UNIVERSITY OF STRATHCLYDE

DEPARTMENT OF MANAGEMENT SCIENCE

Incorporating Expert Judgement into Condition Based Maintenance Decision Support Using a Coupled Hidden Markov Model and a Partially Observable Markov Decision Process.

by

Samaneh Balali

Supervised by

Prof. Tim Bedford & Prof. Lesley Walls

A thesis presented in fulfillment of the requirements for the degree of Doctor of Philosophy

Submitted 2012

Signed: Samaneh Balali Date: 26th Nov. 2012

Acknowledgement

I wish to express my sincere appreciation to my supervisors, Prof. Tim Bedford and Prof. Lesley Walls, for all their support and guidance throughout my PhD study. Without their patience, encouragement and invaluable comments, it would not have been possible to complete this research. I would also like to thank Prof. Ken McNaught, Dr. Kerem Akartunali, and Prof. John Quigley for kindly agreeing to serve as examiners and convener of the examining committee.

Special thanks are due to Prof. Isaac Meilijson for the illuminating conversation about Hidden Markov Models, and for giving me invaluable advice on formulation of the Coupled Hidden Markov Model. I am also grateful to Jim Gaw for the cooperation and helpful discussions on the maintenance system operated in the engineering company.

I would like to acknowledge my friends and colleagues in the Department of Management Science for making the last five years of my life a truly joyful experience. In particular, I wish to express my gratitude to Dr. Farhad Shafti for his continued encouragement, to Alison Kerr and Caroline Sisi for their unlimited support, and to Bernadine Williams for her endless love.

I will be forever thankful to my brother, Hossein, my mother, Mehry, and my father, Reza, for being so supportive from so far away. Without their unconditional love and encouragement this thesis could not have been completed.

Abstract

Preventive maintenance consists of activities performed to maintain a system in a satisfactory functional condition. Condition Based Maintenance (CBM) aims to reduce the cost of preventive maintenance by supporting decisions on performing maintenance actions, based on information reflecting a system's health condition. In practice, the condition related information can be obtained in various ways, including continuous condition monitoring performed by sensors, or subjective assessment performed by humans. An experienced engineer might provide such subjective assessment by visually inspecting a system, or by interpreting the data collected by condition monitoring devices, and hence give an "expert judgement" on the state of the system. There is limited academic literature on the development of CBM models incorporating expert judgement. This research aims to reduce this gap by developing models that formally incorporate expert judgement into the CBM decision process.

A Coupled Hidden Markov Model is proposed to model the evolutionary relationship between expert judgement and the true deterioration state of a system. This model is used to estimate the underlying condition of the system and predict the remaining time to failure. A training algorithm is developed to support model parameter estimation. The algorithm's performance is evaluated with respect to the number of expert judgements and initial settings of model parameters.

A decision-making problem is formulated to account for the use of expert judgement in selecting maintenance actions in light of the physical investigation of the system's condition. A Partially Observable Markov Decision Process is proposed to recommend the most cost-effective decisions on inspection choice and maintenance action in two consecutive steps. An approximate method is developed to solve the proposed decision optimisation model and obtain the optimal policy. The sensitivity of the optimal policy is evaluated with respect to model parameters settings, such as the accuracy of the expert judgement.

Thesis Contents

List of Figures

List of Tables

Notation Table

The table below lists the notations that are introduced and used throughout the thesis.

Coupled Hidden Markov Model	
Notation	***Meaning***
N	Number of the deterioration states of a system
$y_t \in \{1,2,...,N\}$	Expert judgement state at time t
$x_t \in \{1,2,...,N\}$	Deterioration state at time t
p_i	Probability of the system making a self-transition in State i
q_k^i	Probability of the expert judgement remaining in State k, given that the system is in State i
$a_{ik,jl}$	Transition probability from the joint state $(x_t = i, y_t = k)$ to the joint state $(x_{t+1} = j, y_{t+1} = l)$
$V = \{kS, kF\}$	Set of observation symbols; kS means that the expert judgement is in State k and the system is still working, and kF means that the expert judgement is in State k and the system has failed
F_i	Probability of instant failure given that the system is in State i
$b_{ik}(u_t)$	Probability of observing $u_t \in V$, given that the system is in the joint State $(x_t = i, y_t = k)$
$\lambda = \{p_i, q_k^i, F_i\}$	Complete set of parameters used to represent the proposed Coupled Hidden Markov Model

Probabilities related to observation sequences	
Notation	***Meaning***
$P(U\|\lambda)$	Probability of the observation sequence $U = u_1, u_2, ..., u_T$ given the model λ
$\alpha_t(ik)$	Joint probability of the observation sequence up to time t and the coupled states $(x_t = i, y_t = k)$
$\beta_t(ik)$	Probability of the partial observation sequence from time $t+1$ given the coupled states $(x_t = i, y_t = k)$
$\xi_t(ik, jl)$	Probability of the system being in the joint state $(x_t = i, y_t = k)$ at time t and in the joint state $(x_{t+1} = j, y_{t+1} = l)$ at time $t+1$, given the observation sequence U and model λ
$\gamma_t(ik)$	Probability of being in the joint state $(x_t = i, y_t = k)$ at time t, given the observation sequence U and model λ
$f_i(t)$	Probability distribution of time to failure given that the system is in State i
Decision model	
Notation	***Meaning***
Δ	Time duration of a decision interval
K	Total number of decision intervals in a planning horizon
φ	Discount rate

$o_k \in \{1,2,...,Z\}$	Outcome of a simple inspection at decision period k
$b_i(o)$	Probability of observing the inspection outcome o, given that the system is in State i
$B=\left[b_i(o),1\le i\le N,\ 1\le o\le Z\right]$	Observation probability matrix related to the simple inspection
$a^I \in \{0,1,2\}$	Decision variable at decision Step 1, $a^I=0$ denotes taking no action; $a^I=1$ denotes conducting a simple inspection; and $a^I=2$ denotes performing an accurate inspection
$a^M \in \{0,1,2\}$	Decision variable at decision Step 2, $a^M=0$ denotes taking no action; $a^M=1$ denotes performing an imperfect maintenance action; and $a^M=2$ denotes carrying out a preventive replacement
C^S,C^A,C^M,C^R,C^F	Immediate cost of conducting a simple inspection, an accurate inspection, an imperfect maintenance action, a preventive replacement, and a failure, respectively
Maintained system evolution	
Notation	***Meaning***
$\mathrm{P}=\left[p_{ij}=P\left(x_{t+1}=j \middle\| x_t=i\right)\right]$	Probability transition matrix of system's deterioration states over one discrete time unit
q^i_{lm}	Probability of the expert judgement to be in State m at the beginning of decision period $k+1$, given that the expert judgement is in State l at decision period k and the system is in State i at the beginning of decision period $k+1$

$Q(i)=\left[q_{lm}^{i},\ 1\le i,l,m\le N\right]$	Expert judgement state transition probability matrix, over one decision period, given that the system is in State i
x_k'	Deterioration state of a system updated upon performing an imperfect maintenance action at decision period k
y_k'	Expert judgement state updated upon performing an imperfect maintenance action at decision period k
r_{ij}	Probability of restoring the system from State i to State j by an imperfect maintenance action
$\pi^k=\left(\pi_1^k,\pi_2^k,\ldots,\pi_N^k\right)$	Conditional probability distribution of system's deterioration state at decision period k, with π_i^k denoting the probability of the system being in State i at decision period k
H^k	History of information available at the beginning of decision period k
$\pi_i^k\left(y_k\right)$	Belief state or the conditional probability of the system being in State i at period k, given the expert judgement state y_k and all the information available at the beginning of this decision period
$\pi_i^k\left(o_k,y_k\right)$	Probability of the system being in State i at period k, given the expert judgement state y_k and the simple inspection outcome o_k at period k and all other information available at the beginning of this decision period
$\pi_i^k\left(SM\right)$	Probability of the system to be in State i at period k upon a simple inspection followed by an

	imperfect maintenance action
$\pi_i^k(AM)$	Probability of the system to be in State i at period k upon an accurate inspection followed by an imperfect maintenance action
Survival probabilities	
Notation	***Meaning***
$R(k,i,t)$	Probability of the system to survive for at least t time units, given that it is in State i at the beginning of decision period k
$\overline{R}(k,\pi^k,\Delta)$	Probability of the system to survive at the beginning of decision period $k+1$, given the belief state at decision period k
Cost functions	
Notation	***Meaning***
$V(k,\pi^k)$	Minimum expected total discounted cost incurred over $K-k$ periods given the belief state π^k at period k
$W^I(k,\pi^k(y_k))$	Expected total discounted cost if it is chosen to not take any action at period k and make optimal decisions afterwards
$S(k,\pi^k(y_k))$	Expected total discounted cost if it is chosen to conduct a simple inspection at period k and make optimal decisions afterwards
$A(k,\pi^k(y_k))$	Expected total discounted cost if it is chosen to conduct an accurate inspection at period k and make optimal decisions afterwards
$V^S(k,\pi^k(y_k,o_k))$	Minimum expected total discounted cost incurred

	over $K-k$ periods given the belief state $\pi^k(y_k, o_k)$ at period k
$W^S(k, \pi^k(y_k, o_k))$	Expected total discounted cost if it is chosen to not perform any maintenance action after a simple inspection at decision period k and make optimal decisions afterwards
$M^S(k, \pi^k(y_k, o_k))$	Expected total discounted cost if an imperfect maintenance action is carried out after conducting a simple inspection at decision period k and optimal decisions are made afterwards
$r(k)$	Expected total discounted cost if, upon an inspection, a preventive replacement is carried out at decision period k and optimal decisions are made afterwards
$V^A(k, x_k)$	Minimum expected total discounted cost incurred over $K-k$ periods if an accurate inspection is conducted at period k with the outcome x_k
$W^A(k, x_k)$	Expected total discounted cost if, upon an accurate inspection it is chosen to not perform any maintenance action at period k, and make optimal decisions afterwards
$M^A(k, x_k)$	Expected total discounted cost if, upon an accurate inspection an imperfect maintenance action is carried out at decision period k, and optimal decisions are made afterwards

1 Introduction

The goal of this research is to develop a modelling framework that formally incorporates expert judgement into the Condition Based Maintenance (CBM) decision process. The industrial motivation for this research is grounded in the case of CBM in a large engineering company operating fans. The gap in scientific knowledge is established through a review of the existing literature on CBM and the relevant model classes. This chapter describes the context of the research, introduces key definitions and scopes the research objectives. An overview of the remainder of the thesis is presented at the end of this chapter.

1.1 Introduction to Condition Based Maintenance

Maintenance is defined as "the combination of all technical and associated administrative actions intended to retain an item or system in, or restore it to, a state in which it can perform its required function" (EN 13306, 2001). The terms "retain it in" and "restore it to" in this definition imply the broad classification of maintenance types into "preventive" and "corrective". Preventive maintenance can be further classified according to the approaches of scheduling and performing the maintenance actions as "Time Based" and "Condition Based" maintenance, and "direct condition monitoring" and "indirect condition monitoring" as seen in Figure 1.1.

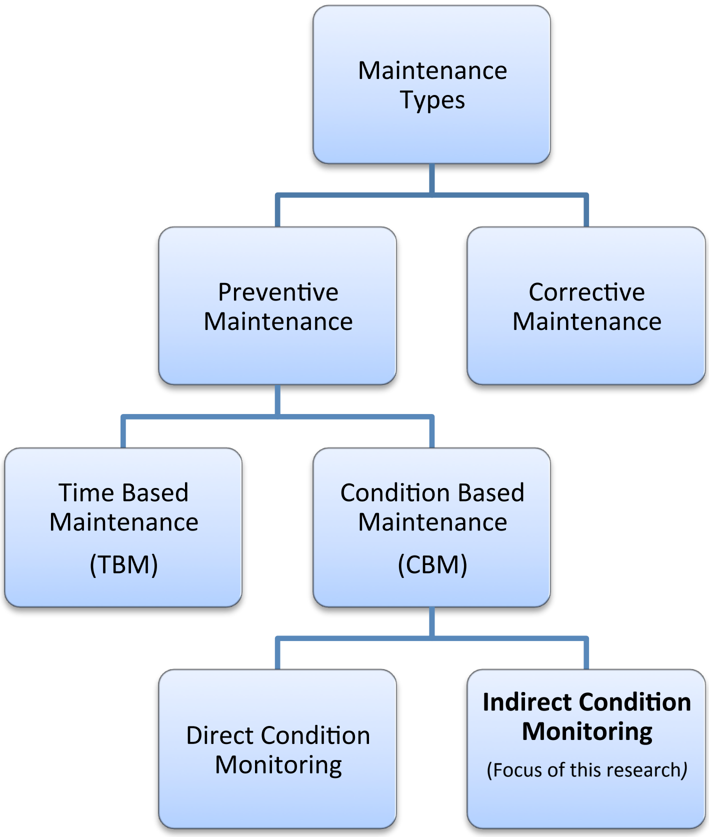

Figure 1.1: A Classification of the maintenance types.

In corrective maintenance, the actions are performed after the occurrence of a failure to restore a system to a condition where it can carry out its required function (Knezevic 1987; Saranga and Knezevic 2000).

In contrast to corrective maintenance, preventive maintenance consists of activities performed before a failure occurs. Preventive maintenance actions are utilised to maintain a system in a satisfactory functional condition, or prevent a defect from developing into more severe conditions. Preventive maintenance is motivated by the need to avoid the significant economic impacts of loss of system availability, quality or safety, caused by failure. For example, in manufacturing processes there are critical components whose failure can lead to the breakdown of the whole production line. In addition to loss of production, the failure of some equipment or systems can decrease safety, and hence cause irreparable damages. Therefore, preventive maintenance, whose aim is to repair or replace the components before a failure occurs, is extremely important.

In Time Based Maintenance, maintenance actions are scheduled based on the calendar time, or operational age, such as “cycles” or “cumulative load” (Cooke and Bedford 2002). In practice, a system is exposed to random disruptions such as unexpected changes of operational conditions or work schedules. These random variations cause uncertainty in the system operational characteristics. In Time Based maintenance the changes in the operational characteristics are not taken into account when scheduling maintenance action. This can lead to substantial failure costs, or high maintenance costs caused by a too conservative maintenance policy.

CBM can reduce the cost of preventive maintenance by systematically taking maintenance actions when evidence of abnormal behaviour is observed from the information collected through “monitoring the condition” of a system (Campbell and Jardine 2001).

Systems subject to condition monitoring are classified into two categories: completely observable systems and partially observable systems (Wang and Christer, 2000; Jardine et al., 2006). For completely observable systems, the health condition can be completely identified through condition monitoring. The process through which the actual condition of a system (e.g. depth of a tooth crack in a gear) is observed is called “direct condition monitoring ” (Wang 2008).

Sometimes it is not possible to observe the true condition of a system, i.e. to measure the exact amount of deterioration, during the process. For instance, the health condition of a system might include the conditions of various internal unobservable components. In addition, sensors used to measure the deterioration may give noise-corrupted readings. Finally, the exact deterioration level may be costly to measure. For instance, it may require stopping the system from operating and this can cause a substantial loss of production. In this case some parameters stochastically correlated with the actual health condition of the system are collected, that is “indirect condition monitoring”. Vibration signals, cutting forces in a machining process or temperature are examples of information obtained by “indirect condition monitoring”. The terms “covariate” or “ condition data” have been used in literature to represent variables measured by indirect condition monitoring (Jardine et al. 1998; Wang 2008; Heng et al. 2009). The focus of this research will be on indirect condition monitoring; henceforward we use the term “condition data” to denote the measurements that are related to the health condition of a system subject to indirect condition monitoring.

1.2 Condition Based Maintenance Decision Support

CBM systems and what they typically involve have been discussed by various researchers (Chinnam and Baruah, 2004; Jardine et al., 2006; Thurston, 2001; Wang, 2008). A CBM system, in general, consists of four stages: (1) Data Acquisition; (2) Data Cleaning and Processing; (3) Diagnostics and Prognostics; (4) Decision-Making. A brief discussion of these stages follows in this section.

1.2.1 Data Acquisition

At the first stage, the information relevant to system’s health condition is obtained, that is condition monitoring. The European maintenance terminology standard (EN 13306, 2001) defines monitoring as “activity, performed either manually or automatically, intended to observe the actual state of an item”. As seen in this definition, condition monitoring can be performed “manually” or “automatically”, this includes subjective assessment performed by human, or continuous monitoring performed by autonomous sensors.

1.2.2 Data Cleaning and Processing

Data collected through condition monitoring are typically contaminated with noise caused by changes of environmental condition (e.g. temperature or torque load) and error in data recording (e.g. caused by the sensors collecting the data or by human error entering data manually). After cleaning condition data, to remove noise and error, cleaned data are analysed and transformed into useful information (e.g. some statistical feature values), and that is data processing.

1.2.3 Diagnostics and Prognostics

Diagnostics consists of the tasks performed to indicate whether something is wrong in the monitored system and if so, to determine the nature of the fault; these tasks are referred to as “fault detection” and “fault identification” respectively (Campbell and

Jardine 2001). For example, in a CBM system, vibration signals can be collected and analysed in order to indicate a fault in a piece of equipment (e.g. a gearbox) and identify the nature of the fault (e.g. a gear tooth fracture).

Prognostics, on the other hand, refers to the tasks carried out before a fault or a failure occurs, to estimate how soon it will happen. For example, in the same CBM system mentioned above, the vibration signals could be also used to estimate the likelihood of having a tooth fracture over a specific duration of time. The result of prognostics can be either provided as the expected value or the probability distribution function of the remaining time to failure (Jardine et al. 2006; Peng et al. 2010; Si et al. 2011).

1.2.4 Maintenance Decision-Making

The main purpose of diagnostics and prognostics is to use condition data to provide useful information to support decision-making on performing maintenance actions (e.g. repair or replacement) with the aim to reduce downtime and preventive maintenance costs. This can be accomplished, for example, by incorporating the information obtained from diagnostics and prognostics into an optimisation model. Maintenance optimisation models are defined as "mathematical models whose aims are to find the optimum balance between the costs and benefits of maintenance, while taking all kinds of constraints into account" (Dekker, 1996). CBM optimisation models can be of help in maintenance decision-making, given the information obtained from condition monitoring, in order to "maintain the system in a cost effective way" (Wang 2008).

1.3 Expert Judgement: Experienced Engineer's Assessment of System's Condition

As mentioned in Sub-Section 1.2.1 condition data can be obtained in various ways. This includes subjective assessments that lead to qualitative results. For instance, the deterioration condition of an auto-greaser in a fan can be assessed by visual inspection of the colour of the grease.

The data cleaning and processing stage is especially important when condition data are continuously recorded by autonomous condition monitoring devices such as vibration sensors. In such systems, condition data are first "cleaned" of noise, and then they are interpreted and transformed into useful information about the system's condition – i.e. data processing is carried out. Interpretation of condition data can also be carried out by a subjective assessment performed by an experienced engineer, that is "expert judgement". For example, this assessment may be done for CBM applications where more than one parameter (e.g. vibration and temperature) is measured to monitor a system's condition. Since these parameter values could contradict one another, an expert might carefully examine the recorded data, along with other available information (e.g. previous condition data or visual inspection results) to gain insight into the system's condition. The result of this subjective assessment, i.e. expert judgement, can be used as the basis for taking further maintenance actions (e.g. performing a more accurate inspection).

The condition monitoring and data interpretation methods explained above have been witnessed in a practical CBM implementation that will be described in Chapter 2. Figure 1.2 aims to represent different stages of a CBM decision-support process where expert judgement might play a role in condition monitoring or data processing.

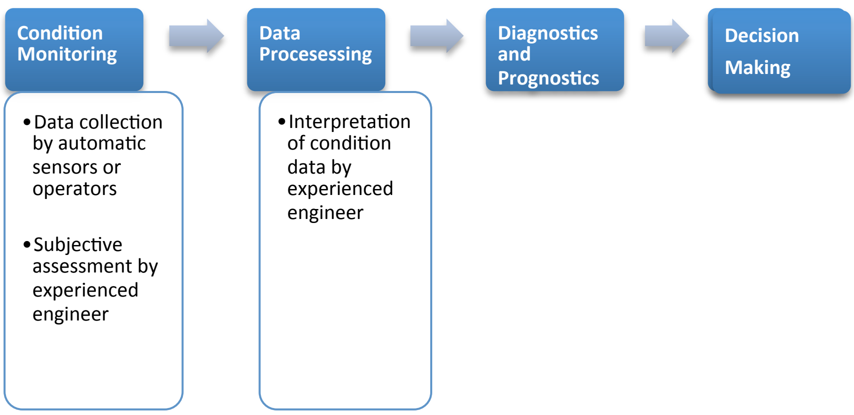

Figure 1.2: Possible roles of expert judgement in the first two stages of a Condition Based Maintenance decision-support process.

Figure 1.2 implies that expert judgement is used for assessing a system's condition and so can contribute to answering questions such as "what is the chance that the system is in a specific health condition?" and "what is the chance of failure within a specific time interval?".

Incorporation of expert judgement in CBM decision support process, as described above, has not been widely addressed in literature. An example is presented by Wang and Zhang (2008) who addressed the incorporation of expert judgement in prognostic modelling. They assumed that expert judgement is provided based on the current condition data and is an indirect assessment of the residual life of a system. Given the possible roles of expert engineering judgement in an industrial practice, and the lack of coverage in the academic literature, we aim to explore the development of models with expert judgement in a CBM context.

1.4 Research Aims and Objectives

The overall aim of this research is to develop a modelling framework to support CBM decision-making by formally incorporating expert judgement together with other relevant data about the deterioration condition of a system. Diagnostics, prognostics and further decisions on maintenance actions are to be supported. To achieve this aim we require both a stochastic model of the condition of the system and a means of making optimal decisions based on a cost-benefit analysis of maintenance policies. This leads us to state the following initial objectives:

1. To develop a stochastic model that captures the evolutionary relationship between expert judgement and the underlying deterioration condition of a system in order to estimate the true deterioration condition and to predict the remaining time to failure;

2. To develop a parameter estimation method for the stochastic model and evaluate its performance with respect to potential application issues that might be faced in practice, such as the number of expert judgements;

3. To develop an optimisation model to select cost-effective maintenance policies based on a trade-off between the costs and benefits of alternative maintenance actions, using the diagnostic and prognostic information provided by the stochastic model;

4. To examine the sensitivity of the optimal maintenance policies with respect to changes in system failure rate, cost and the accuracy of the expert judgement.

Based on a review of the literature relating to both CBM and mathematical modelling relevant to this context, as discussed in Chapters 3 and 4, we shall be able to position this research and refine the statement of objectives.

1.5 Thesis Overview

The thesis is organised in 9 chapters as illustrated in Figure 1.3.

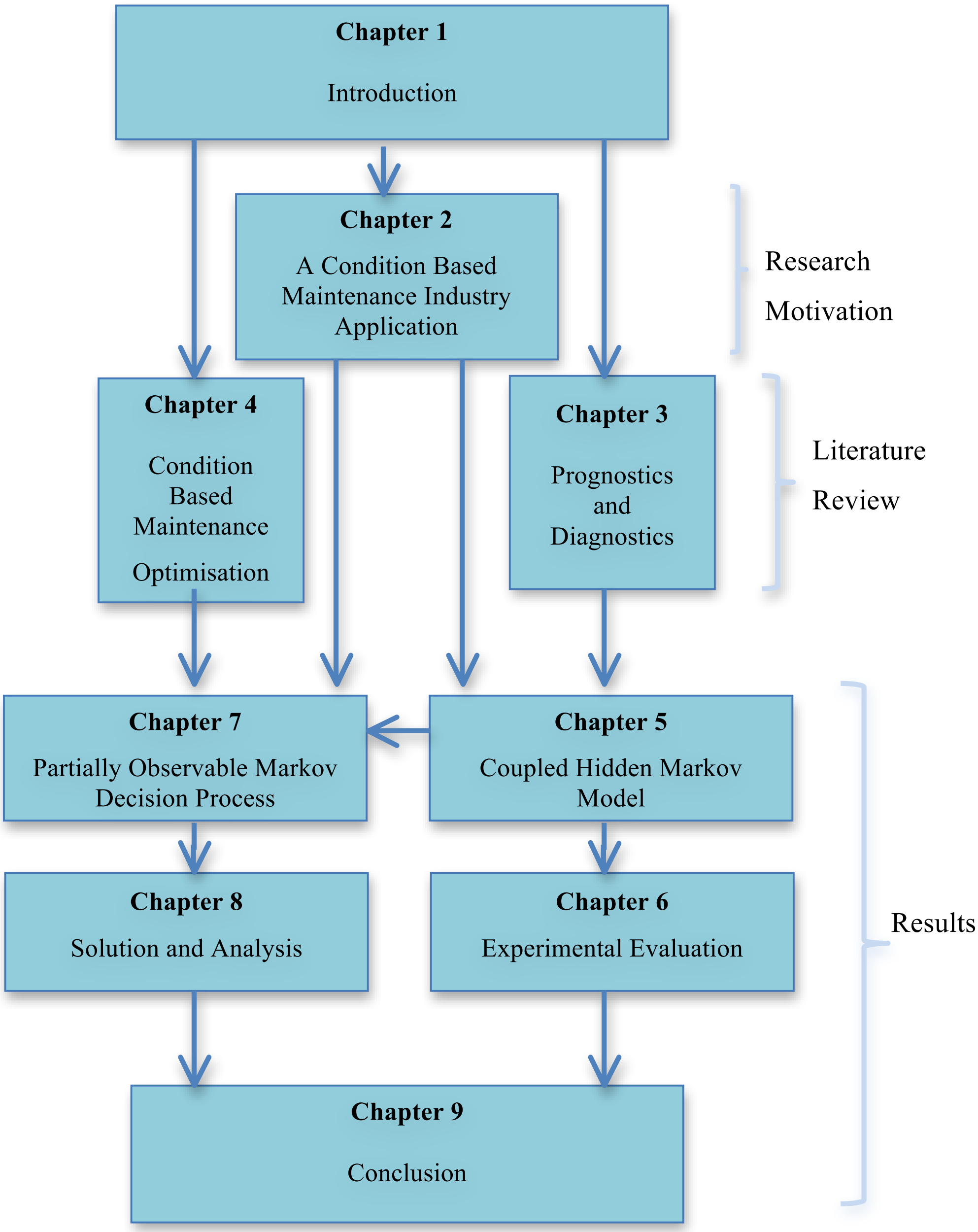

Figure 1.3: Organisation of the thesis chapters. Arrows depict the information flow among the chapters.

An overview of the remainder of the thesis is as follows:

Chapter 2 reports an industrial application that has motivated this research. The maintenance event database and the decision-making process in this CBM application are described and the key observations and possible research directions are summarised. Chapter 3 provides a review of diagnostics and prognostics models in literature. The emphasis is on modelling frameworks for capturing the interaction between condition data and the underlying deterioration condition of a monitored system, and how they are used for diagnostics and prognostics. Chapter 4 clarifies the research gaps and prepares the ground for introducing the specific architecture of Hidden Markov Models, which is established and formulated in Chapter 5. The CBM optimisation models for partially observable systems are reviewed, and the research gaps, which are addressed in Chapter 7, are identified.

In Chapter 5 a Coupled Hidden Markov Model (CHMM) is developed to model the evolution of, and capture the stochastic relationship between, expert judgement and the underlying condition of a system. This model is used as a basis for diagnostics and prognostics to estimate the systems' deterioration state given all expert judgements made to date and predict the remaining time to failure. A training algorithm is developed to estimate the CHMM parameters. In Chapter 6 this training algorithm is demonstrated and evaluated by numerical experiments. The effect of some potential implementation issues (e.g. limited number of training observation sequences) on performance of the algorithm is investigated through experimental sensitivity analysis.

Chapter 7 describes how the intervention of maintenance actions and observations obtained from physical inspections can be incorporated into the model developed in Chapter 5. Motivated from the decision-making mechanism in the CBM application described in Chapter 2, a two-step decision optimisation model is formulated as a Partially Observable Markov Decision Process (POMDP). In Chapter 8 an algorithm is developed to solve the decision optimisation model formulated in Chapter 7. The solution procedure and the optimal policy are demonstrated through numerical experiments. An experimental sensitivity analysis is conducted to study the effect of the influential parameters on the optimal policy. The structural characteristics of the optimal policy, that can ease its implementation in practice, are explored.

Chapter 9 provides a summary of the research to conclude how the research aims have been achieved. Suggestions for future research are discussed.

Details of MATLAB codling are provided in the appendices. Appendix A provides the MATLAB computer codes developed to train the CHMM. Appendix B provides the MATLAB computer codes written to solve the POMDP and illustrate the result.

2 Research Motivation: A Condition Based Maintenance Industry Application

A goal of this research is to align with industry needs so that future tools can be developed from this research. This chapter provides an insight into the Condition Based Maintenance system implemented in a large engineering company operating fans. The main purpose of this chapter is to justify the models developed in this research and, as discussed in Chapter 9, provide direction for future research.

2.1 Description of the Maintenance System of Fans

The large engineering company consists of four independent generating units. In each unit there are two fans, and the failure of one fan will cause a huge loss of generation, which would result in a substantial loss of revenue. The company has established a maintenance strategy in 2005 that includes a "control-limit" CBM policy for these fans. This CBM policy, which is currently implemented at the company, is briefly described in Section 2.1.1.

2.1.1 The Maintenance Policy Implemented at the Company

The failure of bearings has been found to be the predominant cause of failure of the fans over the past few years. An online vibration and temperature monitor is installed on the motor bearings. Data on the amplitude of the vibration and the temperature of the bearings are continuously collected and recorded every second. The maintenance policy is based on alarm levels, referred to as "control-limits" in literature, so that when the vibration or temperature reaches a predetermined level a necessary action is carried out. These alarm levels have been set with a view to provide a managed approach that maximises the available time to decide the best course of action and also to minimise the loss of generation and damage to the fans.

The thresholds and the corresponding actions of the policy are summarised as follows:

Expected operating range: No action is taken while the vibration level is within this range.

Suspicious level: The engineers are called to verify the situation and perform additional inspections.

Alarm level: A short notice is given to reduce the unit load and the engineers perform the necessary operations regarding the condition verification as soon as possible.

Action level: Operations are performed for taking mills out of service and preparing to stop fans.

Trip level: Operations are carried out to stop fans immediately.

When the vibration amplitude reaches the 'suspicious level', an experienced engineer is called to verify the situation by performing additional inspections. He will then decide to either leave the equipment and keep it running until the next inspection occasion, plan for load drop to inspect the bearings, or take the fans out of service immediately. The decision on what action to take is based not just on the overall level and temperature, but also on the frequency and the amplitudes of the harmonics of the vibration signals. This processed information regarding the vibration signals is held on a system administrated by a third party.

2.1.2 Maintenance Event Database

We refer to an assessment made by the engineer as an "expert judgement" hereafter. Expert judgements are not recorded directly but the information regarding the maintenance interventions are recorded in a database called "Work Order Cards (WOCs)", which are considered to be very reliable for fans. The WOCs associated with primary air fan bearings of three units for the duration of July 2008 to September 2010 were supplied. WOCs for six fans (two fans per unit for the total of three units) were available in the form of a table comprising a short description of work orders, the work type, the scheduled start and the target completion dates. There were 29 works orders in total related to bearings. An example of a WOC extracted for a primary air fan is given in Table 2.1. For confidentiality reasons, the details of the work orders have been disguised; however, despite the slight changes, they remain coherent with the original information.

The work types recorded in WOCs are categorised as “corrective maintenance”, “emergency maintenance”, “outage” and “minor outage”. The “scheduled start” refers to the date at which the works were planned to be carried out. The time intervals during which the works were planned to be completed are recorded as “the target completion”, which ranges from 24 hours to 1 month. This is the ordered time duration for completing the work from the scheduled start, without taking into account any commissioning or interlock checks. The status of the works have been recorded as “open”, “close” and “cancelled”, where open states that the work is not yet complete and close means the work has been done. Cancelled means the work was not carried out under the WOC reference, although another WOC may have been used instead.

Table 2.1: Overview of the work orders of a fan, for the period 01/07/2008–30/08/2010. The details have been disguised for confidentiality reasons.

