis uses a data set from Minitab "Data set library". Minitab provides numerous sample data sets taken from real-life scenarios across many different industries and fields of study. The one used in this Master's thesis "insulation.MTW" is a data set of the deterioration of an insulation that is used for electric motors under different temperature. The insulation and the elastomer from an annular preventer are made by similar types of materials, i.e., the lifetime is highly influenced by the temperature. Thus the models used to analyze the "insulation data" in the following sections can also be used for BOP system annular preventer elastomeric annular body. For insulation data set, see Appendix B.4

4.4.1 Weibull regression model

For an observation with a temperature value X, there is a potential lifetime T such that

$$\ln(T) = \beta_0 + \beta_1 x + \sigma U,$$

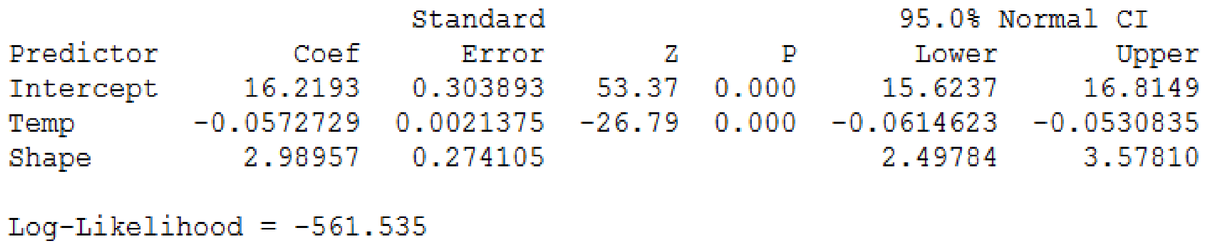
where U is standard normal distribution, standard Gumbel distribution, standard logistic distribution for log-normal regression model, Weibull regression model and log-logistic regression model respectively. See Section 4.3.2. By replacing μ to $\beta_0 + \beta_1 x$, we have the likelihood function

$$L(\mu,\sigma) = \prod_{i:\delta_i=1} \phi(\frac{\ln t_i - \beta_0 - \beta_1 x}{\sigma})\frac{1}{\sigma t} \prod_{i:\delta_i=0} (1-\Phi(\frac{\ln t_i - \beta_0 - \beta_1 x}{\sigma})), \tag{4.7}$$

where $\delta_i = 1$ means the failure time is observed while $\delta_i = 0$ means it is right censored. Note that the temperature here refers to the "average temperature", not instaneous temperature. That is, the temperature x is

$$x = \frac{\int_a^b f(x)dx}{b-a},$$

where a is the time experiments start and b is the time failures are observed or right censored time. For the Weibull regression model, Minitab gives following results. See Figure 4.8. The p-values for the intercept and the coefficient of the temperature are closed to 0. The smaller the p-value, the more evidence we have against the null hypothesis which, in our case, is H_0 : $\beta_0, \beta_1 = 0$. So we have enough evidence to reject the null hypothesis. In other words, it means the estimation is very good.



```
                      Standard                        95.0% Normal CI
Predictor      Coef      Error       Z      P        Lower        Upper
Intercept   16.2193   0.303893   53.37  0.000      15.6237      16.8149
Temp     -0.0572729  0.0021375  -26.79  0.000   -0.0614623   -0.0530835
Shape       2.98957   0.274105                     2.49784      3.57810

Log-Likelihood = -561.535
```

Figure 4.8: Output from Minitab

The estimated lifetime T can be expressed as

$$\ln(\mathrm{T}) = 16.2193 - 0.057x + \frac{1}{2.989}U \quad \text{where} \quad U \sim \text{Gumbel}(0,1) \quad \alpha = 2.989$$

since $\ln(\theta) = \mu = \beta_0 + B_1 x, \quad \theta = e^{16.2193-0.057x}$. According to (Rausand, 2014), MTTF for Weibull model is

$$\theta \times \Gamma(1/\alpha + 1)$$

It is $e^{16.2193-0.057\times 100} \times \Gamma(1/2.989+1) = 33056$ for the temperature x=100. To check if the Weibull regression model conforms to the result, random Gumbel distributed numbers are generated to simulate the lifetime. See following Matlab code.

```
for i=1:100000
lifetime(i,1)=exp(evrnd(0,1)/2.989+16.2193-5.7);
end
lifetime=sum(lifetime)/100000;
```

The average lifetime for temperature is 33039, which is very close to the one obtained by the formula from (Rausand, 2014), showing the Weibull regression model gives a good estimation.

We can also deduce the survival function R(t) and mean residual life MRL(t)

$$\mathrm{R(t)} = e^{-(\frac{t}{\theta})^{\alpha}} = e^{(\frac{t}{e^{\beta_0+\beta_1\times x}})^{\alpha}}, \tag{4.8}$$

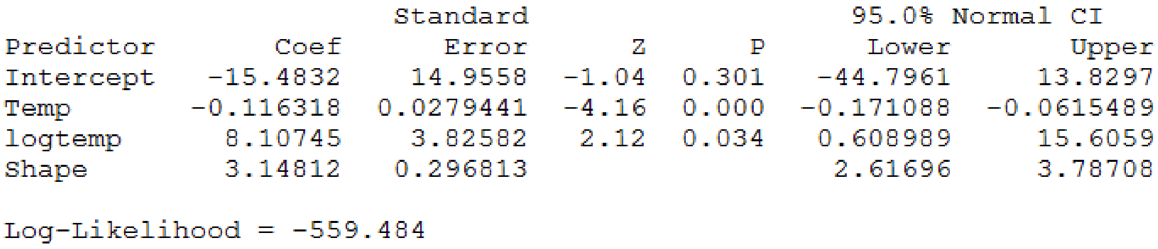
$$\mathrm{MRL}(t) = \frac{1}{R(t)}\int_t^{\infty} R(x)dx = \frac{1}{e^{(\frac{t}{e^{\beta_0+\beta_1\times x}})^{\alpha}}}\int_t^{\infty} e^{(\frac{y}{e^{\beta_0+\beta_1\times x}})^{\alpha}}d\mathrm{y}. \tag{4.9}$$

To find MRL at the temperature 100 and the time T=33039 hours, i.e., at MTTF. Inserting the estimated values into Equation 4.8 and 4.9, it gives MRL(33039)=9953 hours. See the following Matlab code

```
theta=exp(16.2193-5.7); a=2.989; t=33039; % from the previous analysis
fun = @(x) 1/exp(-(t/theta)^a)*exp(-(x/theta).^a);
MRL = integral(fun,t,Inf)
```

Another interesting part is to test whether some covariates are significant or not. For example, if the covariate "loading pressure" to the elastomer is taken into consideration. A null hypothesis that $H_0: \quad \beta_2 = 0$ must be tested where β_2 refers to the coefficient of the second covariate. Previously, the p-value of the coefficient of the temperature tells it is significant. This Master's thesis makes a virtual covariate which is the logarithm of the temperature data. The purpose by doing that is to illustrate how to test this null hypothesis in case two or more influence factors are observed for an annular preventer in the future.

For the Weibull regression model with two covariates, Minitab gives following results. See Figure 4.9.



```
                        Standard                          95.0% Normal CI
Predictor       Coef       Error      Z      P       Lower       Upper
Intercept   -15.4832     14.9558  -1.04  0.301    -44.7961     13.8297
Temp       -0.116318   0.0279441  -4.16  0.000   -0.171088  -0.0615489
logtemp      8.10745     3.82582   2.12  0.034    0.608989     15.6059
Shape        3.14812    0.296813                   2.61696     3.78708

Log-Likelihood = -559.484
```

Figure 4.9: Output from Minitab

For Model 2, the lifetime is expressed as

$$\ln T = -15.483 - 0.116x_1 + 8.107x_2 + \frac{U}{3.148} \quad \text{where} \quad U \sim \text{Gumbel}(0,1)$$

The null hypothesis is done by testing

$$H_0 : \beta_2 = 0 \text{ vs } H_1 : \beta_2 \neq 0$$

Under H_0, 2(difference of log-likelihoods) is χ^2_1 distributed. Thus we reject the null hypothesis if it is larger than 3.84 at 5% significant level. In our case, 2(difference of log-likelihoods)=$2 \times (-559.484 - (-561.535)) = 4.102 > 3.84$. Thus the null hypothesis is rejected at 5% significant level. In other words, the second covariate influences the lifetime significantly as well.

Note that these methods can also be applied in a log-logistic regression model and a log-normal regression model. Nevertheless, to make these two regression models work, $\ln(\theta)$ and α in the Weibull regression model should be replace by μ and $\frac{1}{\sigma}$, respectively.

4.4.2 Proportional hazard model (Cox regression model)

Sir David Cox, in his famous paper from 1972 suggested to use the model

$$z(t;x) = z_0(t)e^{\beta_1 x_1+\beta_2 x_2+\beta_3 x_3+....\beta_k x_k} \tag{4.10}$$

In Equation 4.10, $z(t;x)$ is the failure rate at time t with covariates x_i. $z_0(t)$ can be any positive function of t, i.e., a hazard rate function. $\beta_1,\beta_2,\beta_3,...,\beta_k$ are ordinary parameters, $x_1,x_2,...,x_k$ are the covariates. The model is said to be semi-parametric (Cox, 1987). Suppose the x_1, e.g., the temperature, is increased by 10 degree, then

$$\frac{z(t;x_{\text{new}})}{z(t;x_{\text{old})}} = \frac{z_0(t)e^{\beta_1(x_1+10)+\beta_2 x_2+\beta_3 x_3+....\beta_k x_k}}{z_0(t)e^{\beta_1 x_1+\beta_2 x_2+\beta_3 x_3+....\beta_k x_k}} = e^{\beta_1\times 10}$$

The failure rate will actually increase by $e^{\beta_1\times 10}$ times. Thus e_i^{β} is actually the "risk" (from the failure rate perspective) increased if a covariate x_i is increase by 1 unit. Cox' partial likelihood function is usually calculated to find the estimated β_i at which the likelihood function reaches its maximum value. It would be a time-consuming process to find the estimated value by hand. The partial likelihood function will be much complex when the number of covariates increases. Besides Minitab doesn't support Proportional hazard model. Luckily, Matlab has an embedded function called "coxphfit" to estimate β_i for Proportional hazard model. "Coxphfit" also returns the loglikelihood, "logl", a structure, "stats", that contains additional statistics, and a two-column matrix, "H", that contains the T values in the first column and the estimated baseline cumulative hazard, in the second column (Mathworks). See the following Matlab code

```
% Proportional hazard model parameters estimation
load('proportional_hazard_model_data.mat');
censored=abs(Total(:,3)-1);
[b,logl,H,stats]=coxphfit(Total(:,1), Total(:,2), 'censoring', censored)
```

It is the same data set, which is stored in a matrix called "Total", analyzed by Weibull regression model. For the value of "Total", see Appendix B.4. Minitab used 0 for right censored while Matlab uses 1. The third line in the code changes value 0 to 1 and vice versa from "Total (:,3)" to a new column called "censored". "Total(:,1)" is a column for the covariates, i.e., the temperature. "Total(:,2)" is the column for the response, i.e., the lifetime. "censored" is a column for the censored value. 1 means right censored and 0 means the failure time. Note that the notation is opposite to the one used in Minitab. The estimated $\hat{\beta}$=0.1788. The p-value is close to 0 referring that it is a good estimation, i.e., rejecting the null hypothesis that $\beta = 0$. Thus the failure rate at time t with the temperature x can be expressed as

$$z(t;x) = z_0(t)e^{0.1788x}$$

The estimated value of the baseline hazard rate $z_0(t)$ is not directly shown through Matlab. The cumulative hazard rate is

$$Z(t;x) = \int_0^t z_0(t)e^{0.1788x} = Z_0(t)e^{0.1788x}$$

Thus we can use Proportional hazard model to estimate the survival/reliability function:

$$R(t;x) = e^{-Z(t;x)} = e^{-\hat{Z_0}(t)e^{\hat{\beta}x}} = e^{-\hat{Z_0}(t)e^{0.1788x}}$$

Because the output from Matlab gives the estimated cumulative hazard rate $\hat{Z}_0(t)$, which is stored in the array "H". For "H" value, see Appendix B.5. "H" contains the time t values in the first column and the estimated baseline cumulative hazard rate $\hat{Z}_0(t)$ in the second column. For example, at time 665 with temperature 20. The reliability function is

$$R(t;x) = e^{-Z(t;x)} = e^{-Z_0\hat{(}t)e^{\hat{\beta}x}} = e^{-Z_0(\hat{6}65)e^{0.1788\times 20}}$$

$Z_0(\hat{6}65)$, which can be found from array "H", is estimated as 0.0067. Thus R(665;20)=0.7866. In principle, we can use Proportional hazard model to find $R(t;x)$ at different time with different covariates.