Scheduled Start (date)	**Work Type**	**Work Order**	**Target Completion**	**Status/Notes after the work completion**
16/07/2008	CM	Check all PA fan bearings auto greasers, replace as required	1 Month +	Close
28/08/2009	CM	Remove/Replace bearings	24 Hours	Renewed NDE bearing and lab seals. Refitted coupling guard
18/10/2009	CM	Carry out alignment check on motor/fan	24 Hours	Alignment Check carried out ok
23/03/2010	OU	Bearings remove top halves, investigate, grease and box up.	1 Week	Opened up bearings ok, re-greased and boxed up. Unblocked grease holes in top half.
14/04/2010	CM	Remove/Replace bearings as required	1 Week	PA fan lined up. Coupling springs re-greased and boxed up. Guard refitted. NDE bearing opened up and inspected all ok. Bearings cleaned and re-greased and refitted.
20/05/2010	CM	Check all PA fan bearings auto greasers, replace as required	1 Month +	Closed WOC to comply with KPIs
02/08/2010	CM	Check all PA fan bearings auto greasers, replace as required	1 Month +	Ready
18/08/2010	OUMIN	Open up and inspect/replace bearings	24 Hours	Bearings opened up and inspected. Bearings ok. Bearings re-greased and boxed up.

2.2 Expert Judgement in the Maintenance Policy

This section clarifies the significance of expert judgements in the CBM system practiced in this industry application, and how they are recorded and used in maintenance decision-making.

2.2.1 Expert Judgement on System's Condition State

The work orders are results of the experienced engineers' assessments, i.e. expert judgements, of the condition of bearings. A discussion with the maintenance managers revealed that although expert judgements on equipment condition are provided, mostly at two-week to two-month intervals, they are not always recorded. When the bearings operate normally in the expert's opinion, the result of the assessment and the time that the expert is called are not recorded in WOCs. In fact, an expert judgement is recorded when the engineer schedules further investigation. The combination of the type of actions and their target completion dates recorded in WOCs imply the states of bearings in the expert's opinion.

Table 2.2 shows the works recorded in WOCs, categorised according to the type of maintenance actions and their scheduled completion times. To demonstrate the suggested expert judgement states, the actions are marked by different patterns. These patterns illustrate the expert judgements classified as discrete states. These states represent the severity of the health condition of bearings, in the expert's opinion; that is, the more severe the defect, the higher the number of the state. The discrete expert judgement states can be represented by integer numbers from 1 to N, where 1 represents the non-defective, and N represents the most defective condition. The description of the discrete states is depicted in Table 2.3. When the conditions of the bearings of both fans in a unit are assessed as the worst condition, i.e. action type "outage", both fans are stopped for further investigation. Both action types "outage" and "outage" minor suggest the same bearings condition in the expert's opinion, hence they are classified as the same state.

Table 2.2: Expert judgements as discrete states according to the action type and target completion.

Action Type / *Target Completion*	**Corrective Maintenance**	**Emergency Maintenance**	**Outage Minor (Action to be taken on 1 of the 2 fans of a unit)**	**Outage (Action to be taken on both fans of a unit)**
1 Month	1-Check auto greasers and replace if required 2-Monitor bearings			
1 Week	1-Open up and inspect bearings, replace if required 2-Autolube line is detached, investigate/replace 3-Bearing cartridge is turning, replace cartridge 4-Remove/replace bearing as required		1-Inspect bearings, vibration level rising 2- Bearings slack, investigate/replace	1-Remove top half of bearings, investigate, grease and box up
24 Hours	1-Remove/replace bearings 2-Bearings open up, tighten inner race and box up 3-Carry out alignment check 4-Clean coupling/area prior to alignment check 5-Grease lines reported blocked	1-Bearing temperature high (at 80° C) Investigate/inspect	1-Open up and inspect /replace bearings 2-Carry out alignment check	

Table 2.3: Description of the expert judgement discrete states

State	Description	Action Type
1	Operating normally, no action is taken	No action is taken
2	The condition is not as good as new, monitor bearings	Corrective Maintenance
3	Additional investigation is carried out as soon as possible	Emergency
4	Additional investigation is carried out by stopping one fan	Outage Minor
4	Additional investigation is carried out by stopping both fans	Outage

2.2.2 The Structure of the Expert Judgement-Based Maintenance Policy

In this CBM industry application, expert judgement is used to assess the condition of equipment, e.g. bearings. Based on this assessment, the decision regarding further investigation and maintenance action is made.

Based on the underlying mechanism observed from WOCs, the maintenance policy implemented in this industry application can be described as follows. The temperature and vibration, are continuously monitored by operators. Once they reach the "suspicious" level, the engineer is called for further inspection and verification of any abnormal changes of the condition data. If he diagnoses the condition of equipment to be normal, no action is taken. This is when his assessment is classified as State 1, recalling the state descriptions in Table 2.3. If he is suspicious of the fan bearings being faulty, he orders an action. This action is recorded in WOCs as "corrective maintenance" that might be followed by physical inspection, which is recorded as "emergency", "outage" or "outage minor". The structure of this maintenance policy is graphically depicted in Figure 2.1. In this picture $y \in \{1,2,...,N\}$ denotes the expert judgement provided as an integer number, where $N = 4$ represents the worst health condition.

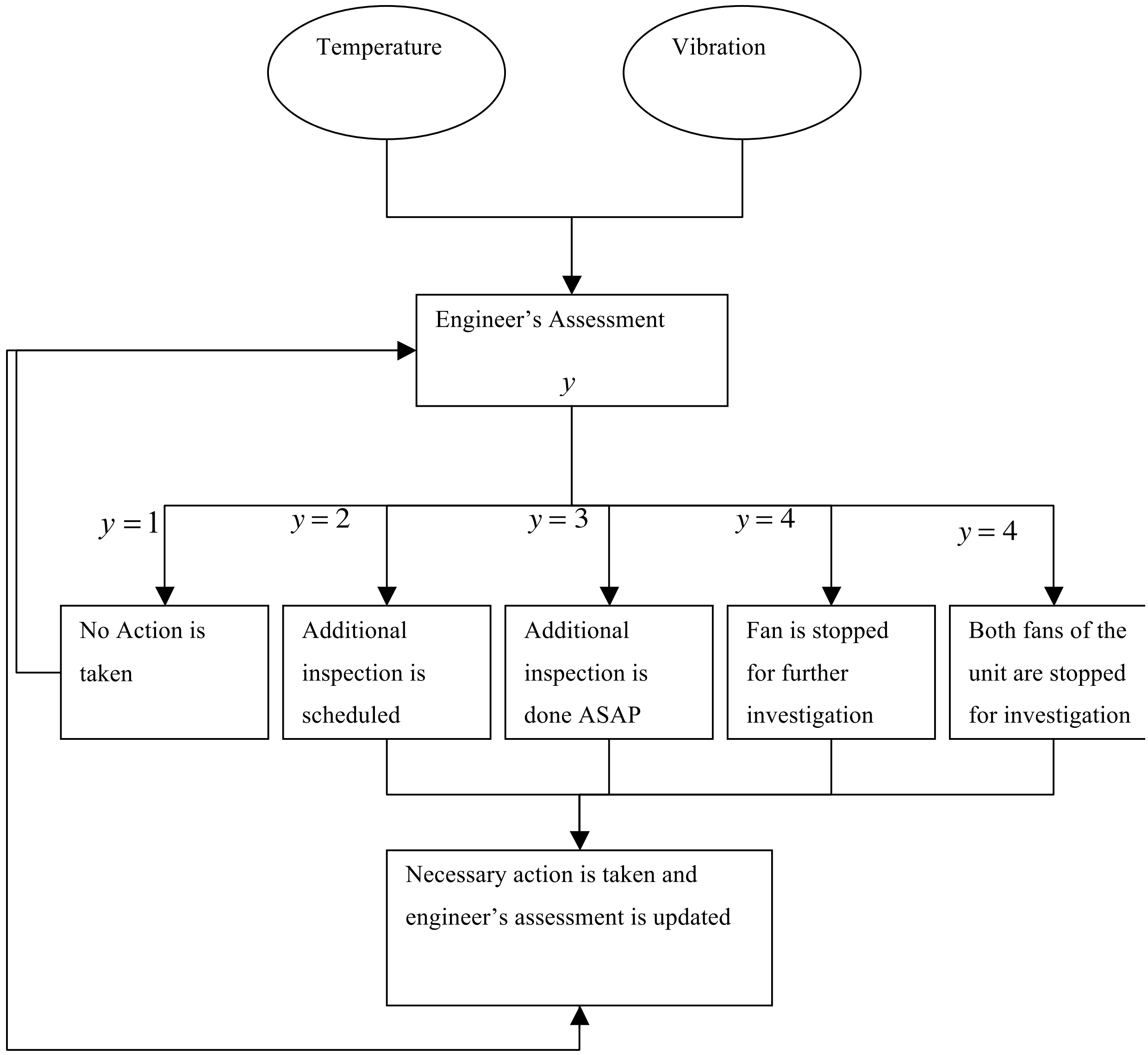

Figure 2.1: Structure of the CBM policy practiced at the company operating fans.

2.3 Discussion of Research Objectives in this Application Context

This section describes the research objectives in this application context, which were identified through examining the CBM system in this industry practice.

2.3.1 Expert Judgement-Based Diagnostics and Prognostics

As already noted, the results of expert assessments are recorded as work orders along with the scheduled start and target completion dates. The status of some WOCs have been recorded as cancelled and other work orders, mainly flagged as emergency, have been raised instead. This is mostly due to the fact that the potential faults propagated to a worse condition before the scheduled investigation was carried out. This demonstrates that knowledge regarding the variations in the system's condition would be helpful for the maintenance managers when scheduling the actions as "target completion" in WOCs. In other words, the information in terms of the likelihood of having a fault and how soon it will propagate to failure would support decision-making in terms of planning for further investigation.

This key observation was the motivation for developing a modelling framework to estimate the true condition of a system and the time to failure based on the available expert judgements. The need for a prognostic model was also stated by the maintenance managers at the preliminary meeting held at this company.

2.3.2 Maintenance Decision Optimisation

As explained earlier, the action types along with the target completions imply the severity of the condition of the fan bearings in the expert's opinion. The actions are mostly recorded as further inspections, meaning that the necessary corrections or replacements need to be carried out based on additional inspection of fan bearings. The information recorded in WOCs shows that some work orders have been recorded for further investigation by stopping the fans, but after undertaking this costly action, physical inspection has confirmed that the system is not in a faulty condition.

Different preparations need to be made for further investigation, according to each category of actions. Specifically, to carry out an additional inspection recorded as "monitor bearings", the unit load needs to be dropped. Physical inspection recorded as "minor outage" requires the necessary preparations for stopping the fan, which is much more costly than dropping the load.

Once further investigation is carried out and the engineer verifies the system to be faulty, he either performs minor corrections, such as reconnecting auto greasers, or replaces the bearing with a new spare. If he verifies that the fan is not faulty he either does nothing or conducts preventive maintenance, such as greasing the bearings manually.

The scenario explained above is the motivation for establishing a decision model to support cost-effective decision-making with regard to the inspection choice and maintenance action type. The decision choices are optimised by making a trade-off between the cost due to unnecessary actions and the cost of consequences of unexpected fault propagation and failure.

2.4 Summary and Conclusion

The CBM system practiced at a large engineering company operating fans was described in this chapter. A careful examination of the maintenance event database from the past two years demonstrated the role of expert judgement in the CBM decision support process.

As mentioned in Sub-Section 2.1.2, the Work Order Cards associated with six fans from the duration of July 2008 to September 2010 were supplied; data related to the time before July 2008 is not available since the data management system has been changed in 2008. The failure of bearings has been found to be the predominant cause of failure of the fans. The average lifetime of bearings is almost two years. In total three lifetime data sets of expert judgements related to bearings could be extracted from the available data. However, the data does not contain expert judgements during normal operation of the fans, as data is currently only collected in connection with maintenance interventions.

Furthermore, a simulation study is conducted in Chapter 6 to examine the effect of the number of expert judgement sequences, and the quantity of data associated with equipment condition states, on the performance of the proposed model. The simulation results confirm that the amount of real data available are not enough for training reliable models and hence cannot be used for quantitative evaluation of the models proposed in this thesis. Hence, for these two reasons, the CBM system in this industry application is used as a motivation for formulating plausible models, but it is not used further to populate the models.

In order to formally incorporate expert judgement into the CBM decision-support process, a stochastic model is needed to relate it to the underlying system's condition. Chapter 3 reviews the existing modeling approaches in literature for capturing the interaction between condition data and the underlying deterioration condition of a monitored system.

3 Condition Based Maintenance Diagnostics and Prognostics Modelling Approaches

This chapter reviews existing modelling approaches in the literature used for diagnostics and prognostics, with emphasis on statistical modelling techniques. The aim is to obtain an insight into the methods used for establishing the stochastic relationship between the underlying condition of a system and condition data, and find potential modelling methods for incorporating expert judgement into the CBM decision support process.

Section 3.1 presents a brief review of diagnostics and prognostics approaches and highlights their merits and limitations for different problem situations. Sections 3.2 to 3.5 review the modelling approaches for establishing the stochastic relationship between condition data and the unobservable condition of a monitored system. Section 3.6 concludes the chapter by highlighting the gaps in literature.

3.1 Diagnostics and Prognostics Approaches

Recall from Section 1.2 that diagnostics and prognostics are two important aspects of CBM that use condition data to provide useful information in order to support decision-making on performing maintenance actions. Diagnostics consists of the tasks performed to indicate whether something is wrong in the monitored system and if so, to determine the nature of the fault. Prognostics refers to the tasks carried out before a fault or a failure occurs, to estimate how soon it will happen. Depending on the implementation objective of CBM, condition data can be used for diagnostics, prognostics, or both.

Diagnostics and prognostics approaches have been broadly classified into three groups: Physics-based models, data-driven, and hybrid approaches (Jardine et al., 2006; Heng et al., 2009; Pecht, 2008). The classification of different diagnostics and prognostics approaches is illustrated in Figure 3.1.

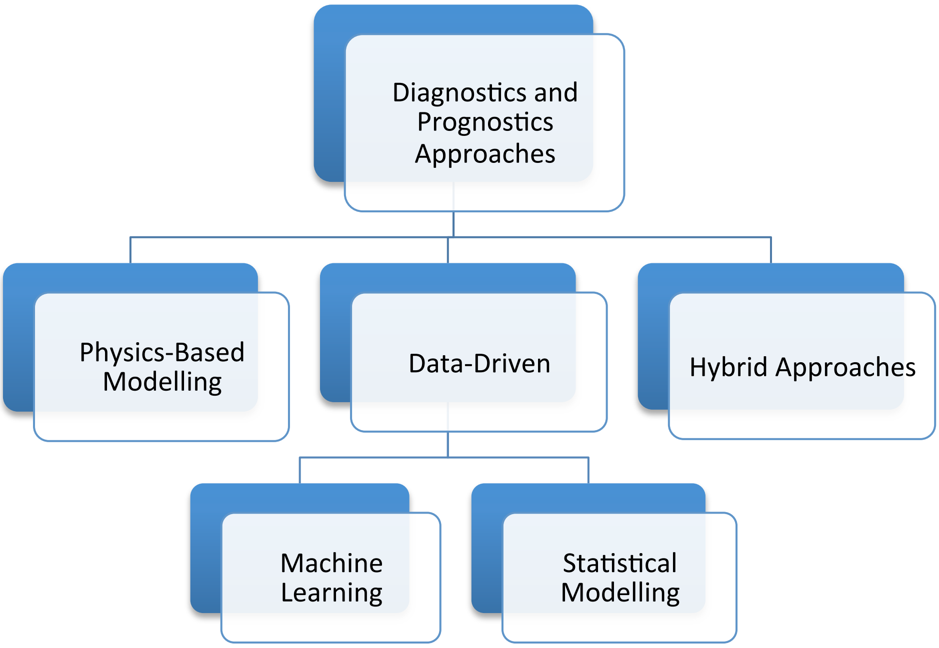

Figure 3.1: Diagnostics and prognostics approaches.

3.1.1 Physics-Based Modelling

Physics-based models utilise the mechanistic knowledge of the monitored system to obtain diagnostic and prognostic information. In this approach, the initiation and propagation of a fault, e.g. crack growth, is modelled based on known properties of the dynamics of a fault, drawn from the fields of physics or mechanics. Since the underlying degradation processes are not physically observable in indirectly monitored systems, residual generation methods, such as the Kalman filter, are used to obtain useful information from condition data, i.e. residuals, and relate them to the fault type and its physical severity, e.g. size of a crack. An example of this modelling approach for prognostics is reported by Qiu et al. (2002), in which the natural frequency and acceleration amplitude of a bearing is related to its stiffness. The relationship between the operating time, the failure time and the stiffness is established from damage mechanics, and subsequently the residual life of a bearing is predicted based on vibration measurements. Other examples of the application of physics-based models have been reported for diagnostics and prognostics of crack growth in rotor shafts (Oppenheimer and Loparo, 2002), fracture growth on gas turbine blades (Kumar, 2010), fracture in helicopter gearboxes (Kacpryznski et al., 2004) and prognostics of the failure mechanisms of electronic products (Pecht and Gu, 2009).

Although physics-based modelling has the advantage of accuracy for diagnostics and prognostics, it is not practical to use this method for complex systems because of the hard to model relationships between the failure mechanisms of different components in such systems (Heng et al., 2009). Also, these models are developed based on mechanistic knowledge of specific mechanical component degradations, with regard to the material and geometrical features of the component. In other words, these models are tailored for individual components with specific degradation processes, and this limits generalising their application to other types of components (Brotherton et al., 2000; Oppenheimer and Loparo, 2002; Kumar, 2010).

3.1.2 Data-Driven Approach

In data-driven approaches, prognostics and diagnostics are performed by direct analysis of condition data without requiring knowledge of physics or engineering principles (Heng et al., 2009; Si et al., 2011). Data-driven methods can be further classified into two categories: statistical modelling and machine learning approaches.

In the statistical modelling approach, prognostics and diagnostics are performed by fitting a model to available lifetime failure data and condition data (Si et al., 2011). Recall from Chapter 1 that the focus of this thesis is on CBM for partially observable systems. For such systems, since the actual system's condition is not observable, a stochastic model is needed to relate condition data to the unobserved condition of a monitored system. This model is used for estimating the actual system's condition and predicting the time to failure, given condition data. In Sections 3.2 to 3.5 we review the existing modelling approaches for capturing the stochastic relationship between condition data and the underlying system's condition, and how they are used for diagnostics and prognostics.

In machine learning techniques the complex non-linear relationship between condition data and the system's condition is approximated, i.e. learned by repetitive examples. In contrast to statistical modelling, the machine learning approach does not require distributional assumptions about the underlying failure and the stochastic relationship between condition data and system's actual condition; hence it can avoid "potentially large errors" due to incorrect assumptions (Tse and Atherton, 1999). However, machine learning approaches require a relatively large amount of data to provide accurate results (Heng et al., 2009). For some applications it might be feasible to use accelerated life tests and obtain a large amount of condition data (e.g. vibration signals). However, for most CBM applications it is not practical to obtain these amount of data, and this can limit the implementation of machine learning approaches.

A comprehensive recent review of machine learning diagnostic and prognostic techniques can be found in Pandian and Ali, 2010. The most commonly known machine learning technique for diagnostics and prognostics is Artificial Neural Networks (ANNs) (Jardine et al., 2006; Heng et al., 2009; Pandian and Ali, 2010). An ANN consists of a layer of input nodes, one or more layers of intermediate hidden nodes, at least one layer of output nodes, and connecting weights. The unknown functions that relate the inputs to outputs are trained by adjusting the connecting weights, with repetitive observations of inputs and outputs.

ANNs have been demonstrated to capture the complicated relationship between condition data and the actual health condition of a system without requiring a prior knowledge of the physics or distributional assumptions regarding their stochastic evolution (Rao et al., 2012). Also, their ability to perform "robustly" in noisy environments has made them a popular technique in diagnostics based on waveform type condition data (e.g. vibration signals) that are contaminated with noise (Jardine et al., 2006; Rao et al., 2012). However, the main limitation of ANNs is their "lack of transparency" or "black box" nature (Heng et al., 2009; McNaught and Zagorecki, 2011). According to McNaught and Zagorecki (2011) this would particularly be a problem for applications with the "training" intention.

3.1.3 Hybrid Approaches

Hybrid approaches attempt to combine various methods, in order to apply the merits of them, and utilise all useful information such as mechanistic knowledge of a system's degradation process or maintenance event data along with condition data, to increase accuracy in diagnostics and prognostics.

An example of a hybrid diagnostics and prognostics approach is Dynamic Bayesian Networks (DBN), in which probabilistic graphical models are combined with Bayesian statistics. Bayesian Networks (BNs) are probabilistic graphical models representing the joint probability distributions over a sequence of random variables (Pearl, 1997; Jensen, 2001). DBNs are an extension of standard BNs that model the evolution of variables of BNs over time. The flexible structure of a DBN allows the incorporation of other useful information, such as the effect of changes in operational conditions (e.g. load) and maintenance actions, in addition to condition data, to support diagnostics and prognostics. DBN-based prognostics have been investigated by Dong and Yang (2008), and McNaught and Zagorecki (2009).

Other hybrid approaches combining data-driven approaches with physics-based modelling have been reviewed in Jardine et al., 2006; Pecht, 2008; Heng et al., 2009; and Peng et al., 2010. An example of combining machine learning and statistical modelling is proposed by Mohanty et al. (2008).

In the following sections we review the existing statistical modelling approaches for capturing the stochastic relationship between condition data and the underlying system's condition, and discuss how they are used for diagnostics and prognostics. The classification of these modelling approaches is illustrated in Figure 3.2.

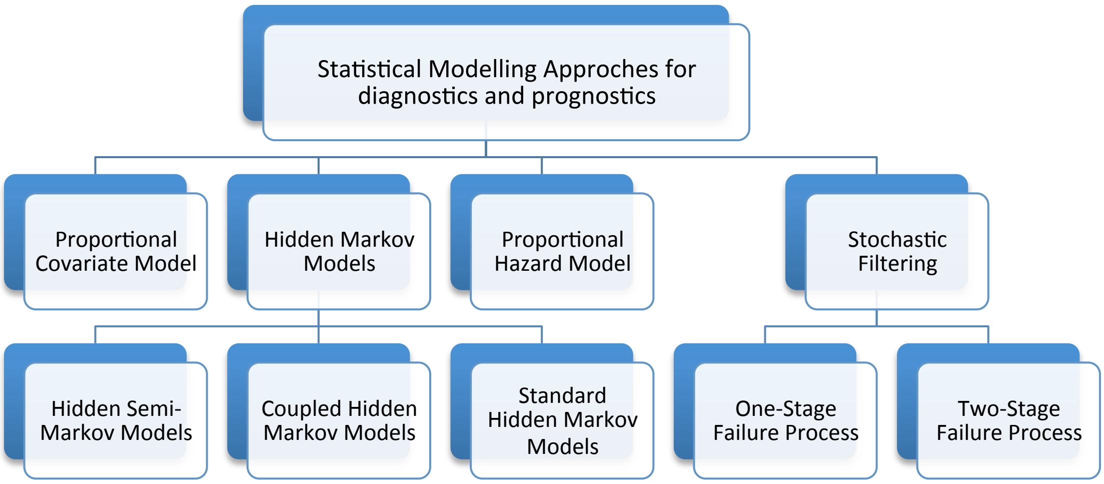

Figure 3.2: Statistical modelling approaches for diagnostics and prognostics.

3.2 Proportional Hazard Model (PHM)

The Proportional Hazard Model (PHM) is a popular model in CBM for partially observable systems, in which both age and condition data are taken into account to determine the underlying condition of a system (Wang 2008; Heng et al. 2009). In this model hazard rate is used as a measure to represent the underlying health condition of a system, and is assumed to change proportionally with factors termed as "covariates". Covariates could be parameters indicating operational and environmental conditions that affect system's degradation (e.g. temperature, or torque load), or condition data reflecting system's condition, depending on the model application. The basis of the PHM is the simple assumption that the hazard rate is affected by covariates in a multiplicative way. This model was first initiated by Cox (1972) and started to be used in the area of CBM in the 1980s (Bendell et al., 1991; Jardine et al., 1987).

The hazard rate at age t is given by the following function:

$$h(t) = h_0(t)\exp(\gamma_1 z_1(t) + \ldots + \gamma_n z_n(t)),$$

where $h_0(t)$ is a baseline hazard function, $z_1(t),\ldots,z_n(t)$ are covariates at time t, and $\gamma_1,\ldots\gamma_n$ are constant coefficients. A common parametric form used for the baseline hazard function is Weibull, which is given in the form of $h_0(t) = \frac{\beta}{\alpha}\left(\frac{t}{\alpha}\right)^{\beta-1}$, where β and α are "shape" and "scale" parameters, respectively. The covariates $z_1(t),.....z_n(t)$ can be any condition data such as features extracted from vibration data through signal processing, or measurements of operational and environmental conditions influencing system's degradation such as cutting force in a machining process.

It is important to note that in PHMs only the current condition data is used, as opposed to other methods such as stochastic filtering method and Hidden Markov Models where previous condition data is also considered for diagnostics and prognostics. We will discuss these modelling frameworks later in this chapter.

PHMs have been applied to different types of condition data, such as metal particle levels in engine oil (Jardine et al., 1987) and the significant condition indicators extracted from vibration signals (Lin et al., 2004; Banjevic et al., 2001; Mazzuchi, 2008; Vlok et al., 2002). Jardine et al. (Jardine et al., 1997; Banjevic and Jardine, 2006) used the PHM for calculating the reliability function and the remaining residual life of rolling element bearings and engine given the condition data. They used a Weibull hazard function and a non-homogeneous Markov process to model the evolution of condition data.

Based on this framework, optimal maintenance policies were also proposed to support the maintenance decision-making, in a series of works (Makis and Jardine, 1992; Jardine et al., 1997; Vlok et al., 2002; Tian and Liao, 2009; Wong et al., 2011). These PHM-based optimal policies were proposed as control-limit policies where some thresholds are set for the hazard rate and once the hazard rate reaches these thresholds, maintenance actions such as replacement are carried out accordingly. A software program called EXACT (Jardine et al., 1997) was developed to compute the optimal policy for hazard rate thresholds and inspection intervals, given the cost values and the historical reliability and maintenance event (lifetime data) and condition data.

Maximum Likelihood Estimation (MLE) is used to estimate the parameters of a PHM, based on lifetime data and condition data. To estimate reliable parameter values for a PHM, large data sets are required. However, these data are often limited in practice due to the monitored components being replaced before failure, and this can restrict the use of PHMs in practice (Gorjian, 2009; Heng et al., 2009; Sun et al., 2006). Recently a parameter estimation method based on expert's prior knowledge of the model has been proposed by Zuashkiani et al. (2008) to overcome this problem.

According to Gorjian et al. (2009) and Wang (2008), PHM mixes different types of covariates and this can cause problems. For instance, the changes of vibration signals are caused by the changes of the health condition of a system, i.e. vibration signals are influenced by the hazard rate. Some other covariates, such as the amount of metal particles measured in oil analysis, are influenced by the deterioration condition of a monitored system (e.g. an engine) and hence represent condition data reflecting system's health condition. These metal particles may also accelerate the deterioration process. PHMs assume that covariates that are influenced by system's condition (e.g. vibration signals), covariates that have bilateral relationship with system's condition (e.g. metal particles in oil analysis) and covariates representing environmental conditions (e.g. temperature), all affect the hazard rate in a multiplicative way.

3.3 Proportional Covariate Model

As mentioned above, PHMs assume that the hazard rate changes proportionally with condition data, which are termed as covariates. In other words condition data are assumed to be "explanatory variables" and the hazard rate is the response variable in a PHM. Sun et al. (2006) argued that in practice, condition data changes due to changes of the underlying system's condition, which is represented by the hazard rate in a PHM. They proposed the Proportional Covariate Model (PCM) in which condition data is modelled as "explanatory variables" and the hazard rate is modelled as the "response variable". A function of condition data, denoted by $\Psi\left(z(t)\right)$ in PCM, is expressed as follows:

$$\Psi\left(z(t)\right) = C(t)h(t),$$

where $z(t)$ denotes the covariate at time t, and $C(t)$ represents a baseline covariate function used to describe the relationship between the covariates and the hazard rate, and $h(t)$ denotes the hazard rate.

According to Sun et al. (2006), since the baseline covariate function $C(t)$ in PCM is dependent on "both historical failure data and historical condition data and can be updated according to newly observed failure data and covariates", it can be applied when the historical failure data are limited. They demonstrated the implementation of this model for estimating the hazard rate of gears using vibration data obtained through an accelerated life test.

Recently Cai et. al. (2012) applied PCM to estimate the failure rate of cutting tools based on vibration signals. They obtained the baseline of the covariate function from a small sample of historical failure data.

3.4 Stochastic Filtering

In the stochastic filtering based methods, the unobservable condition of a monitored system given condition data is estimated through recursive equations over time. As opposed to the PHMs and PCM where only the current condition data is considered, ,in this method the whole history of condition data is used to estimate the system's condition and predict its value in future.

3.4.1 Stochastic Filtering Method Based on a One-Stage Failure Process

Christer et al. (1997) employed the Kalman filter to estimate and predict the erosion level of the inductors in an induction furnace, using all condition data to date. Let x_t be the unobservable system's condition and $z(t)$ be the measured condition data at time t. Christer et al. (1997) assumed that both x_t and $z(t)$ are stochastic processes and their evolutionary relationship follows the state space model defined as $x_t = \alpha x_{t-1} + \varepsilon_t$ and $z(t) = \beta x_t + \eta_t$, where α and β are state space parameters and ε_t and η_t are Gaussian noises. They employed Kalman filter to estimate the actual system's condition x_t, and predict its value at any time in future, given $Z_t = \{z(1), z(2), ..., z(t)\}$ that is the condition data obtained to date.

Wang and Christer (2000) proposed a stochastic filtering method that relaxes the Gaussian and linear assumptions of the Kalman filtering approach proposed by Charister et al. (1997). In this method, the underlying condition of a system, x_t, is modelled as the conditional residual life, that is the interval from any time point that condition monitoring information is obtained, to the time that the system "may be declared to be failed" given $Z_t = \{z(1), z(2), ..., z(t)\}$, in the absence of maintenance interventions. They assumed that the relationship between the conditional residual life at time t and the conditional residual life at time $t + \Delta$ can be described as $x_{t+\Delta} = x_t - \Delta$, where Δ is the condition monitoring interval. They assumed that there is a negative correlation between x_t and $Z(t)$ throughout the lifetime of a system. Consequently they showed that the probability distribution function of the conditional residual life could be updated through the following recursive equation:

$$p_t\left(x_t \middle| Z(t)\right) = \frac{p\left(z(t) \middle| x_t\right) p_{t-\Delta}\left(x_t + \Delta \middle| Z(t-\Delta)\right)}{\int_0^\infty p\left(z(t) \middle| x_t\right) p_{t-\Delta}\left(x_t + \Delta \middle| Z(t-\Delta)\right) dx_t}.$$

Wang (2002) applied this filtering approach to vibration signals of rolling element bearings obtained from a laboratory fatigue experiment.

3.4.2 Stochastic Filtering Method Based on a Two-Stage Failure Process

In the stochastic filtering based method described in Sub-Section 3.4.1, it is assumed that there is a negative correlation between the actual system's condition and condition data. However, the operating process of equipment can be classified into two stages, namely "normal" and "defective", where this negative correlation is only valid at the defective stage. This is due to the fact that when the system is in the normal condition, the value type conditions data (e.g. temperature) and the features extracted from waveform type condition data (e.g. kurtosis of the residual signal) are stable and fluctuate around a constant value. But when a defect initiates, and propagates in the second stage, condition data show abnormal changes and trend.

Wang (2003) improved the stochastic filtering based method by incorporating a two-stage failure process, known as the "delay time concept" in literature.

The delay time concept considers the failure process as a two-stage stochastic process. The first stage is the initiation of the defect where a detectable defect arises, while during the second stage a defect propagates and leads to a failure. The time between when a defect is identified until a failure occurs is called "delay time" (Baker and Wang, 1993; Christer and Waller, 1984). The delay time is a random variable, which is not directly measurable. Rather, it is a characteristic of the inspected system, the nature and type of the inspection, and sometimes of the person carrying out the inspection. For example, the initiation of a fault might be recognisable at different times according to different inspection methods.