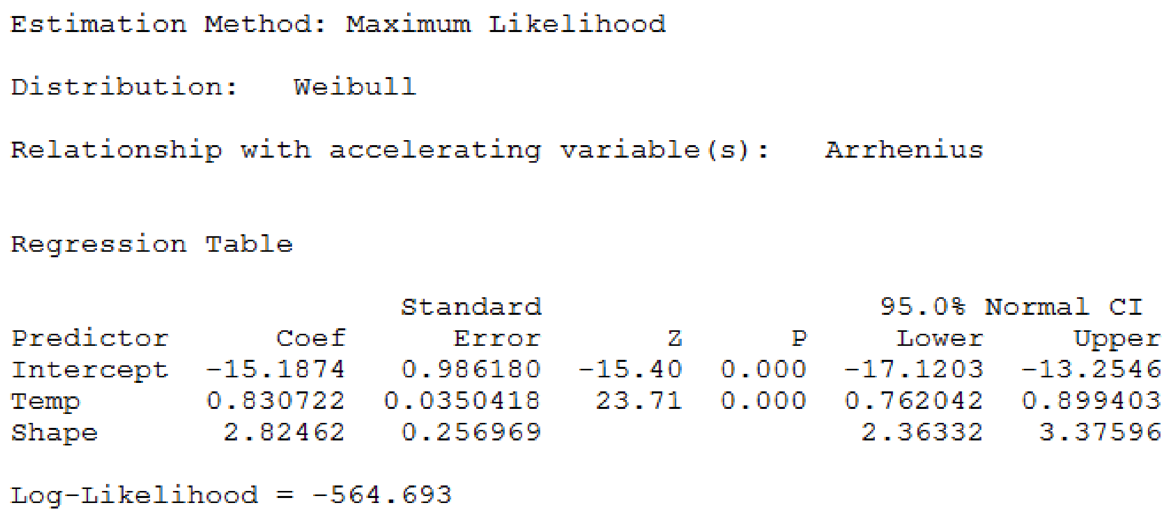
4.4.3 Arrhenius model

Arrhenius model is one of the earliest and most successful acceleration models predicts how time-to-fail varies with the temperature. The model used in this Master's thesis is

$$\ln\mathrm{T} = \beta_0 + \beta_1 \cdot \frac{c}{x+273.16} + \frac{1}{\alpha} \cdot \mathrm{W}$$

where W is standard Gumbel distributed and x is the temperature in Celsius. So $x + 273.16$ is the absolute temperature. For the data set from Appendix B.4, Minitab gives

$$\ln\mathrm{T} = -15.1874 + 0.8307 \cdot \frac{11604.83}{x+273.16} + \frac{1}{2.82462} \cdot \mathrm{W} \tag{4.11}$$



```
Estimation Method: Maximum Likelihood

Distribution:   Weibull

Relationship with accelerating variable(s):   Arrhenius


Regression Table

                       Standard                          95.0% Normal CI
Predictor      Coef       Error       Z      P       Lower      Upper
Intercept  -15.1874    0.986180  -15.40  0.000   -17.1203   -13.2546
Temp       0.830722   0.0350418   23.71  0.000   0.762042   0.899403
Shape       2.82462    0.256969                   2.36332    3.37596

Log-Likelihood = -564.693
```

Figure 4.10: Output from Minitab

Figure 4.11 is the the probability plot for failure T. Cox (1987) gives the probability function for the Weibull model

$$\ln(-\ln(\mathrm{R(t)}) = \alpha \ln\mathrm{T} - \alpha \ln\theta.$$

$\alpha = 2.82462$, which is the slope. Thus all lines share the same slope. $\ln\theta = \beta_0 + \beta_1 \cdot \frac{11604.83}{x+273.16}$ is based on the temperature. Higher temperature results in a lower θ, which makes the line more left. The failure time under each temperature is scattered clearly near the corresponding probability plot line, showing that the temperature is a significant influencing factor for the insulation lifetime. Note the α in Arrhenius model is estimated as 2.8246 which is very close to 3.148, the one gotten from the Weibull regression model.

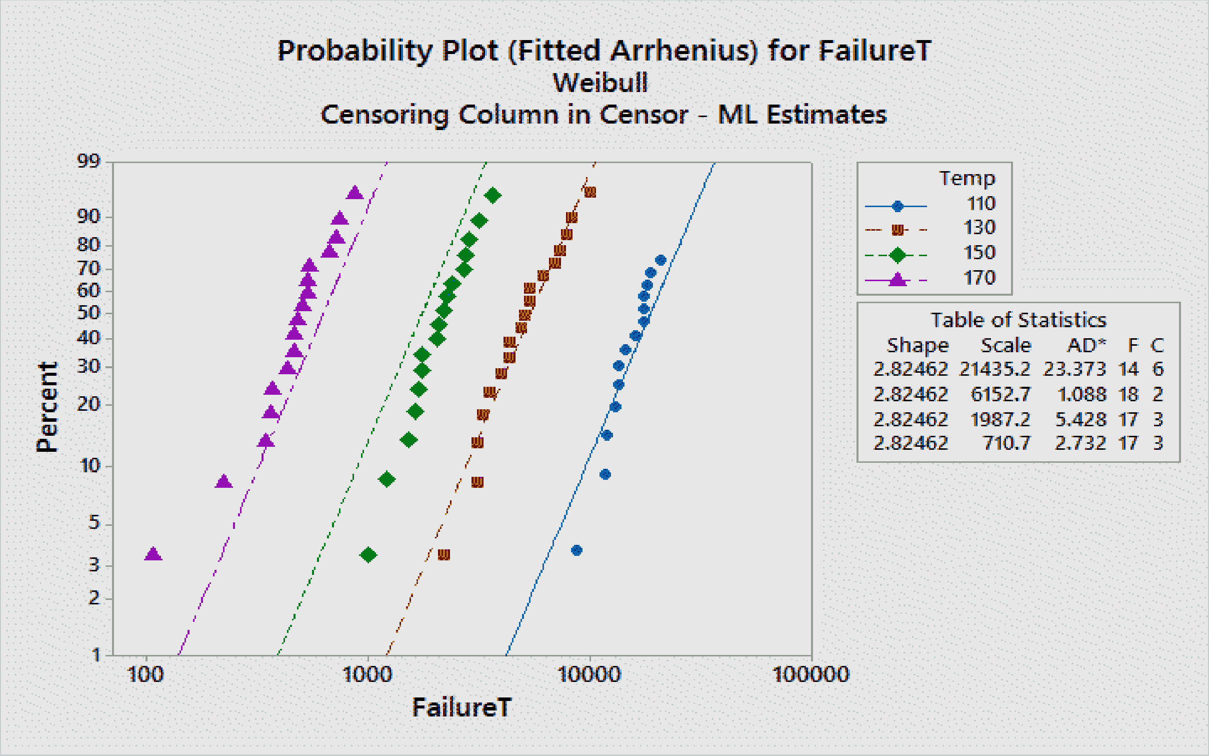


Table of Statistics

Shape	Scale	AD*	F	C
2.82462	21435.2	23.373	14	6
2.82462	6152.7	1.088	18	2
2.82462	1987.2	5.428	17	3
2.82462	710.7	2.732	17	3

Figure 4.11: probability plot for Arrhenius model

4.4.4 Brownian motion considering the temperature

Jackson et al. (2009) gives the expected value of the first hitting time for $\mu > 0$, $\sigma = 1$ and $X(0) = 0$, i.e., MTTF

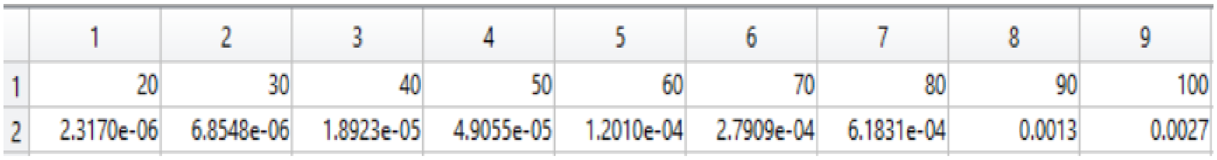
$$E(T_{FL}) = \frac{FL}{\mu}. \tag{4.12}$$

Failure level (FL) is set to be 100 as the default. We will use Equation 4.12 to find the estimated μ instead of using the maximum likelihood estimation. One can use the Weibull regression model, Proportional hazard model or Arrhenius model to find this μ. Nevertheless, this section only choose Arrhenius model to find the drift value μ in the Brownian motion process. Firstly MTTF is calculated under each temperature (from 20 to 100 Celsius). Secondly μ is estimated from Equation 4.12. See the following Matlab code

```
% To find mu under each temperature (model; Brownian motion and Arrhenius model)
x=[20:10:100]; % Define the temperature range
theta=exp(-15.1874+0.8307*11604.83./(x+273.16));
alpha=2.82462;
MTTF=theta*gamma(1/alpha+1);
```

```
FL=100;
mu=FL./MTTF;
Output=[x;mu];
```

The values of μ for 9 temperature points are shown in Figure 4.4.4.



	1	2	3	4	5	6	7	8	9
1	20	30	40	50	60	70	80	90	100
2	2.3170e-06	6.8548e-06	1.8923e-05	4.9055e-05	1.2010e-04	2.7909e-04	6.1831e-04	0.0013	0.0027

Figure 4.12: The μ value under each temperature

If an annular preventer has been operated under the average temperature 60 Celsius for 8760 hours without any failure. Then The degradation level can be estimated as 39. See the following Matlab code

```
mu=0.00012; % Find u for each temperature
sigma=1;
t=1;
NumSteps=8760;
D=zeros(1,NumSteps);
D(1)=0;% Write your starting point of the degradation level under an inspection.
Fail_level=100; % Write your failure level above which the system fails.
NumSimulations=250; % Write the number of simulations.
DL=zeros(1,NumSimulations);
for j=1:NumSimulations
for i=1:NumSteps-1
    D(i+1)=D(i)+normrnd(mu*t,sqrt(t)*sigma);% MAIN PRINCIPLE, to generate a
      % random normally distributed number
end
if D(NumSteps)<0
      DL(j)=0
else
DL(j)=D(NumSteps)
end
end
ave=sum(DL)/NumSimulations;
if ave>Fail_level
```

```
disp(['It fails before ' num2str(NumSteps) 'hours']);
else
disp(['The estimated failuer level is ' num2str(ave) ]);
end
```

In the previous sections, the degradation level is found by the expert judgement during each inspection. This section gives an another alternative. The final degradation level can be determined based on both the observation and the simulation. After the degradation level is determined, the drift value μ for the future is crucial to find important parameters, e.g., RUL, to make maintenance decisions. Unlike the μ in the previous sections, here μ is always changing depending on the temperature. To make the model simple, we assume we know the average temperature for the next 14 days. In reality, it is possible to have a real-time on-line temperature monitoring system to alter the value of μ with the fluctuation of the temperature. The path of the degradation can be simulated continuously. In principle, it is likely to make diagnostics and prognostics. For example, given the degradation level is 39 as obtained previously and the average temperature for the following 14 days is 90 Celsius, the drift value μ is found to be 0.0013. See Figure 4.4.4. Then the toolbox from Chapter 3 can give RUL with standard deviation and the probability to survive till the next maintenance interval as 993.452, 50.47 and 1 respectively.

4.5 Model the degradation in a system level

Usually two annular preventers are installed in a BOP system, i.e., the upper annular and the lower annular. See Figure 4.1. They function together as a 1oo2 system. This 1oo2 system will not fail until both of them fail. Previous sections give degradation models in a component level, with or without the covariates. This section presents degradation models in a system level.

4.5.1 Markov model

Section 3.1 has elaborated Markov model to model a 2oo2 system with degradation levels. To make condition based maintenance decisions, accurate estimation of the mean residual life (the mean value of remaining useful life or mean time to the first DU failure after last inspection), the probability to survive till the next maintenance interval and PFD value is crucial. Markov model is often used. Mean residual life can be found both analytically (by using Laplace transforms, see (Rausand, 2014) page 352) and numerically (by using the toolbox from Section 3.1). The probability to survive till the next maintenance interval can be found by the toolbox as well. So does the PFD value. See Chapter 3. There is no need to make more repetitions here. This section only discusses the question related to "demands with prolonged duration" for a 1oo2 system. Same topic for a 1oo1 system can be found in (Rausand, 2014) page 354. It is assumed that demands occur according to a homogeneous Poisson process with demand rate "λ_{de}". The Markov diagram for two annular preventers with DU failure "internal leakage" and the demand rate "λ_{de}" is Figure 4.13

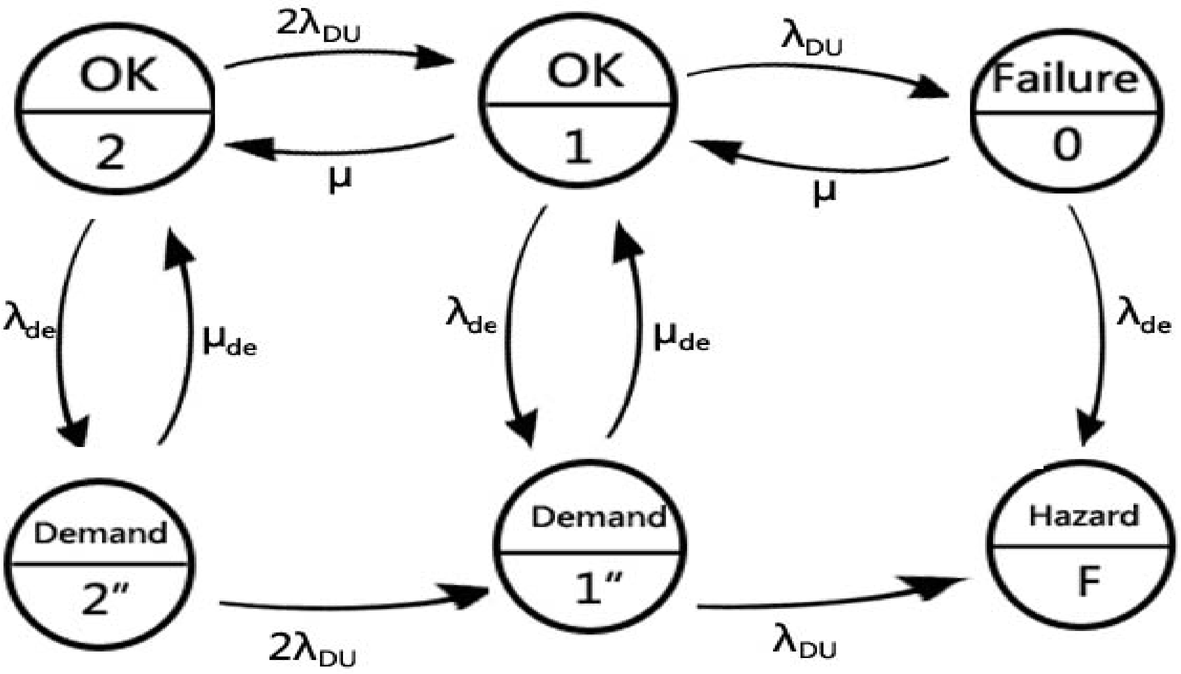


Figure 4.13: Markov transition diagram for a 1oo2 system with the demand rate

State description

2 Fully functioning state

1 The system has one DU failure

0 Two Du failures, unavailable state

2" Demand state: 2 can function when the demand occurs

1" Demand state: 1 can function when the demand occurs

F Hazardous state

λ_{DU} is the failure rate for an annular preventer with DU failure "internal leakage". λ_{de} is the demand rate for an annular preventer to operate. μ is the repair rate. We assume the repair time is same regardless of the number of the failure components. We also assume when a demand occurs, no maintenance task can be executed.

Each time a demand occurs, it has a demand duration with mean value MDD. We assume that the demand duration is exponentially distributed and introduce the demand duration rate μ_{de} = 1/MDD. A hazardous event (HE) can occur in two ways:
1. A demand occurs while the voted group has a fault
2. A dangerous (D) failure occurs while the demand is active (i.e., within the demand duration) (Rausand, 2014).

State F is an absorbing state. For the proof test interval (0, τ), the average PFD is

$$\mathrm{PFD} = \frac{1}{\tau}\int_0^{\tau} P_0(t)\,\mathrm{d}t$$

since the state 0 is a failure state. The frequency of hazardous events (HEF) is the visit frequency to state F. To come to state F, the subsystem must be in either state 0 or state 1" and then makes a “jump”. Thus The average HEF in the proof test interval (0, τ) is

$$\mathrm{HEF} = \frac{1}{\tau}\int_0^{\tau} (P_0(t)\lambda_{de} + P_{1"}(t)\lambda_{\mathrm{DU}})$$

However when the event "HEF" occurs, it does not necessarily lead to a hazardous event. it only refers to both annular preventer fail to operate when a demand occurs. The pipe rams in Lower BOP stack will function as a safety barrier afterwards. It will most likely to prevent the occurrence of the hazardous event, i.e., the blowout. Modelling pipe rams is out of the scope of this Master’s thesis.

The frequency of the demand varies based on various drilling situations. This should be taken into consideration. Condition based monitoring gives an opportunity to alter the value of the demand rate based on the real time drilling condition. Any abnormal phenomena, e.g., the change of the temperature and pressure in a well hole, could lead to a huge difference of the demand rate. Thus, in Figure 4.13, λ_{de} can be altered based on the condition. So does the failure rate λ_{DU}. Previous sections have shown the method to estimate the failure rate under various temperature condition. To sum up, the conventional Markov model has predetermined transition rates. The "Markov model" for CBM should have changable transition rates, including the demand rate and the failure rate. If such model is applied successfully, when making maintenance decisions, HEF is more persuasive than PFD. The last perspective has been always overlooked.

4.5.2 Brownian motion process for a koon system with covariates

Normally two annular preventers are in one BOP system act as a 1oo2 system. As mentioned before, the elastomers in an annular preventers are extremely sensitive to the temperature. And the degradation process can be modelled as a Brownian motion process with a drift in a component level. This section gives a solution to model the degradation process of a 1oo2 system or any koon systems by using Brownian motion processes. The drift value is always changing based on the observation. here, we use "usage" as the only covariate instead of the temperature. The reasons are

1. The temperature for the two annular preventers are almost same which leads to the degradation process similar. However, the usage for two annular preventers are not same. Usually, the upper annular preventer is used more often than the lower annular preventer. This difference leads to total different two degradation process paths.

2. After enquiring a BOP expert Geir-Ove Strand, it is believed that comparing to the temperature, the "usage" is a more significant influence factor for the annualar preventer lifetime. However, aforementioned models can still be used by simply changing temperature values to usage values.

The result form Figure 4.4.4 can be used directly. The temperature value is changed to the times of the use. It is OK to make such change since we use simulated data. The purpose is to illustrate the model. The usage for the upper annular preventer increases faster than the lower annualr preventer. In reality, these numbers can be recorded automatically. The increment of the usage values can increase the drift value μ, which leads to two different degradation paths. The MTTF for this 1oo2 system is modelled as the maximum value of the two hitting times. Generally, for a koon system, MTTF can be modelled as the maximum value of the first n-k+1 hitting times. The Logic diagram of modelling a koon system is below.

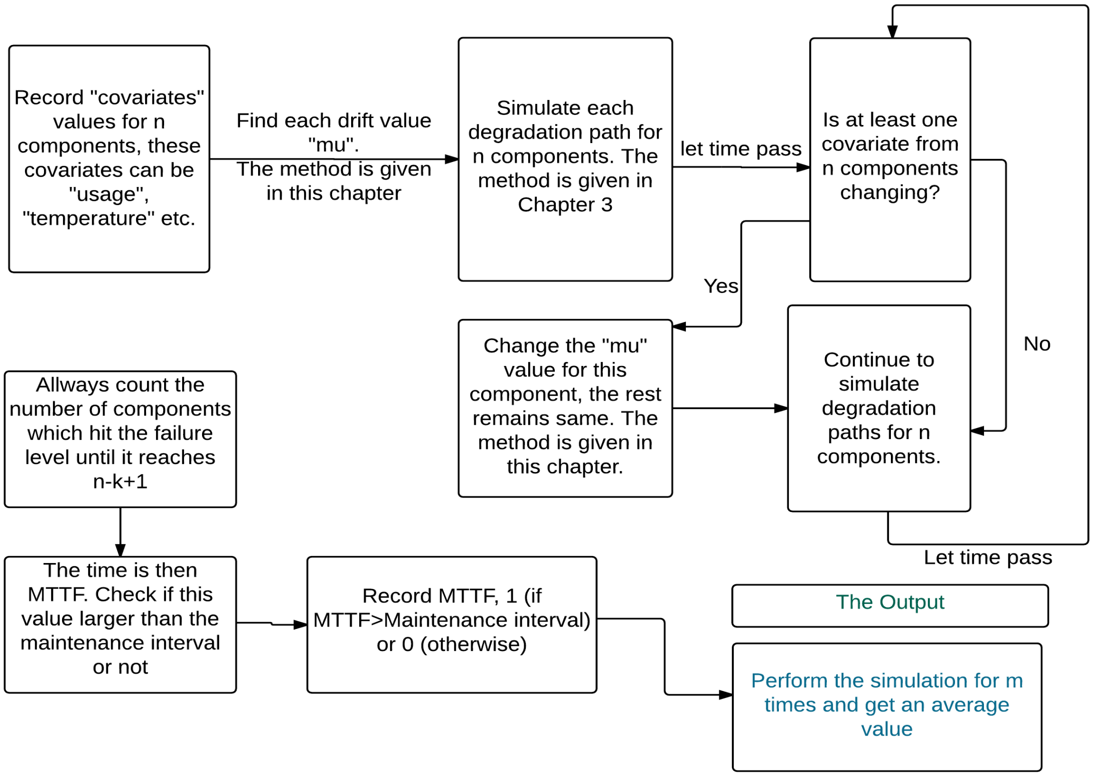

Record "covariates" values for n components, these covariates can be "usage", "temperature" etc.
Find each drift value "mu". The method is given in this chapter
Simulate each degradation path for n components. The method is given in Chapter 3
let time pass
Is at least one covariate from n components changing?
Yes
No
Change the "mu" value for this component, the rest remains same. The method is given in this chapter.
Continue to simulate degradation paths for n components.
Let time pass
Allways count the number of components which hit the failure level until it reaches n-k+1
The time is then MTTF. Check if this value larger than the maintenance interval or not
Record MTTF, 1 (if MTTF>Maintenance interval) or 0 (otherwise)
The Output
Perform the simulation for m times and get an average value

Figure 4.14: Logic diagram for a koon system modelled by Brownian motion with several covariates

4.5.3 Condition based maintenance strategy

Maintenance strategy is easy to make, since most efforts are done before this step. Conventional maintenance strategies, e.g., age based maintenance or block based maintenance is not suitable in our case. This section gives two maintenance policies for CBM.

The previous models usually give two results

- RUL with the standard deviation
- The probability to survive till the next maintenance interval

Foulliaron et al. (2014) presents two maintenance policies which can fit our situation well. However, some changes are made here. We define the RUL of the system at the inspection time t is RUL(t) and the probability to survive till the next maintenance interval is P(t). The maintenance interval is defined as τ

RUL based policy

At the inspection time t for the annular preventers, this policy gives

- if the degradation level is less than the failure level, and RUL(t) $< \tau$, do preventive maintenance.
- if the degradation level is less than the failure level, and RUL(t) $> \tau$, do nothing.
- if the degradation level larger than the failure level, do corrective maintenance

In our case, we can add the standard deviation of RUL as factor for this policy in future work.

Probability based policy

- if the degradation level is less than the failure level, and P(t) $< Q$, do preventive maintenance.
- if the degradation level is less than the failure level, and P(t) $> Q$, do nothing.
- if the degradation level is larger than the failure level, do corrective maintenance.

Here Q is a predetermined value between 0 and 1. However, if we define a high Q value, it can lead to high reliability (low risk) of the system but high preventive maintenance cost, vice versa.

4.6 BOP condition monitoring systems

This section gives some industry applications in BOP condition monitoring systems. All these applications have a same principle which is to monitor BOP systems based on the information collected by sensors. This information can lead to predictable alarm. In addition, those systems can automatically collect the data and transfer to onshore facilities. Onshore personnel can analyze this data and convert it to useful information at any time.

4.6.1 NOV BOP dashboard system

The BOP dashboard, shown in Figure 4.15), aims to simplify complex BOP diagnostics in an easy-to-understand format that facilitates a joint assessment of the issue. In early 2011, BP, Ensco and National Oilwell Varco (NOV) collaborated on a project to consider preliminary development of a BOP dashboard that takes existing alarms, analog data and events from the BOP EWS and translates them into a high-level “traffic light” status. The traffic light logic is based on levels of system redundancy that allow the user to understand when critical functions are impaired. Originally three automated tiers of colors were envisioned to provide the health status of the BOP. “Red” status would mean no functionality, “yellow” status would mean functional but no redundancy, and “green” status would be fully functional and with redundancy (Mckay, 2012).

4.6.2 GE’s drilling ibox system

GE’s drilling ibox system is a combination of hardware and software solution which is used to convert existing data from event logger into reports, status updates (GE). It is a tool to make preventive maintenance and condition based maintenance. The BOP can be monitored in real time from an onshore facility to help with diagnostics and troubleshooting.