Wang and Zhang (2008) applied the above stochastic filtering method to predict the residual life of a system over the failure delay time, for the scenario when the judgements provided by maintenance engineers are used as condition data. They assumed that an experienced engineer provides judgement on system's condition state as an integer number and there is a negative correlation between this number and the residual life of the system during the failure delay time. To establish the relationship between the expert judgement and the residual life, they proposed for the expert judgement a Normal probability distribution function with a constant variance, σ, and a mean that varies against the residual life, $\mu = A + Be^{-Cx_t}$, where x_t is the residual life at time t, and A, B and C are constant values. The discrete expert judgement variable is converted into a continuous variable by dividing the continuous space into integer intervals, where each interval represents a discrete expert judgement value.

Carr and Wang (2010) used the stochastic filtering based method and the delay time concept for the prediction of a residual life of a component subject to different failure modes. In this method, an individual stochastic filter is used to predict the residual life according to each potential failure mode and then the outputs from each filter are weighted with respect to the probability that the failure mode is the actual underlying failure mode. This stochastic filtering based model was extended by Wang et al. (2010) to predict the residual life and recommend the best inspection time for a system with multiple components subject to different failure modes. Wang et al. (2010) proposed modelling each component and failure mode individually and then pooling them together to form the system inspection model.

The successful application of stochastic filtering based method in prognostics relies heavily on specifying the right distribution form of the conditional probability of the condition data given the residual life. The formulation of the general likelihood function for parameter estimation of this method was given by Wang (2007). The author explored the choice between Weibull and Gamma distributions as the best-fitted model for the probability distribution function of the condition data given the residual life, based on experimental vibration signals of rolling element bearings.

The failure delay time is a natural methodology within the maintenance-engineering context because of its" easy to understand" nature. However, the application of this concept along with the stochastic filtering method requires the specification of the functions and parameter estimation for the probability distribution of the time duration of the two stages, in addition to the distribution form and parameter estimation of the conditional probability of the condition data given the residual life. The beginning of the second stage, the defect initiation, is often hard to identify and is usually not recorded in practice, and this can restrict the application of the stochastic filtering based prognostics (Heng et al. 2009).

3.5 Hidden Markov Models (HMMs)

A Markov process is a discrete random process with Markov properties, that is, the probability distribution of the system state at any time, given the state in the previous time step, is independent from all other previous states. A Hidden Markov Model (HMM) is a Markov process "observed in noise" i.e. the Markov process is not directly observable (Cappé et al., 2005). What is available to the observer is another stochastic process whose distribution is determined by the Hidden Markov process. This modelling framework fits well with the hidden process of fault progression of mechanical systems that goes through different deterioration phases before failure, and the observations obtained from monitoring their condition.

Because of the rich mathematical structure and the availability of computer implementation of HMMs, this framework has become a very popular technique for temporal pattern recognition such as speech recognition (Rabiner, 1989), handwriting recognition (Brown and Turin, 1996) and gesture recognition (Marcel et al., 1997).

The success of HMM in the area of pattern recognition motivated Bunks and McCarthy (2000) to investigate the use of this method in CBM modelling. They studied the feasibility of utilising HMMs in machine health diagnostics and prognostics by comparing CBM to speech processing. It was demonstrated that this model can be applied to the problem of machine health monitoring by treating fault classes as hidden states in the HMM and by condition data such as vibration measurements as the observations process. Their study concluded that HMM is a powerful framework for CBM, particularly because of its ability to differentiate between the changes in condition data due to defects and changes due to operational conditions such as torque loads.

3.5.1 Theoretical Background of Hidden Markov Models

A schematic representation of a first-order HMM rolled out in time is depicted in Figure 3.3. The square nodes represent the observation variables and circular nodes represent the hidden state variables. The horizontal arrows denote the conditional probabilities between two consecutive hidden states, while the vertical arrows denote the conditional dependency between hidden states and observations.

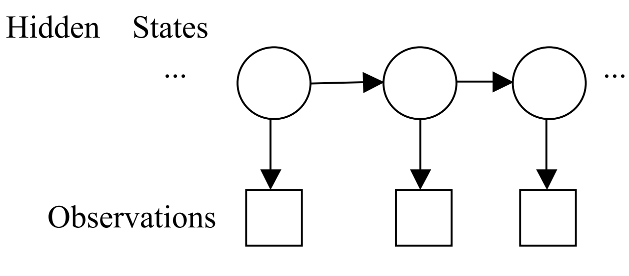

Figure 3.3: Schematic representation of Hidden Markov Models

HMMs can be classified into continuous and discrete models, according to the observations and hidden processes. Since the model developed in this research is based on a discrete-time discrete-state HMM, the theoretical background of this type of HMMs is given in this section.

3.5.1.1 Elements of an HMM With Discrete States and Observations

A discrete-time, discrete-state HMM can be characterised by the following parameters (Rabiner, 1989):

N is the number of states in the model. The states are interconnected according to the specific topology of the model. The set of discrete hidden states is denoted by $S = \{S_1, S_2, ..., S_N\}$. $q_t \in S$ is a time-indexed discrete variable denoting the state of the system at time t.

M is the number of distinct observation symbols per state. The observation symbols represent the physical output of the system modelled as an HMM. The individual symbols, denoted by $V = \{v_1, v_2, ..., v_M\}$. $o_t \in V$ are time indexed discrete variables denoting the output of the system at time t.

$A = \{a_{ij}\}$ is the state-to-state transition probability distribution, where each element a_{ij} in Matrix A, is the probability of making a transition from State i to State j , that is: $a_{ij} = P\left(q_{t+1} = S_j \middle| q_t = S_i\right)$, $1 \le i, j \le N$.

$B = \{b_j(k)\}$ is the observation probability distribution in State j, where $b_j(k) = P\left(o_t = v_k \middle| q_t = S_j\right)$, $1 \le j \le N$, $1 \le k \le M$.

$\pi = \{\pi_i\}$ is the initial state probability distribution, where $\pi_i = P(q_1 = S_i)$, $1 \le i \le N$.

All the parameters listed above are required for a complete specification of an HMM. A compact representation of an HMM denoted by $\lambda = (\pi, A, B)$ is used to indicate the complete set of parameters for the model.

3.5.2 Hidden Markov Models Implementation for Diagnostics

HMMs have been applied as a diagnostic method for fault recognition and classification of tools in drilling processes, which are subject to gradual wear (Baruah and Chinnam, 2005; Akhilesh Kumar et al., 2011). The diagnostic model assesses the health state of the cutting tool during the machining process based on thrust-force and torque signals. In this method, HMM states represent different phases the equipment might go through before failure. In other words, the higher indexed states represent the higher level of severity of a defect. Diagnostics is performed by solving the decoding problem, described in Paragraph 3.5.6.2, to estimate the current health state given a sequence of observations. Similar applications of HMMs for CBM have been also investigated for bearing wear and structural damage (Rammohan and Taha, 2005).

A different method of using an HMM in CBM has been reported by Ocak et al., (2007) for bearing wear tracking. In this method, an HMM with three states is trained to represent a normal (non-defective) bearing, using the vibration signals collected through an accelerated life test while it is working in a normal condition. The condition of a bearing in this method is assessed by the probability of the features extracted from the vibration signals given the normal HMM. As bearing wear increases, this probability, which represents the similarity between the current signals and those related to a normal bearing, decreases. Ocak et al. (2007) proposed to use this probability value as a measure indicating the severity of a defect. Their experiment showed that this probability drops gradually as a defect initiates and propagates and hence could be a predictive indication of an upcoming failure.

3.5.3 Coupled Hidden Markov Models (CHMMs)

3.5.3.1 General Architecture and Modelling

CHMMs are an extension of HMMs that contain multiple Hidden Markov chains and observation processes. The interaction between Hidden Markov chains is modelled by introducing conditional dependencies between their states, across time. A widely used class of CHMMs, referred to as " standard fully coupled" in the literature, was introduced by Brand (1997) to capture the interactions among multiple HMMs in action recognition. The CHMM topology proposed by Brand is illustrated in Figure 3.4.

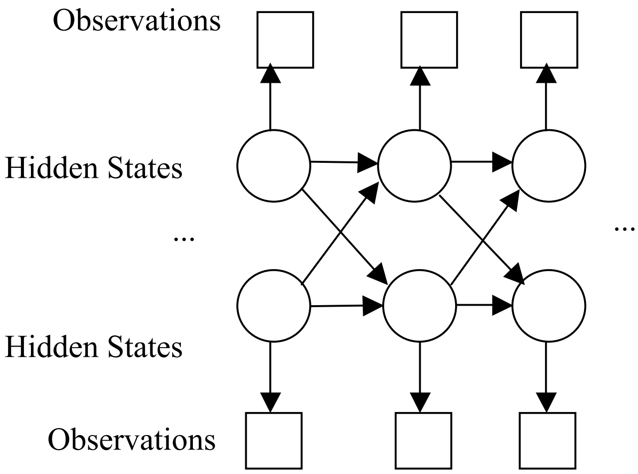

Figure 3.4: Topology of Standard Coupled Hidden Markov Model.

In this model, multiple processes are coupled by bridging their states across time. In other words, the state of each process depends on the states of all of the processes, including itself, at the previous time slice. Each process has individual observation outputs. The state transition probability for C HMMs coupled together is given by the joint conditional dependency $P\left(q_t^c \middle| q_{t-1}^1, q_{t-1}^2, \ldots, q_{t-1}^C\right)$, where q_t^c is the state of the c^{th}, $1 \le c \le C$, HMM. In Brand's formulation (Brand, 1997), this joint conditional dependency is given by the product of all marginal conditional probabilities, represented by: $P\left(q_t^c \middle| q_{t-1}^1, q_{t-1}^2, \ldots, q_{t-1}^C\right) = \prod_{c'=1}^{C} P\left(q_t^c \middle| q_{t-1}^{c'}\right)$.

This topology of CHMMs have been applied to human action recognition (Brand et al., 1997), image processing (Gai et al., 2007), suspect interaction modelling (Brewer et al., 2006), freeway traffic modelling (Kwon and Murphy, 2000) and more recently to rotating machinery diagnostics (Xiao et al., 2011) (this will be discussed in Paragraph 3.5.3.2).

A different formulation of CHMMs, based on the architecture depicted above, was proposed by Zhong and Ghosh (2001). They assumed that the HMMs can influence each other in different levels, as opposed to the standard fully coupled HMMs, and the level of coupling is directly characterised by some parameters. They introduced the new parameter "Coupling Coefficient" to model the degree of coupling between two HMMs. The authors proposed to model the joint conditional dependency of C HMMs coupled together as "a linear combination of all marginal dependencies", i.e. $P\left(q_t^c \middle| q_{t-1}^1, q_{t-1}^2, \ldots, q_{t-1}^C\right) = \sum_{c'=1}^{C} \theta_{c'c} P\left(q_t^c \middle| q_{t-1}^{c'}\right)$, where $\theta_{c'c}$ is the coupling coefficient representing the level of influence from model c' on model c.

We will introduce a new topology of CHMMs in Chapter 5 to capture the interaction between expert judgement and the deterioration process of a system. The formulation of our CHMM has the advantage of reduced parameter space compared to the CHMMs mentioned above.

3.5.3.2 CHMMs Implementation for Diagnostics

In many CBM applications more than one condition monitoring method is used, in order to obtain more accurate diagnostics. An example is multi-channel vibration signals collected from rolling bearings for fault diagnostics (Jardine et al., 2006). In this case HMMs are not appropriate for modelling CBM diagnostics since there is only one process to describe the underlying failure mechanism and one process to model the observation.

Recently the standard fully coupled HMM depicted in Figure 3.4 has been used in rotating machinery diagnostics. Xiao et al. (2011) used the CHMM to combine the features extracted from horizontal and vertical vibration signals for fault diagnostics of bearings. Each observation sequence corresponds to a feature sequence extracted from the vibration signals of one channel. A separate CHMM was trained for all possible bearing conditions, i.e. normal and three different fault types. The diagnosed condition is the one associated with the model that maximises the likelihood of the observation data. Their experiment on data from a rolling bearing vibration test machine validated the feasibility of this method and its effectiveness for the fault diagnostics of rotating machinery based on multi-channel fusion monitoring. In this experimental study, 80 datasets were used to train the model. However, in real practice the number of observation sequences available for training the model is much more limited. This is mainly due to replacements carried out before failure. It would be interesting to see the effect of limited datasets on the trained model and hence its diagnostic performance.

3.5.4 HMMs Represented as Dynamic Bayesian Networks

The hidden state in a DBN is modelled in terms of a set of variables, instead of a single random variable; this allows a flexible structure for the interaction between hidden states for complex systems. Murphy (2002) proposed a method of representing multiple interacting HMMs as DBNs. This method has the advantage of reduced computational complexity of inference. Camci and Chinnman (2005) used this method of implementing HMMs by DBNs to estimate the health state of the drill-bits on a CNC drilling machine, using the thrust-force and torque sensors as monitoring information. More recently, Tobon-Meja et al. (2012) used this modelling framework for the prognostics and diagnostics of the cutting tools in a CNC machine. They modelled the evolution of wear in the cutting tools by a Mixture of Gaussian Hidden Markov Models (MoG-HMMs) represented by a DBN and used the features extracted from condition data to train this model. As condition data are continuously collected, the current health condition of the cutting tools is continuously estimated online and the residual life is predicted, based on the DBN inference algorithms provided by Murphy (2002).

3.5.5 Hidden Semi Markov Models

One limitation of HMMs is that this framework is unable to explicitly model the time spent in each state. Based on the Markovian property, in an HMM the probability of staying in State i for d time units is computed as the joint probability of the self-transition in State i for $d-1$ unit times and an outgoing transition once, or ${a_{ii}}^{d-1}(1-a_{ii})$, which is the geometric distribution function of d. Since in most real practices the state duration does not follow this function, Hidden Semi Markov Models (HSMM) were introduced to overcome this drawback by explicitly modelling the state durations using an appropriate probability density function (Russel and Moore, 1985). An HSMM can be represented by the compact form of $\lambda = (\pi, A, B, D)$ where D is the state duration probability distribution and, as for the standard HMM, A is the transition matrix, B is the observation matrix and π is the initial probability distribution. Unlike an HMM in which each hidden state generates a single observation, each state in an HSMM can emit a sequence of observations.

Since an HSMM allows modelling the time duration of each hidden state, it has the potential capability of performing prognostics – that is, of estimating the residual useful life. Dong et al. (2006) proposed a diagnostics and prognostics approach based on HSMM method. An individual HSMM is trained for all possible fault types in addition to the HSMM for normal conditions. After training the models, the state durations are estimated from the training data. Given a sequence of observations a fault can be diagnosed using the standard HMM classification technique. After identifying the fault type and estimating the current health state, the residual life of the system is computed using a recursive equation. Dong and He (2007) evaluated the proposed methodology through an experimental case study on hydraulic pumps and showed that the HSMM-based diagnostics is superior to HMM-based diagnostics. Recently, a non-stationary version of this model has been proposed by Peng and Dong (2011) to improve the prognostic performance of this method. In this model, the probabilities of transition to less healthy states increase by aging. Hazard rate is used to combine the aging factor with the state transition probabilities. Using the same experimental data, the prognostic performance of this method has been evaluated and has been shown to be an improvement over the HSMM used without the aging factor.

3.5.6 Three Basic Problems in HMMs

In order to utilise HMMs in CBM modelling, three basic problems have to be solved. These problems along with their best-known solutions in literature are given in the following paragraphs.

3.5.6.1 Evaluation (Classification)

Consider the problem where there are a number of HMMs. Given a sequence of observations of length T, $O = o_1, o_2, \ldots, o_T$, the problem is to find out which HMM most likely generated the given sequence of observations. In other words, for a HMM, the problem is to compute the probability of the observation sequence $P(O|\lambda)$, and hence choose which HMM was the most likely to have produced that sequence. In the industry application described in Chapter 2, O denotes the sequence of expert judgements, where o_t denotes the expert judgement provided at time t as an integer number.

A naive method of calculating the probability of the observation sequence would be to consider the probability of this sequence given all N^T possible sequences of hidden states. Calculation in this way is computationally expensive, particularly for models with a large number of states or long observation sequences. Rather, an efficient procedure called the forward algorithm, based on dynamic programming, is used to calculate the probability of an observation sequence given a particular HMM (Rabiner, 1989; Cappe, 2005). The forward algorithm uses the forward variable, denoted by $\alpha_t(i)$, that represents the joint probability of observations up to time t and $q_t = i$, $\alpha_t(i) = P(o_1, o_2, \ldots, o_t, q_t = i|\lambda)$. In other words, $\alpha_t(i)$ is the probability of observing the partial sequence $o_1, o_2, \ldots, o_t$ and the system being in State i at time t.

The name of this algorithm originates from the way it processes a sequence of observations. It moves forward from the first observation in the sequence to the last. At each single observation in the sequence, probabilities to be used for calculations at the next observation are computed. The algorithm comprises the following steps:

Initialisation: Define $\alpha_1(i)$ as:

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \le i \le N.$$

Induction: Compute the forward variable at each time $1 < t \le T-1$ through the following recursive equation:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(o_{t+1}),\ 1 \le j \le N.$$

Termination: The probability of the observation sequence is given by summing the likelihood on all possible paths, that is,

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i).$$

The complexity of this method is proportional to N^2T, while the direct calculation mentioned before has an exponential complexity. When there is a collection of HMMs and a sequence of observations, the model with the maximum $P(O|\lambda)$ is chosen as the model that most likely generated the observation sequence.

3.5.6.2 Decoding (Recognition)

Another problem, which is usually of most interest, is to find the hidden states that generated a sequence of observations. Within the context of CBM, this problem is solved for diagnostics to estimate the current health state of a system given condition data (Baruah and Chinnam, 2005; Akhilesh Kumar et al., 2011). Recalling the industry application described in Chapter 2, this corresponds to finding the actual condition state of the system, given a sequence of expert judgements.

Given a sequence of observations $O = o_1, o_2, \ldots, o_T$, and an HMM, the problem is to find the most likely sequence of hidden states. We could do this by listing all possible sequences of hidden states and finding the probability of the observed sequence for each of the combinations. The most probable sequence of hidden states is then that combination that maximises the probability of the observation sequence given the hidden state combination. This approach of exhaustively calculating each combination is computationally expensive.

The Viterbi algorithm is a dynamic programming algorithm used to determine the most probable sequence of hidden states given a sequence of observations and an HMM (Forney, 2005). It defines the partial probability δ as the probability of reaching a particular intermediate state in the trellis. This probability is different from the one in the forward algorithm, since it represents the probability of the most probable path to a state at time t, and not the total over all possible paths to the state. $\delta_t(i)$ is defined as the maximum probability of all sequences ending in State i at time t and the best partial path is the sequence that achieves this maximum probability. Each state at time $t = T$ will have a partial probability and a partial best path. The Viterbi algorithm finds the overall best path by choosing the state with the maximum partial probability and choosing its partial best path.

At each intermediate and end state we know the partial probability $\delta_t(i)$. However, the aim is to find the most probable sequence of states; therefore we need some way of recording the partial best paths through the trellis. This recording is carried out by holding, for each state, a back pointer, denoted by $\Psi_t(i)$, which points to the predecessor state that optimally provokes the current state. The following steps formally define the Viterbi algorithm:

Initialisation: Calculate $\delta_t(i)$ and $\Psi_t(i)$ for $t = 1$ as:

$$\delta_1(i) = \pi_i b_i(o_1),$$
$$\Psi_1(i) = 0, \quad 1 \le i \le N.$$

Induction: Calculate $\delta_t(j)$ and $\Psi_t(j)$, for $2 \le t \le T$, using the following recursive equations, where the operator "argmax" selects the index that maximises the bracketed expression:

$$\delta_t(j) = \max_{1 \le i \le N}\left[\delta_{t-1}(i) a_{ij}\right] b_j(o_t),$$
$$\Psi_t(j) = \arg\max_{1 \le i \le N}\left[\delta_{t-1}(i) a_{ij}\right], \quad 1 \le j \le N,\ 2 \le t \le T.$$

Termination: Find the most likely final state q_T^*, by selecting the highest $\delta_T(i)$:

$$q_T^* = \arg\max_{1 \le i \le N}\left[\delta_T(i)\right].$$

Back tracing: Find the most likely sequence of states by back tracing, starting from q_T^* via the back pointers:

$$q_t^* = \Psi_{t+1}\left(q_{t+1}^*\right), \ \ t = T-1, T-2, ..., 1.$$

The Viterbi algorithm looks at the whole sequence, before deciding on the most likely final state and then tracing back through the pointers to indicate how it might have arisen. This algorithm makes the assumption that the most likely sequence of hidden states up to time t must depend only on the observation at time t, and the most likely sequence of states at time $t-1$. This assumption is satisfied in a first-order HMM.

3.5.6.3 Training (Learning)

The training or learning problem is to adjust the HMM parameters to maximise the probability of a given observation sequence (training sequence), i.e. to maximise $P(O|\lambda)$ where $\lambda = (\pi, A, B)$. The problem of computing the optimal parameters, or a set of parameters that globally maximise $P(O|\lambda)$, is intractable in practice (Chen et al., 2010). However, using some iterative procedure such as the Expectation Maximisation algorithm (EM), it is possible to improve the model parameters to locally maximise $P(O|\lambda)$ (Baum, 1970; Cappé et al., 2005; Rabiner, 1989). In this paragraph we discuss the Baum–Welch algorithm (Welch, 2003), which is the most successful, widely cited, HMM training method in the literature.

The Baum–Welch algorithm is a generalised Expectation Maximisation (EM) algorithm that locally maximises $P(O|\lambda)$. Using an initial guess of model parameters, the Baum–Welch algorithm first estimates the likelihood of hidden states given the observation sequence, and then uses the expected counts of state transitions and observations to estimate the parameters. Since the expected counts can be derived from the parameters and vice versa, the procedure can be iterated to move from an initial guess of the parameters to a better estimate that (locally) maximises $P\left(U|\lambda\right)$ (Cappé et al., 2005). The Baum–Welch algorithm uses forward and backward variables to calculate the expected counts. The calculation of the forward variable $\alpha_t(i)$ is given in paragraph 3.5.6.1. The backward variable $\beta_t(i)$ is defined as the probability of the partial observation sequence $o_{t+1}, o_{t+2},..., o_T$ given that the current state q_t is i:

$$\beta_t(i) = P(o_{t+1}, o_{t+2},..., o_T | q_t, \lambda).$$

The backward algorithm starts from the last observation in the sequence and moves backward to the first observation. As in the case of the forward variable, a recursive equation is used to calculate $\beta_t(i)$:

$$\beta_T(i) = 1, \ 1 \le i \le N,$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \qquad 1 \le i \le N, \qquad t = T-1,...,1.$$

To re-estimate the parameters the new variable $\xi_t(i, j)$ is defined as the probability of a path being in State i at time t and in State j at time $t+1$, given the observation sequence O and the model:

$$\xi_t(i, j) = P(q_t = i, q_{t+1} = j | O, \lambda).$$

Using Bayes' rule, $\xi_t(i, j)$ can be given by:

$$\xi_t(i, j) = \frac{P(q_t = i, q_{t+1} = j, O | \lambda)}{P(O|\lambda)}.$$

From the definitions of forward and backward variables, $\xi_t(i, j)$ can be given by:

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)},$$

where $\alpha_t(i)$ accounts for the partial observation sequence $o_1, o_2,..., o_t$ ending in State i at time t; a_{ij} represents the transition to State j; $b_j(o_{t+1})$ represents observing o_{t+1} at time $t+1$; and $\beta_{t+1}(j)$ accounts for observing the partial observation sequence $o_{t+2}, o_{t+3}, ..., o_T$.

By summing $\xi_t(i,j)$ over j, we obtain the probability of being in State i at time t given the observation sequence and the model, denoted by $\gamma_t(i)$:

$$\gamma_t(i) = P(q_t = i | O, \lambda) = \sum_{j=1}^{N} \xi_t(i,j).$$

By summing $\gamma_t(i)$ over the time index t, we get a quantity that can be interpreted as the expected number of times that State i is visited or the expected number of transitions made from state i:

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{ Expected number of transitions made from State } i.$$

Also, the quantity we get from summation of $\xi_t(i,j)$ over the time index from $t = 1$ to $t = T-1$ can be interpreted as the expected number of transitions made from State i to State j:

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{Expected number of transitions made from State } i \text{ to State } j.$$

Based on the concept of counting the event occurrences, and using the above formulas, the parameters of the model can be re-estimated given the training observation sequence as follows:

$\bar{\pi}_i$ = The expected frequency, or number of times, in State i at time $t = 1$

$= \gamma_1(i)$;

$$\bar{a}_{ij} = \frac{\text{Expected number of transitions from State } i \text{ to State } j}{\text{Expected number of transitions from State } i}$$

$$= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)};$$

$$\bar{b}_j(k)=\frac{\text{Expected number of times visiting State } j \text{ and observing symbol } v_k}{\text{Expected number of times visiting State j}}$$

$$=\frac{\sum_{\substack{t=1\\ \text{s.b. } o_t=v_k}}^{T-1}\gamma_t(j)}{\sum_{t=1}^{T-1}\gamma_t(j)};$$

The Baum-Welch algorithm starts from the initial model $\lambda=(\pi,A,B)$ to compute the forward and backward variables and then re-estimates the parameters using the above formula. Denoting the re-estimated model as $\bar{\lambda}=\left(\bar{\pi},\bar{A},\bar{B}\right)$, Baum and his colleagues (1970) proved that $P\left(O\middle|\bar{\lambda}\right)\geq P\left(O\middle|\lambda\right)$, that is, the probability of the observation sequence given the re-estimated model is greater than or equal to the probability of the observation sequence given the initial model. Using $\bar{\lambda}$ in place of λ iteratively and repeating the re-estimation calculation forms an expectation maximisation algorithm that, when repeated until convergence, adjusts the parameters of an HMM corresponding to a local maximum in model likelihood.

By following the steps analogous to the Baum-Welch algorithm, in Section 5.3 we develop an algorithm to train the proposed CHMM using sequences of expert judgements.

3.5.7 Challenges in Implementing HMMs in Condition Based Maintenance

A challenge associated with implementing HMM in practice that yet needs to be investigated is training the models. An issue associated with implementing EM training algorithms (such as the Baum–Welch algorithm, and other event occurrence-based parameter estimation methods in general) in CBM is the availability of the training data. In practice, there is often insufficient number of event occurrences to train reliable model parameters, since systems are subject to preventive maintenance and a replacement is most likely carried out before a failure occurs. Therefore, there are usually a small number of observations available for training, particularly for higher indexed states representing worse levels of deterioration.

For some specific applications it might be feasible to use accelerated life tests and obtain the condition data related to all condition states of a machine. However, for most CBM applications it is not practical to experimentally obtain the condition data associated with all of the deterioration states and operational conditions that a machine undergoes. An example of this would be complex systems for which the accelerated life test is impractical, or when the condition monitoring is contracted out. Therefore, it is important to investigate the amount of training data required for training reliable models in order to have effective diagnostics and prognostics performance.

We will examine this implementation issue for our proposed CHMM in Chapter 6. The effect of the number of observation sequences on the trained model will be examined and possible solutions to improve the training performance will be discussed.

3.6 Summary of Research Gaps

This chapter reviewed the existing modelling approaches for diagnostics and prognostics with emphasis on how they model the relationship between condition data and the underlying health condition of a monitored system.

Recall from Section 1.4, that one of the objectives of this research is to estimate the true condition of a system and to predict the remaining time to failure given expert judgements. In Section 2.4 the role of expert judgement in assessing the system's condition in an industrial practice was described. The subjective assessment of systems's condition leads to qualitative results that can be can be stated by discrete measures, such as grades or classes, and this results in the definition of discrete states.

Incorporation of expert judgement in prognostics and diagnostics has not been widely addressed in academic literature. As mentioned in Sub-Section 3.4.2, Wang and Zhang (2008) addressed the use of expert judgement in prognostics. They considered that expert judgement is provided based on the current condition data and is an indirect assessment of the residual life of a system. Thus, they assumed that the current expert judgement is independent of previous judgements, given the current residual life. They used a stochastic filtering based method to predict the residual life of a monitored system at regular time epochs given expert judgements. The application of this method requires the specification of the probability distribution function and parameter estimation of the probability of expert judgement given the residual life, i.e. $P(Y_t|x_t)$ in addition to the probability distribution function of the delay time.

HMMs can provide a flexible way to model the evolutionary relationship between expert judgement and the underlying condition of a system. The stochastic evolution of the actual system's state and the stochastic relationship between expert judgement and the underlying condition state can be described by the transition and observation probability matrices whose elements can be estimated by the well-established HMMs training algorithm, such as the one described in Paragraph 3.5.6.3. In Chapter 5, we will propose a new CHMM formulation to model the stochastic relationship between expert judgement and the unobservable deterioration condition of a system.

Chapter 4 reviews the models that are related to the final stage of CBM that is maintenance decision-making given the information available from diagnostics and prognostics.

4 Condition Based Maintenance Optimisation

In Chapter 3, we reviewed the models that describe the stochastic relationship between condition data and the actual condition of a monitored system, to be used as a basis for diagnostics and prognostics. These models are related to the technical interpretation of the condition data answering the question "how should the condition monitoring results be interpreted?" (Campbell and Jardine, 2001). This chapter is focused on the final stage of CBM, that is, maintenance decision-making given that the condition data and their interpretation are available. In other words, this chapter reviews the models answering the question "how should the condition monitoring results be acted on?". The aim is to provide an insight on, and identify the existing gaps of CBM optimisation models for partially observable systems.

4.1 Introduction

A maintenance optimisation model is defined as "a mathematical model in which both costs and benefits of maintenance are quantified and an optimum balance between both is obtained" (Dekker, 1996). In this definition maintenance costs include all the costs incurred by activities "intended to retain an item or system in, or restore it to, a state in which it can perform its required function" (EN 13306, 2001). These include, for instance, the costs of performing repairs or conducting inspections. Maintenance benefits consist of the savings on costs that are prevented by maintenance (e.g. costs of replacement of failed equipment, or production loss due to unscheduled breakdowns). The outcome of maintenance optimisation models can be obtained as optimal maintenance policies, or derived as analytical results regarding their structural characteristics (e.g. stationary control-limit policy) (Dekker et al., 1997).

Maintenance optimisation models originated in the early sixties with models optimising Time Based maintenance policies, such as the well-known "Age replacement" and "block replacement" models (Barlow and Proschan, 1965). A number of surveys of Time Based maintenance optimisation models and their applications have been presented by several researchers: Barlow and Proschan (1965), Valdez and Feldman (1989), McCall (1965), Dekker (1996), Scarf (1997), Wang (2002), Nakagawa (2006), and more recently Ahmad and Kamaruddin (2012). In Time Based maintenance, maintenance actions (e.g. preventive replacements) are scheduled based on calendar time, or operational age, such as "cycles" or "cumulative load" (Cooke and Bedford, 2002). Nakagawa (2006) classifies Time Based optimisation models according to the factor used for scheduling the maintenance actions: (1) "time-dependent models" i.e. maintenance actions are scheduled based on the total amount of calendar time the system has been in operation; and (2) "number-dependent models" i.e. maintenance actions are scheduled based on the number of cycles completed by a property of the system.

CBM uses condition data to determine "more precisely the most advantageous moment" for performing maintenance actions (Campbell and Jardine, 2001). Hence CBM optimisation models can help to indicate the best decisions regarding maintenance actions, given the information available from condition monitoring. Recall from Chapter 1 that systems subject to condition monitoring fall into two categories: completely observable systems and partially observable systems. The focus of this research is CBM for partially observable systems, i.e. the system's condition cannot be fully observed or identified through condition monitoring. As discussed and reviewed in Chapter 3, a model is needed to describe the stochastic relationship between condition data and the underlying condition of the monitored system. This model (e.g. PHM, HMM) is used as a basis for diagnostics and prognostics, i.e. to estimate the actual system's condition and its residual life, given condition data. When optimising maintenance policies, interventions of maintenance activities are incorporated into diagnostics and prognostics. This "makes the situation more complicated since extra effort is needed to describe the nature of maintenance policies" (Jardine et al., 2006).

Depending on problem context, CBM optimisation models for partially observable systems have different structures. Apart from the modelling approach for describing the stochastic relationship between condition data and the underlying system's condition, there are some influential factors that characterise these models. We will discuss these factors in Section 4.3. In Section 4.2 we first discuss some general aspects and features in maintenance optimisation models. We will address these later in Section 4.4 to review the CBM optimisation models for partially observable systems and clarify the gaps in literature.