4.6.3 Rig watcher

Ashford Technology has developed Rig watcher for real time BOP condition monitoring. It allows personnel to monitor BOP 24/7 from onshore, making maintenance plans and identi-

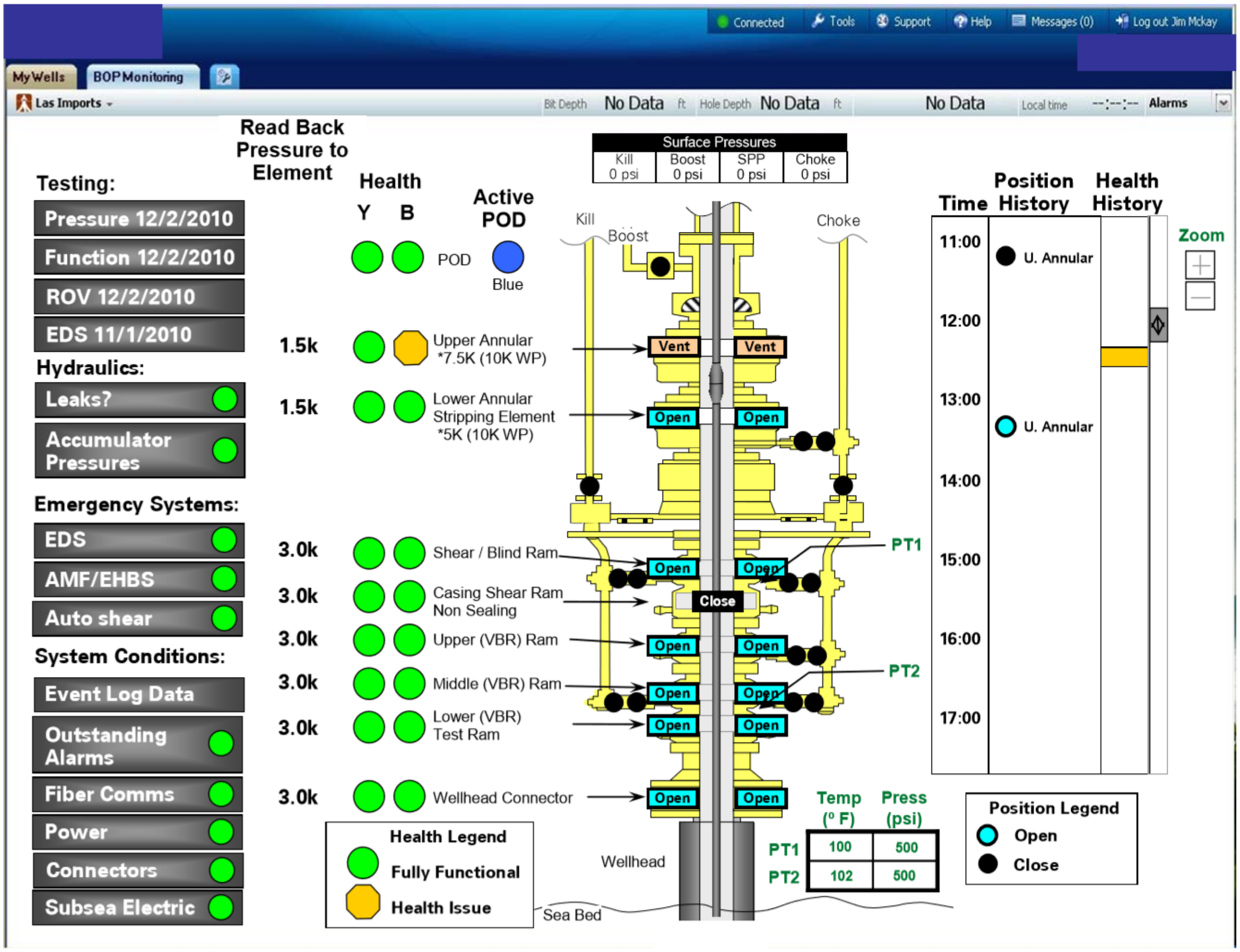


Figure 4.15: BOP dashboard

fying incipient problems. It can view current and historical BOP information anytime, anywhere, and generate accurate valve cycle information for preventive maintenance. See http://www.rigwatcher.com/.

4.6.4 Ram position monitoring

A typical drilling BOP system designed for deep water application is equipped with five to six ram type preventers. These rams are used to shear the drill pipes, to shear the casing, to seal the well, to control and monitor oil and gas in the well. Rams are critical for BOP systems. If the ram position can be monitored accurately, it will enhance the BOP systems reliability and, at the same time reduce the maintenance cost. Ram position monitoring is offered by many

BOP vendors. Many of them provide the exact position of ram in real time. Some representative ram position monitoring systems are GE's Ramtel (GE), EFC ram sensor, NOV's ram position indicator.

Chapter 5

Summary and Recommendations for Further Work

5.1 Summary and Conclusions

This Master's thesis covers almost all aspects of Condition Based Maintenance (CBM). All objectives in Chapter 1 are met. The thesis is mainly comprised of three parts. First part introduces the world of CBM to readers. This part presents data acquisition, data processing and databases, which are the foundation to CBM. Then it highlights models which are divided into physics based models, data-driven models and hybrid models, for diagnostic and prognostic use. Three promising diagnostic and prognostic models are specified, i.e., Markov model, Artificial neural networks and the time-dependent proportional hazard model. Afterwards, CBM main steps are presented in Figure 2.2. This figure is made based on a large quantity of literature review and can function as an index when readers are querying CBM data, diagnostic and prognostic models and steps. It can also give readers a whole picture of CBM. Next, introductions of Prognostic and Health Management (PHM), CBM industry applications and CBM state of the art are followed. Specific challenges, phenomena and questions are summarized.

Second part presents a Matlab toolbox made by the writer. This toolbox estimates components' Remaining Useful Life (RUL) with a standard deviation, the probability to survive till the next maintenance time and Probability of Failure on Demand (PFD) based on numerous simulations. The stochastic processes behind are the (continuous time) Markov model, the Brownian motion process and the Gamma process. Users can choose among them in the toolbox. Specially, MTTF and the steady-state-probability can be achieved by using the Markov model. This toolbox is used often in the next part when data is analyzed. The writer makes the graphic interface of this toolbox easy for people to use. All instructions are given. All code is also attached, from the whole toolbox code to a tiny simulation step with detailed explanations. This toolbox makes it possible for the people with little knowledge in statistics and maintenance to make their own maintenance plans. This toolbox can be download on-line.

The third part of this Master's thesis uses 7 statistic models and 3 stochastic processes to model the degradation process of the elastomeric annular body from the annular preventer of a BOP system. To make these models, many relevant papers and books are studied. In this Master's thesis, these models are not just "theories" or "formulas". Instead, for each model, the writer gives a vivid example by analyzing the data with all Matlab code and detailed explanations following. The writer believe by doing this readers can have a deeper understanding of each model. They may use one of these models for their own data in the future. To make it easier for readers to follow, the difficulty of these models is increasing one after one. The complex model can give a more precise estimation of the lifetime with more influence factors being taken into consideration. As to the structure of the third part, firstly, much literature about BOP is read. A brief description about BOP systems is followed. Virtual failure data is simulated based on a trustful BOP reliability report. The exponential model is firstly used to give a preliminary understanding of the data. Afterwards, the Weibull model, the log-logistic model and the log-normal model are used. All these models use Maximum likelihood Estimation (MLE). Minitab is the analysing software used here. Then, the Brownian motion process is introduced to model the degradation process. Next, the covariates are introduced (e.g., the temperature). The Weibull regression model is elaborated followed by Proportional hazard model (Cox regression model) and Arrhenius model. These three are very promising models used in CBM. Brownian motion is used again to model the degradation. However, this time, the covariates are taken into account. It leads to the change of the path of the Brownian motion process each time when covariates are changing. It is more complex but more realistic. This is the final step to model the degradation in a component level. To model the degradation in a system level, two extra models are included. That is the Markov model and the Brownian motion process for a koon system with covariates. They are shown in the same chapter. Finally, relevant maintenance plans are made based on the result of "RUL" and "the probability to survive till the next maintenance interval.

5.2 Discussion and recommendations

- Change different elastomer types for various temperature range. For example, choosing softer elastomers for lower temperature wells while harder elastomers for higher temperature wells. By applying this, the temperate may not be the most significant influence factor. Instead, the frequency of the usage of the annular preventer may become the most significant influence factor. Because, wear and tear of the elastomer, caused by pipes passing by the BOP system, can lead to the internal leakage under demand. Aforementioned models can still work only by replacing the temperature value to the usage.

- There is an almost infinite combination of drilling fluids, considering the large number of additives and the temperature ranges of operations. It is very difficult and impractical to test all critical elastomers with every drilling fluid combination possible, especially at all the possible temperature ranges. Thus, when a drilling fluid is selected for a program, the operator should ensure that it is compatible with the specific elastomers in the intended BOP. The equipment manufacturers should document the effect of temperature on the elastomers used as sealing components.

- The exponential model, the Weibull model, log-scale-family models and the Brownian motion process can be used to model the degradation process without considering covariates, in a component level. Parameters like RUL, the probability to survive till the next maintenance interval, can be estimated based on the failure data.

- The Weibull regression model, the Proportional hazard (Cox regression) model, Arrhenius models and the Brownian motion process can be used to model the degradation process when considering covariates, in a component level. Parameters like RUL, the probability to survive till the next maintenance interval, can be estimated based on the failure data and the condition monitoring data.

- The Markov model and the Brownian motion process can be used to model the degradation process with covariates, in a system level. Parameters like RUL, the probability to survive till the next maintenance interval, can be estimated.

- Data and models are essential for making accurate diagnosis and prognosis. Many industry applications within CBM concepts can be found. Many condition monitoring systems are built recently in oil & gas industry.

- For accurate machine health diagnostic and prognostic, hidden (semi) Markov model, Artificial neural network and time-dependent proportional hazard model are good choices.

- CBM is a very wide concept. To be a good CBM engineer, one needs to have some knowledge in signal processing, data processing, stochastic processes, statistic models, machine mechanism, data mining and maintenance optimization. To make an analogy, if CBM is a house, then maintenance optimization knowledge is only the roof. That is the last step of CBM. The aforementioned knowledge is the bricks to build that house, and they are essential. It is not wise to only focus on the roof.

- CBM up to the step "diagnostic" is well developed. See Figure 2.2. That is to say in step "raw data acquisition", "feature extraction" and "diagnostic", choosing the correct models and approaches is efficient to make machine fault diagnostic. However, the step "prognostic" has not reached its maturity. Remaining Useful Life (RUL) prediction is always with high uncertainty. Thus, instead of giving an uncertain RUL, CBM contributes more on optimal maintenance strategy. Maintenance suggestions are made based on minimizing cost per time unit. Nevertheless, we have no best tool to validate whether it is the best decision. Especially for dynamic system, conditions are changing over time. To achieve more accurate prognostic, on-line condition monitoring is necessary.

- Remaining useful life (RUL) is the useful life left before equipment failure. Its estimation is a key indicator for the health condition of equipment both in CBM and PHM. In CBM, RUL is usually estimated through various sources (i.e., event data and condition monitoring data). Various methods on estimating RUL have been shown on many publications. However, there is no such best approach which suits universally and reaches most accurate outcome.

- Due to the huge uncertainty in prognostics, there are certain risks in making maintenance decisions based on the prediction of RUL in CBM. Methods for fault diagnostic are numerous and the results always show these methods are trustful. As to prognostics, methods are scarce, and the results show that RUL is usually not so trustful. Usually validations for RUL are undertaken in laboratory only.

- It is important to acquire correct data at correct position with correct measurement.

- What type of Data is needed to make models work? The answer can be found at Figure 2.2. To build these models, historical data is needed. Take ANN and HSMM for example, historical data including event data (historical performance data, e.g., the speed) and condition monitoring data (e.g., vibration) are used to "train" the models. See suggested papers (Tian et al., 2010) and (Wang et al., 2004). And the on-line condition monitoring data are used for diagnostic and prognostic.

5.3 Recommendations for Further Work

- **More rams**

  This Master's thesis only consider annular preventers in a BOP system. Pipe rams and other types of rams should be taken into consideration in future work.

- **Improve the toolbox**

  The self-made toolbox needs more modification work. It can be improved to a new level, in which all models are included and data can be analyzed directly. In this phase, Matlab and Minitab are used to analyze the data.

- **Improve prognostic models**

  There is need to improve prognostic models, i.e., gathering data from sensors, extract features and comparing prognostics models against actual incidents or malfunctions. The development of an equipment degradation prognostic model needs historical data and equipment conditions. Thus a database (e.g., an extended OREDA database) that holds both event data and condition monitoring data can help both build/validate models and make prognosis.

- **Applications of CBM in industry has just started**

  Numerous advanced mathematical models for CBM are proposed so far. Usually these models are validated by laboratory experiment and the results indicate corresponding methods are satisfied. With the help of modern computer, these sophisticated mathematics models can be easily applied. Many steps (except the step "prognostic") of CBM as showed in Figure 2.2 are well established. However, it is believed that the theory/model of CBM has reached its bottleneck and application of CBM in the industry has only started.

- **On-line condition monitoring is the future**

  For complex dynamical system, physics based models can seldom be built with high accuracy, while data driven models are prevalently applied. Vibration analysis is the most effective/popular method for machine condition monitoring both in the laboratory and the industry. However, vibration sensors are usually costly. This makes vibration sensors being installed only at the main components in most cases. Another suggestion for dynamic system is that on-line condition monitoring is extremely important to get an accurate prediction. According to (Jardine et al., 2006): Developing advanced sensor and on-line condition monitoring to continuously monitoring its own health using on-line data acquisition, on-line signal processing and on-line diagnostic tools.

- **Research on Prognostic and health management**

  Methods for diagnostic and prognostic are two different aspects. Papers for diagnostic are vast. Nevertheless, researches on prognostics have not received its prominence compared to the other steps in CBM. The number of papers for prognostic started to increase steadily around year 2005, many of which are under a concept called "Prognostic and Health Management (PHM)". Prognostic is superior to diagnostic. However, outcome of prognostic is not always trustful at this phase.

Appendix A

Acronyms

ANN Artificial Neural Network

CBM Condition Based Maintenance

HMM Hidden Markov Model

HSMM Hidden Semi Markov Model

IFR Increasing Failure Rate

IVHM Integrated Vehicle Health Management

LBOP Lower Blowout Prevent

LMRP Lower Marine Riser Package

MTTF Mean Time to Failure

PFD Probability of Failure on Demand

PHM Prognostic and Health Management

RAMS Reliability, Availability, Maintainability, and Safety

RUL Remaining Useful Life

SHM System Health Management

Appendix B

Additional Information

B.1 Toolbox dashboard

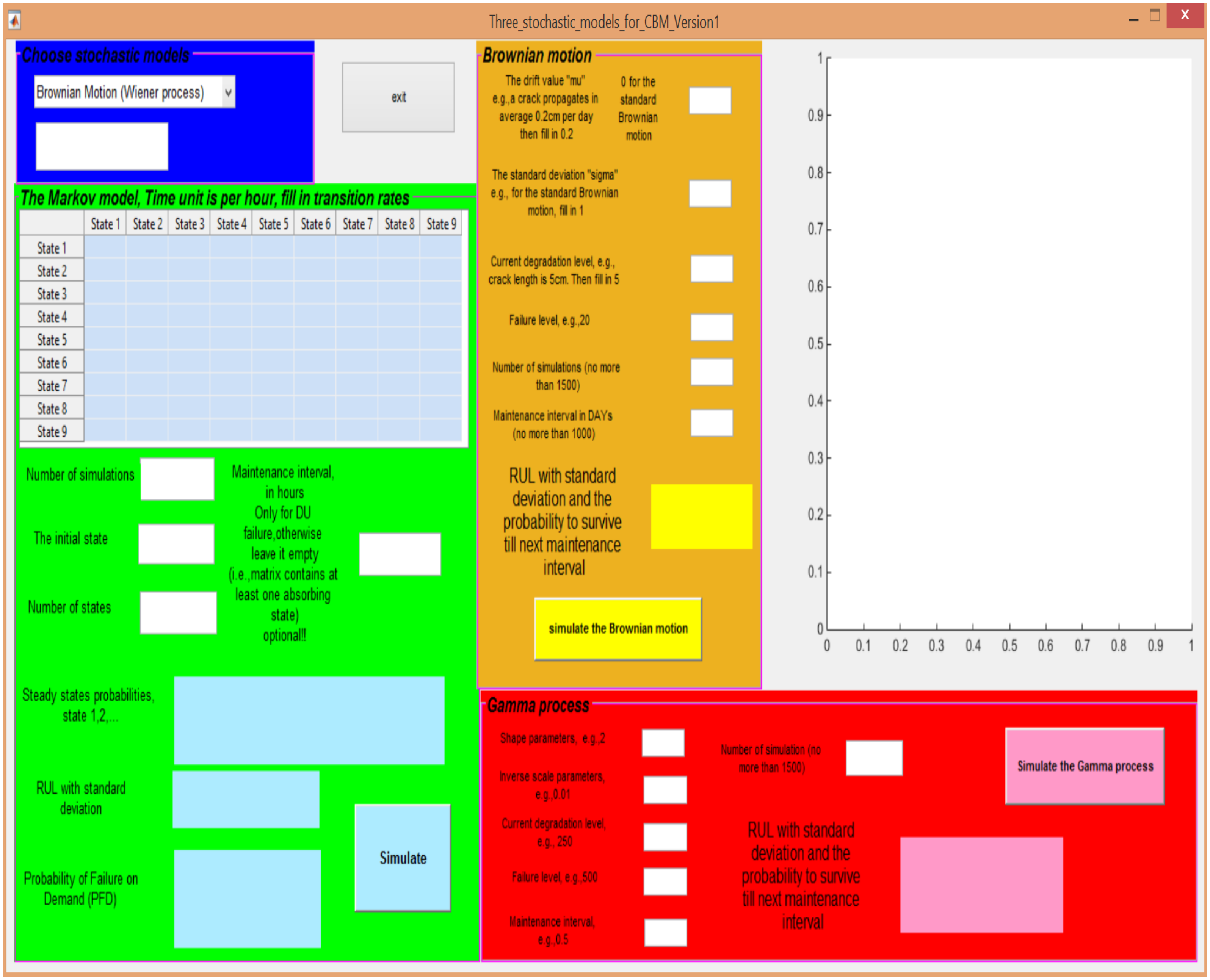

Three_stochastic_models_for_CBM_Version1
Choose stochastic models
Brownian Motion (Wiener process)
exit
The Markov model, Time unit is per hour, fill in transition rates
State 1
State 2
State 3
State 4
State 5
State 6
State 7
State 8
State 9
Number of simulations
The initial state
Number of states
Maintenance interval, in hours Only for DU failure,otherwise leave it empty (i.e.,matrix contains at least one absorbing state) optional!!
Steady states probabilities, state 1,2,...
RUL with standard deviation
Probability of Failure on Demand (PFD)
Simulate
Brownian motion
The drift value "mu" e.g.,a crack propagates in average 0.2cm per day then fill in 0.2
0 for the standard Brownian motion
The standard deviation "sigma" e.g., for the standard Brownian motion, fill in 1
Current degradation level, e.g., crack length is 5cm. Then fill in 5
Failure level, e.g.,20
Number of simulations (no more than 1500)
Maintenance interval in DAYs (no more than 1000)
RUL with standard deviation and the probability to survive till next maintenance interval
simulate the Brownian motion
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Gamma process
Shape parameters, e.g.,2
Inverse scale parameters, e.g.,0.01
Current degradation level, e.g., 250
Failure level, e.g.,500
Maintenance interval, e.g.,0.5
Number of simulation (no more than 1500)
RUL with standard deviation and the probability to survive till next maintenance interval
Simulate the Gamma process

Figure B.1: Toolbox dashboard

B.2 Matlab code for the toolbox

```
function varargout = Brownian_motion_simulation_GUI(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Brownian_motion_simulation_GUI_OpeningFcn, ...
                   'gui_OutputFcn',  @Brownian_motion_simulation_GUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% --- Executes just before Brownian_motion_simulation_GUI is made visible.
function Brownian_motion_simulation_GUI_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;
guidata(hObject, handles);
% --- Outputs from this function are returned to the command line.
function varargout = Brownian_motion_simulation_GUI_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function edit1_Callback(~, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)

```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
mu = str2double(get(handles.edit1,'String'));
sigma = str2double(get(handles.edit2,'String'));
t=1;
NumSteps=1000;
D=zeros(1,NumSteps);
Time=1:NumSteps;
D(1) = str2double(get(handles.edit3,'String'));
Fail_level = str2double(get(handles.edit4,'String'));
NumSimulations=str2double(get(handles.edit6,'String'));
MTTF=1000.*ones(1,NumSimulations);
prop=zeros(1,NumSimulations);
maintenance_interval=str2double(get(handles.edit18,'String'));
for j=1:NumSimulations

for i=1:NumSteps-1
    D(i+1)=D(i)+normrnd(mu*t,sqrt(t)*sigma);
end

if D(NumSteps)>Fail_level
MTTF(j)=find(D==min(D(D>=Fail_level)));%because the time unit here is one
% disp(['It fails in ' num2str(MTTF) 'seconds']);
else
% disp(['It will not fail in ' num2str(NumSteps*t) 'seconds' ]);
end
if D(maintenance_interval)>Fail_level
```

```
    prop(j)=0;
else
    prop(j)=1;
end
prop_to_survive=sum(prop)/NumSimulations;
RUL=sum(MTTF)/NumSimulations;
variance=(sum((MTTF-RUL).^2))/NumSimulations;
SD=sqrt(variance);
% plot(Time,D,'Parent',handles.axes1);
plot(Time,D);
hold on;
end
hold off;
set(handles.text3,'string',[RUL SD prop_to_survive]);

% --- Executes on button press in exit.
function exit_Callback(hObject, eventdata, handles)
% hObject    handle to exit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
delete(handles.figure1);

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

function edit3_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit3 (see GCBO)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)


% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% markov
f=get(handles.uitable2,'data')
dimension=str2num(get(handles.d,'String'));

for i=1:dimension
    for j=1:dimension
      r(i,j)=f(i,j)
    end
end
ll=cell2mat(r); % Your transition matrix here.
% lambdas=f(find(~isnan(f))); % Your transition matrix here.
for d=1:length(ll)
    if sum(ll(d,:))~=0
        ll(d,d)=-1;
    else
    end
end
lambdas=1./ll;
% lambdas(find(lambdas==inf))=0;
numStates = length(lambdas); % Here number of states are specified.
numSimulations = str2num(get(handles.edit7,'String')); % Choose the number of simulation
numSteps = 19000;% Here the chain length is specified.
States = zeros(1,numSteps);
% States(1) = randi(numStates)%;if the initial state is unknow
States(1) = str2num(get(handles.edit8,'String')); % if the initial state is 2
%Changing the following variables are not recommended.
warn=ones(1,numStates)*inf;% If an absorbing state exists, this "warn" vector
% helps to find MTTF
MTTF_vector=zeros(1,numSimulations);
Time = zeros(1,numSteps); %To record time spent in each simulation
sojourn_time= zeros(1,numStates);% Find "virtual" sojourn time
rate = zeros(1,numStates);% Find transition rate for each state.
Count = zeros(1,numStates);
```

```
absorb=sum(ismember(lambdas,warn,'rows')); %This value is to check if there
% is one or more obsorbing states. "obsorb=0" means that there is no
% absorbing state. Otherwise, it means that there is at least one
% absorbing state.
%Case one: no obsorbing states
if absorb==0
    for i = 1:numSteps-1
    rate = lambdas(States(i),:);
    for n=1:numStates
    sojourn_time(n) = exprnd(rate(n));
    end
    m = min(sojourn_time(sojourn_time>0));
    % From state i to i, the transition rate is
    % negative. in such case, function "exprnd" will not give the value.
    % Thus it will not influence the "min" function.
    index = find(sojourn_time == m);% find the next state
    States(i+1) = index;
    Count(States(i)) = Count(States(i))+m; %To sum each sojourn time
    % for the each state
    Time(i+1) = Time(i) + m; % To record the time
    end
plot(Time,States,'x');
prop = Count/Time(end); % To find the steady state probabilities
% disp(sum(prop*lambdas));% This value shall be 0 if it is a good simulation
% disp(['The steady-state probabilities are ' num2str(prop)]);
%   if abs(sum(prop*lambdas))<=2e-16
%      disp('Congratulations!It is a good simulation');
%   else abs(sum(prop*lambdas))>=2e-16
% disp('Opps!It is a bad simulation. Try again or increase value "numSteps"');
%   end
set(handles.text12,'string',num2str(prop));
set(handles.text26,'string',[]);
set(handles.edit15,'string', 'PFD is the steady-state probabilities of failure states');



```

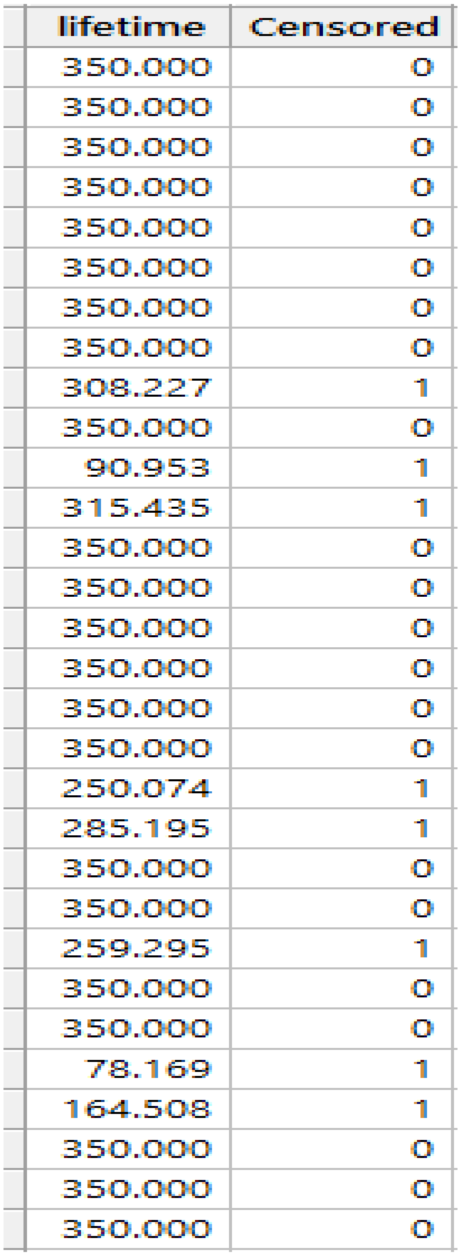
```

%Case two: at least one obsorbing state
else
for j=1:numSimulations
    for i = 1:numSteps-1
    rate = lambdas(States(i),:);
          for n=1:numStates
           sojourn_time(n) = exprnd(rate(n));
          end
       if isequal(sojourn_time,warn)==0
       m = min(sojourn_time(sojourn_time>0));
      % From state i to i, the transition rate is
      % negative. in such case, function "exprnd" will not give the value.
      % Thus it will not influence the "min" function.
      index = find(sojourn_time == m);% find the next state
      States(i+1) = index;
      Count(States(i)) = Count(States(i))+m; %To sum each sojourn time
      % for the each state
      Time(i+1) = Time(i) + m; % To record the time
      else
      MTTF_vector(j)=Time(i);
      break;
      end
    end
end
MTTF=sum(MTTF_vector)/numSimulations;
variance=sum((MTTF_vector-MTTF).^2)/numSimulations;
sd=sqrt(variance);
set(handles.text26,'string', [MTTF; sd]);
set(handles.text12,'string',[])
tao=str2num(get(handles.edit16,'String'));
failuretime=(tao-MTTF_vector)./tao;
failuretime(find(failuretime<0))=0;
pfd=sum(failuretime)/numSimulations;
set(handles.edit15,'string', num2str(pfd));
% disp('An absorbing state is found!');
% disp(['MTTF is ' num2str(MTTF)]);
end