4.2 General Aspects in Maintenance Optimisation Models

According to Dekker (1996), the following aspects are generally covered by maintenance optimisation models:

1. Description of a technical system (e.g. a simple item or a complex mechanical system).
2. Modelling the deterioration of system in time and the possible consequences (e.g. failure probability and costs of replacement or repair of failed equipment and production loss during unscheduled breakdown).
3. Description of the available information about the system and the actions open to management (e.g. replacement and physical inspections revealing the actual deterioration condition of system).
4. An objective function and an optimisation technique that helps in finding the best balance (e.g. the total expected discounted cost over a finite horizon minimised by dynamic programming).

These aspects, along with the modelling approaches for addressing them, characterise maintenance optimisation models. For instance, the deterioration of a system can be considered as continuous stochastic process and modelled as a gamma process or assumed to be a discrete-state stochastic process and modelled as a Markov process. The admissible action could be considered as replacement and inspection conducted at discrete time epochs revealing the actual deterioration condition. Among these aspects, there are some that are specifically related to CBM, such as the types of condition monitoring (e.g. perfect or imperfect). We will discuss these in Section 4.3. There are also some aspects in maintenance optimisation models that are applied to all types of maintenance, i.e. possible maintenance actions in a policy and the planning horizon for policy optimisation. We discuss these aspects in the following sub-sections.

4.2.1 Planning Horizon

Maintenance policies can be optimised over finite or infinite planning horizons and the optimal policies can be obtained as “stationary” or “dynamic” policies (Dekker et al., 1997; Nicolai and Dekker, 2008).

According to Wagner (1970), when optimising “investment” decisions that are repeated in future, it is better to consider an infinite horizon and include the long-term future consequences. Since the optimisation situations (e.g. systems behaviour) usually become steady over the long-term, optimising maintenance policies over an infinite planning horizon makes it possible to obtain a stationary optimal policy. A stationary policy is provided as static decisions for maintenance actions that do not change over the planning horizon. This facilitates derivation of the analytical expressions that specify the optimal policy, as well as its implementation in practice. However, a stationary policy is only valid for a steady situation and hence “short-term” variations such “unexpected opportunities” cannot be taken into account (Dekker et al., 1997) .

When short-term, time-dependent variations need to be taken into account the policies are optimised over a finite horizon and the optimal policy is obtained as a dynamic (time-dependent) policy that may vary over the planning horizon (Dekker et al., 1997). For instance, season-based environmental condition variations such as weather climate can significantly change the lead time to prepare repair resources and hence change the availability of wind turbines (McMillan and Ault, 2008; Byon and Ding, 2010). Therefore, when modelling the policy optimisation, the consequences of short-term availability variations (i.e. production loss) need to be taken into account. This can be done, for instance by assuming varied failure cost in the maintenance optimisation model. In a manufacturing company (e.g. producing ice-cream) the revenue loss could vary due to the short-term changes of the production demand (e.g. reduced demand of ice-cream during winter). These short-term variations can be captured in a dynamic maintenance policy.

4.2.2 Effectiveness of Maintenance Actions

When optimising maintenance policies, maintenance actions with different levels of effectiveness can be considered. Maintenance effectiveness is the degree to which the condition of a system is restored. Pham and Wang (1996) classify maintenance actions according to their effectiveness as :

- *Perfect maintenance*: the operating condition of a system is restored to "as good as new" condition, meaning that the lifetime distribution, deterioration level and failure rate are the same as a brand new system after the maintenance action.
- *Minimal repair or minimal maintenance*: the operating condition of a system is restored to "as bad as old" meaning that the failure rate of a system is not changed and is the same as before performing the maintenance action.
- *Imperfect repair or imperfect maintenance*: the operating condition of a system is restored to somewhere between "as good as new" and "as bad as old".
- *Worse repair or worse maintenance*: the deterioration level of a system or the failure rate increases by performing a maintenance action, but the system does not break down.
- *Worst repair or worst maintenance*: performing the maintenance action causes a breakdown.

In reality most preventive maintenance actions (e.g. cleaning or greasing fan bearings) and repairs (e.g. realigning fan bearings) fall into the category of imperfect maintenance, i.e. if effectively done, they can improve the system health condition. In general, the effectiveness of admissible maintenance actions depends on the system complexity and the maintenance resources available. For instance, for simple non-repairable components (e.g. gears), when preventive maintenance actions such as greasing are not available, the maintenance action choices in an optimisation model are limited to "replacement" and "no action".

4.3 Condition Based Maintenance Optimisation Models for Partially Observable Systems

CBM optimisation models try to optimise the choices of maintenance actions or the time for performing them on the basis of the information collected through condition monitoring. Here we characterise CBM optimisation models by the choices of decisions to be optimised (i.e. maintenance action choices, or the time of performing them) and by the control factors (e.g. residual life) on the basis of which the decisions are optimised. The control factors and decision choices basically depend on the mechanism of the condition monitoring technique. Figure 4.1 illustrates this suggested characterization.

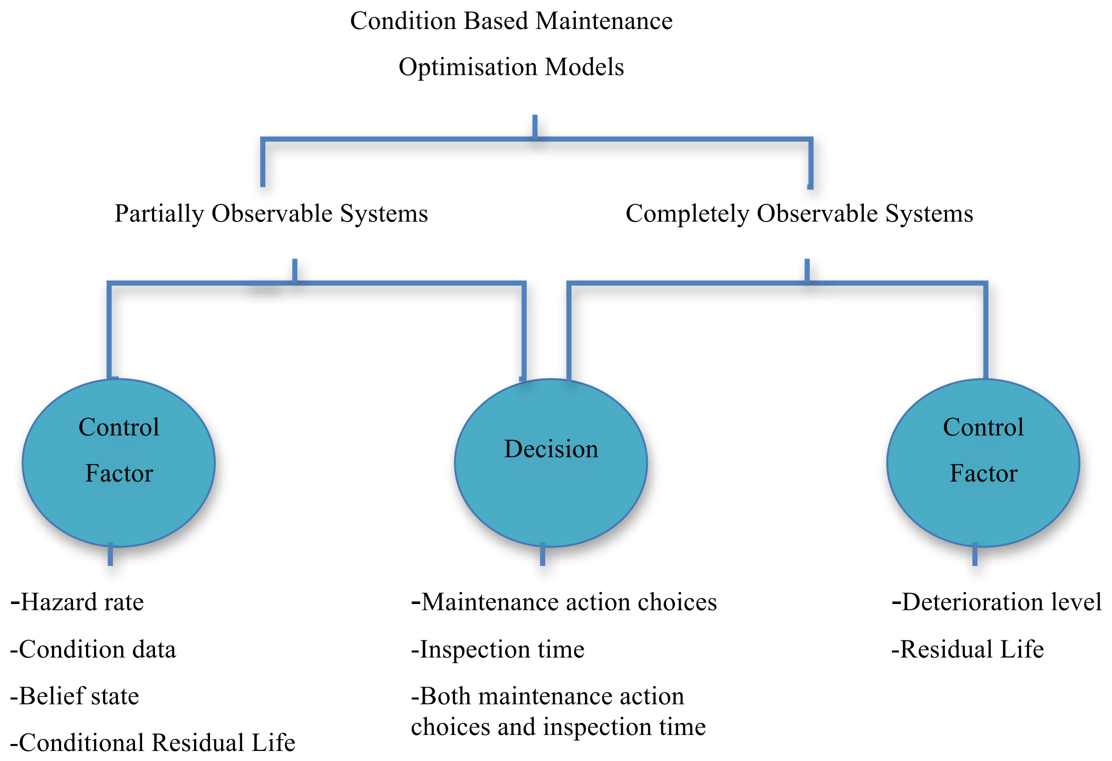

Figure 4.1: Condition Based Maintenance optimisation models.

The decision variables optimised depend on the condition monitoring mechanism. For instance, when the condition data are collected continuously (e.g. continuously recorded vibration signals), the optimisation model tries to optimise the “triggering threshold” for performing maintenance actions. For a system whose condition is monitored through sequential inspections (e.g. oil analysis), the optimisation model can be applied to optimise the time of conducting the inspections and/or the choices of maintenance actions on the basis of the inspection outcomes. For such systems condition monitoring is considered a costly activity and decisions as to what maintenance action to perform and when to conduct the next inspection (i.e. condition monitoring conducted at discrete time epochs) are optimised, on the basis of the condition monitoring outcome. These decisions are limited to the maintenance action choices, if the cost incurred by condition monitoring is not considered in the optimisation models – that is, condition monitoring is assumed to be carried out at predetermined intervals.

A large class of CBM optimisation models assumes that the optimal policy has a control-limit structure and tries to optimise the policy decision variables according to this structure. A “control-limit” policy is defined by a set of thresholds such that if a control factor reaches the threshold certain maintenance actions are carried out. Control-limit optimal CBM policies for completely observable systems are obtained as optimal thresholds of the system deterioration level for performing maintenance actions or scheduling the condition monitoring intervals, see for example Wang (2000); Barata et al. (2002); Dieulle et al. (2003); and Huynh et al. (2011).

For partially observable systems, however, the exact deterioration level is unknown. Several CBM optimisation models for partially observable systems consider condition data (e.g. temperature) or a function of them (e.g. proportional hazard rate) as thresholds to be optimised. In a category of these models referred to as “hazard control-limit type” the optimal CBM policy is obtained as optimal thresholds of hazard rate for performing maintenance actions (Makis and Jardine, 1992; Kumar and Westberg, 1997; Jardine et al., 1999; Vlok et al., 2002). In this approach PHM (see Section 3.5) is used to model the hazard rate as a combination of age and condition data; hence the optimal policy is defined as control-limits over these two factors.

Hazard "control-limit" models were criticised by Scarf (1997), Christer et al. (1997) and Wang (2003) in that only the current value of condition data are taken into account and the rest of the condition history is ignored in maintenance decision-making. To address this problem, Christer et al. (1997) used a state space model and the Kalman filter to estimate and predict the erosion level of the inductors in an induction furnace conditional on all condition data to date. A dynamic optimal policy was proposed to determine the optimal time to perform preventive replacement, on the basis of the estimated erosion condition, when the latest condition data becomes available. Wang (2003) applied a stochastic recursive control model to predict the residual life given all the condition data available and proposed a dynamic policy to determine the monitoring intervals given the predicted residual life distribution.

The Partially Observable Markov Decision Process (POMDP) (Monahan, 1982) is another approach to modelling CBM optimisation in which the whole history of information, including all maintenance actions and condition data, is taken into account. When optimising maintenance policies, POMDPs do not assume certain predetermined structural characteristics of the optimal policies. Hence results can be either obtained as optimal policies, or their structural characteristics, or both. In the following section, we review POMDP models applied to CBM optimisation for partially observable systems.

4.4 Partially Observable Markov Decision Processes (POMDPs)

When system's conditions are monitored at discrete time epochs, CBM policies can be described as a sequential decision-making process through which decisions regarding maintenance activities are made, given the information available from condition monitoring. To exemplify such sequential decision-making, consider a system whose deterioration level can be classified into N discrete states. At each decision time epoch t, the system is in a state $x_t \in \{1,2,...,N\}$, and an action $a_t \in \Phi$ has to be made, where Φ denotes the set of all actions available to the decision maker. Taking the action a_t makes the system randomly move to a new state x_t', incurring a corresponding cost that is dependent on the current state, the new state, and the action taken. The goal is to find the optimal policy for taking the actions, to optimise an objective function, such as the minimum expected total cost over a finite planning horizon. Markov Decision Processes (MDPs) (Puterman, 1994; Sheskin, 2010) enable maintenance policies to be formulated as sequential decision processes; and they find the optimal policy by evaluating a trade-off between immediate and future benefits and costs. These models are originally based on the basic sequential decision model introduced by Derman (1962).

POMDPs (Monahan, 1982; Lovejoy, 1991) are generalisations of MDPs in which it is not assumed that the system state at each decision time epoch is precisely known. Hence, they provide a natural way for formulating CBM policies for partially observable systems. States in POMDPs, i.e. "belief states", are represented by the conditional probability distributions of the current condition states given the historical observations (e.g. condition data and maintenance actions).

In what follows we discuss the POMDPs applied to CBM optimisation for partially observable systems in two categories classified according to the nature of the obtained results. The first category uses the POMDP to establish conditions under which the optimal policy has certain structural characteristics. The second category obtains the optimal policy without assuming a particular structure.

4.4.1 Results on the Optimal Policy Structure

There are many structural results on POMDP models that suggest, given some fairly realistic assumptions about the model parameters such as increasing failure rate, that the optimal policy has a control-limit structure. This feature allows the optimal policy to be represented by a collection of decision rules characterised by functions that relate the belief states to maintenance actions. A control-limit policy for partially observable systems is basically defined as follows (Ohnishi et al., 1986): For each action, there exists a region (control-limit) denoting the sets of all the belief states at which this action is optimal. The regions for POMDPs with two condition states, namely "good" and "bad", are expressed as fixed probability rules stating that if the posterior probability of the system operating in the "good" condition is less than or equal to some threshold value, then a certain action should be taken (Shiryaev, 1968). The regions for POMDPs with more than two states are defined with respect to "likelihood ratio ordering" (Rosenfield, 1976) between the belief states.

Some of the earliest papers investigating the structure of optimal maintenance policies for a two-state discrete time production processes were reviewed by Monahan (1982). Among them, Ross (1971) and White (1979) studied the POMDP for quality control problems for production processes with two condition states. The actions to be optimised at decision epochs were considered as inspection, do nothing or replacement. They analysed the optimal discounted cost over a finite horizon and derived conditions under which the optimal policies have monotone control-limit structures. Ohnishi et al. (1986) extended these models by introducing the concept of condition monitoring and investigated an optimal inspection and replacement policy. However, they assumed that inspection reveals the exact state of a system. They studied the characteristics of the optimal policy that minimises the expected total discounted cost over an infinite horizon and they showed that under certain conditions it has a "monotonic control-limit" structure. Grosfeld-Nir (1996) investigated the replacement policy optimisation for a two-state production process and derived simple equations to find the control-limit policy, under the assumption that the observed value follows a uniform distribution.

Ohnishi et al. (1994) generalised the previous POMDPs by including minimal-repair as a maintenance action choice. The deterioration was modelled as a discrete-time, discrete-state Markov process stochastically related to condition data through a known observation probability matrix. They assumed that the system could be in "available" condition or fail to "unavailable condition" with a certain probability depending on the current deterioration state. Minimal-repair was assumed to recover the system to an available condition preserving it in the same deterioration state. They obtained the condition under which the policy that minimises the expected total discounted cost over an infinite horizon has a control-limit structure.

Ivy and Pollock (2005) included the concept of imperfect repair into the POMDP model and studied the structure of the optimal policy over a finite horizon. They modelled the system deterioration process as a Markovian discrete-time, discrete-state process and used a binomial distribution to describe the stochastic relationship between condition data and the system deterioration condition. Failures were assumed to be "silent", meaning that the system could continue to operate under the failed condition unless some action is taken, and they considered a cost associated with such operation. They established conditions under which the policy that minimises the total expected cost has a control-limit structure. Tamura et al. (2010) also included imperfect repair into a POMDP for a partially observable system with two deterioration conditions, namely "good" and "bad". Imperfect maintenance was assumed to transfer the system to the "good" state with a state-dependent probability, i.e. the result of repair was assumed to be uncertain. They showed that under some constraints of the model parameters, the optimal maintenance policy has monotone properties and can be classified into one of six specified structures.

4.4.2 Derivation of the Optimal Policy

Another class of POMDPs focuses on the applicability of the POMDP models in practice rather than the theoretical results, and obtains the optimal CBM policy by solving the model using approximate methods without assuming a predetermined structure.

Maillart (2006) formulated a POMDP for the optimisation of a CBM policy, including inspection as a decision choice, in addition to replacement. She studied the problem for both perfect inspection, revealing the actual system's condition, and imperfect inspection, obtaining condition data, and obtained the policy that minimises the total expected cost per unit time over an infinite horizon by numerically solving the proposed POMDP. She also derived a closed-form heuristic expression for the perfect inspection problem.

Ghasemi et al. (2007) studied a replacement policy optimisation model formulated as a POMDP in which the stochastic relationship between the condition data and the actual system's condition was modelled by PHM (see Section 3.4). Based on the results obtained by White (1979) and Ohnishi (1994) they derived the optimality conditions for the replacement policy that minimises the long-run expected cost per unit time. They used dynamic programming methods to numerically solve the proposed POMDP.

Recently Zhou et al. (2011) proposed a POMDP with an extended decision space, modelled as a combination of the maintenance action choice (inspection; imperfect repair; replacement) and their corresponding waiting time. To optimise the time to perform actions, instead of the classic Markovian deterioration process they considered a continuous-state deterioration process and adopted a Gamma-based state space model (Zhou et al., 2009) to describe its stochastic behaviour. They used a combination of Monte Carlo-based density projection (Brooks et al., 2006) and policy iteration (Puterman, 1994) to optimise the choices of maintenance actions and the corresponding waiting durations to minimise the long-run expected cost per unit time.

4.5 Summary of Research Gaps

The focus of this chapter was on the decision-making stage of CBM systems, where decisions regarding maintenance actions are made on the basis of the information available from condition monitoring. The existing modelling approaches in CBM policy optimisation, with the focus on partially observable systems, have been reviewed.

The big gap between theory and practice in the field of maintenance optimisation modelling has been pointed out by several researchers who reviewed maintenance optimisation models (Dekker, 1996; Dekker et al., 1997; Sherwin, 2000; Nicolai and Dekker, 2008; Horenbeek et al., 2010; Sharma et al., 2011). Dekker (1996), and Nicolai and Dekker (2008) specifically concluded that case studies are not well represented in maintenance optimisation literature, although this field is applied mathematics. Several papers reported mathematical extensions of existing models, obtained by relaxing some assumptions or adding new features. The papers presenting new models developed or validated based on case studies are very few. Therefore, one of the objectives of this research is to develop a CBM decision optimisation model motivated from an industry application.

In the following paragraphs we discuss a few research directions that have been motivated from the CBM decision-making mechanism described in Section 2.3, and which have not been addressed in literature.

One of the important observations arising from studying the maintenance event database in Chapter 2 was a two-step decision-making procedure carried out at time epochs when an experienced engineer (i.e. expert) interprets the condition data. The result of this decision-making procedure is logged as an inspection (e.g. direct monitoring of the condition of the fan bearings) followed by a maintenance action (e.g. replacement or cleaning the fan bearings). In this CBM policy, an expert's assessment of the system's condition is provided at discrete time epochs, and based on this a choice of inspection is selected to further investigate and verify the system's condition. If an inspection is carried out, a maintenance action is then selected on the basis of the inspection outcome. Such a sequential procedure is also applicable to other CBM systems, where different condition monitoring techniques with different costs and precision, and different maintenance action types with different costs and effectiveness are performed. In such situations, a policy optimisation model can be of help to support cost-effective decision-making with regard to the inspection choices, and maintenance action choices selected on the basis of the inspection results.

POMDPs provide a natural way for formulating and optimising sequential CBM decision-making procedures for partially observable systems, when the condition data or their interpretations are provided at discrete time epochs. Many POMDP models have been proposed and developed for CBM optimisation, with a wide variety of underlying assumptions (as reviewed in Section 4.4). However, the sequential decision-making throughout a decision interval, i.e. selection of the inspection types followed by selection of the maintenance actions, has not been investigated within this body of literature.

Maillart (2006) addressed the problem of sequential decision-making during a decision period but with limited action choices. She only considered "no action", "imperfect inspection" and "replacement" as the action choices at the beginning of a decision period, and "no action" and "replacement" as the admissible action choices after the inspection (if any) is performed. Most of the CBM optimisation models developed as POMDPs assume replacement as the only admissible maintenance action. Although in reality most of the preventive maintenance actions and repairs are classified as imperfect maintenance, very few considered imperfect maintenance as an action choice in the POMDPs. For instance, Zhou et al. (2011) included imperfect repair into the POMDP model, but, similar to Maillart (2006), they only considered one type of inspection and assumed that the selection of inspection and maintenance actions is made at a single decision step.

Motivated by the CBM industry application in Chapter 2, we develop a CBM optimisation model in Chapter 7 that explicitly addresses the gaps discussed above. Essentially, we propose to model a POMDP where at the beginning of sequential decision time epochs, followed by expert judgement, the decisions regarding inspection choices and maintenance actions are made in two consecutive steps. Particularly, we consider "no inspection", "perfect inspection" and "imperfect inspection" as the inspection choices, and "no action", "imperfect repair" and "replacement" as the maintenance action choices admissible after the inspection. We obtain the optimal policy over a finite planning horizon and explore its structural characteristics in Chapter 8.

5 A Coupled Hidden Markov Model Capturing the Interaction Between Expert Judgement and the Deterioration Condition of a System

In this chapter a new Coupled Hidden Markov Model (CHMM) formulation is proposed to model the stochastic relationship between expert judgement and the underlying deterioration state of the system. The proposed CHMM aims to enrich the capabilities of standard HMMs by introducing a new modelling structure while utilising the well-established methodologies of HMMs (e.g. forward-backward procedure).

Consider a mechanical system subject to deterioration and sudden failure, where the deterioration condition of the system can be classified into N unobservable states, with State 1 representing the "as good as new" condition and State N denoting the worst health condition of the system. The system can be a component (e.g. a gear) or a piece of mechanical equipment (e.g. a gearbox). We assume an experienced engineer assesses the condition of the system at regular discrete time epochs. We refer to the experienced engineer's assessment on the system's state as "expert judgement state" and we assume that it is provided as a positive integer number, $y_t \in \{1, 2, ..., N\}$ at time t.

In Section 5.1 we describe the conceptual architecture of the problem we are concerned with. In Section 5.2 we present the formulation of the proposed CHMM. In Section 5.3 we develop a training algorithm to estimate the model parameters. The derivation of the conditional probability distribution of the remaining time to failure, based on the proposed modelling approach, is presented in Section 5.4.

5.1 Conceptual Architecture of the Model

Let $x_t \in \{1, 2, ..., N\}$ denote the system's deterioration state at time t. In this modeling framework, we assume that the state transition of the deterioration process takes place at the beginning of a time epoch. When assessing the system's condition at the beginning of a time epoch, the expert is also influenced by his judgement at the previous time epoch. That is, the expert judgement state at time t, y_t, is dependent on the expert judgement state at the previous time epoch y_{t-1}, and the deterioration state of the system x_t. We also assume that system's failure occur at the beginning of a time epoch, and if it occurs, it will be detected by the expert when evaluating the system's condition. Hence, an observation in this model is considered to be the combination of the expert judgement state, i.e. a positive integer number, and the status of the system as failed or survived.

The probabilistic inference graph in Figure 5.1, which is rolled out in time, illustrates the interaction between the true deterioration state of the system and the expert judgement over discrete time intervals. The square nodes represent the observation variable u_t and the circular nodes represent the true deterioration state of the system x_t and the expert judgement state y_t, at time t. The horizontal arrows represent the conditional probabilities between variables. The vertical arrows coming from x_t to y_t and u_t model the influence imposed by the system's deterioration state to the expert judgement state and the observation.

An alternative assumption, which could generalise this modelling framework, would be assuming that the assessments are carried out by multiple experts. This, for example, could represent the scenario when engineers with different levels of experience, or different general viewpoints (e.g. optimist or pessimist), assess the deterioration condition of the system. In this case, multiple expert judgement processes would be considered. Therefore, the interaction between the expert judgement states and the system's deterioration states, across time, would be modeled by different conditional probabilities.

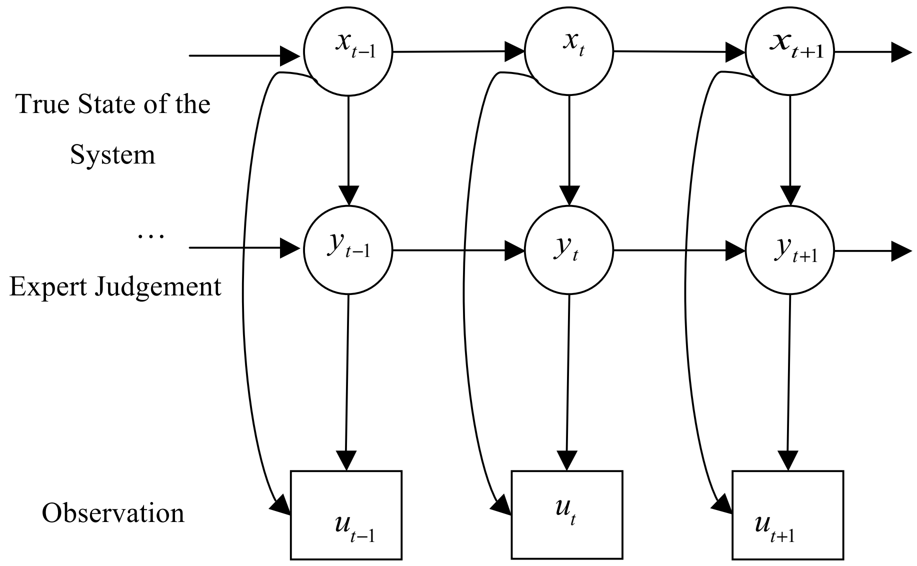

Figure 5.1: Probabilistic inference graph demonstrating the interaction between the true states of the system, expert judgement states and observations over regular discrete time intervals.

5.2 Theoretical Structure

We assume that the evolution of the system's condition and expert judgement follow a Markovian stochastic process. The Markov process is chosen because of its ability to graphically and mathematically describe the evolution of the system and expert judgement with discrete states over time. In this section we formulate the hierarchical structure depicted in Figure 5.1 as a CHMM in which the deterioration condition of the system and expert judgement are considered as coupled Markov processes, and an observation process is affected by both expert judgement and systems' deterioration processes. These two processes are coupled by introducing the conditional probabilities between their state variables. Let (X_t, Y_t) denote the joint state of the proposed CHMM at time t. X_t and Y_t are random variables, and x_t and y_t denote their realisations. To simplify the notation we usually drop the random variables; so for example $P(X_t = x_t, Y_t = y_t | X_{t-1} = x_{t-1}, Y_{t-1} = y_{t-1})$ will be written as $P(x_t, y_t | x_{t-1}, y_{t-1})$. The transition probabilities between the joint states and the observation probabilities for the proposed CHMM are obtained in Sub-Section 5.2.1 and 5.2.2 respectively.

5.2.1 Transition Probabilities

The probabilistic inference graph in Figure 5.1 describes the underlying assumption regarding the interaction between the true deterioration states of the system and expert judgement states. The vertical arrows coming from x_t to y_t represent the probabilistic relationship between the current expert judgement state and the current deterioration state of the system. The transition probability between the joint states over one time unit can be given as follows, using the chain rule:

$$P\left(x_t, y_t \middle| x_{t-1}, y_{t-1}\right) = P\left(y_t \middle| x_t, x_{t-1}, y_{t-1}\right) P\left(x_t \middle| x_{t-1}, y_{t-1}\right). \tag{5.1}$$

As illustrated in Figure 5.1, the current expert judgement is only dependent on the current true state of the system and the expert judgement state provided in the previous time epoch. Also, the deterioration process of the system is assumed to follow Markovian evolution, and hence the probability of the current system's deterioration state, given the system's deterioration state in the previous time epoch, is independent of other information. Therefore Equation (5.1) can be simplified to

$$P\left(x_t, y_t \middle| x_{t-1}, y_{t-1}\right) = P\left(y_t \middle| x_t, y_{t-1}\right) P\left(x_t \middle| x_{t-1}\right). \tag{5.2}$$

We assume that the true state of the system, i.e. the system's deterioration state, can either degrade to the next deterioration state or remain in the same state, over one time unit. This assumption implies that the system cannot improve on its own (without any maintenance intervention). Assuming that the system is in State i, $1 \le i \le N$, at time $t-1$, let p_i denote the probability of the system remaining at the same state and not moving to another state, at the end of the time epoch starting at time $t-1$, and $\overline{p}_i$ be the complementary probability of p_i,

$$\begin{aligned} p_i &= P\left(x_t = i \middle| x_{t-1} = i\right), \\ \overline{p}_i &= P\left(x_t \ne i \middle| x_{t-1} = i\right), \quad 1 \le i \le N. \end{aligned} \tag{5.3}$$

The above assumption regarding the transition of the system's deterioration states is valid in practice if the time unit is defined small enough (e.g. as a day). Alternatively, we could assume that the system could degrade to any higher indexed states, over one time unit. In this case the transition probabilities between the deterioration states would be represented by the elements of an upper triangular matrix.

The final state is considered to be the absorbing state, i.e. $p_N = 1$. Figure 5.2 graphically shows the evolution of the true state of the system as a "Left-to-Right" Markov process with N discrete states. The oval nodes represent the state labels while the arrows represent the transition probabilities between the states.

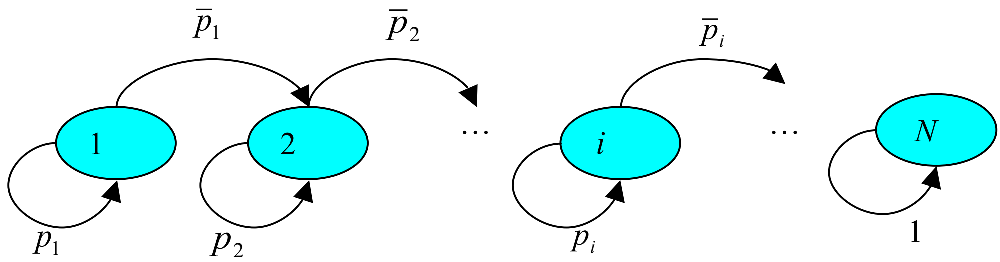

Figure 5.2: State transition diagram of the deterioration process of a system with N states.

As for the deterioration process, we assume that the evolution of the expert judgement also follows Markovian behaviour. As seen in Figure 5.1, it is assumed that the expert judgement state at time t, y_t, is dependent on the expert judgement state at the previous time epoch, y_{t-1}, and the current true state of the system, x_t. We introduce q_k^i, $1 \le i, k \le N$, to denote the probability that the expert judgement remains in State k if it was in this state at the previous time epoch, given that the system is in State i. Note that in this definition the timing has been considered to reflect the assumption that the state transition of the deterioration process takes place at the end of the time epoch starting at time $t-1$, just before expert judgement y_t is provided at the beginning of the time epoch starting at time t. Let $\bar{q}_k^i$ be the complementary probability of q_k^i, they are formally defined as

$$q_k^i = P\left(y_t = k \middle| y_{t-1} = k, x_t = i\right),$$
$$\bar{q}_k^i = P\left(y_t \neq k \middle| y_{t-1} = k, x_t = i\right),\ 1 \le i, k \le N. \tag{5.4}$$

As for the deterioration process, the final expert judgement state, State N, is considered to be the absorbing state, $q_N^i = 1,\ 1 \le i \le N$. Defining the transition probabilities for the expert judgement states in this manner comes directly from the conceptual structure of dependencies depicted in Figure 5.1. It reflects the modelling assumption that the expert judgement process has its own internal dynamic while still being influenced by the deterioration process of the system. Particularly, it implies that the expert makes only "degrading judgements" meaning that he believes the condition of the system cannot be improved over time without any maintenance intervention. Hence assuming that $y_{t-1} = k,\ 1 \le k \le N,\ y_t$ will either remain in State k or degrades to the next state. Figure 5.3 demonstrates the state transition of the expert judgement as a "Left-to-Right" Markov process. The oval nodes represent the expert judgement state labels, and the arrows represent the transition probabilities between the expert judgement states, with i representing the current deterioration state of the system.

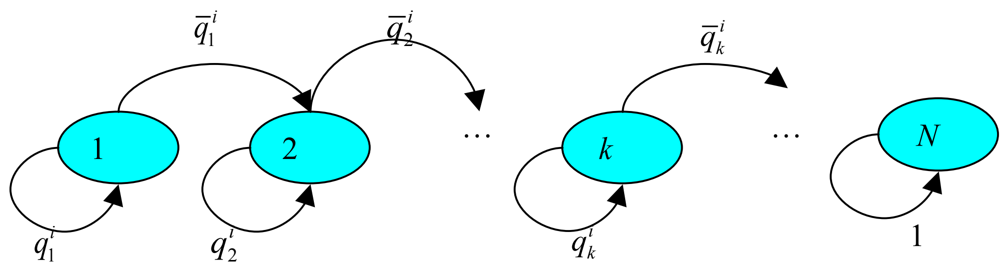

Figure 5.3: State transition diagram of the expert judgement process, given that the system is in State i.