% set(handles.text3,'string',[ff])

function edit7_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
     set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)

function edit8_CreateFcn(hObject, eventdata, handles)

 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
     set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)

contents = get(handles.popupmenu1,'Value');

 switch contents
         case 1
           set(handles.text14, 'String', 'Go yellow');
         case 2
           set(handles.text14, 'String', 'Go green');
         case 3
           set(handles.text14, 'String', 'Go red');
         otherwise
```

```
        delete(handles.figure1);
 end

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
t=0.01; % t=1;% MTTF changes when the delta t changes to 0.1 or 0.01..very odd why??
%ok I find it out: t needs to be small enough for example 0.001 and
%numSteps needs to be large enough
  A = str2double(get(handles.edit9,'String'));
  B = 1/str2double(get(handles.edit10,'String'));
  numSteps=3000;
  y=zeros(1,numSteps);
  y(1)=str2double(get(handles.edit11,'String'));
  time=zeros(1,numSteps);
  numSimulation=str2double(get(handles.edit12,'String'));
  Failure_level=str2double(get(handles.edit13,'String'));
  maintenance_in=str2double(get(handles.edit19,'String'));
  MTTF=ones(1,numSimulation).*(numSteps*t);
  prop=ones(1,numSimulation);
  for j=1:numSimulation

  for i=1:numSteps-1
      y(i+1)=y(i)+gamrnd(A*t,B);
      time(i+1)=time(i)+t;
  end

  if y(numSteps)>Failure_level
  first_time=find(y>Failure_level,1,'first');
  MTTF(j)=first_time*t;
    if first_time*t < maintenance_in
      prop(j)=0;
    else
    end
  else
  end
  plot(time,y);
  hold on;
  end
  hold off;
  Mttf=sum(MTTF)/numSimulation;
  variance=sum((MTTF-Mttf).^2)/numSimulation;
  sd=sqrt(variance);
  survive=sum(prop)/numSimulation;
  set(handles.edit14, 'string', num2str([Mttf sd survive]));

function edit9_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)

```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit11_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit12_Callback(hObject, eventdata, handles)

function edit12_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit13_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit13 as text
%        str2double(get(hObject,'String')) returns contents of edit13 as a double
% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
function edit15_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes when entered data in editable cell(s) in uitable2.
function uitable2_CellEditCallback(hObject, eventdata, handles)
% --- Executes on button press in d.
function d_Callback(hObject, eventdata, handles)
% set(handles.uitable2,'String', NaN)
function edit16_Callback(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit16 as text
%        str2double(get(hObject,'String')) returns contents of edit16 as a double
% --- Executes during object creation, after setting all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes during object creation, after setting all properties.
function d_CreateFcn(hObject, eventdata, handles)
% hObject    handle to d (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function edit18_Callback(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit18 as text
%        str2double(get(hObject,'String')) returns contents of edit18 as a double
% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function edit19_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function edit19_CreateFcn(hObject, eventdata, handles)
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

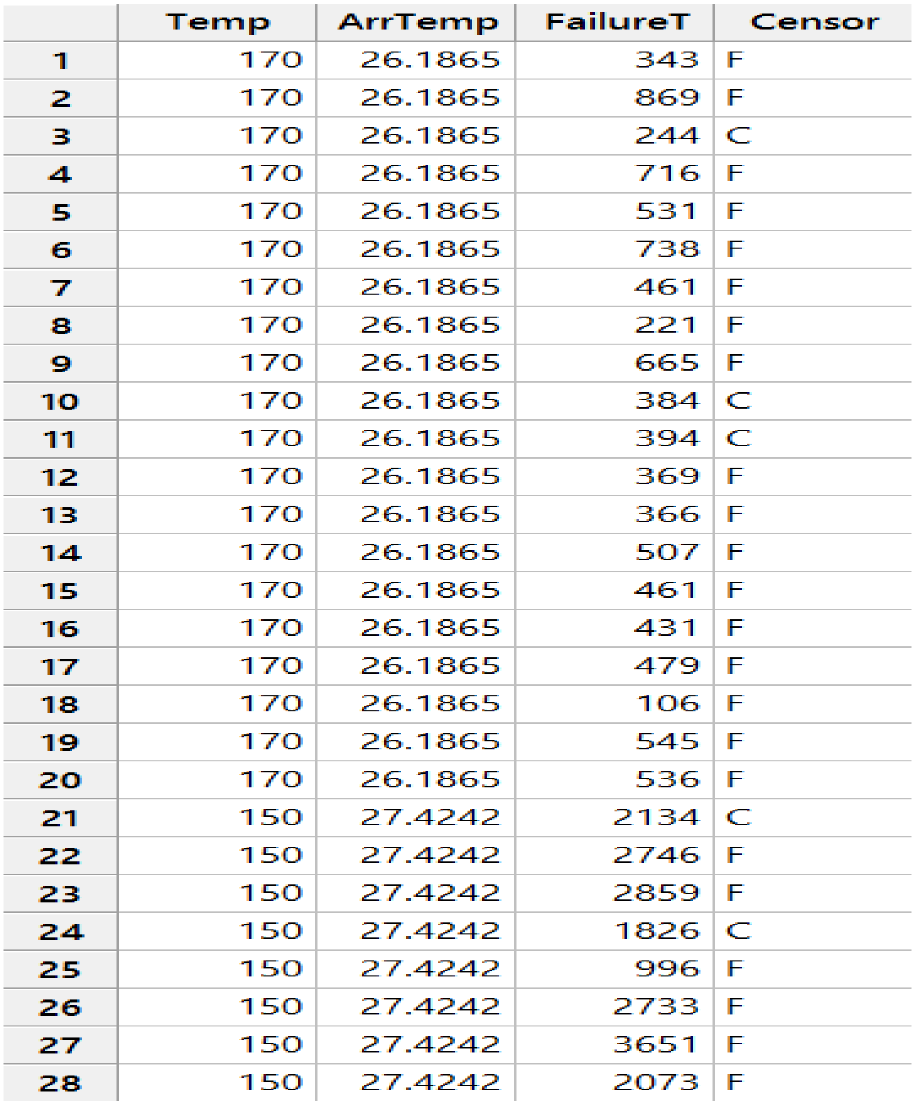
B.3 Simulated lifetime for the annular preventer with the failure mode "internal leakage" without considering the temperature



lifetime	Censored
350.000	0
350.000	0
350.000	0
350.000	0
350.000	0
350.000	0
350.000	0
350.000	0
308.227	1
350.000	0
90.953	1
315.435	1
350.000	0
350.000	0
350.000	0
350.000	0
350.000	0
350.000	0
250.074	1
285.195	1
350.000	0
350.000	0
259.295	1
350.000	0
350.000	0
78.169	1
164.508	1
350.000	0
350.000	0
350.000	0

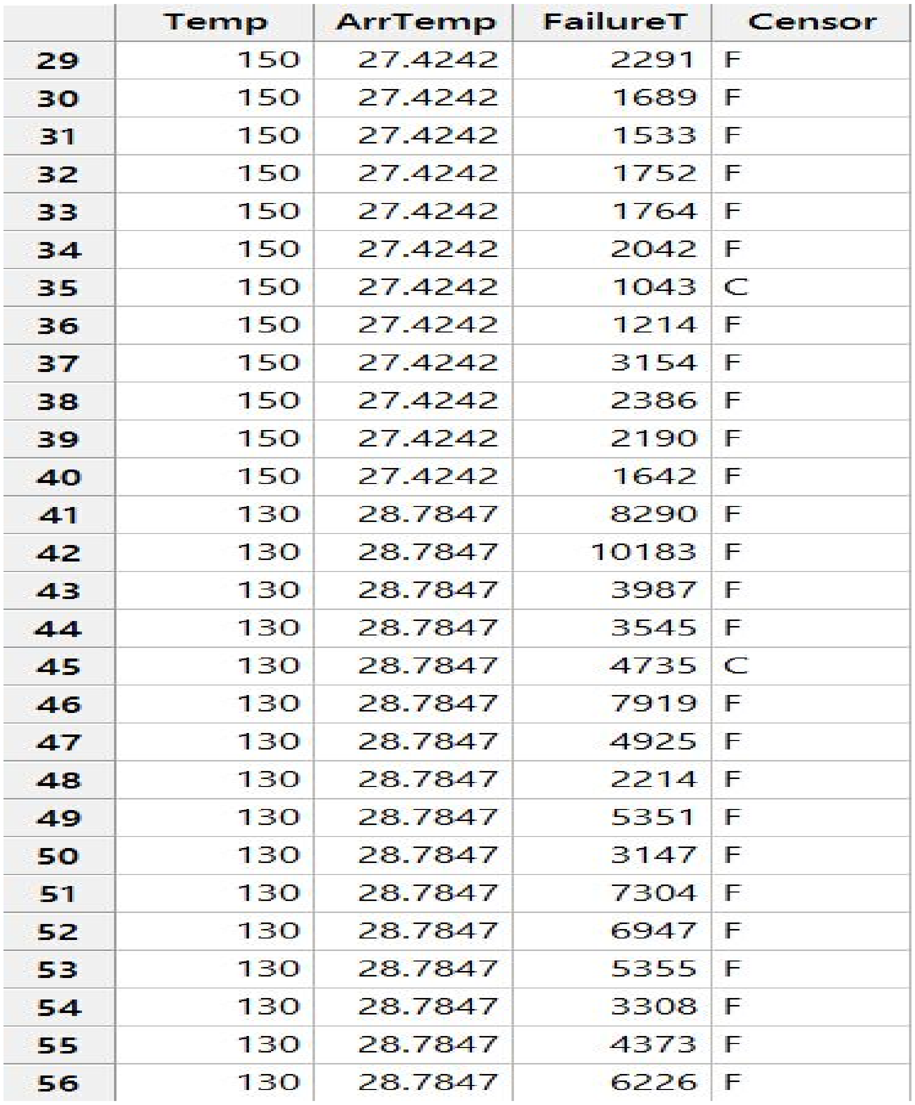
Figure B.2: Simulated lifetime for the annular preventer with the failure mode "internal leakage"

B.4 Lifetime for the annular preventer with the failure mode "internal leakage" considering the temperature



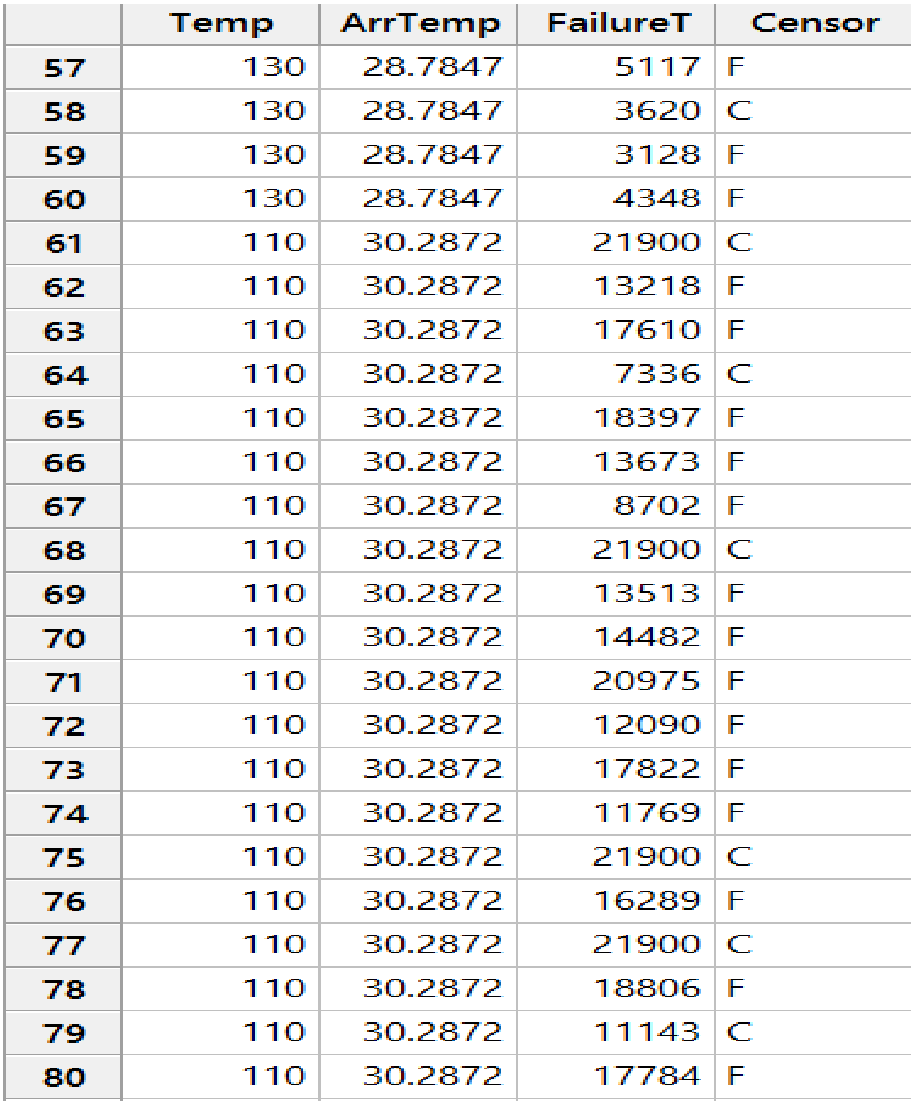
	Temp	ArrTemp	FailureT	Censor
1	170	26.1865	343	F
2	170	26.1865	869	F
3	170	26.1865	244	C
4	170	26.1865	716	F
5	170	26.1865	531	F
6	170	26.1865	738	F
7	170	26.1865	461	F
8	170	26.1865	221	F
9	170	26.1865	665	F
10	170	26.1865	384	C
11	170	26.1865	394	C
12	170	26.1865	369	F
13	170	26.1865	366	F
14	170	26.1865	507	F
15	170	26.1865	461	F
16	170	26.1865	431	F
17	170	26.1865	479	F
18	170	26.1865	106	F
19	170	26.1865	545	F
20	170	26.1865	536	F
21	150	27.4242	2134	C
22	150	27.4242	2746	F
23	150	27.4242	2859	F
24	150	27.4242	1826	C
25	150	27.4242	996	F
26	150	27.4242	2733	F
27	150	27.4242	3651	F
28	150	27.4242	2073	F

Figure B.3: Lifetime for the annular preventer with the failure mode "internal leakage" part 1



	Temp	ArrTemp	FailureT	Censor
29	150	27.4242	2291	F
30	150	27.4242	1689	F
31	150	27.4242	1533	F
32	150	27.4242	1752	F
33	150	27.4242	1764	F
34	150	27.4242	2042	F
35	150	27.4242	1043	C
36	150	27.4242	1214	F
37	150	27.4242	3154	F
38	150	27.4242	2386	F
39	150	27.4242	2190	F
40	150	27.4242	1642	F
41	130	28.7847	8290	F
42	130	28.7847	10183	F
43	130	28.7847	3987	F
44	130	28.7847	3545	F
45	130	28.7847	4735	C
46	130	28.7847	7919	F
47	130	28.7847	4925	F
48	130	28.7847	2214	F
49	130	28.7847	5351	F
50	130	28.7847	3147	F
51	130	28.7847	7304	F
52	130	28.7847	6947	F
53	130	28.7847	5355	F
54	130	28.7847	3308	F
55	130	28.7847	4373	F
56	130	28.7847	6226	F

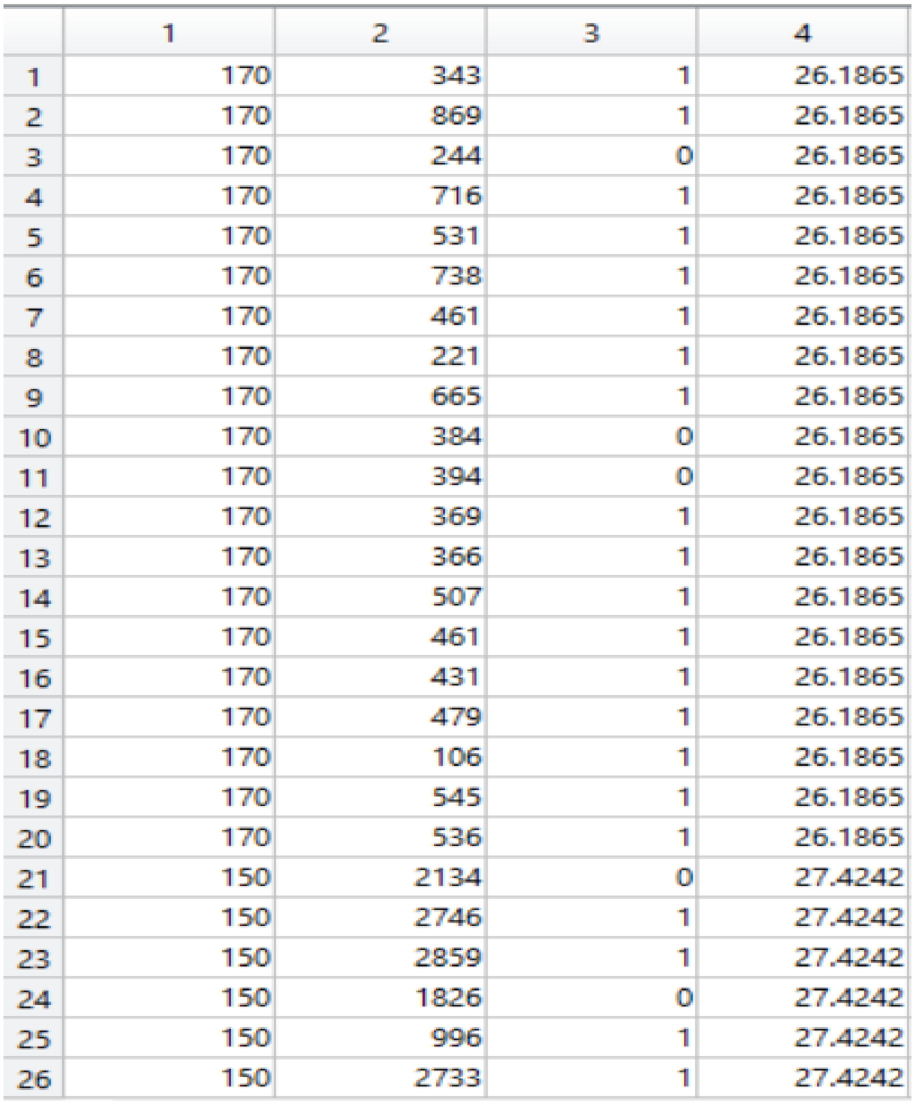
Figure B.4: Lifetime for the annular preventer with the failure mode "internal leakage" part 2



	Temp	ArrTemp	FailureT	Censor
57	130	28.7847	5117	F
58	130	28.7847	3620	C
59	130	28.7847	3128	F
60	130	28.7847	4348	F
61	110	30.2872	21900	C
62	110	30.2872	13218	F
63	110	30.2872	17610	F
64	110	30.2872	7336	C
65	110	30.2872	18397	F
66	110	30.2872	13673	F
67	110	30.2872	8702	F
68	110	30.2872	21900	C
69	110	30.2872	13513	F
70	110	30.2872	14482	F
71	110	30.2872	20975	F
72	110	30.2872	12090	F
73	110	30.2872	17822	F
74	110	30.2872	11769	F
75	110	30.2872	21900	C
76	110	30.2872	16289	F
77	110	30.2872	21900	C
78	110	30.2872	18806	F
79	110	30.2872	11143	C
80	110	30.2872	17784	F

Figure B.5: Lifetime for the annular preventer with the failure mode "internal leakage" part 3

This data set can also be found in Minitab data set library with the name "Insulate.MTW". See http://support.minitab.com/en-us/datasets/.

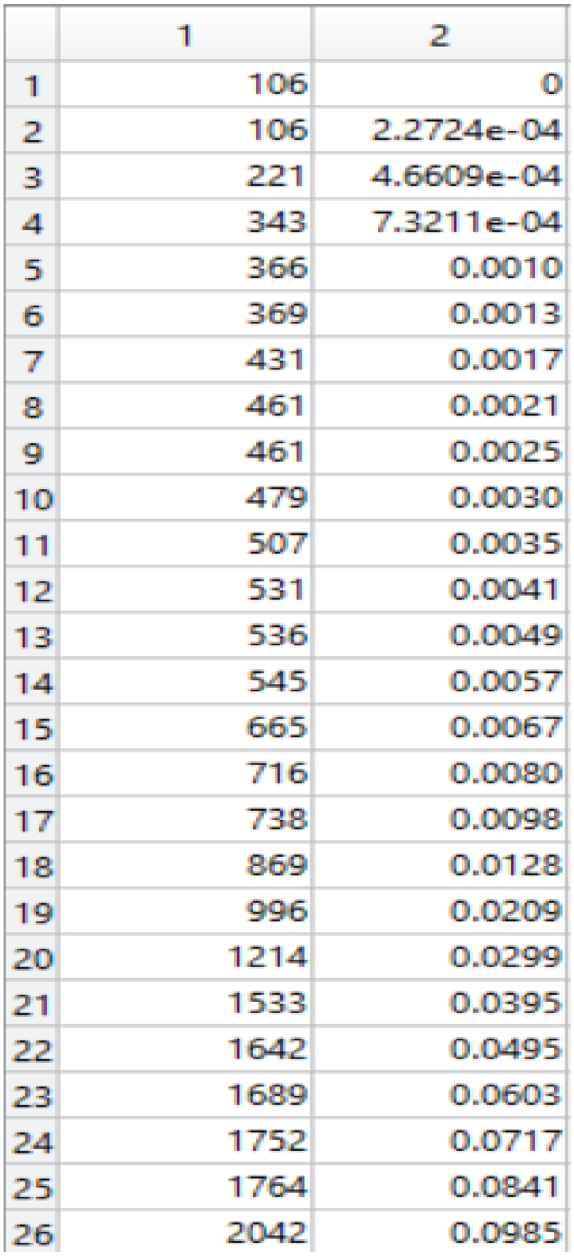


	1	2	3	4
1	170	343	1	26.1865
2	170	869	1	26.1865
3	170	244	0	26.1865
4	170	716	1	26.1865
5	170	531	1	26.1865
6	170	738	1	26.1865
7	170	461	1	26.1865
8	170	221	1	26.1865
9	170	665	1	26.1865
10	170	384	0	26.1865
11	170	394	0	26.1865
12	170	369	1	26.1865
13	170	366	1	26.1865
14	170	507	1	26.1865
15	170	461	1	26.1865
16	170	431	1	26.1865
17	170	479	1	26.1865
18	170	106	1	26.1865
19	170	545	1	26.1865
20	170	536	1	26.1865
21	150	2134	0	27.4242
22	150	2746	1	27.4242
23	150	2859	1	27.4242
24	150	1826	0	27.4242
25	150	996	1	27.4242
26	150	2733	1	27.4242

Figure B.6: Matrix "Total" used in Proportional hazard model

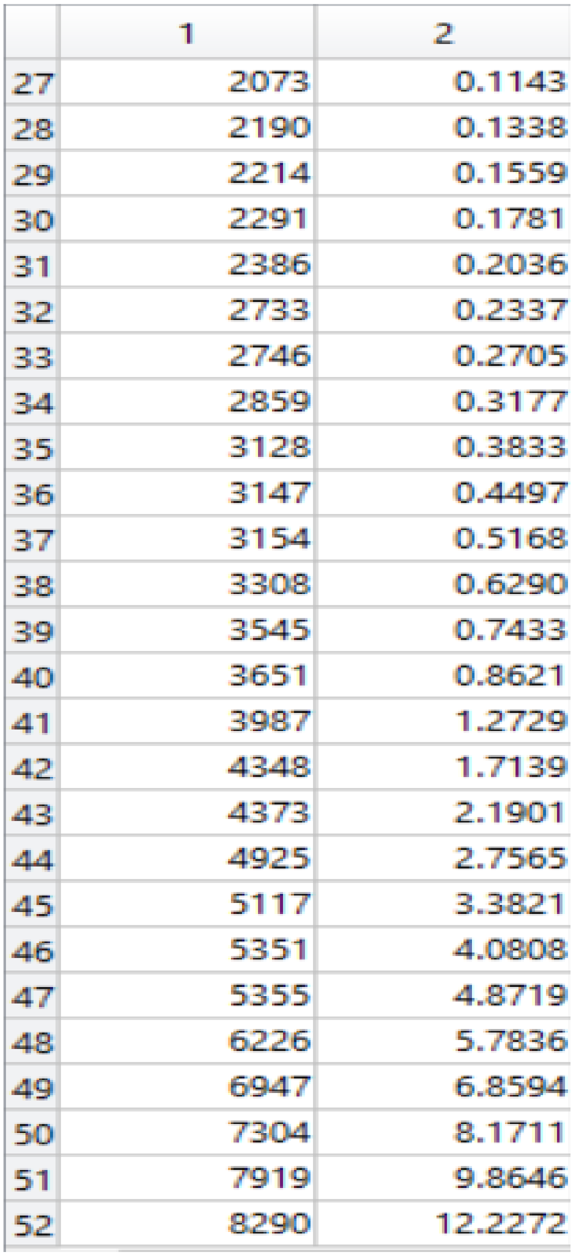
The values of entries in Matrix "Total" represent the temperature, the lifetime and censored value, which are the same as shown before. However, the sequence of the column is changed. Thus one screen shot of Matrix "Total" is given here.

B.5 Cumulative baseline hazard rate from Proportional hazard model (H)



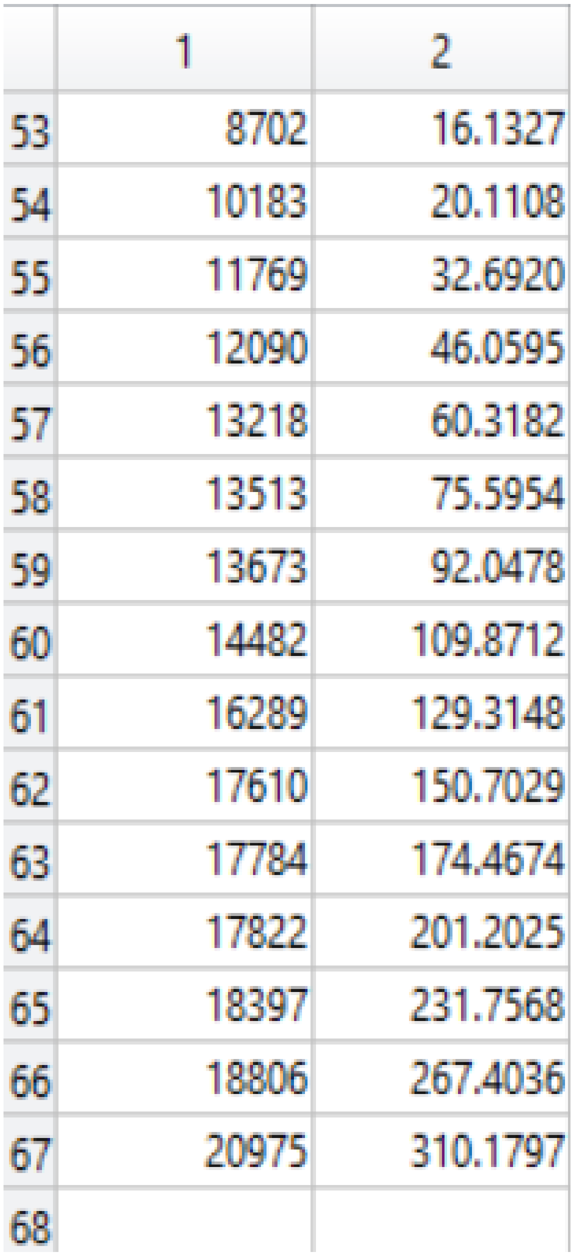
	1	2
1	106	0
2	106	2.2724e-04
3	221	4.6609e-04
4	343	7.3211e-04
5	366	0.0010
6	369	0.0013
7	431	0.0017
8	461	0.0021
9	461	0.0025
10	479	0.0030
11	507	0.0035
12	531	0.0041
13	536	0.0049
14	545	0.0057
15	665	0.0067
16	716	0.0080
17	738	0.0098
18	869	0.0128
19	996	0.0209
20	1214	0.0299
21	1533	0.0395
22	1642	0.0495
23	1689	0.0603
24	1752	0.0717
25	1764	0.0841
26	2042	0.0985

Figure B.7: Cumulative baseline hazard rate from Proportional hazard model (H) part 1



	1	2
27	2073	0.1143
28	2190	0.1338
29	2214	0.1559
30	2291	0.1781
31	2386	0.2036
32	2733	0.2337
33	2746	0.2705
34	2859	0.3177
35	3128	0.3833
36	3147	0.4497
37	3154	0.5168
38	3308	0.6290
39	3545	0.7433
40	3651	0.8621
41	3987	1.2729
42	4348	1.7139
43	4373	2.1901
44	4925	2.7565
45	5117	3.3821
46	5351	4.0808
47	5355	4.8719
48	6226	5.7836
49	6947	6.8594
50	7304	8.1711
51	7919	9.8646
52	8290	12.2272

Figure B.8: Cumulative baseline hazard rate from Proportional hazard model (H) part 2



	1	2
53	8702	16.1327
54	10183	20.1108
55	11769	32.6920
56	12090	46.0595
57	13218	60.3182
58	13513	75.5954
59	13673	92.0478
60	14482	109.8712
61	16289	129.3148
62	17610	150.7029
63	17784	174.4674
64	17822	201.2025
65	18397	231.7568
66	18806	267.4036
67	20975	310.1797
68		

Figure B.9: Cumulative baseline hazard rate from Proportional hazard model (H) part 3

Column 1 is the time at which failures are observed. Column 2 contains the value of the cumulative baseline hazard rate at the failure time in Column 1.

B.6 Academic research center focusing on maintenance

The following table contains academic research center focusing on maintenance and universities with adequate papers regarding to CBM or relevant topics. Table B.1 gives six renowned research centers focusing on CBM.

Center for Intelligent Maintenance System (IMS): This center focuses on technologies in embedded and remote monitoring, prognostics technologies, and intelligent decision support tools (Watchdog Agent® Toolbox).
The Centre for Maintenance Optimization and Reliability Engineering: This center focuses in the areas of condition-based maintenance, spares management, protective devices, mathematical modeling, statistical analysis, software for CBM applications, and failure-finding intervals. This center is within the Department of Mechanical and Industrial Engineering at the University of Toronto.
NASA prognostic Center of Excellence (PCoE): This center works on prognostic technology within the application areas of aeronautics and space exploration
Center for advanced Life Cycle engineering, University of Maryland
The Integrated Vehicle Health Management (IVHM) Centre, Cranfield University, UK
Department of Engineering and Design ,University of Sussex, UK

Table B.1: research centers focusing on CBM