Returning to the joint state transition probability in Equation (5.2), we introduce $a_{ik,jl},\ 1 \le i, j, k, l \le N$, as the transition probability from the joint state $\left(x_t = i, y_t = k\right)$ to the joint state $\left(x_{t+1} = j, y_{t+1} = l\right)$ over a unit of time,

$$a_{ik,jl} = P\left(x_{t+1} = j, y_{t+1} = l \middle| x_t = i, y_t = k\right),\ 1 \le i, j, k, l \le N. \tag{5.5}$$

According to the definitions in Equations (5.3) and (5.4) the transition probability between the joint states can be explicitly given by the model parameters as:

$$a_{ik,jl} = P\left(x_{t+1} = j, y_{t+1} = l \middle| x_t = i, y_t = k\right)$$

$$= \begin{cases} p_i q_k^j & \text{if } j = i \text{ and } l = k, \\ \bar{p}_i q_k^j & \text{if } j = i+1 \text{ and } l = k, \\ p_i \bar{q}_k^j & \text{if } j = i \text{ and } l = k+1, \\ \bar{p}_i \bar{q}_k^j & \text{if } j = i+1 \text{ and } l = k+1, \\ 0 & \text{otherwise.} \end{cases} \qquad 1 \le i, j, k, l \le N, \tag{5.6}$$

5.2.2 Observation Probabilities

Recall from Section 5.1, we assume that failures will be only identified by the expert when evaluating the condition of the system at discrete time epochs, and hence the observation in this model is considered to be a combination of the expert judgement state and the status of the system as failed or survived. Thus we define the set of observation symbols, denoted by V, as

$$V = \left\{kS, kF\right\},\ 1 \le k \le N, \tag{5.7}$$

where kS means that the expert judgement is in State k, $1 \le k \le N$, and the system is still working, and kF means that the expert judgement is in State k and the system has failed.

We assume that the system fails in State i, $1 \le i \le N$, with probability F_i, and $\bar{F}_i$ denotes its complementary probability or the probability of survival in State i, i.e. $\bar{F}_i = 1 - F_i$.

Let $b_{ik}(u_t)$ be the probability of observing u_t given that the Coupled Hidden Markov process is in the joint State $\left(x_t = i, y_t = k\right)$, $1 \le i, k \le N$, hence the probability of the observation symbols is given by

$$\begin{aligned} b_{ik}\left(u_t = kS\right) &= P\left(u_t = kS \middle| x_t = i, y_t = k\right) = \bar{F}_i, \\ b_{ik}\left(u_t = kF\right) &= P\left(u_t = kF \middle| x_t = i, y_t = k\right) = F_i,\ \ 1 \le i, k \le N. \end{aligned} \tag{5.8}$$

As presented in this section, the dimensionality of the problem is reduced considerably by the way in which the transition and observation probabilities are parameterised. Gathering together all the parameters, the model can be indicated by a set of parameters, as

$$\lambda = \left\{ p_i, q_k^i, F_i \right\},\ 1 \le i, k \le N, \tag{5.9}$$

where λ is the complete set of parameters and is used to represent the proposed CHMM. The initial probabilities across the states are not crucial for this model. This is because we assume that the deterioration process of the system starts in time 0 when the system is put into service in the "as good as new" condition, i.e. State 1.

5.3 CHMM Training Algorithm

Given an observation sequence $U = u_1, u_2, \ldots u_T$, where $u_t \in V$ is the observation symbol at time t, the problem is to find the optimal set of model parameters that maximises the probability of the observation sequence given the model, $P(U|\lambda)$. Recall from Paragraph 3.5.6.3 that there is no known way for an HMM to analytically solve this problem. However, using some iterative procedure such as the Expectation Maximisation algorithm (EM), known as Baum–Welch for the standard HMMs, it is possible to improve the model parameters to locally maximise the likelihood of the observation sequence (Baum, 1970; Cappé et al., 2005; Rabiner, 1989)

Using an initial guess of model parameters, the Baum–Welch algorithm first estimates the likelihood of hidden states given the observation sequence, and then uses the expected counts of state transitions and observations to estimate the parameters. Since the expected counts can be derived from the parameters and vice versa, the procedure can be iterated to move from an initial guess of the parameters to a better estimate that (locally) maximises $P(U|\lambda)$(Cappé et al., 2005). Dempster et al. (1977) proved that the EM algorithm is guaranteed to increase the likelihood of the observation sequence at each iteration until a local maximum is reached. In practice, convergence is declared when the difference of the probabilities of the observation sequence given the trained model parameters at consecutive iterations becomes less than a predefined convergence threshold.

In this section we develop an EM algorithm to train the proposed CHMM by following steps analogous to the Baum–Welch algorithm (as described in Paragraph 3.5.6.3). In Sub-Section 5.3.1 we derive forward and backward variables, to perform the inference for our CHMM (forward and backward variables for standard HMMs were defined in Sub-Section 3.5.6). In Sub-Section 5.3.2 we introduce and derive the posteriori probability measures using the forward and backward variables. The computation of the forward and backward variables and the posteriori probability measures form the E (expectation) step of the EM algorithm. At the M (maximisation) step, the parameters are re-estimated using the probability measures computed at the E step. In Sub-Section 5.3.3 we derive the re-estimation formulas for our CHMM parameters by intuitive Bayesian posteriori re-estimation.

5.3.1 Extended Forward-backward Procedure

Let $U = u_1, u_2, \ldots u_T$ be a sequence of observations where $u_t \in V$ is the observation symbol at time t. To solve the likelihood $P\left(U \middle| \lambda\right)$ and compute the posteriori probability measures, we modify the standard forward-backward procedure for our CHMM. Let $\tilde{\lambda}$ denote the set of an initial guess of model parameters, i.e. $\tilde{\lambda} = \left\{\tilde{p}_i, \tilde{q}_k^j, \tilde{F}_i\right\}$. Given the initial parameter values, the transition and observation probabilities, denoted by $\tilde{a}_{ik,jl}$ and, $\tilde{b}_{ik}(u_t)$, can be computed using Equations (5.6) and (5.8).

Given a sequence of observations $U = u_1, u_2, \ldots u_T$ at each time t, we define the forward variable $\alpha_t(ik)$ as the joint probability of the observations up to time t and the coupled states $\left(x_t = i, y_t = k\right)$ given the model $\tilde{\lambda}$. That is

$$\alpha_t(ik) = P(u_1, u_2, \ldots u_{t,} x_t = i, y_t = k \mid \tilde{\lambda}), \quad 1 \le i, k \le N, 1 < t \le T. \tag{5.10}$$

Starting from the first observation at time $t = 1$, we have

$$\alpha_1(ik) = P(u_1, x_1 = i, y_1 = k \mid \tilde{\lambda})$$
$$= P(x_1 = i, y_1 = k \mid \tilde{\lambda}) P(u_1 \mid x_1 = i, y_1 = k, \tilde{\lambda}), \quad 1 \le i,k \le N. \tag{5.11}$$

As mentioned before, it is assumed that the deterioration and the expert judgement processes start in the "as good as new" state, therefore we have:

$$\alpha_1(ik) = \begin{cases} \tilde{b}_{ik}(u_1) & \text{for } i,k = 1 \text{ and} \\ 0 & \text{otherwise.} \end{cases} \tag{5.12}$$

At each time t, $1 < t \le T$, we can calculate the forward variable $\alpha_{t+1}(jl)$ as:

$$\alpha_{t+1}(jl) = P\left(u_1, u_2, ..., u_{t+1}, x_{t+1} = j, y_{t+1} = l \middle| \tilde{\lambda}\right)$$
$$= \sum_{i=1}^{N} \sum_{k=1}^{N} P\left(u_1, u_2, ..., u_t, u_{t+1}, x_t = i, y_t = k, x_{t+1} = j, y_{t+1} = l \middle| \tilde{\lambda}\right)$$
$$= \sum_{i=1}^{N} \sum_{k=1}^{N} P\left(u_1, u_2, ..., u_t, x_t = i, y_t = k \middle| \tilde{\lambda}\right) \underbrace{P\left(u_{t+1}, x_{t+1} = j, y_{t+1} = l \middle| u_1, u_2, ..., u_t, x_t = i, y_t = k, \tilde{\lambda}\right)}_{\text{Term 1}}. \tag{5.13}$$

Term 1 in Equation (5.13) can be given as follows, using the chain rule:

$$P\left(u_{t+1}, x_{t+1} = j, y_{t+1} = l \middle| u_1, u_2, ..., u_t, x_t = i, y_t = k, \tilde{\lambda}\right)$$
$$= P\left(x_{t+1} = j, y_{t+1} = l \middle| u_1, u_2, ..., u_t, x_t = i, y_t = k, \tilde{\lambda}\right)$$
$$\times P\left(u_{t+1} \middle| x_{t+1} = j, y_{t+1} = l, u_1, u_2, ..., u_t, x_t = i, y_t = k, \tilde{\lambda}\right). \tag{5.14}$$

As seen in Figure 5.1, it is assumed that the observation at each time period, given the current deterioration and expert judgement states, is independent of previous observations and coupled states. Also, according to Markovian evolution of the coupled states, Equation (5.13) can be given as

$$\alpha_{t+1}(jl) = \sum_{i=1}^{N} \sum_{k=1}^{N} P\left(u_1, u_2, ..., u_t, x_t = i, y_t = k \middle| \tilde{\lambda}\right) P\left(x_{t+1} = j, y_{t+1} = l \middle| x_t = i, y_t = k, \tilde{\lambda}\right)$$
$$\times P\left(u_{t+1} \middle| x_{t+1} = j, y_{t+1} = l, \tilde{\lambda}\right), \qquad 1 \le j,l \le N, \ 1 < t \le T, \tag{5.15}$$

and therefore, based on the definitions in Equations (5.6), (5.8) and (5.10), the forward variable at each time t, $1 < t \le T$, can be computed through the following recursive equation:

$$\alpha_{t+1}(jl) = \left[\sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_t(ik) \tilde{a}_{ik,jl} \right] \tilde{b}_{jl}(u_{t+1}), \qquad 1 \le j,l \le N,\ 1 < t \le T. \tag{5.16}$$

It follows from the definition of the forward variable that the likelihood of the observation sequence given the model λ can be given by

$$\begin{aligned} P(U|\lambda) &= \sum_{i=1}^{N} \sum_{k=1}^{N} P(u_1, u_2, ..., u_T, x_T = i, y_T = k|\lambda) \\ &= \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_T(ik). \end{aligned} \tag{5.17}$$

We define the backward variable $\beta_t(ik)$ as the probability of the partial observation sequence $u_{t+1}, u_{t+2}, ..., u_T$ given the coupled states $(x_t = i, y_t = k)$ and the model $\tilde{\lambda}$. Starting from the last observation in the sequence, and going back to the first one, $\beta_t(ik)$ is formally defined as:

$$\beta_t(ik) = P(u_{t+1}, u_{t+2}, ..., u_T|x_t = i, y_t = k, \lambda),\ 1 \le i,k \le N,\ 1 \le t < T,$$

and $\beta_T(ik) = 1$ for all $1 \le i,k \le N$.

(5.18)At

each time t, $1 \le t < T$, we can compute the backward variable $\beta_t(ik)$ as follows:

$$\begin{aligned} \beta_t(ik) &= P(u_{t+1}, u_{t+2}, ..., u_T|x_t = i, y_t = k, \tilde{\lambda}) \\ &= \sum_{j=1}^{N} \sum_{l=1}^{N} P(x_{t+1} = j, y_{t+1} = l, u_{t+1}, u_{t+2}, ..., u_T|x_t = i, y_t = k, \tilde{\lambda}) \\ &= \sum_{j=1}^{N} \sum_{l=1}^{N} P(x_{t+1} = j, y_{t+1} = l|x_t = i, y_t = k, \tilde{\lambda}) \underbrace{P(u_{t+1}, u_{t+2}, ..., u_T|x_{t+1} = j, y_{t+1} = l, x_t = i, y_t = k, \tilde{\lambda})}_{\text{Term 1}}, \end{aligned}$$

(5.19)

Using the chain rule, Term 1 in Equation (5.19) can be given as

$$
\begin{aligned}
&P(u_{t+1},u_{t+2},...u_T \mid x_{t+1}=j, y_{t+1}=l, x_t=i, y_t=k, \tilde{\lambda}) \\
&= P(u_{t+1} \mid x_{t+1}=j, y_{t+1}=l, x_t=i, y_t=k, \tilde{\lambda}) P(u_{t+2},...,u_T \mid u_{t+1}, x_{t+1}=j, y_{t+1}=l, x_t=i, y_t=k, \tilde{\lambda})
\end{aligned}
\tag{5.20}
$$

Recall that the observation at each time given the current coupled state is independent of previous coupled states and observations, thus Equation (5.20) can be simplified to

$$
\begin{aligned}
&P(u_{t+1},u_{t+2},...u_T \mid x_{t+1}=j, y_{t+1}=l, x_t=i, y_t=k, \tilde{\lambda}) \\
&= P(u_{t+1} \mid x_{t+1}=j, y_{t+1}=l, \tilde{\lambda}) P(u_{t+2},...,u_T \mid x_{t+1}=j, y_{t+1}=l, \tilde{\lambda})
\end{aligned}
\tag{5.21}
$$

By substituting Equation (5.21) into (5.19), the backward variable, for $1 \le t < T$, can be computed by recursion as

$$
\begin{aligned}
\beta_t(ik) &= P\left(u_{t+1},u_{t+2},...,u_T \middle| x_t=i, y_t=k, \tilde{\lambda}\right) \\
&= \sum_{j=1}^{N}\sum_{l=1}^{N} P\left(x_{t+1}=j, y_{t+1}=l \middle| x_t=i, y_t=k, \tilde{\lambda}\right) \\
&\qquad \times P\left(u_{t+1} \middle| x_{t+1}=j, y_{t+1}=l, \tilde{\lambda}\right) P\left(u_{t+2},...,u_T \middle| x_{t+1}=j, y_{t+1}=l, \tilde{\lambda}\right) \\
&= \sum_{j}\sum_{l} \tilde{a}_{ik,jl} \tilde{b}_{jl}(u_{t+1}) \beta_{t+1}(jl), \qquad 1 \le i,k \le N,\ 1 \le t < T.
\end{aligned}
\tag{5.22}
$$

5.3.2 Posteriori Probability Measures

Given the observation sequence, initial model $\tilde{\lambda} = \left\{ \tilde{p}_i, \tilde{q}_k^j, \tilde{F}_i \right\}$ and the corresponding transition and observation probabilities $\tilde{a}_{ik,jl}$ and $\tilde{b}_{ik}\left(u\right)$, we introduce two posteriori probabilities to compute the expected counts needed for parameter re-estimation.

Let $\xi_t(ik, jl)$ be the probability of being in the joint state $(x_t = i, y_t = k)$ at time t and in the joint state $(x_{t+1} = j, y_{t+1} = l)$ at time $t+1$, given the observation sequence and the model parameters, that is

$$\begin{aligned}\xi_t(ik, jl) &= P\left(x_t = i, y_t = k, x_{t+1} = j, y_{t+1} = l \middle| U, \lambda\right) \\ &= \frac{P\left(x_t = i, y_t = k, x_{t+1} = j, y_{t+1} = l, U \middle| \lambda\right)}{P\left(U \middle| \lambda\right)} \\ &= \frac{P\left(x_t = i, y_t = k, x_{t+1} = j, y_{t+1} = l, U \middle| \lambda\right)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N} P\left(x_t = i, y_t = k, x_{t+1} = j, y_{t+1} = l, U \middle| \lambda\right)}.\end{aligned} \tag{5.23}$$

From the definitions of the forward and backward variables, $\xi_t(ik, jl)$ can be given by

$$\xi_t(ik, jl) = \frac{\alpha_t(ik)\tilde{a}_{ik,jl}\tilde{b}_{jl}(u_{t+1})\beta_{t+1}(jl)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N}\alpha_t(ik)\tilde{a}_{ik.jl}\tilde{b}_{jl}(u_{t+1})\beta_{t+1}(jl)}, \tag{5.24}$$

where $\alpha_t(ik)$ accounts for the partial observation sequence $u_1, u_2, ..., u_t$, ending at the joint state $(x_t = i, y_t = k)$ at time t; $\tilde{a}_{ik,jl}$ represents the transition to the joint state $(x_{t+1} = j, y_{t+1} = l)$ from the joint state $(x_t = i, y_t = k)$; $\tilde{b}_{jl}(u_{t+1})$ represents observing u_{t+1} at time $t+1$; and $\beta_{t+1}(jl)$ accounts for observing the partial observation sequence $u_{t+2}, u_{t+3}, ..., u_T$.

By summing $\xi_t(ik, jl)$ over the joint state $(x_{t+1} = j, y_{t+1} = l)$ we obtain the probability of being in the joint state $(x_t = i, y_t = k)$ at time t, given the observation sequence and the model parameters, denoted by $\gamma_t(ik)$,

$$\gamma_t(ik) = P\left(x_t = i, y_t = k \middle| U, \lambda\right) = \sum_{j=1}^{N}\sum_{l=1}^{N}\xi_t(ik, jl). \tag{5.25}$$

The quantity obtained from summing $\gamma_t(ik)$ over the time index for $1 \le t \le T-1$ can be interpreted as the expected number of times that the joint state (i, k) is visited or the expected number of transitions made from the joint state (i,k), given the observation sequence and $\tilde{\lambda}$,

$$\sum_{t=1}^{T-1} \gamma_t(ik) = \text{Expected number of transitions from the coupled state } (i,k). \quad (5.26)$$

Similarly, if we sum $\xi_t(ik, jl)$ over time, $1 \le t \le T-1$, we obtain a quantity that can be interpreted as the expected number of transitions from the joint state (i,k) to the joint state (j,l) given the observation sequence U, and model $\tilde{\lambda}$,

$$\sum_{t=1}^{T-1} \xi_t(ik, jl) = \text{Expected number of transitions from the joint state } (i,k) \text{ to } (j,l). \quad (5.27)$$

5.3.3 Parameter Re-Estimation

The probability measures in Equations (5.24) to (5.27) are computed using the initial value of the parameters; $\tilde{p}_i, \tilde{q}_k^j, \tilde{F}_i$. Based on the concept of event occurrences, it is possible to re-estimate the parameters using the posteriori probabilities computed in Sub-Section 5.3.2.

Recall from Equation (5.3) that p_i is the probability of the system remaining at State i over one unit of time. Based on the concept of frequencies of event occurrences, the estimated value of p_i, denoted by $\hat{p}_i$, is given by

$$\hat{p}_i = \frac{\text{Expected number of times that the system remains in State } i \text{ and does not move to another state at the begining of the next time epoch.}}{\text{Expected number of times that State } i \text{ is visited.}} . \quad (5.28)$$

This equation can be expressed in terms of the transitions between the coupled states:

$$\hat{p}_i = \frac{\sum_{k=1}^{N}\sum_{l=1}^{N} \text{Expected number of transitions from the joint State } (i,k) \text{ to } (i,l)}{\sum_{k=1}^{N} \text{Expected number of transitions from the joint State } (i,k)} . \quad (5.29)$$

Using Equations (5.26) and (5.27), we have:

$$\hat{p}_i = \frac{\sum_{t=1}^{T-1}\sum_{l=1}^{N}\sum_{k=1}^{N}\xi_t(ik,il)}{\sum_{t=1}^{T-1}\sum_{k=1}^{N}\gamma_t(ik)}. \tag{5.30}$$

Recall from Equation (5.4) that q_k^j is the probability that the expert judgement remains in State k if it was in this state at the previous time epoch, given that the system is in State j. Based on the concept of frequencies of event occurrences, the estimated value of q_k^j, denoted by $\hat{q}_k^j$, can be computed using the expected number of transitions between the joint states as:

$$\hat{q}_k^j = \frac{\sum_{i=1}^{N}\text{Expected number of transitions from the joint State }(i,k)\text{ to }(j,k)}{\sum_{i=1}^{N}\sum_{l=1}^{N}\text{ Expected number of transitions from the joint State }(i,k)\text{ to }(j,l)}. \tag{5.31}$$

From the definitions in Equation (5.27), Equation (5.31) can be given as:

$$\hat{q}_k^j = \frac{\sum_{t=1}^{T-1}\sum_{i=1}^{N}\xi_t(ik,jk)}{\sum_{t=1}^{T-1}\sum_{i=1}^{N}\sum_{l=1}^{N}\xi_t(ik,jl)}. \tag{5.32}$$

Finally, F_j, i.e. the probability of failure at State j, based on the concept of frequencies of event occurrences, can be estimated as:

$$\hat{F}_j = \frac{\text{Expected number of times that the system fails in State } j}{\text{Expected number of times that State } j \text{ is visited}}. \tag{5.33}$$

This expression can be given in terms of the events according to the joint states:

$$\hat{F}_j = \frac{\sum_{l=1}^{N}\text{Expected number of times that the system fails in the joint State }(j,l)}{\sum_{l=1}^{N}\text{ Expected number of times that the joint State }(j,l)\text{ is visited}}. \tag{5.34}$$

As defined in Equation (5.7) the observation symbol kF means that the expert judgement state is k, $1 \le k \le N$, and the system has failed. Assuming that the system cannot improve on its own, and upon a failure a replacement is carried out, the observation symbol kF can only be observed at the end of an observation sequence, i.e. $u_t = kS$ for all $1 \le t < T$, where T is the length of the observation sequence. In other words, the last observation is either $u_T = kF$ when the observation sequence has ended with a failure, or $u_T = kS$ when the observation sequence completed before failure occurred.

We previously defined $\xi_t(ik, jl)$ as the probability of being in the joint state $(x_t = i, y_t = k)$ at time t and in the joint state $(x_{t+1} = j, y_{t+1} = l)$ at time $t+1$, given the observation sequence and the model parameters. Therefore, if we sum $\xi_t(ik, jl)$ over time t, for $1 \le t \le T-1$ that $u_{t+1} = lF$, we obtain a quantity that can be interpreted as the expected number of transitions from the joint state (i,k) to the joint state (j,l) and observing $u_{t+1} = lF$. If we sum this quantity again over the joint state (i,k) we obtain what could be interpreted as the expected number of times that the joint state (j,l) is visited and the symbol lF is observed, i.e. the expected number of times that the joint state (j,l) is visited and the system fails in the deterioration State j. Hence Equation (5.34) can be given by:

$$\hat{F}_j = \frac{\sum\limits_{\substack{t=1 \\ s.b. \\ u_{t+1}=lF}}^{T-1} \sum_{i=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N} \xi_t(ik, jl)}{\sum_{t=1}^{T-1}\sum_{i=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N} \xi_t(ik, jl)}. \tag{5.35}$$

5.3.4 Summarising the Training Algorithm

Through Equations (5.10)–(5.35) we developed an Expectation Maximisation procedure, by following steps analogous to the Baum-Welch algorithm. To summarise, starting with an initial set of parameter values, $\tilde{\lambda} = \{\tilde{p}_i, \tilde{q}_k^j, \tilde{F}_i\}$, we first compute the forward and backward variables and the posteriori probability measures, and then re-estimate the parameter values $\hat{\lambda} = (\hat{p}_i, \hat{q}_k^j, \hat{F}_i)$. Since the expected event counts in the re-estimation equations are derived from the parameters and vice versa, we can iteratively use $\hat{\lambda}$ in place of $\tilde{\lambda}$ in the right hand side of Equations (5.10)-(5.22) to move from an initial guess of the parameters to a better estimate that (locally) maximises $P(U|\lambda)$.

The Baum-Welch algorithm could not be applied directly, because our model has coupled states, i.e. (x_t, y_t), and the Markov transition probabilities are parameterized; Baum-Welch assumes single hidden states, and an unconstrained Markov transition matrix.

We developed this training algorithm by following steps analogous to the Baum–Welch algorithm, which is the most successful, widely cited, method for training the standard HMMs. To perform the inference for our CHMM, in Sub-Section 5.3.1, we modified the standard forward-backward procedure described in Paragraph 3.5.6.3. This modification was made with regard to the joint states, i.e. (x_t, y_t), based on the model assumptions according to their conditional dependency across time. Similar modification was also made to derive the posteriori probability measures in Sub-Section 5.3.2. The final step of the Baum-Welch algorithm is re-estimating the elements of the transition and observation probability matrices. In the proposed training algorithm we derived the re-estimation formulas for our CHMM parameters, i.e. p_i, q_k^i, F_i , $1 \le i, k \le N$.

Baum et al. (1970) and Cappe et al. (2005) described the derivation of the re-estimation formulas for the HMM parameters in the Baum-Welch algorithm, by standard optimization method. A self-mapping transformation is constructed based on the optimality equations from the Lagrange multiplier method, and it is proved by Baum and his colleagues (1970) that this transformation leads to an increase in the objective function, i.e. $P(U|\lambda)$. The re-estimation formulas in the Baum-Welch algorithm can be also explained by intuitive Bayesian posteriori re-estimation and the concept of counting event occurrences. Based on this interpretation, we defined the re-estimation formulas for the proposed CHMM.

Note that in the proposed training algorithm the model parameters, p_i, q_k^j, F_i, are re-estimated directly, instead of the joint state transition and observation probabilities, $a_{ik,jl}$ and $b_{ik}(o)$. Thus, the total number of parameters re-estimated at each iteration is

$\underbrace{N-1}_{p_i} + \underbrace{N(N-1)}_{q_k^i} + \underbrace{N}_{F_i} = N(N+1)-1$, instead of $\underbrace{(N^2 \times N^2)}_{a_{ik,jl}} + \underbrace{N^2 \times (2N)}_{b_{ik}(o)} = N^3(N+2)$

, and this has two advantages. First, the training algorithm is considerably faster and hence more efficient. For a model with a small number of deterioration states (e.g. $N=3$) this would be a matter of a few seconds. However, for a relatively larger model (e.g. $N=6$) and dataset (e.g. $T=100$) our algorithm will be several minutes faster. Secondly, the trained model is more robust. This is greatly important for implementing the model in practice where there are usually insufficient lifetime data available for training the models (as discussed in Sub-Section 3.5.7). What counts in training HMMs (or the proposed CHMM in particular) is not the total number of observation sequences, but the number of sample data per parameter. Therefore, when the number of parameters is reduced, a training dataset is shared among less number of statistics and hence the estimated model parameters will be more robust.

The algorithm presented in the next page codifies the training method explained above. In this training algorithm we declare the convergence when the difference between the probabilities of the observation sequence given the trained model parameters, $P\left(U \middle| \hat{\lambda}\right)$, at consecutive iterations becomes less than a predefined threshold.

Proposed CHMM Training Algorithm

Given a sequence of observations $U = u_1, u_2, ..., u_T$ and an initial guess of model parameters $\tilde{\lambda} = \left\{ \tilde{p}_i, \tilde{q}_k^j, \tilde{F}_i \right\}$,

Repeat:

E Step:

1. Compute $\tilde{a}_{ik,jl}$ and $\tilde{b}_{ik}(u)$, $\forall i,k,j,l \in \{1,2,...,N\}$ and $\forall u \in V$.
2. Compute $\alpha_t(ik)$ and $\beta_t(jl)$, $\forall i,k,j,l \in \{1,2,...,N\}$ and $\forall t \in \{1,2,...,T\}$.
3. Compute $P(U|\tilde{\lambda})$ and, **Return** $\tilde{\lambda} = \left\{ \tilde{p}_i, \tilde{q}_k^j, \tilde{F}_i \right\}$ if $P(U|\tilde{\lambda})$ has converged.
4. Compute $\xi_t(ik, jl)$ and $\gamma_t(ik)$ $\forall i,k,j,l \in \{1,2,...,N\}$ and $\forall t \in \{1,2,...,T\}$.

M Step:

5. Compute $\hat{F}_i, \hat{p}_i, \hat{q}_k^i$, $\forall i,k \in \{1,2,...,N\}$.
6. Use $\hat{\lambda} = \left\{ \hat{p}_i, \hat{q}_k^i, \hat{F}_i \right\}$ in place of $\tilde{\lambda} = \left\{ \tilde{p}_i, \tilde{q}_k^j, \tilde{F}_i \right\}$.

5.3.5 Multiple Observation Sequences

Because of the "Left-to-Right" transient nature of the proposed CHMM only a small number of observations for each state will be available in a single observation sequence. Recall from Sub-Section 3.5.6 that in maintenance practice this is specifically an issue for higher indexed states representing worse levels of deterioration, due to preventive replacements performed before failure. Therefore, in order to have sufficient data to estimate reliable model parameters, multiple observation sequences need to be used as training data.

Let $U=\left[U^{(1)},U^{(2)},...,U^{(G)}\right]$ denote the set of G independent sequences of observations, which could be different in length, where $U^{(g)}=u_1^{(g)},u_2^{(g)},...,u_{T_g}^{(g)}$ is the observation sequence number g, $1\le g\le G$, with length of T_g. Levinson et al. (1983) presented an EM algorithm for training the standard HMMs based on multiple independent observation sequences. Since the re-estimation formulas in Baum–Welch algorithm are obtained based on the concept of frequencies of event occurrence, the re-estimation formulas for multiple observation sequences are modified by adding together the event counts associated with individual observation sequences. Levinson et al. (1983) proved that the re-estimation equations modified in this way guarantee the convergence of $P\left(U\middle|\hat{\lambda}\right)=\prod_{g=1}^{G}P\left(U^{(g)}|\lambda\right)$ to local maxima.

Since we obtained the re-estimation formulas for the proposed CHMM based on the concept of frequencies of event occurrence, we can modify them based on the method that Levinson et al. used, to train the model based on multiple observation sequences. Let $\tilde{\lambda}$ denote the set of an initial guess of model parameters. Let $\xi_t^{(g)}(ik,jl)$ denote the probability of being in the joint state $(x_t=i,y_t=k)$ at time t and in the joint state $(x_{t+1}=j,y_{t+1}=l)$ at time $t+1$, given the observation sequence $U^{(g)}$, $1\le g\le G$, and $\tilde{\lambda}$, which can be obtained using Equation (5.24). As given by Equation (5.27), if we sum $\xi_t^{(g)}(ik,jl)$ over time, $1\le t\le T_g-1$, we obtain a quantity that can be interpreted as the expected number of transitions from the joint state (i,k) to the joint state (j,l) given the observation sequence $U^{(g)}$, and $\tilde{\lambda}$. If we compute $\xi_t^{(u)}(ik,jl)$ for all of the independent observations sequence in the data set $\left[U^{(1)},U^{(2)},...,U^{(G)}\right]$ and add them together, we obtain a quantity that can be interpreted as the expected number of transitions from the joint state (i,k) to the joint state (j,l) given the data set $\left[U^{(1)},U^{(2)},...,U^{(G)}\right]$, and $\tilde{\lambda}$. By modifying the corresponding expected event counts in Equations (5.30), (5.32) and (5.35) in this manner, the re-estimation equations for multiple observation sequences can be given as follows:

$$\hat{p}_i = \frac{\sum_{u=1}^{G}\sum_{t=1}^{T_g-1}\sum_{l=1}^{N}\sum_{k=1}^{N}\xi_t^{(g)}\left(ik, jl\right)}{\sum_{u=1}^{G}\sum_{t=1}^{T_g-1}\sum_{k=1}^{N}\gamma_t^{(g)}\left(ik\right)}, \tag{5.36}$$

$$\hat{q}_k^j = \frac{\sum_{u=1}^{G}\sum_{t=1}^{T_g-1}\sum_{i=1}^{N}\xi_t^{(g)}\left(ik, jk\right)}{\sum_{u=1}^{G}\sum_{t=1}^{T_g-1}\sum_{i=1}^{N}\sum_{l=1}^{N}\xi_t^{(g)}\left(ik, jl\right)}, \tag{5.37}$$

$$\hat{F}_j = \frac{\sum_{u=1}^{G}\sum_{\substack{t=1 \\ s.b. \\ u_{t+1}=lF}}^{T_g-1}\sum_{i=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N}\xi_t^{(g)}\left(ik, jl\right)}{\sum_{u=1}^{G}\sum_{t=1}^{T_g-1}\sum_{i=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N}\xi_t^{(g)}\left(ik, jl\right)}. \tag{5.38}$$

5.4 Conditional Probability Distribution of Time to Failure

Modelling the evolutionary interaction between expert judgement and the deterioration state of a system within the HMM framework allows us to use the established methods of this framework, such as the Viterbi algorithm (as described in Paragraph 3.5.6.2), to find the most likely sequence of deterioration states based on a particular sequence of observations. As mentioned in Section 3.5.3, this problem is related to diagnostics, i.e. to estimate the current system's deterioration state given a sequence of expert judgement states.