B.7 CBM key words

Key words used for searching for papers within CBM concepts:
Condition Based Maintenance (CBM), Prognostic and Health Management (PHM), System Health Management (SHM), Integrated Vehicle Health Mangement (IVHM) and Engine Health Management (EHM)

Key words for CBM steps and and outputs:
Data acquisition, data processing, extract features, signal processing, diagnostic/diagnosis, prognostic/prognosis, pattern recognition, remaining useful life (RUL), condition monitoring, E-maintenance, etc.

Key words for CBM data:
Waveform data, vibration data, acoustic emission, motor current, stator current, value type data, oil analysis data, event data, velocity, strain, voltage, etc.

Key words for CBM data processing:
Time-domain analysis, time synchronous average, frequency-domain analysis, spectrum analysis, fast Fourier transform, time-frequency analysis, wavelet transform, principal component analysis, trend analysis, etc.

Key words for CBM models:
Markov, hidden Markov model, Petri net, artificial neural network, proportional hazard model, fuzzy logic system, expert system, support vector machine, auto-regression model, etc.

Bibliography

https://www.geoilandgas.com/drilling/offshore-drilling/ramteltm-plus.

Baruah, P. and Chinnam, R. (2005). Hmms for diagnostics and prognostics in machining processes. *International Journal of Production Research*, 43(6):1275–1293. cited By 106.

Bunks, C., McCarthy, D., and Al-Ani, T. (2000). Condition-based maintenance of machines using hidden markov models. *Mechanical Systems and Signal Processing*, 14(4):597–612. cited By 126.

Cox, D. R.; Oakes, D. (1987). Analysis of survival data. *Chapman and Hall, London – New York*.

Dong, M. and He, D. (2007). A segmental hidden semi-markov model (hsmm)-based diagnostics and prognostics framework and methodology. *Mechanical Systems and Signal Processing*, 21(5):2248–2266. cited By 111.

Foulliaron, J., Bouillaut, L., Aknin, P., Barros, A., and Rozas, R. (2014). A prognostic algorithm based on probabilistic graphical models for a periodically observable system. pages 849–857. cited By 0.

Heng, A., Zhang, S., Tan, A., and Mathew, J. (2009). Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing*, 23(3):724–739. cited By (since 1996)0.

Holand, P. (1999). Reliability of subsea bop systems for deepwater application, phase ii dw unrestricted version. *SINTEF report*.

Huang, R., Xi, L., Li, X., Richard Liu, C., Qiu, H., and Lee, J. (2007). Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods. *Mechanical Systems and Signal Processing*, 21(1):193–207. cited By 173.

I.A.E.A (2007). Implementation strategies and tools for condition based maintenance at nuclear power plants. *IAEA,TECDOC*. cited By 33.

Jackson, K., Kreinin, A., and Zhang, W. (2009). Randomization in the first hitting time problem. *Statistics and Probability Letters*, 79(23):2422–2428. cited By 8.

Jardine, A., Lin, D., and Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7):1483–1510. cited By (since 1996)910.

Kenny, M. (2004). *Assessment of BOP stack sequencing, monitoring and kick detection technology*. Bureau of Safety and Environmental Enforcement, Houston, Texas, 2nd edition.

Lawless, J. F. (2002). *Statistical Models and Methods for Lifetime Data*. Wiley, Hoboken, NJ.

Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., and Siegel, D. (2014). Prognostics and health management design for rotary machinery systems - reviews, methodology and applications. *Mechanical Systems and Signal Processing*, 42(1-2):314–334. cited By (since 1996)0.

Li, B., Chow, M.-Y., Tipsuwan, Y., and Hung, J. (2000). Neural-network-based motor rolling bearing fault diagnosis. *IEEE Transactions on Industrial Electronics*, 47(5):1060–1069. cited By 269.

Li, Z., Wu, Z., He, Y., and Fulei, C. (2005). Hidden markov model-based fault diagnostics method in speed-up and speed-down process for rotating machinery. *Mechanical Systems and Signal Processing*, 19(2):329–339. cited By 64.

Lin, D., Banjevic, D., and Jardine, A. (2006). Using principal components in a proportional hazards model with applications in condition-based maintenance. *Journal of the Operational Research Society*, 57(8):910–919. cited By (since 1996)28.

Lin, D., Wiseman, M., Banjevic, D., and Jardine, A. (2004). An approach to signal processing and condition-based maintenance for gearboxes subject to tooth failure. *Mechanical Systems and Signal Processing*, 18(5):993–1007. cited By (since 1996)54.

Loutas, T., Roulias, D., Pauly, E., and Kostopoulos, V. (2011). The combined use of vibration, acoustic emission and oil debris on-line monitoring towards a more effective condition monitoring of rotating machinery. *Mechanical Systems and Signal Processing*, 25(4):1339–1352. cited By 33.

Mahamad, A., Saon, S., and Hiyama, T. (2010). Predicting remaining useful life of rotating machinery based artificial neural network. *Computers and Mathematics with Applications*, 60(4):1078–1087. cited By 17.

Mathworks. Mathworks documentation. http://se.mathworks.com/help/stats/coxphfit.html.

Mckay, J. E., S. C. H. T. S. G. D. M. . P. A. L. (2012). Blowout preventer (bop) health monitoring. *Society of Petroleum Engineers*.

Medjaher, K., Tobon-Mejia, D., and Zerhouni, N. (2012). Remaining useful life estimation of critical components with application to bearings. *IEEE Transactions on Reliability*, 61(2):292–302. cited By 24.

Minitab. Data set library. http://support.minitab.com/en-us/datasets/.

Peng, Y. and Dong, M. (2011). A prognosis method using age-dependent hidden semi-markov model for equipment health prediction. *Mechanical Systems and Signal Processing*, 25(1):237–252. cited By 27.

Rafiee, J., Arvani, F., Harifi, A., and Sadeghi, M. (2007). Intelligent condition monitoring of a gearbox using artificial neural network. *Mechanical Systems and Signal Processing*, 21(4):1746–1754. cited By 88.

Rausand, M. (2014). *Reiability of Safety-Critical Systems: Theory and Applications*. Wiley, Hoboken, NJ.

Rausand, M. and Høyland, A. (2004). *System Reliability Theory: Models, Statistical Methods, and Applications*. Wiley, Hoboken, NJ, 2nd edition.

Ross, S. (2014). *Introduction to Probability Models: Eleventh Edition*. cited By (since 1996)1.

Samanta, B. and Al-Balushi, K. (2003). Artificial neural network based fault diagnostics of rolling element bearings using time-domain features. *Mechanical Systems and Signal Processing*, 17(2):317–328. cited By 236.

Si, X.-S., Wang, W., Hu, C.-H., and Zhou, D.-H. (2011). Remaining useful life estimation - a review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1):1–14. cited By (since 1996)0.

Sun, Y., Ma, L., Mathew, J., Wang, W., and Zhang, S. (2006). Mechanical systems hazard estimation using condition monitoring. *Mechanical Systems and Signal Processing*, 20(5):1189–1201. cited By (since 1996)53.

Tian, Z., Wong, L., and Safaei, N. (2010). A neural network approach for remaining useful life prediction utilizing both failure and suspension histories. *Mechanical Systems and Signal Processing*, 24(5):1542–1555. cited By 35.

U.S. Chemical Safety And Hazard Investigation Board (2010). Explosion and fire at the macondo well. *Society of Petroleum Engineers*.

Wang, W., Golnaraghi, M., and Ismail, F. (2004). Prognosis of machine health condition using neuro-fuzzy systems. *Mechanical Systems and Signal Processing*, 18(4):813–831. cited By 159.

Woo, C., Choi, S., Lee, S., and Kim, H. (2010). Useful lifetime prediction of rubber components using accelerated testing. *IEEE Transactions on Reliability*, 59(1):11–17. cited By 14.

Yang, B.-S., Han, T., and Kim, Y.-S. (2004). Integration of art-kohonen neural network and case-based reasoning for intelligent fault diagnosis. *Expert Systems with Applications*, 26(3):387–395. cited By 73.

Zhang, X., Xu, R., Kwan, C., Liang, S., Xie, Q., and Haynes, L. (2005). An integrated approach to bearing fault diagnostics and prognostics. volume 4, pages 2750–2755. cited By 38.

Curriculum Vitae

Name:	**Zehan Han**
Gender:	male
Date of birth:	24. November 1989
Temporary address:	Sjøveien 12, 303, Sandnes
Nationality:	Chinese
Email (1):	zehanh@stud.ntnu.no
Email (2):	han.zehan.norge@gmail.com
Telephone:	+47 47441734

Language Skills

Chinese; English; Norwegian

Education

- Norwegian University of science and technology, Norway

Computer Skills

- Microsoft Office package
- Matlab
- Minitab

Experience

- SINTEF, Summer Internship, 2014
- NTNU, Teaching Assistant, 2014
- National Oilwell Varco, Condition monitoring analyst, since June 1st, 2015