Another problem, usually of most interest in maintenance management, concerns predicting the remaining time to failure, given the current deterioration state of a system. This is particularly important when cost-effective decisions need to be taken regarding the maintenance actions (as described in Chapter 7). In this section, we explain how to derive the probability density function of the remaining time to failure given the estimated deterioration state.

Suppose that the model has been diagnosed to be in State i, $1 \le i \le N$. We introduce the random variable T_f to denote the time to failure. Let $f_i(t)$ denote the probability distribution function of the remaining time to failure given that the system is currently in State i, that is, the probability of the system failing in t discrete time units, given that it is in State i.

$$f_i(t) = P\left(T_f = t \middle| x_0 = i\right),\ 1 \le i \le N. \tag{5.39}$$

Assuming that the current deterioration state of the system is $x_0 = i$, the system will either remain at the current state until it fails, or it will move to a higher indexed state and then fail. Since it is assumed that the system cannot improve on its own, Equation (5.39) can be given as:

$$f_i(t) = P\left(T_f = t, x_t = i \middle| x_0 = i\right) + P\left(T_f = t, x_t > i \middle| x_0 = i\right),\ 1 \le i \le N. \tag{5.40}$$

Recall from Section (5.1) that, the state transition is assumed to take place at the beginning of a time epoch. Given that the system is in State i, the probability that the system remains at the current state for t time units and then fails in this state is:

$$P\left(T_f = t, x_t = i \middle| x_0 = i\right) = \left(\bar{F}_i p_i\right)^t F_i,\ \ 1 \le i \le N, \tag{5.41}$$

where F_i is the probability of failure when the system is in State i and $\bar{F}_i$ is the complementary probability of F_i.

Now consider a situation when the system fails in a higher indexed state, given that it is currently in State i. In this case, for $t \ge 1$, the system can remain in State i for s, $1 \le s \le t$, time units before moving to the next state, $i+1$. Once it moves to State $i+1$, the probability of the remaining time to failure will be $f_{i+1}(t-s)$. Therefore, given that the system is in State i, the probability that it fails in $t \ge 1$ time units in a higher indexed state is:

$$P\left(T_f = t, x_t > i \middle| x_0 = i\right) = \sum_{s=1}^{t} \bar{F}^s p_i^{s-1} \bar{p}_i f_{i+1}(t-s),\ 1 \le i < N,\ t \ge 1. \tag{5.42}$$

By substituting Equations (5.41) and (5.42) into Equation (5.40), the probability of the system failing in $t \geq 1$ discrete time units, given that it is in State i, can be given by the following iterative equations:

$$
\begin{aligned}
f_i(t) &= \left(\overline{F}_i p_i\right)^t F_i + \sum_{s=1}^{t} \overline{F}^s p_i^{s-1} \overline{p}_i f_{i+1}(t-s), \ \ 1 \leq i < N, \\
f_N(t) &= \overline{F}_N^t F_N,
\end{aligned}
\tag{5.43}
$$

where F_i is the instant failure or the probability of the system failing in zero time units, when it is in State i.

5.5 Summary

In this Chapter we proposed a CHMM to describe the evolutionary relationship between expert judgement and the unobservable deterioration condition of a system. The proposed formulation has the advantage of reduced parameter space compared to standard CHMMs in literature (as reviewed in Sub-Section 3.5.3). To estimate the model parameters, we developed a training algorithm by following steps analogous to Baum-Welch algorithm. The performance of the training algorithm will be experimentally evaluated in Chapter 6. The experimental results in Chapter 6 empirically confirm that the proposed training algorithm converges to a local maximum and thus it can be used as an efficient practical method for training the proposed CHMM. However, there remains an opportunity to prove that theoretically.

6 Experimental Evaluation of the Coupled Hidden Markov Model Training Algorithm

The algorithm developed to train the proposed CHMM in Chapter 5 is demonstrated and evaluated in this chapter by numerical experiments. The performance of the training algorithm is evaluated in terms of fitting the training data to the trained model, and the deviation of the trained parameter values from the original parameter values (which are used to generate the training data). The training algorithm is first illustrated and its performance is explored by numerical experiments in Section 6.2. The effect of the number of training observation sequences and the initial parameter values on the performance of the algorithm is then investigated by experimental sensitivity analysis in Section 6.3 and Section 6.4, respectively.

6.1 Experimental Setup

A program was coded using the MATLAB software package to execute the training algorithm developed in Section 5.3. The program code is presented in Appendix A. As summarised in Sub-Section 5.3.4, starting with an initial model $\tilde{\lambda}$, the algorithm iteratively updates the parameter values until the difference between the probabilities of the training data given the trained model, $P\left(U\middle|\hat{\lambda}\right)$, at consecutive iterations becomes less than a predefined threshold. For the simulation studies in this chapter we set the convergence threshold of the log-likelihood, $\log\left(P\left(U\middle|\hat{\lambda}\right)\right)$, to 0.0001 and if convergence does not take place, the re-estimation procedure is repeated for a maximum number of 100 times. Running the training algorithm with several different parameter settings and different number of observation sequences showed that the convergence often takes place when the re-estimation procedure is repeated less than 50 times. Therefore, the limit of 100 times is set as a reasonable compromise way of terminating the algorithm.

To conduct the numerical experiment and the sensitivity analysis, random sequences of observations are generated in MATLAB to be used as the training data. The parameter values used to generate the data are assigned with regard to the following assumptions. These assumptions, which only applied to the model parameters generating the training data, are made to help the intuitive representation of real lifetime data.

Assumption 1: F_i, $0 \leq \mathrm{i} \leq \mathrm{N}$, i.e. the probability of the system failing in State i, is non-decreasing in i, meaning that the system is more likely to fail in a higher indexed deterioration level reflecting a worse condition.

Assumption 2: p_i, $1 \leq \mathrm{i} < \mathrm{N}$, i.e. the probability of self-transition when the system is in State i, is non-increasing in i. This means that, as the deterioration level becomes worse, it is more likely to make a transition to a higher indexed deterioration level over one time unit. Note that this assumption does not apply to State N, i.e. the absorbing state, where $p_N = 1$.

Assumption 3: q_k^i, the probability of the expert judgement making a self-transition in State k given that the system is in State i, is non-increasing in i for $1 \leq k < N$ and $1 \leq i \leq N$. This implies that the expert is assumed to provide reasonable judgement tracing the actual state of the system. Therefore, given that the expert judgement is in State k, if the system degrades to a worse condition, i.e. a higher indexed State $j > i$, then the expert judgement is less likely to stay in the same state, that is $q_k^j \leq q_k^i$ for $j > i$.

6.2 Numerical Experiment Results

First, to illustrate the training algorithm and explore its performance in training the model, numerical experiments are conducted. Consider a simple model where the deterioration condition of a system is classified into three states, i.e. $N = 3$. Ten random sequences of observations are generated to be used as training data (detail on generating observation sequences is included in Appendix A). We have performed further numerical studies for different values of N, but these studies show qualitatively similar behaviour, and for space reasons are not presented. The effect of the number of observation sequences on the performance of the training algorithm is examined through a numerical sensitivity analysis presented in Section 6.4.

The values assigned to the model parameters, referred to as original parameter values, are listed in Table 6.1. To help the intuitive representation of a real scenario, we assigned a relatively large value to the probability of self-transition when the system is in the least deteriorated condition, i.e. p_1, and a small value to the probability of the system failing in this condition, i.e. F_1. Based on these values, we then set the other parameter values according to the assumptions listed in Section 6.1. The maximum length of the sequences is set to 30, sufficiently large enough to have sequences ending with a failure given the original parameter values. Sequences not ending with a failure represent censored data.

Table 6.1: Original parameter values assigned to the model for generating the training data.

Parameter	p_1	p_2	q_1^1	q_1^2	q_1^3	q_2^1	q_2^2	q_2^3	F_1	F_2	F_3
Value	*0.95*	*0.85*	*0.8*	*0.2*	*0.15*	*0.7*	*0.7*	*0.2*	*0.03*	*0.1*	*0.3*

The initial values assigned to the model parameters, along with the output parameter values, are given in Table 6.2. The log-likelihood increased at every iteration and converged after 23 iterations to the model parameter values given below. Figure 6.1 shows the log-likelihood as a function of the number of iterations.

Table 6.2: The initial, and output parameter values after the convergence of the algorithm . The training algorithm converged after 23 iterations.

Parameter	p_1	p_2	q_1^1	q_1^2	q_1^3	q_2^1	q_2^2	q_2^3	F_1	F_2	F_3

Initial guess	*0.8*	*0.7*	*0.7*	*0.4*	*0.3*	*0.7*	*0.4*	*0.3*	*0.1*	*0.3*	*0.5*
Output parameter	*0.8144*	*0.7593*	*0.5832*	*0.6152*	*0.0293*	*0.7369*	*0.8009*	*0.2182*	*0.1175*	*0.0106*	*0.2004*

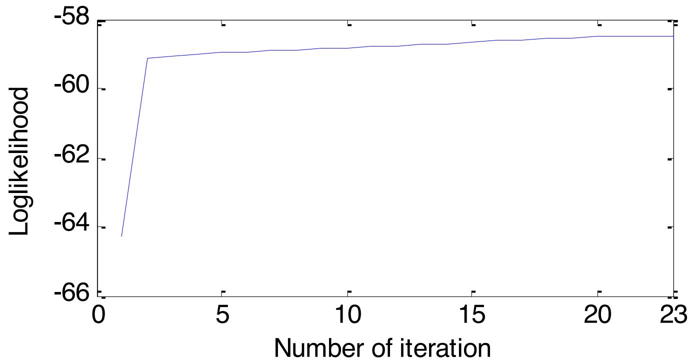

Figure 6.1: Log-likelihood of the model as a function of the algorithm iteration.

The numerical experiment was repeated using different setups for parameter values and different numbers of observation sequences. A summary of the results of this extensive simulation study is as follows:

1. The numerical experiments with different setups empirically confirm that the training algorithm converges to a local maximum.
2. When the number of observation sequences is reduced, some of the parameters are adjusted to unreasonable values. Parameters with small original values, q_1^3 and F_1 in particular, are mostly adjusted to 0 when three or fewer observation sequences are used to train the model. This happens even when the initial values are assigned to be very close to the original values, which is suggestive of overfitting because of insufficient training data. An experimental sensitivity analysis is given in Section 6.3 to investigate the sensitivity of performance of the algorithm to the number of observation sequences.

3. As mentioned in Section 5.3, the training algorithm is only able to find the local maximum. Therefore, the performance of the trained model is sensitive to the initial model. The numerical experiments show that the algorithm converges to inaccurate models when the model is initialised improperly, i.e. when the initial parameter values are largely deviated from the true values. To study the effect of the initial parameter values on the performance of the training algorithm, a numerical sensitivity analysis is presented in Section 6.4.

6.3 Experimental Sensitivity Analysis of the Training Algorithm with Regard to the Number of Observation Sequences

6.3.1 Simulation Procedure

To investigate the sensitivity of the training algorithm with regard to the number of observation sequences an experimental sensitivity analysis is conducted. The experimental setup explained in Section 6.1 and the original parameter values listed in Table 6.1 are used to generate the training data. The initial parameter values listed in Table 6.2 are used to initialise the model. We will study the effect of the initialised model on the performance of the algorithm in Section 6.4. The Absolute Deviation (AD) between the estimated and the actual parameter values is used as a measure to evaluate the efficiency of the training algorithm, i.e. $|\omega - \overline{\omega}|$, where ω denotes the actual parameter value and $\hat{\omega}$ represents the estimated value of a parameter. The AD represents the validation error of the trained model to perform well on unseen data, and not only on the training data.

The experimental sensitivity analysis is conducted as follows. First, 50 random observation sequences are generated based on the original parameter values given in Table 6.1. Then, based on the initial parameter values listed in Table 6.2, at each step the first $n = 1,2,...,50$ observation sequences, in the same order, are used to train the model. The AD between the estimated and the original values is calculated for all of the parameters at each step.

6.3.2 Results and Discussion

The simulation results are illustrated in Figure 6.2–Figure 6.12.

Figure 6.2: Absolute Deviation between the original and estimated value of $p_1 = 0.95$ according to the number of the training sequences.

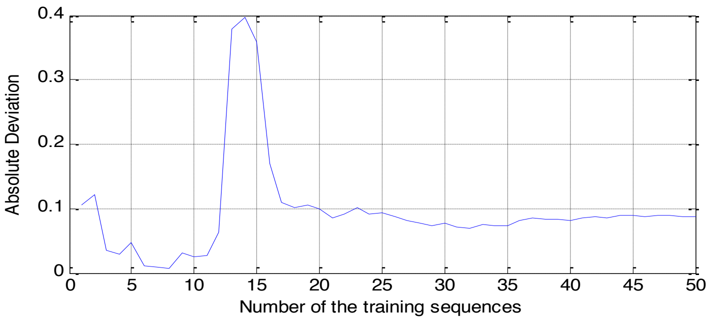

Figure 6.3: Absolute Deviation between the original and estimated value of $p_2 = 0.85$ according to the number of the training sequences.

Figure 6.4: Absolute Deviation between the original and estimated value of $F_1 = 0.03$ according to the number of the training sequences.

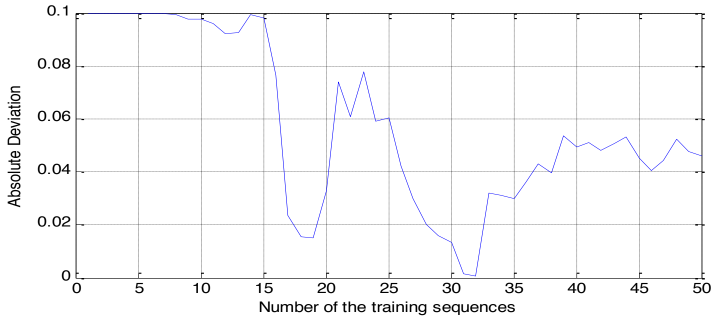

Figure 6.5: Absolute Deviation between the original and estimated value of $F_2 = 0.1$ according to the number of the training sequences.

Figure 6.6: Absolute Deviation between the original and estimated value of $F_3 = 0.3$ according to the number of the training sequences.

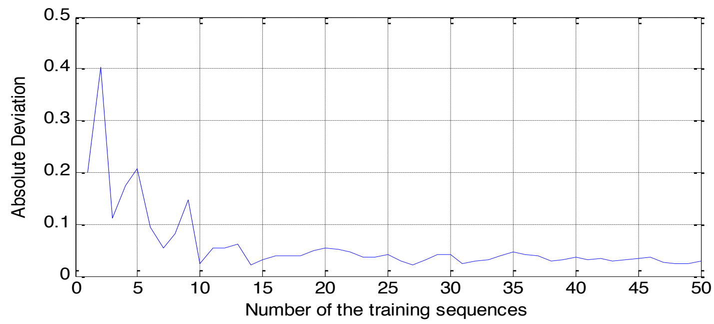

Figure 6.7: Absolute Deviation between the original and estimated value of $q_1^1 = 0.8$ according to the number of the training sequences.

Figure 6.8: Absolute Deviation between the original and estimated value of $q_1^2 = 0.2$ according to the number of the training sequences.

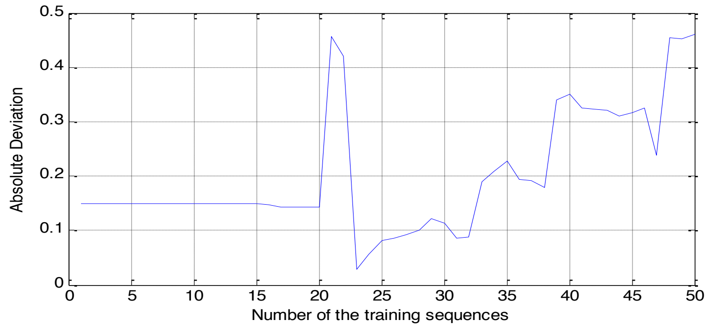

Figure 6.9: Absolute Deviation between the original and estimated value of $q_1^3 = 0.15$ according to the number of the training sequences.

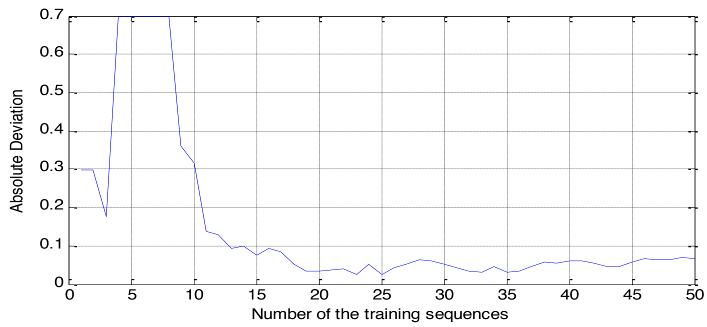

Figure 6.10: Absolute Deviation between the original and estimated value of $q_2^1 = 0.7$ according to the number of the training sequences.

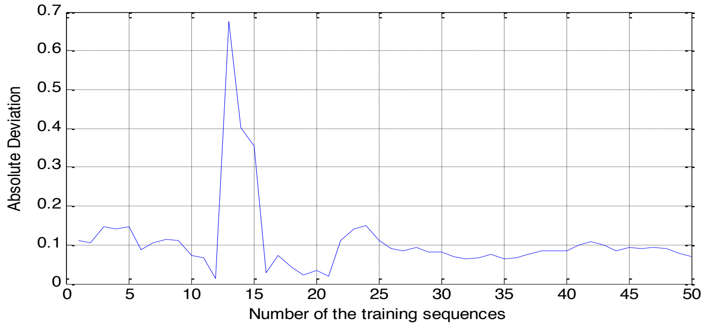

Figure 6.11: Absolute Deviation between the original and estimated value of $q_2^2 = 0.7$ according to the number of the training sequences.

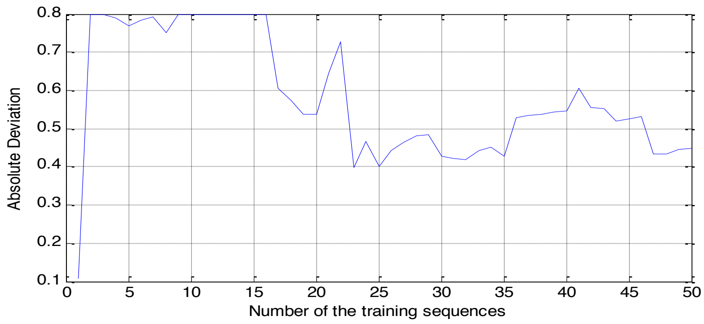

Figure 6.12: Absolute Deviation between the original and estimated value of $q_2^3 = 0.2$ according to the number of the training sequences.

The results of this experimental sensitivity analysis show that in general as the number of the observation sequences increases, the Absolute Deviation between the true and the estimated parameter values decreases. This means that, as expected, the greater the quantity of training data used to train the model, the better the algorithm will perform in terms of training a generalised model.

As seen from the above figures, the validation error for the larger probabilities reaches its minimum faster than for the relatively lower probabilities. For example, as shown in Figure 6.2, the AD of p_1, with the original value of 0.95, reduces to less than 0.005 when 10 sequences are used to train the model and it becomes almost steady afterwards. On the other hand, for F_1, with the original value of 0.03, the validation error reaches its minimum and the almost steady value only when 22 or more observation sequences are used. The relatively large validation error of q_1^3 with the original value of 0.15, and q_2^3 with the original value of 0.2, is also suggestive of overfitting, where the parameter values converge to unreasonable values, even when the initial values are close to the original values.

Overfitting takes place when the parameter values are adjusted to random features of the training data. Recall from Sub-Section 6.3.1, at each step of this simulation study the first $n = 1,2,...,50$ observation sequences are used in the same order. The high peaks in some of the figures are due to the high level of variability that some of these observation sequences show. These high peaks are mostly observed when less than 20 number of observation sequences are used to train the model. As more sequences are added to the training dataset, the number of sample data, i.e. event counts, for re-estimating the parameter values increases and sensitivity to individual sequences decreases. Therefore, the ADs become almost steady afterwards.

Although the structure of the proposed CHMM has the advantage of a small parameter space and consequential computation efficiency, the possibility of overfitting still exists, particularly for small probabilities. This is because the model is trained by maximising its performance on the training data while its efficiency is evaluated by its performance on unseen data, and not the training data, that is represented by the deviation of the parameters from their true values.

Overfitting is particularly a problem for parameters with smaller values since, compared to other parameters, the training data contain smaller amount of sample data for re-estimating their values. In this situation the training algorithm iteratively adjusts these parameters to the specific random features of the training data. Thus, the performance of the trained model, i.e. the log-likelihood, increases iteratively while the performance on new data, i.e. AD, decreases.

Intuitively, the most generalised and fitted model trained with this training algorithm would be where the generalisation error over unseen data has its global minimum. A solution would be to define a generalisation performance (or validation error) for the model and use that as a condition to terminate the re-estimation algorithm.

Some methods in literature have used model entropy as a measure of generalising performance to overcome overfitting in training standard HMMs (Brand, 1999; Walder, Kootsookos, and Lovell, 2003). The entropy of an observation sequence of length T produced by model λ is given by $H(\lambda,T) = -\sum_{\forall U \in \tilde{U}^T} P(U|\lambda)\log(U|\lambda)$, where $\tilde{U}^T$ is the set of all sequences of length T that can be produced by model λ (Walder et al., 2003). A brief discussion on these methods and how they can be applied to the proposed training algorithm is presented in Sub-Section 9.2.2.

Another solution to the problem of overfitting would be to add constraints to the training algorithm such that, at each iteration, the estimated parameter values would not fall below specific values. However, a prior knowledge of the model is needed in order to apply this method of dealing with lack of training data.

6.4 Experimental Sensitivity Analysis of the Training Algorithm with Regard to the Initial Parameter Values

6.4.1 Simulation Procedure

The proposed training algorithm converges to a local maximum and therefore it is sensitive to the random values assigned to model parameters at initialisation of training. In this section we study the effect of initialising the parameters on the performance of the training algorithm, on both the training and unseen data. Log-likelihood is used to evaluate the performance of the algorithm on training data. To assess the performance of the trained model on unseen data we use the Absolute Deviation between the estimated and the actual parameter values.

The experimental setup explained in Section 6.1 and the original parameter values listed in Table 6.1 are used to generate the training data. First, 30 random observation sequences were generated based on the parameter values given in Table 6.1, to train the model. To study the effect of initialising individual parameters on the training performance, at each simulation step the initial value of one parameter is varied while the value of the rest of the parameters are initialised as their original values. The initial value of the parameters with relatively higher original values, $p_1 = 0.95$, $p_2 = 0.85$ and $q_1^1 = 0.8$, are varied between 0.1 and 1.0 with steps of 0.02, while the initial value of the parameters with relatively smaller original values, $q_1^2 = 0.2$ and $F_1 = 0.03$, are varied between 0.01 and 0.5 with steps of 0.01.

6.4.2 Results and Discussion

The simulation results for some of the parameters follow in the next page in Figure 6.13 and Figure 6.14. The log-likelihoods of the trained model according to the initial value of the parameters are shown in Figure 6.13. The Absolute Deviation between the original and estimated values of the parameters according to their initial values are shown in Figure 6.14. The log-likelihood and Absolute Deviation for each simulation are displayed in the same row, and the parameter for which the initial value is varied is tagged along with its original value.

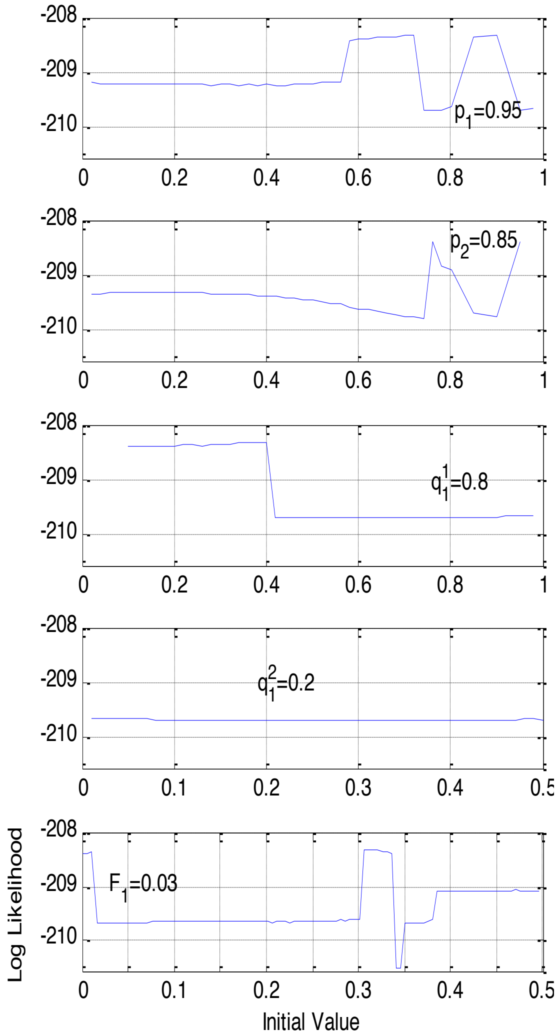

Figure 6.13: Log-likelihood of the trained model according to the initial value of the parameters. The original value of the parameter, for which the initial value is varied, is shown on each diagram.

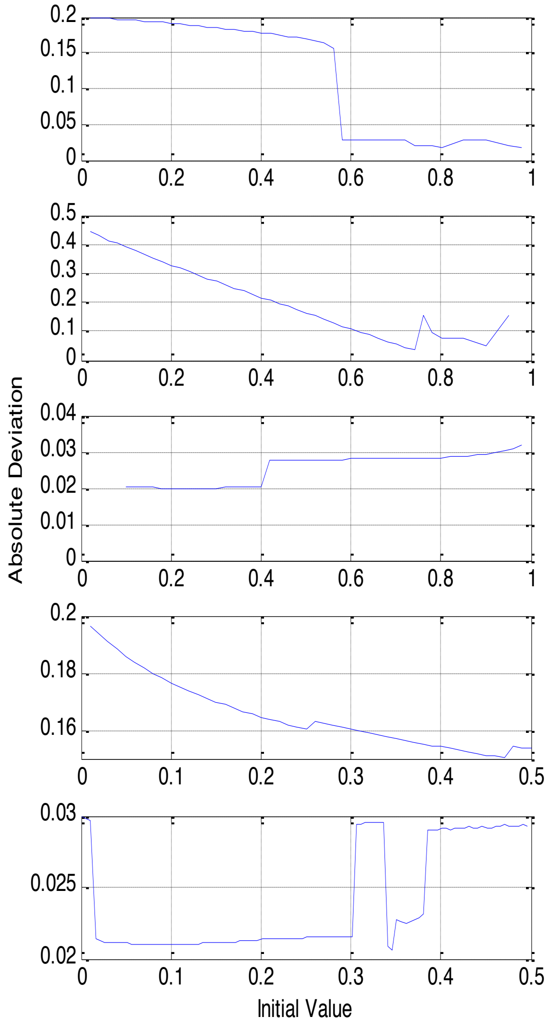

Figure 6.14: Absolute Deviation between the original and estimated values of the parameters according to their initial values.

The simulation results show that while the performance of the training algorithm has the highest level of sensitivity to initial values of the transition probabilities of the true states of the system, i.e. p_1 and p_2, their related Absolute Deviation errors are significantly lower when their initial values are assigned close to their original values. For example, the Absolute Deviation error of p_1 varies between 0.0001 and 0.2, i.e. between 0.01% and almost 21% of its original value, with the lowest Absolute Deviation around its original value of 0.95. But for q_1^1, with the original value of 0.8, the AD error only varies between 0.02 and 0.03. This means that the training algorithm for this parameter performs most robustly with the lowest level of sensitivity to the initial value.

The results for q_1^2 and F_1 with relatively lower original values, are different. For q_1^2 while the log-likelihood shows slight variations, the AD error varies between 0.2 and 0.001. For F_1 with an original value of 0.03, the AD error varies between 0.2 and 0.029. This means that the algorithm performance on unseen data, irrespective of their initial values, is not satisfactory for these parameters.

The fluctuation of log-likelihood over the range of the initial values, even around the original value of the parameters, reflects the fact that the training algorithm converges to a local maximum. This property of the training algorithm makes its performance on training data sensitive to the initial model. When model training is repeated for different setups, changing the original and initial parameter values randomly, it is observed that when the model parameters are initialised such that the order between their true values are preserved, the training algorithm performs better than when the model is initialised randomly. Therefore, using prior knowledge of the model in initialising the parameters will improve the performance of the training algorithm.

A superior choice of initialisation could be obtained by (1) using the expert knowledge to initialise some models; (2) evaluating the quality of the initialised models on the basis of the probability of each model generating the training data, i.e. $P\left(U\middle|\tilde{\lambda}\right)$, using Equation (5.17); and (3) train the model by starting with the initialised model having the largest $P\left(U\middle|\tilde{\lambda}\right)$. This will have the effect of initialising the model closer to the global maximum, and hence the algorithm has a higher chance of converging to it. Although this method of initialisations is computationally expensive, it can be considered in practice when the quality of the trained model outperforms its computation cost.

6.5 Summary and Conclusion

The performance of the proposed training algorithm for the CHMM, and its sensitivity to initial parameter values and the number of training observation sequences were evaluated using simulated data. The observation sequences were generated based on some assumptions with regard to model parameters in order to intuitively represent real data. The numerical experiments empirically confirmed that the training algorithm converges to a local maximum. Therefore, if the initial model is chosen near the global maximum, the algorithm performs better in terms of fitting to both the training and unseen data. Initialisation becomes more important when there are insufficient data available for training the model.

Although the structure of the proposed CHMM has the advantage of a small parameter space and consequently computation efficiency, the possibility of overfitting still exists, particularly for parameters with small values. When there are limited training data available, the parameter values are adjusted to the specific random features of the training data and hence the performance of the trained model on new (unseen) data is reduced. This motivates us to investigate alternative methods for training the proposed CHMM in future research. In Sub-Section 9.2.2, a brief discussion on the existing training algorithms in literature, that aim to solve the problems mentioned above for the standard HMMs, is presented.

7 A Two-Step Partially Observable Markov Decision Process

In Chapter 5 a model was developed to describe the stochastic relationship between the deterioration condition of a system and expert judgement, in the absence of maintenance intervention. Motivated from the maintenance policy described in Section 2.2, we now consider a decision-making problem where upon expert judgements, decisions regarding physical investigation of the system's condition and maintenance actions are to be made. In this chapter we formulate the decision optimisation problem as an MDP with partially observable states.

We first describe the maintenance policy as a two-step decision-making process in Section 7.1. We then take the CHMM, developed in Chapter 5, a step further by incorporating the intervention of maintenance actions. The evolution of the deterioration process and expert judgement process subject to maintenance actions are described in Section 7.2. In Section 7.3 we present the formulation of the decision problem as a two-step POMDP. Detailed derivation of the optimal cost by dynamic programming is given in Section 7.4.

7.1 Description of a Maintenance Policy as a Two-Step Decision Process

We assume that an experienced engineer, i.e. expert, is called at predetermined regular discrete time decision epochs $t = 0, \Delta, 2\Delta, ..., k\Delta, ...$ to assess the condition of a system based on the monitoring information such as temperature and vibration. He provides his assessment as a positive integer number, $y \in \{1, 2, ..., N\}$. We refer to this preliminary assessment as “expert judgement” in order to distinguish from inspection.

Following a preliminary assessment it is then decided either to leave the system until the next decision occasion, to carry out simple inspection, or to conduct accurate inspection. Performing an inspection allows the expert to more accurately determine the condition of the system; however, the level of accuracy of the inspections depends on the chosen inspection type. An accurate inspection is assumed to provide perfect information revealing the true condition of the system through precise physical investigations. A simple inspection is assumed to provide partial information, meaning that only the probability of being in each health condition state could be derived from the outcome of the inspection. A simple inspection costs less than an accurate inspection due to their different types of preparation. For example, the system load needs to be dropped in order to conduct a simple inspection, but the system has to be stopped altogether to be able to perform an accurate inspection.

Based on the outcome of the inspection, the maintenance manager will select one of the following actions: to leave the system; to carry out imperfect maintenance action; or to replace the system. When an imperfect maintenance action is carried out, the system is restored to an intermediate state between the “as good as new” and the pre-maintenance state, with different probabilities governing which restored state is likely to be attained. If the monitored system is non-repairable, the imperfect maintenance refers to some preventive maintenance action that can decrease the failure rate, if it is performed effectively. For example, consider the maintenance practice described in Section 2.2. Although bearings could be considered to be non-repairable systems, unblocking the grease holes or correcting the shaft misalignment can decrease the failure rate. If the monitored system is repairable then imperfect maintenance implies both corrective and preventive actions. Therefore we use the term “imperfect maintenance” to denote both corrective and preventive maintenance actions that transfer a system to an intermediate state. It is assumed that the system can fail from within any state and upon failure, replacement is carried out which returns the system to “as good as new” condition, State 1.

This two-step decision process, as described above, is graphically illustrated in Figure 7.1.

Since the exact state of the system is unknown at each decision time epoch, it is first inferred from the expert judgement. Following a preliminary diagnosis, the first step of decision-making is to select which type of inspection is to be carried out. Let a^I denote the decision variable regarding the inspection choice. If the decision is to not conduct any inspection then $a^I = 0$; in this case nothing is done until the next time the expert is called to assess the system.

When the simple inspection is chosen then $a^I = 1$ and the system is observed partially through a simple inspection procedure. The outcome of the simple inspection is assumed to be a positive integer value $o \in \{1, 2, ..., Z\}$ that is observed with probability $b_i(o)$ when the system is in the true State i, $1 \le i \le N$.

The second step of decision-making is carried out when the maintenance action is selected, conditioned on the observations obtained from the inspection, if any. The decision regarding the maintenance action will be based on the belief derived from the inspection regarding the true deterioration state of the system. Let a^M be the decision variable denoting the maintenance action choice; $a^M = 0$ when the decision is to do nothing, $a^M = 1$ when the decision is to perform an imperfect maintenance, and $a^M = 2$ when the decision is to carry out a preventive replacement.

If the decision is to conduct an accurate inspection, $a^I = 2$, the true state of the system, $x \in \{1, 2, ..., N\}$ is determined through precise physical inspection, and so the maintenance action decision is made in light of the true state of the system.

Let C^F be the cost of a failure replacement, C^S the cost incurred to conduct a simple inspection, C^A the cost of an accurate inspection, C^M the cost of an imperfect maintenance and C^R the cost of carrying out a replacement. We wish to find the optimal policy for choosing inspection types and maintenance actions so that the total expected discounted cost over a finite planning horizon is minimised. Discounting future costs means that all future costs are re-calculated to the equivalent value at the present time. Therefore, costs incurred far in the future will not affect current decision-making since the present value of these costs is small.

In Section 7.3, we will formulate the maintenance optimisation problem described above as a POMDP. Modelling the problem using an MDP framework enable us to evaluate a trade-off between immediate and future costs and benefits.

As shown in Figure 7.1, the decision process consists of two steps at each decision period:

Step 1: the selection of inspection type; and

Step 2: the selection of maintenance actions.

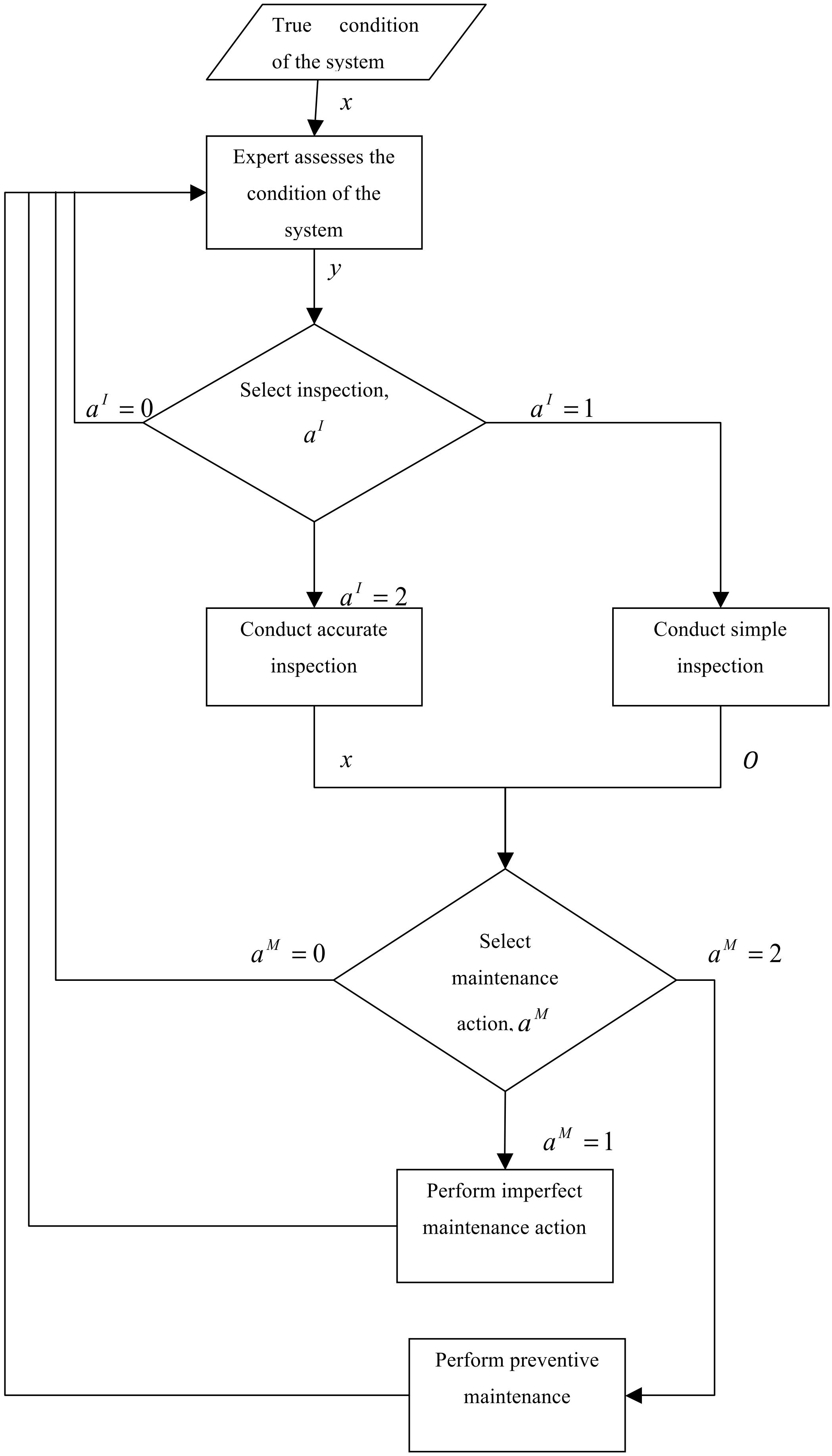

Figure 7.1: Decision process for selecting inspection type and maintenance action. ↗ represents the flow of information; ◇ symbolises decision-making; ▭ represents performing inspection or maintenance action.

7.2 Description of the Maintained System Behaviour

In this section we describe the behaviour of a maintained system that is subject to imperfect maintenance. It was mentioned in the previous section that imperfect maintenance means that, after a maintenance action, the system is restored to an intermediate state between the "as good as new" and the pre-maintenance condition. The evolution of the system's deterioration and the expert judgement is assumed to follow a Markovian stochastic process. The Markov process is chosen because of its ability to graphically and mathematically describe the evolution of a system that enters different discrete states over time. This also makes it convenient to model the effect of maintenance actions in a plausible way. In the following sub-sections the intervention of imperfect maintenance action is incorporated into the evolution of deterioration of the system and expert judgement that was modelled in Chapter 5.

7.2.1 Evolution of the Deterioration Process

Recall that in Chapter 5 we considered a system with N non-observable deterioration states, where State 1 represents the "as good as new" condition and State N is the absorbing state, which is considered the final state with respect to deterioration. Suppose that the system is in State $i \in \{1, 2, ..., N\}$ at time t; then in the absence of any maintenance action, we assume that the system either degrades to the next deterioration state with probability $\bar{p}_i$ or remains at the current state with probability p_i, at time $t+1$, where $\bar{p}_i$ is the complementary probability of p_i. In other words, the system cannot improve on its own. As defined in Chapter 5, we have:

$$p_i = P(x_{t+1} = i | x_t = i),$$

$$\bar{p}_i = P(x_{t+1} \neq i | x_t = i), \; 1 \leq i \leq N. \tag{7.1}$$

Based on Equation (7.1) we define the transition probability matrix $\mathbf{P}$, the probability transition matrix of the system states over one discrete time unit, as:

$$P = \left[p_{ij} = P(x_{t+1} = j | x_t = i), 1 \leq i, j \leq N \right], \tag{7.2}$$

where:

$$p_{ij} = \begin{cases} p_i & \text{for } j = i \\ \bar{p}_i & \text{for } j = i+1 \\ 0 & \text{Otherwise.} \end{cases} \tag{7.3}$$

Since the expert judgement is provided at decision time epochs $t = 0, \Delta, 2\Delta, ..., k\Delta, ...$, we consider the evolution of the system deterioration over Δ time units. Since the evolution of the true state of the system is assumed to follow a time-homogeneous Markov chain, the Δ-step transition probability matrix can be computed as the Δ power of the transition matrix P, thus:

$$P^{\Delta} = \left[p_{ij}^{(\Delta)} \right], \quad 1 \le i, j \le N. \tag{7.4}$$

In the absence of maintenance interventions, the system will transit to State j with probability $p_{ij}^{(\Delta)}$ at the end of the decision period k, given that it is in State i at the beginning of this decision period k:

$$P\left(x_{k\Delta+\Delta} = j \middle| x_{k\Delta} = i\right) = p_{ij}^{(\Delta)}. \tag{7.5}$$

Convention: Since the decisions are made at regular intervals, to simplify the notation we drop Δ from the time index; for example:

$$P\left(x_{k\Delta+\Delta} = j \middle| x_{k\Delta} = i\right) = P\left(x_{k+1} = j \middle| x_k = i\right).$$

If a failure occurs, a corrective replacement is taken which always restores the system to "as good as new" condition, State 1. When an imperfect maintenance action is carried out, the system is restored to an intermediate state between the pre-maintenance state and State 1 with different probabilities accordingly. We define x'_k as the true state of the system after an imperfect maintenance action at period k. Given that the system is in State i at period k, it is assumed that an imperfect maintenance action restores it to an intermediate state, $x'_k = j, 1 \le j \le i$, with probability r_{ij}, thus:

$$r_{ij} = P\left(x'_k = j \middle| x_k = i, a_k^M = 1\right),\ 1 \le i, j \le N, \tag{7.6}$$

where a_k^M denotes the decision variable at period k regarding the maintenance action, and $a_k^M = 1$ indicates that an imperfect repair is carried out at period k. It is assumed that the maintenance action does not worsen the condition of the system, i.e. $r_{ij} = 0$ for $j > i$.

For example, suppose that the system is in State 4. Then performing an imperfect maintenance action transfers the system to the intermediate States 2 with probabilities r_{42}. The state transition for a system with N deterioration states subject to imperfect maintenance is graphically depicted in Figure 7.2.

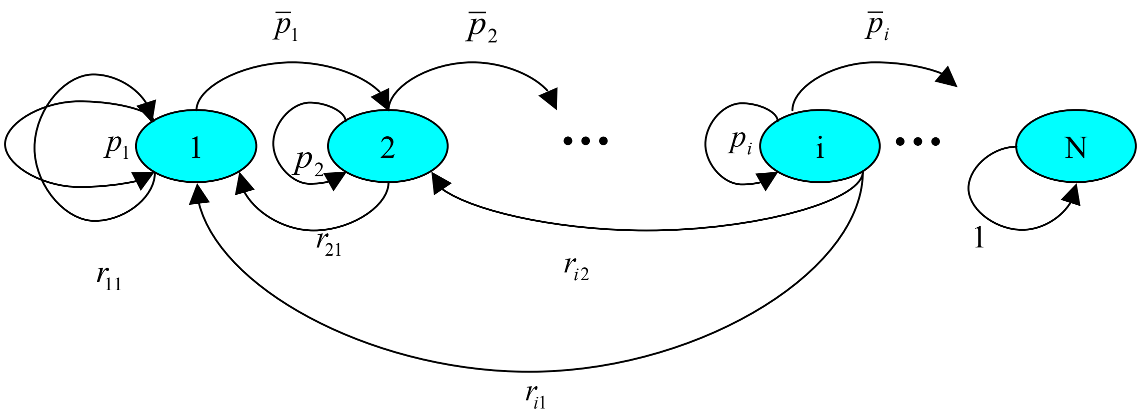

Figure 7.2: State transition diagram of a system subject to imperfect maintenance.

7.2.2 Evolution of the Expert Judgement Process

We assume that the true state of the system is not observable and expert judgement at period k is provided as a positive integer number $y_K \in \{1, 2, ..., N\}$, and that the expert judgement process possesses Markovian properties. We also assume that the transitions of the expert judgement states occur at the beginning of the decision intervals. Given that the expert believes that the system is in State l at period k, and the system at period $k+1$ is in State i, we define the transition probability q_{lm}^i as the probability that the expert judgement moves to State m at the beginning of the decision period $k+1$, shown as:

$$q_{1m}^{i} = P\left(y_{k+1} = m \middle| x_{k+1} = i, y_k = l\right),\ 1 \le i, l, m \le N. \tag{7.7}$$

Suppose that at decision period k the system is in State i, $x_k = i$, and the expert believes that the system is in State l, i.e. $y_k = l$, $1 \le l \le N$. When an imperfect maintenance action is conducted the system is restored to State j, $1 \le j \le i \le N$, with probability r_{ij}. This probability reflects the uncertainty related to the effectiveness of the imperfect maintenance action. We assume that the expert is aware of this uncertainty and hence, after an imperfect maintenance, he provides a judgement regarding the state of the system as the most likely state that the system is restored to by an imperfect maintenance action. We define y'_k as the expert judgement state at period k after an imperfect maintenance. Assuming that, at decision period k, the expert believes that the system is in State l, i.e. $y_k = l$, $1 \le l \le N$, we have:

$$y'_k = m = \arg\max_j \left(r_{lj}\right),\ \ 1 \le l, j \le N, \tag{7.8}$$

where, r_{ij} is defined in Equation (7.6). In other words, after an imperfect repair, the expert judgement moves from State l to State m, $1 \le l, m \le N$, so that if the system was actually in State l the imperfect maintenance action would most likely restore it to State m, that is $r_{lm} = \max_j \left(r_{lj}\right)$.

7.3 Formulation of a Two-Step Partially Observable Markov Decision Process

7.3.1 State Space

Since the true deterioration state of the system is hidden, we should infer it from the history including all past expert judgements, observations and maintenance actions. The conditional probability distribution of the system state is defined as:

$$\pi^k = \left(\pi_1^k, \pi_2^k, ..., \pi_N^k\right),$$

$$\pi_i^k = P\left(x_k = i \middle| H^k\right), 1 \le i \le N \text{ and } k = 0,1,2,3,... \tag{7.9}$$

where π_i^k denotes the probability of the system being in State i at period k, i.e. time $t = k\Delta$ given all the information available at the beginning of period k. We refer to π^k as the "belief state" hereafter. Let H^k represent all information available at the beginning of period k, comprising the sum of knowledge regarding the starting situation, all actions performed, all expert judgements made and all observations made up to time $t = k\Delta$. We assume that at the beginning of the process the system is in "as good as new" state and we define:

$$\pi_i^0 = \begin{cases} 1 & \text{for i=1,} \\ 0 & \text{otherwise.} \end{cases} \tag{7.10}$$

7.3.2 Decision Space

We define the decision space of the two decision steps as follows:

Step 1: Select inspection

Let $a^I \in \{0,1,2\}$ be the decision variable, where:

$a^I = 0$ means to not conduct any inspection,

$a^I = 1$ means to conduct a simple inspection, and

$a^I = 2$ means to carry out an accurate inspection.

Let a_k^I denote the decision variable for the inspection choice at period k.

Step two: Select maintenance action

Let $a^M \in \{0,1,2\}$ denote the decision variable, where:

$a^M = 0$ means to not take any maintenance action,

$a^M = 1$ means to perform imperfect repair, and

$a^M = 2$ means to replace the system.

Let a_k^M denote the decision variable for the maintenance action at period k.

7.3.3 State Transition

The conditional probability distribution of the system state is updated when an action is taken or information is obtained. That is, when new expert judgement is provided, an inspection is conducted, or a maintenance action is performed. The sequence of obtaining the information over a decision interval is illustrated in Figure 7.3. Note that the time taken for the decision to be made, the inspection to be conducted and the imperfect maintenance action to be carried out, are assumed to be negligible compared to the length of a discrete decision period. For example, consider a situation when expert judgement is provided every month, after which observation from inspection is obtained and imperfect maintenance action is carried out in less than 24 hours. Considering the discrete time unit, t, as a day, that is $\Delta = 30$, the time taken for the decision to be made, the inspection to be conducted and imperfect maintenance action to be carried out is negligible. If this time is assumed to be non-negligible, the probability of the deterioration state transition over this time duration need to be taken into account when updating the conditional probability distribution of the system state.

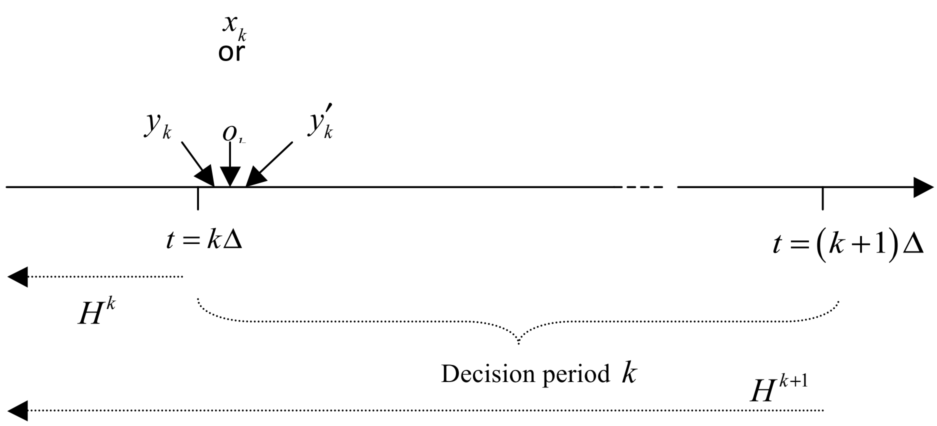

Figure 7.3: Sequence of obtaining the information during a decision period.

We assume that after a replacement the system is recovered to 'as good as new' state (State 1), and it takes almost one decision period to carry out the replacement due to its complicated preparations. Hence, when a replacement is carried out at decision period k, the history at the beginning of decision period $k+1$ will the same as the history at the beginning of the decision-making process, that is, H^0.

Let $\pi^k\left(y_k\right)$ represent the conditional probability distribution of the system at decision period k updated after the expert judgement is provided, $\pi^k\left(y_k, o_k\right)$ denote the conditional probability distribution of the system at decision period k updated after a simple inspection is conducted, $\pi^k\left(SM\right)$ denote the conditional probability distribution of the system at decision period k updated when an imperfect maintenance action is carried out after a simple inspection and $\pi^k\left(AM\right)$ denote the conditional probability distribution of the system at decision period k updated when an imperfect maintenance action is carried out after an accurate inspection.

Further definitions and calculations of these conditional probability distributions follow for each in turn.

7.3.3.1 Updating the Conditional Probability Distribution of a System when Expert Judgement is Provided at the Beginning of a Decision Period

At period $k+1$, when the expert judgement y_{k+1} is provided, the conditional probability distribution of the system is updated. We define $\pi_j^{k+1}\left(y_{k+1}\right)$ as the probability that the system is in State j at period $k+1$, given the expert judgement provided at period $k+1$ and all information available at the beginning of this period:

$$\pi_j^{k+1}\left(y_{k+1}\right) = P\left(x_{k+1} = j \middle| H^{k+1}, y_{k+1}\right). \tag{7.11}$$

Using Bayes' theorem gives,

$$\pi_j^{k+1}\left(y_{k+1}\right)=P\left(x_{k+1}=j\middle|H^{k+1},y_{k+1}\right)=\frac{P\left(x_{k+1}=j,y_{k+1}\middle|H^{k+1}\right)}{P\left(y_{k+1}\middle|H^{k+1}\right)}=\frac{P\left(x_{k+1}=j,y_{k+1}\middle|H^{k+1}\right)}{\sum_{x_{k+1}}P\left(x_{k+1},y_{k+1}\middle|H^{k+1}\right)}. \tag{7.12}$$

As illustrated in Figure 7.3, H^{k+1}, the history at the beginning of period $k+1$, is equal to the history available at the beginning of period k plus the information obtained during period k. The information obtained during period k depends on the decisions made during this period regarding the inspections and maintenance actions. Possible scenarios that could take place at each decision period and the corresponding history are listed in Table 7.1.

Table 7.1: Possible histories at the beginning of decision period k+1, based on the decisions made at period k.

Scenario Number	a_k^I	a_k^M	*Decision choices taken at period* k	*History of information at the beginning of period* $k+1$
1	0	0	No inspection and no maintenance action	$H^{k+1}=H^k,y_k$
2	1	0	Simple inspection followed by no maintenance action	$H^{k+1}=H^k,y_k,o_k$
3	1	1	Simple inspection followed by an imperfect maintenance action	$H^{k+1}=H^k,y_k,o_k,y_k'$
4	1	2	Simple inspection followed by a replacement	$H^{k+1}=H^0$
5	2	0	Accurate inspection followed by no maintenance action	$H^{k+1}=H^k,y_k,x_k$
6	2	1	Accurate inspection followed by an imperfect maintenance action	$H^{k+1}=H^k,y_k,x_k,y_k'$
7	2	2	Accurate inspection followed by a	$H^{k+1}=H^0$

	replacement

For each scenario in Table 7.1, we can calculate the updated conditional probability distribution of the system after the expert judgement is provided at the beginning of a decision period. We take each row of Table 7.1 in turn.

Scenario 1 – *No inspection is conducted at period* k

First consider a scenario when at decision period k it is decided to not conduct any inspection or maintenance action, as shown in the first row of Table 7.1. In this scenario, the numerator of the right hand side of Equation (7.12) can be written as:

$$P\left(y_{k+1}, x_{k+1} \middle| H^{k+1}\right) = P\left(y_{k+1}, x_{k+1} \middle| H^{k}, y_k\right). \tag{7.13}$$

Using the chain rule, we have:

$$P\left(y_{k+1}, x_{k+1} \middle| H^{k}, y_k\right) = P\left(y_{k+1} \middle| x_{k+1}, H^{k}, y_k\right) P\left(x_{k+1} \middle| H^{k}, y_k\right). \tag{7.14}$$

It is assumed that (1) the expert judgement, given the current true state and the expert judgement at the previous decision time epoch, is independent of the rest of the history of expert judgements and true states and (2) that the true state of the system follows Markovian evolution and is independent of the history of expert judgements and the true deterioration states, given the previous true state. Recalling that H^k includes all the information available at the beginning of period k, Equation (7.14) can be given by the following, using the chain rule:

$$\begin{aligned} P\left(y_{k+1}, x_{k+1} \middle| H^{k}, y_k\right) &= P\left(y_{k+1} \middle| x_{k+1}, y_k\right) P\left(x_{k+1} \middle| H^{k}, y_k\right) \\ &= \sum_{x_k} P\left(y_{k+1} \middle| x_{k+1}, y_k\right) P\left(x_{k+1} \middle| x_k, H^{k}, y_k\right) P\left(x_k \middle| H^{k}, y_k\right) \\ &= \sum_{x_k} P\left(y_{k+1} \middle| x_{k+1}, y_k\right) P\left(x_{k+1} \middle| x_k\right) P\left(x_k \middle| H^{k}, y_k\right). \end{aligned} \tag{7.15}$$

Based on the definitions in Equations (7.5) and (7.7), Equation (7.15) can be given by:

$$P\left(x_{k+1}=j, y_{k+1}=m \middle| H^{k}, y_{k}=l\right)=\sum_{i=1}^{N} q_{lm}^{j} p_{ij}^{(\Delta)} \pi_{i}^{k}\left(y_{k}=l\right),\ 1 \leq j, l, m \leq N. \tag{7.16}$$

By incorporating Equation (7.16) into Equation (7.12), we can calculate $\pi_{j}^{k+1}\left(y_{k}\right)$ when no inspection is conducted at decision period $k+1$ as:

$$\begin{aligned}\pi_{j}^{k+1}\left(y_{k+1}=m\right) &= P\left(x_{k+1}=j \middle| H^{k}, y_{k}=l, y_{k+1}=m\right) \\ &= \frac{\sum_{i=1}^{N} q_{lm}^{j} p_{ij}^{(\Delta)} \pi_{i}^{k}\left(y_{k}=l\right)}{\sum_{n=1}^{N} \sum_{i=1}^{N} q_{lm}^{n} p_{in}^{(\Delta)} \pi_{i}^{k}\left(y_{k}=l\right)}, \quad 1 \leq l, m, j \leq N.\end{aligned} \tag{7.17}$$

***Scenario 2** – Simple inspection is conducted at period k, followed by no maintenance action*

In this scenario the history at the beginning of period $k+1$ includes all the history available at the beginning of decision period k and the expert judgement and the outcome of the simple inspection obtained in this decision period. In this case the numerator of Equation (7.12) can be written as:

$$P\left(y_{k+1}, x_{k+1} \middle| H^{k+1}\right)=P\left(y_{k+1}, x_{k+1} \middle| H^{k}, y_{k}, o_{k}\right), \tag{7.18}$$

which can be given as follows, using the chain rule:

$$\begin{aligned}P\left(y_{k+1}, x_{k+1} \middle| H^{k+1}\right) &= P\left(y_{k+1} \middle| x_{k+1}, H^{k}, y_{k}, o_{k}\right) P\left(x_{k+1} \middle| H^{k}, y_{k}, o_{k}\right) \\ &= \sum_{x_{k}} P\left(y_{k+1} \middle| x_{k+1}, H^{k}, y_{k}, o_{k}\right) P\left(x_{k+1} \middle| x_{k}, H^{k}, y_{k}, o_{k}\right) P\left(x_{k} \middle| H^{k}, y_{k}, o_{k}\right) \\ &= \sum_{x_{k}} P\left(y_{k+1} \middle| x_{k+1}, y_{k}\right) P\left(x_{k+1} \middle| x_{k}\right) P\left(x_{k} \middle| H^{k}, y_{k}, o_{k}\right).\end{aligned} \tag{7.19}$$

Using the definitions in Equations (7.5) and (7.7), Equation (7.19) can be given by:

$$P\left(y_{k+1}=m, x_{k+1}=j \middle| H^{k}, y_{k}=l, o_{k}\right)=\sum_{i=1}^{N} q_{lm}^{j} p_{ij}^{(\Delta)} \pi_{i}^{k}\left(y_{k}=l, o_{k}\right),\ 1 \leq j, l, m \leq N, \tag{7.20}$$

in which $\pi_i^k\left(y_k,o_k\right)$ is the belief state of the system at decision period k, updated based on the expert judgement y_k and the simple inspection outcome o_k. Further definition and calculation of $\pi_i^k\left(y_k,o_k\right)$ will be given in Paragraph 7.3.3.2. By substituting Equation (7.20) into Equation (7.12) we can calculate $\pi_j^{k+1}\left(y_{k+1}\right)$ for Scenario 2 as:

$$\pi_j^{k+1}\left(y_{k+1}=m\right)=P\left(x_{k+1}=j\middle|H^k,y_k=l,y_{k+1}=m,o_k\right)$$
$$=\frac{\sum_{i=1}^{N} q_{lm}^{j} p_{ij}^{(\Delta)} \pi_i^k\left(y_k=l,o_k\right)}{\sum_{n=1}^{N}\sum_{i=1}^{N} q_{lm}^{n} p_{in}^{(\Delta)} \pi_i^k\left(y_k=l,o_k\right)}, \qquad 1\le l,m,j\le N. \tag{7.21}$$

***Scenario 3** – Simple inspection is conducted at period k, followed by an imperfect repair*

In this scenario, the history at the beginning of period $k+1$ includes all the history available at the beginning of period k, and the expert judgement, the outcome of the simple inspection, and y'_k that is the expert judgement state updated after an imperfect maintenance action is carried out at period k. Therefore, $P\left(y_{k+1},x_{k+1}\middle|H^{k+1}\right)$ in Equation (7.12) is given by:

$$P\left(y_{k+1},x_{k+1}\middle|H^{k+1}\right)=P\left(y_{k+1},x_{k+1}\middle|H^k,y_k,o_k,y'_k\right). \tag{7.22}$$

which can be given as follows, using the chain rule:

$$P\left(y_{k+1},x_{k+1}\middle|H^{k+1}\right)=P\left(y_{k+1}\middle|x_{k+1},H^k,y_k,o_k,y'_k\right)P\left(x_{k+1}\middle|H^k,y_k,o_k,y'_k\right), \tag{7.23}$$

Again it is assumed that the expert judgement is dependent only on the current true state of the system and the expert judgement at the previous decision epoch. Therefore, the probability of y_{k+1} given x_{k+1} and y'_k is independent of other information and hence, Equation (7.23) can be simplified to:

$$P\left(y_{k+1},x_{k+1}\middle|H^{k+1}\right)=P\left(y_{k+1}\middle|x_{k+1},y'_k\right)P\left(x_{k+1}\middle|H^k,y_k,o_k,y'_k\right). \tag{7.24}$$

Recall from Sub-Section 7.2.1 that x'_k is the deterioration state of the system after an imperfect maintenance action at period k. Using the chain rule, Equation (7.24) can be given as:

$$\begin{aligned}
P\left(y_{k+1},x_{k+1}\middle|H^{k+1}\right) &= \sum_{x'_k} P\left(y_{k+1}\middle|x_{k+1},y'_k\right)P\left(x_{k+1}\middle|x'_k,H^k,y_k,o_k,y'_k\right)P\left(x'_k\middle|H^k,y_k,o_k,y'_k\right)\\
&=\sum_{x'_k} P\left(y_{k+1}\middle|x_{k+1},y'_k\right)P\left(x_{k+1}\middle|x'_k\right)P\left(x'_k\middle|H^k,y_k,o_k,y'_k\right)\\
&=\sum_{x_k}\sum_{x'_k} P\left(y_{k+1}\middle|x_{k+1},y'_k\right)P\left(x_{k+1}\middle|x'_k\right)P\left(x'_k\middle|x_k,H^k,y_k,o_k,y'_k\right)P\left(x_k\middle|H^k,y_k,o_k,y'_k\right)\\
&=\sum_{x_k}\sum_{x'_k} P\left(y_{k+1}\middle|x_{k+1},y'_k\right)P\left(x_{k+1}\middle|x'_k\right)P\left(x'_k\middle|x_k\right)P\left(x_k\middle|H^k,y_k,o_k\right).
\end{aligned} \tag{7.25}$$

Based on the definitions in Equations (7.5) and (7.7), Equation (7.25) can be given by:

$$\begin{aligned}
&P\left(y_{k+1}=m,x_{k+1}=n\middle|H^k,y_k,o_k,y'_k=l\right)\\
&=\sum_{i=1}^{N}\sum_{j=1}^{N} P\left(y_{k+1}=m\middle|x_{k+1}=n,y'_k=l\right)P\left(x_{k+1}=n\middle|x'_k=j\right)P\left(x'_k=j\middle|x_k=i\right)P\left(x_k=i\middle|H^k,y_k,o_k\right)\\
&=\sum_{i=1}^{N}\sum_{j=1}^{N} q_{lm}^{n}p_{jn}^{(\Delta)}r_{ij}\pi_i^k\left(y_k,o_k\right),\ 1\le l,m,n\le N,
\end{aligned} \tag{7.26}$$

where $\pi_i^k\left(y_k,o_k\right)$ is the belief state of the system at decision period k, updated based on the expert judgement y_k and the simple inspection outcome o_k. The calculation of $\pi_i^k\left(y_k,o_k\right)$ will be given in Paragraph 7.3.3.2.

By incorporating Equation (7.26) into Equation (7.12), we can calculate $\pi_j^{k+1}\left(y_k\right)$ when a simple inspection and an imperfect maintenance action are conducted at decision period $k+1$ as:

$$\pi_j^{k+1}\left(y_{k+1}=m\right)=P\left(x_{k+1}=n\middle|H^k,y_{k+1}=m,o_k\right)=\frac{\sum_{i=1}^{N}\sum_{j=1}^{N} q_{lm}^{n}p_{jn}^{(\Delta)}r_{ij}\pi_i^k\left(y_k,o_k\right)}{\sum_{s=1}^{N}\sum_{i=1}^{N}\sum_{j=1}^{N} q_{lm}^{s}p_{js}^{(\Delta)}r_{ij}\pi_i^k\left(y_k,o_k\right)}. \tag{7.27}$$

Scenario 4 *– A replacement, after a simple inspection, is carried out at decision period k*

Recall that after a replacement the system is recovered to the 'as good as new' state, State 1. It is also assumed that it takes almost one decision period to carry out the replacement due to the complicated nature of its preparations, and hence when a replacement is carried out at decision period k, the history at the beginning of decision period $k+1$ is the same as the history at the beginning of the process, H^0. In other words, at the beginning of period $k+1$ the system will be in State 1, when a replacement is carried out at decision period k. Therefore, the conditional probability distribution of the system at the beginning of decision period $k+1$ is as follows:

$$\pi_j^{k+1}\left(y_{k+1}\right) = \pi_j^0 = \begin{cases} 1 & \text{for } j = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{7.28}$$

Scenario 5 *– Accurate inspection is conducted at period k, followed by no maintenance action*

In this scenario the history at the beginning of period $k+1$ includes all the history available at the beginning of period k, the expert judgement and the true state of the system at this decision period revealed by accurate inspection. Therefore: $P\left(y_{k+1}, x_{k+1} \middle| H^{k+1}\right)$ in Equation (7.12) is

$$P\left(y_{k+1}, x_{k+1} \middle| H^{k+1}\right) = P\left(y_{k+1}, x_{k+1} \middle| H^k, y_k, x_k\right), \tag{7.29}$$

which can be given as follows, using the chain rule:

$$P\left(y_{k+1}, x_{k+1} \middle| H^{k+1}\right) = P\left(y_{k+1} \middle| x_{k+1}, H^K, x_k, y_k\right) P\left(x_{k+1} \middle| H^k, x_k, y_k\right), \tag{7.30}$$

Since the true state of the system is assumed to follow Markovian evolution, it is independent of the history of information given the previous true state. Also, it is assumed that the expert judgement, given the current true state and the expert judgement at the previous decision period, is independent of the rest of the history of expert judgements and true states. Therefore, Equation (7.30) can be simplified to:

$$P\left(y_{k+1}=m,x_{k+1}=j\middle|H^{k+1}\right)=P\left(y_{k+1}=m\middle|x_{k+1}=j,y_k=l\right)P\left(x_{k+1}=j\middle|x_k=i\right)$$
$$=q^j_{lm}p^{(\Delta)}_{ij} \quad ,1\le i,j,l,m\le N. \tag{7.31}$$

Based on Equation (7.12), the conditional probability distribution of the system at the beginning of decision period $k+1$ for this scenario is given as follows:

$$\pi^{k+1}_j\left(y_{k+1}\right)=P\left(x_{k+1}=j\middle|H^k,y_k=m,x_k=i,y_{k+1}=l\right)=\frac{q^j_{lm}p^{(\Delta)}_{ij}}{\sum_{n=1}^{N}q^n_{lm}p^{(\Delta)}_{in}},\ 1\le i,j,l,m\le N. \tag{7.32}$$

***Scenario 6** – Accurate inspection is conducted at period k, followed by a repair*

In this scenario the history at the beginning of period $k+1$ includes all the history available at the beginning of period k, the expert judgement, the true state of the system and also the updated expert judgement after the imperfect maintenance action during this decision period. Recalling from Section 7.2.2, on the assumption that the system is in State $x_k=i$ at decision period k, after an imperfect maintenance the expert judgement will be $y'_k=l,\ 1\le l\le i,$ where $l=\arg\max_j\left(r_{ij}\right)$. The numerator of the right hand side of Equation (7.12) for this scenario is:

$$P\left(y_{k+1},x_{k+1}\middle|H^{k+1}\right)=P\left(y_{k+1},x_{k+1}\middle|H^k,y_k,x_k,y'_k\right). \tag{7.33}$$

Again, the expert judgement is dependent only on the current true state of the system and the expert judgement at the previous decision epoch and hence the probability of y_{k+1}, given x_{k+1} and y'_k, is independent of other information. Therefore, using the chain rule, Equation (7.33) can be given as:

$$\begin{aligned}
&P\left(y_{k+1}\middle|x_{k+1},H^k,y_k,x_k,y'_k\right)P\left(x_{k+1}\middle|H^k,y_k,x_k,y'_k\right)\\
&\quad=P\left(y_{k+1}\middle|x_{k+1},y'_k\right)P\left(x_{k+1}\middle|H^k,y_k,x_k,y'_k\right)\\
&\quad=\sum_{x'_k}P\left(y_{k+1}\middle|x_{k+1},y'_k\right)P\left(x_{k+1}\middle|x'_k,H^k,y_k,x_k,y'_k\right)P\left(x'_k\middle|H^k,y_k,x_k,y'_k\right)\\
&\quad=\sum_{x'_k}P\left(y_{k+1}\middle|x_{k+1},y'_k\right)P\left(x_{k+1}\middle|x'_k\right)P\left(x'_k\middle|x_k\right),
\end{aligned} \tag{7.34}$$

that is:

$$
\begin{aligned}
&P\left(y_{k+1}=m, x_{k+1}=n \middle| H^k, y_k, x_k=i, y_k'=l\right) \\
&\quad =\sum_{j=1}^{N} P\left(y_{k+1}=m \middle| x_{k+1}=n, y_k'=l\right) P\left(x_{k+1}=n \middle| x_k'=j\right) P\left(x_k'=j \middle| x_k=i\right) \\
&\quad =\sum_{j=1}^{N} q_{lm}^{n} p_{jn}^{(\Delta)} r_{ij},\ \ 1 \le i,n,l,m \le N.
\end{aligned} \tag{7.35}
$$

Substituting Equation (7.35) into Equation (7.12) yields the conditional probability distribution of the system at the beginning of decision period $k+1$ for this scenario as:

$$
\begin{aligned}
\pi_n^{k+1}\left(y_{k+1}\right) &= P\left(x_{k+1}=n \middle| H^k, y_k, x_k=i, y_k'=l, y_{k+1}=m\right) \\
&= \frac{\sum_{j=1}^{N} q_{lm}^{n} p_{jn}^{(\Delta)} r_{ij}}{\sum_{s=1}^{N}\sum_{j=1}^{N} q_{lm}^{s} p_{js}^{(\Delta)} r_{ij}},\ \ 1 \le i,n,l,m \le N.
\end{aligned} \tag{7.36}
$$

***Scenario** 7 – Accurate inspection is conducted at period k, followed by a replacement*

When a replacement is carried out after an accurate inspection, for the same reason explained for Scenario 4, the system will be in State 1 at the beginning of decision period $k+1$ and hence, the conditional probability distribution of the system will be as given in Equation (7.10).

7.3.3.2 Updating the Conditional Probability Distribution of System Based on the Outcome of an Inspection

At decision period k after the expert judgement y_k is provided and before conducting the inspection, the probability of observing the simple inspection outcome o_k through the simple inspection, given all the information available is given by:

$$
P\left(o_k \middle| H^k, y_k\right) = \sum_{i=1}^{N} P\left(o_k, x_k=i \middle| H^k, y_k\right). \tag{7.37}
$$

Using the chain rule, $P\left(o_k, x_k = i \middle| H^k, y_k\right)$ can be given by:

$$\begin{aligned} P\left(o_k, x_k = i \middle| H^k, y_k\right) &= P\left(o_k \middle| x_k = i, H^k, y_k\right) P\left(x_k = i \middle| H^k, y_k\right) \\ &= P\left(o_k \middle| x_k = i\right) P\left(x_k = i \middle| H^k, y_k\right) = b_i\left(o_k\right) \pi_i^k\left(y_k\right), \end{aligned} \tag{7.38}$$

where $b_i\left(o_k\right)$ is the probability of observing o_k as the output of the simple inspection when the system is in State i. Incorporating Equation (7.38) into Equation (7.37) yields:

$$P\left(o_k \middle| H^k, y_k\right) = \sum_{i=1}^{N} b_i\left(o_k\right) \pi_i^k\left(y_k\right). \tag{7.39}$$

Assuming that the outcome of the simple inspection at period k is o_k, the belief state of the system is updated based on the outcome of the simple inspection, that is:

$$\pi_i^k\left(o_k, y_k\right) = P\left(x_k = i \middle| H^k, y_k, o_k\right). \tag{7.40}$$

Using Bayes' theorem, Equation (7.40) can be given by:

$$\pi_i^k\left(o_k, y_k\right) = \frac{P\left(x_k = i, o_k \middle| H^k, y_k\right)}{P\left(o_k \middle| H^k, y_k\right)}. \tag{7.41}$$

Substituting Equation (7.38) and Equation (7.39) into Equation (7.41) yields:

$$\pi_i^k\left(o_k, y_k\right) = \frac{\pi_i^k\left(y_k\right) b_i\left(o_k\right)}{\sum_{j=1}^{N} \pi_j^k\left(y_k\right) b_j\left(o_k\right)}. \tag{7.42}$$

When an accurate inspection is conducted, the true state of the system is revealed. Therefore the outcome of the accurate inspection at period k will be the true state of the system, x_k. In other words, the conditional probability of the system upon an accurate inspection is updated as,

$$\pi_i^k\left(x_k\right) = P\left(x_k = i \middle| H^k, y_k, x_k = j\right) = \begin{cases} 1 & \text{for } i = j, \\ 0 & \text{otherwise,} \end{cases} \quad 1 \le i, j \le N. \tag{7.43}$$

7.3.3.3 Updating the Conditional Probability Distribution of System after Imperfect Maintenance Action

7.3.3.3.1 Imperfect Maintenance Action Conducted after Simple Inspection

First, consider the scenario where an imperfect maintenance action is conducted after simple inspection. Again, the belief state of the system is updated based on all the information available after carrying out an imperfect maintenance action. The history after a simple inspection at period k consists of the history at the beginning of period k, expert judgement and the outcome of the simple inspection at this period. Let $\pi_j^k(SM)$ denote the updated conditional probability of the system being in State j given all the information available after conducting an imperfect maintenance action, where "SM" implies simple inspection followed by imperfect maintenance action. $\pi_j^k(SM)$ is defined as:

$$\begin{aligned}\pi_j^k(SM) &= P\left(x_k' = j \middle| H^k, y_k, o_k\right) \\ &= \sum_{i=1}^{N} P\left(x_k = i \middle| H^k, y_k, o_k\right) P\left(x_k' = j \middle| x_k = i, H^k, y_k, o_k\right),\end{aligned} \tag{7.44}$$

where x_k' denotes the deterioration state of the system after an imperfect maintenance action at decision period k. As defined in Equation (7.6), given that the system is $x_k = i$ at period k, it is assumed that an imperfect maintenance action transfers the system to an intermediate state $x_k' = j$ with probability r_{ij}. Therefore $\pi_j^k(SM)$ can be given by:

$$\pi_j^k(SM) = \sum_{i=1}^{N} P\left(x_k = i \middle| H^k, y_k, o_k\right) P\left(x_k' = j \middle| x_k = i\right) = \sum_{i=1}^{N} \pi_i^k(o_k, y_k) r_{ij}. \tag{7.45}$$

7.3.3.3.2 Imperfect Maintenance Action Conducted after Accurate Inspection

When an accurate inspection is conducted, the true state of the system is revealed. Let $\pi_j^k(AM)$ be the updated probability of the system at period k when imperfect maintenance action is carried out after accurate inspection, where "AM" implies accurate inspection followed by imperfect maintenance action. Then:

$$\pi_j^k\left(AM\right)=P\left(x_k'=j\middle|H^k,y_k,x_k=i\right)=P\left(x_k'=j\middle|x_k=i\right)=r_{ij}. \tag{7.46}$$

7.3.4 Conditional Reliability of the System over a Decision Period

Recalling from Chapter 5 that the underlying deterioration process of the system is assumed to follow discrete-time, discrete-state Markovian evolution, the probability distribution of the time to failure given that the system is in State i, $f_i(t)$, is given by:

$$\begin{aligned} f_i\left(t\right)&=\left(\overline{F}_i p_i\right)^t F_i+\sum_{s=1}^{t}\overline{F}^s p_i^{s-1}\overline{p}_i f_{i+1}(t-s),\ \ 1\le i<N,\\ f_N\left(t\right)&=\overline{F}_N^t F_N, \end{aligned} \tag{7.47}$$

where T_f denotes the random variable representing the time to failure, F_i denotes the probability of failure when the system is in State i and $\overline{F_i}$ is the complementary probability of F_i.

The probability that the system survives for at least t time units, given that it is in State i at the beginning of period k, is denoted by $R\left(k,i,t\right)$:

$$R\left(k,i,t\right)=P\left(T_f>k\Delta+t\middle|T_f>k\Delta,x_k=i\right). \tag{7.48}$$

Based on the definition of $f_i\left(t\right)$ in Equation (7.47), $R\left(k,i,t\right)$ can be calculated as:

$$R\left(k,i,t\right)=1-P\left(T_f\le k\Delta+t\middle|T_f>k\Delta,x_k=i\right)=1-\sum_{s=k\Delta}^{k\Delta+t}f_i\left(s\right). \tag{7.49}$$

Let $\overline{R}\left(k,\pi^k,\Delta\right)$ denote the conditional reliability of the system over one decision period – i.e. the probability that the system is still working at the beginning of period $k+1$, given the probability distribution of the true state of the system π^k:

$$\overline{R}\left(k,\pi^k,\Delta\right)=\sum_{i=1}^{N}\pi_i^k R\left(k,i,\Delta\right). \tag{7.50}$$

7.4 Calculation of the Optimal Cost Function

We introduce the notation $V(.,.)$ to represent the optimal cost function, that is, the minimum expected total discounted cost over a finite horizon. Let K be the total number of periods in the decision-making horizon. We define $V(k,\pi^k)$ as the minimum expected total discounted cost incurred over $K-k$ periods, $1 \le k \le K$, given that the belief state is π^k at period k. We wish to find the optimal inspection and maintenance action choice at each decision epoch so that the expected total discounted cost over the horizon of K decision periods is minimised.

According to the optimality principle in dynamic programming, also known as "Bellman's optimality principle", $V(k,\pi^k)$ can be stated in a recursive form that relates it to the optimal cost function at the next decision epoch. To clarify this optimality principle, let us consider a classical POMDP model with a single decision step per decision period. At the beginning of decision period k, observation z_k, is obtained. Once this observation is obtained, a decision choice a from a set of admissible actions, must be selected so that the total expected discounted cost over

K period planning horizon is minimised. Let $V(k,\pi^k)$ denote the corresponding optimal cost function at decision period k, and $\tau(\pi^k, z_{k+1}, a)$ denote the belief state updated based on the observation z_{k+1} given that the decision choice a has been selected at decision period k. Let φ denote the discount rate, according to the optimality principle in dynamic programming, $V(k,\pi^k)$ satisfies the following recursive equation (Lovejoy, 1991):

$$V(k,\pi^k) = \underbrace{\min}_{a} \left[\underbrace{C_{imd}(\pi^k, a)}_{\text{Term1}} + \varphi \sum_{z_{k+1}} \underbrace{P(z_{k+1} | \pi^k, a) V(k+1, \tau(\pi^k, z_{k+1}, a))}_{\text{Term 2}} \right], \qquad (7.51)$$

where Term 1 denotes the immediate cost incurred by selecting the decision choice a, in the belief state π^k, at the beginning of the decision period k. The bracket term specifies the expected future discounted cost incurred by selecting the decision choice a in the belief state π^k, at decision epoch k, and making optimal decisions afterwards.

In the context of our problem, Bellman's optimality principle implies that the optimal inspection and maintenance action choice must be selected by taking into account the costs that are immediately incurred by acting on these choices, and the expected future costs from the beginning of the next decision epoch incurred by making optimal decisions afterwards. In what follows we compute the optimal cost functions according to each decision step, by breaking them down into immediate and future expected costs.

7.4.1 Decision step 1: Inspection Type Selection

At the beginning of the decision period k the conditional probability distribution of the system is updated based on the expert judgement, $\pi^k(y_k)$. Recall Section 7.1 that, following the expert judgement at the beginning of a decision period, the decision choices are (1) to leave the system until the next decision occasion; (2) to carry out simple inspection; and (3) to conduct accurate inspection.

Let $W^I(k,\pi^k(y_k))$ denote the expected total discounted cost over $K-k$ periods if it is chosen to not conduct any inspection but wait until the next decision period when the next expert judgement is provided and make optimal decisions afterwards, $S(k,\pi^k(y_k))$ denote the expected total discounted cost over $K-k$ periods when simple inspection is conducted at decision period k and optimal decisions are made afterwards, and $A(k,\pi^k(y_k))$ denote the expected total discounted cost over $K-k$ periods when at decision period k accurate inspection is conducted and optimal decisions are made afterwards.

The optimal cost function $V(k,\pi^k(y_k))$ satisfies the following optimality equation according to the optimality principle in dynamic programming:

$$V(k,\pi^k(y_k)) = \min\left[W^I(k,\pi^k(y_k)), S(k,\pi^k(y_k)), A(k,\pi^k(y_k))\right], \tag{7.52}$$

The calculation of $W^I\left(k,\pi^k\left(y_k\right)\right)$, $S\left(k,\pi^k\left(y_k\right)\right)$ and $A\left(k,\pi^k\left(y_k\right)\right)$ is given in the following paragraphs.

7.4.1.1 No Inspection Selected

$W^I\left(k,\pi^k\left(y_k\right)\right)$ in Equation (7.52) is the minimum expected total discounted cost over $K-k$ periods when at decision period k it is decided to not conduct any inspection, that is, when the inspection decision variable $a_k^I=0$ (recall Figure 7.1). Based on the optimality principle in dynamic programming, it is equal to the expected cost of a failure occurring before period $k+1$, plus the minimum expected future cost starting from period $k+1$. To evaluate the future cost we should consider all possible expert judgement outcomes at period $k+1$, the likelihood of their occurrence $P\left(y_{k+1}\middle|H^k,y_k\right)$, and their resulting future minimum expected discounted cost, $V\left(k+1,\pi^{k+1}\left(y_k+1\right)\right)$, given that the system is still working at the beginning of decision period $k+1$:

$$W^I\left(k,\pi^k\left(y_k\right)\right)=\left(C^F+\varphi V\left(k+1,\pi^0\right)\right)\left(1-\bar{R}\left(k,\pi^k\left(y_k\right),\Delta\right)\right)$$

$$+\varphi\left(\sum_{y_{k+1}}P\left(y_{k+1}\middle|H^k,y_k\right)V\left(k+1,\pi^{k+1}\left(y_{k+1}\right)\right)\right)\bar{R}\left(k,\pi^k\left(y_k\right),\Delta\right), \qquad (7.53)$$

where φ is the discount rate. $\left(C^F+\varphi V\left(k+1,\pi^0\right)\right)$ is the expected discounted cost of a failure replacement, i.e. the immediate cost of failure plus the future expected discounted cost. Recall from Paragraph 7.3.3.1 that a replacement renews the system to "good as new" condition and hence the belief state is updated to π^0. $\bar{R}\left(k,\pi^k\left(y_k\right),\Delta\right)$ is the conditional reliability of the system over decision period k given the belief state $\pi^k\left(y_k\right)$, and $\left(1-\bar{R}\left(k,\pi^k\left(y_k\right),\Delta\right)\right)$ is the probability of having a failure at period k. Then $\varphi\left(\sum_{y_{k+1}}P\left(y_{k+1}\middle|H^k,y_k\right)V\left(k+1,\pi^{k+1}\left(y_{k+1}\right)\right)\right)$ is the expected discounted future cost at the beginning of decision period $k+1$ when the next expert judgement is provided, given that the system has survived.

7.4.1.2 Simple Inspection Selected

Returning to Equation (7.52), $S\left(k,\pi^k\left(y_k\right)\right)$ is the total expected discounted cost, if a simple inspection is conducted after expert judgement is provided at decision interval k. It is equal to the immediate cost of conducting a simple inspection plus the future expected discounted cost. To calculate the future cost, we need to consider all possible conditional probability distributions of the system. As given in Equation (7.42), when a simple inspection is conducted, the conditional belief state is updated based on the inspection outcome. Therefore we need to take an expectation over all possible observations and their resulting future costs, thus:

$$S\left(k,\pi^k\left(y_k\right)\right)=C^S+\sum_{o_k}P\left(o_k\left|H^k,y_k\right.\right)\,V^S\left(k,\pi^k\left(y_k,o_k\right)\right), \tag{7.54}$$

where C^S is the immediate cost of conducting a simple inspection; $\sum_{o_k}P\left(o_k\left|H^k,y_k\right.\right)\,V^S\left(k,\pi^k\left(y_k,o_k\right)\right)$ is the future expected cost if we choose to conduct a simple inspection; and $V^S\left(k,\pi^k\left(y_k,o_k\right)\right)$ is the minimum future expected discounted cost once the observation o_k is provided through the simple inspection (and will be calculated in Paragraph 7.4.2.1). Equation (7.54) can be given as:

$$\begin{aligned}S\left(k,\pi^k\left(y_k\right)\right)&=C^S+\sum_{o_k}\sum_{i=1}^{N}P\left(x_k=i\left|H^k,y_k\right.\right)\,P\left(o_k\left|x_k,H^k,y_k\right.\right)V^S\left(k,\pi^k\left(y_k,o_k\right)\right)\\&=C^S+\sum_{o_k}\sum_{i=1}^{N}P\left(x_k=i\left|H^k,y_k\right.\right)\,P\left(o_k\left|x_k\right.\right)V^S\left(k,\pi^k\left(y_k,o_k\right)\right)\\&=C^S+\sum_{o_k}\sum_{i=1}^{N}\pi_i^k\left(y_k\right)\,b_i\left(o_k\right)V^S\left(k,\pi^k\left(y_k,o_k\right)\right).\end{aligned} \tag{7.55}$$

7.4.1.3 Accurate Inspection Selected

Returning to Equation (7.52), $A\left(k,\pi^k\left(y_k\right)\right)$ is the total expected discounted cost if we choose to conduct an accurate inspection at period k, after the expert judgement is provided. It is equal to the immediate cost of conducting an accurate inspection plus the expected future cost. The expected future cost is computed by taking the expectation over all the possible outcomes of the accurate inspection and their corresponding costs. Recalling that the accurate inspection reveals the true state of the system, we have:

$$\begin{aligned} A\left(k,\pi_k\left(y_k\right)\right) &= C^A + \sum_{i=1}^{N} P\left(x_k = i \middle| H^k, y_k\right) V^A\left(k, x_k = i\right) \\ &= C^A + \sum_{i=1}^{N} \pi_i^k\left(y_k\right) V^A\left(k, x_k = i\right), \end{aligned} \tag{7.56}$$

where C^A is the immediate cost incurred by an accurate inspection; $V^A(k, x_k = i)$ is the minimum total expected discounted cost, assuming that the output of the accurate inspection is the true state $x_k = i$ (and will be calculated in Paragraph 7.4.2.2); and $\sum_{i=1}^{N} \pi_i^k\left(y_k\right) V^A\left(k, x_k = i\right)$ is the future expected cost if an accurate inspection is conducted at period k, after the expert judgement is provided.

7.4.2 Decision Step 2: Maintenance Action Selection

7.4.2.1 Selecting Maintenance Action after a Simple Inspection

At the second decision step, when the choice of the maintenance action is to be selected, let $V^S\left(k, \pi^k\left(y_k, o_k\right)\right)$ denote the minimum expected total discounted cost over $K-k$ periods, given that the simple inspection has been conducted and the output of the simple inspection is o_k. Based on the optimality principle in dynamic programming, $V^S\left(k, \pi^k\left(y_k, o_k\right)\right)$ satisfies the following optimality equation:

$$V^S\left(k,\pi^k\left(y_k,o_k\right)\right) = \min\left[W^S\left(k,\pi^k\left(y_k,o_k\right)\right), M^S\left(k,\pi^k\left(y_k,o_k\right)\right), r\left(k\right)\right], \tag{7.57}$$

where, $W^S\left(k, \pi^k\left(y_k, o_k\right)\right)$ is the expected cost if, after a simple inspection at period k, it is chosen to not perform any maintenance action and instead wait until the next decision epoch; $M^S\left(k, \pi^k\left(y_k, o_k\right)\right)$ is the expected cost if, at period k, an imperfect maintenance action is performed upon a simple inspection; and $r\left(k\right)$ is the expected cost if a preventive replacement is carried out at decision period k. The calculations of these costs are given in the following paragraphs.

7.4.2.1.1 No Maintenance Action Carried Out after a Simple Inspection

$W^S\left(k,\pi^k\left(y_k,o_k\right)\right)$ is the expected cost if it is chosen to not perform any maintenance action and instead wait until the next decision epoch, after a simple inspection is conducted at decision period k. It is calculated as the expected cost of having a failure before period $k+1$ plus the expected future cost starting from period $k+1$. To evaluate the future cost we should consider all expert judgements at period $k+1$, the likelihood of their occurrence given all the information available, $P\left(y_{k+1}\middle|H^k,y_k,o_k\right)$, and their resulting future expected cost, $V\left(k+1,\pi^{k+1}\left(y_k+1\right)\right)$, provided that the system is still working at the beginning of period $k+1$. Thus:

$$W^S\left(k,\pi^k\left(y_k,o_k\right)\right)=\left(C^F+\varphi V\left(k+1,\pi^0\right)\right)\left(1-\bar{R}\left(k,\pi^k\left(y_k,o_k\right),\Delta\right)\right)$$
$$+\varphi\left(\sum_{y_{k+1}}P\left(y_{k+1}\middle|H^k,y_k,o_k\right)V\left(k+1,\pi^{k+1}\left(y_{k+1}\right)\right)\right)\bar{R}\left(k,\pi^k\left(y_k,o_k\right),\Delta\right), \quad (7.58)$$

where $\left(C^F+\varphi V\left(k+1,\pi^0\right)\right)$ is the total expected discounted cost of a failure replacement; $\bar{R}\left(k,\pi^k\left(y_k,o_k\right),\Delta\right)$ is the probability that the system is still working at the beginning of period $k+1$, given the probability distribution of the true state of the system updated based on the simple inspection outcome $\pi^k\left(y_k,o_k\right)$; $\left(1-\bar{R}\left(k,\pi^k\left(y_k,o_k\right),\Delta\right)\right)$ is the probability of having a failure at decision period k; and $\left(\sum_{y_{k+1}}P\left(y_{k+1}\middle|H^k,y_k,o_k\right)V\left(k+1,\pi^{k+1}\left(y_{k+1}\right)\right)\right)$ is the expected future cost at the beginning of period $k+1$ when the expert judgement y_{k+1} is provided, given that the system has survived. The likelihood of the expert judgement y_{k+1} given all the information available after a simple inspection is conducted at period k is calculated as follows:

$$P\left(y_{k+1}\middle|H^k,y_k,o_k\right)=\sum_{x_{k+1}}P\left(y_{k+1},x_{k+1}\middle|H^k,y_k,o_k\right). \quad (7.59)$$

Using Equation (7.20), Equation (7.59) can be given by:

$$P\left(y_{k+1}=m \middle| H^k, y_k=l, o_k\right)=\sum_{j=1}^{N} P\left(y_{k+1}=m, x_{k+1}=j \middle| H^k, y_k=l, o_k\right)$$
$$=\sum_{j=1}^{N}\sum_{i=1}^{N} q_{lm}^{i} p_{ij}^{(\Delta)} \pi_i^k\left(y_k=l, o_k\right),\ 1 \le l, m \le N. \tag{7.60}$$

7.4.2.1.2 *Imperfect Maintenance Action Carried Out after a Simple Inspection*

Returning to Equation (7.57), $M^S\left(k, \pi^k\left(y_k, o_k\right)\right)$ is the expected total discounted cost if imperfect maintenance action is carried out after a simple inspection at period k. It is equal to the immediate cost of performing an imperfect maintenance action plus the future expected cost. To calculate the future cost, we need to consider all the possible belief states and their corresponding future costs. Therefore, we take the expectation over all possible expert judgements at the next decision epoch, given all the information available after the imperfect maintenance action, which includes the updated expert judgement y_k'. Thus:

$$M^S\left(k, \pi^k\left(y_k, o_k\right)\right)=C^M+\left(C^F+\varphi V\left(k+1, \pi^0\right)\right)\left(1-\bar{R}\left(k, \pi^k\left(SM\right), \Delta\right)\right)$$
$$+\varphi\left(\sum_{y_{k+1}} P\left(y_{k+1} \middle| H^k, y_k, o_k, y_k'\right) V\left(k+1, \pi^{k+1}\left(y_{k+1}\right)\right)\right) \bar{R}\left(k, \pi^k\left(SM\right), \Delta\right),$$

(7.61)

where C^M is the immediate cost incurred by performing an imperfect maintenance action; $\pi^k\left(SM\right)$ is the conditional probability distribution of the system updated when an imperfect maintenance action is performed after a simple inspection, and is given by Equation (7.45), $R\left(k, \pi^k\left(SM\right), \Delta\right)$ is the probability that the system is still working at the beginning of period $k+1$ given the updated conditional probability distribution of the system $\pi^k\left(SM\right)$, and is computed by substituting the updated probability distribution $\pi^k\left(SM\right)$ in place of π^k in Equation (7.50);

$\left(C^F + \varphi V\left(k+1, \pi^0\right)\right)$ is the expected total discounted cost of a failure occurring before period $k+1$; $\left(\sum_{y_{k+1}} P\left(y_{k+1} \middle| H^k, y_k, o_k, y'_k\right) V\left(k+1, \pi^{k+1}\left(y_{k+1}\right)\right)\right)$ is the expected future cost at period $k+1$, given that the system is still working at the beginning of this decision period; and $P\left(y_{k+1} \middle| H^k, y_k, o_k, y'_k\right)$ is the probability of the expert judgement at period $k+1$, given all the information available after an imperfect maintenance action is performed at period k. That is:

$$P\left(y_{k+1} \middle| H^k, y_k, o_k, y'_k\right) = \sum_{x_{k+1}} P\left(y_{k+1}, x_{k+1} \middle| H^k, y_k, o_k, y'_k\right). \tag{7.62}$$

Recalling Scenario 3 in Table 7.1, Equation (7.62) can be given as follows, using Equation (7.26):

$$\begin{aligned} P\left(y_{k+1} = m \middle| H^k, y_k, o_k, y'_k = l\right) &= \sum_{n=1}^{N} P\left(y_{k+1} = m, x_{k+1} = n \middle| H^k, y_k, o_k, y'_k = l\right) \\ &= \sum_{n=1}^{N} \sum_{i=1}^{N} \sum_{j=1}^{N} q_{lm}^{n} p_{jn}^{(\Delta)} r_{ij} \pi_i^k\left(y_k, o_k\right),\ 1 \le l, m \le N. \end{aligned} \tag{7.63}$$

7.4.2.1.3 Replacement Carried Out after a Simple Inspection

Returning to Equation (7.57), $r(k)$ is the total expected discounted cost if, upon an inspection at period k, a preventive replacement is carried out. It is evaluated as the immediate cost incurred by a preventive replacement plus the expected future cost once a replacement is carried out. That is:

$$r(k) = C^R + \varphi V\left(k+1, \pi^0\right), \tag{7.64}$$

where C^R is the immediate cost incurred by a preventive replacement and $\varphi V\left(k+1, \pi^0\right)$ is the expected future discounted cost once a replacement is carried out at decision period k.

7.4.2.2 Selecting Maintenance Action after an Accurate Inspection

At decision Step 2, when selecting maintenance action after an accurate inspection, let $V^A(k,x_k)$ denote the minimum expected total discounted future cost, given that an accurate inspection has been conducted at period k and the output of the accurate inspection is the true state of the system x_k. Based on the optimality principle in dynamic programming we have:

$$V^A(k,x_k) = \min\left[W^A(k,x_k), M^A(k,x_k), r(k)\right], \tag{7.65}$$

where $W^A(k,x_k)$ is the expected cost if, after conducting an accurate inspection, no maintenance action is performed; $M^A(k,x_k)$ is the expected cost if an imperfect maintenance action is carried out after an accurate inspection at period k; and $r(k)$ is the expected cost if a preventive replacement is carried out at period k as given by Equation (7.64). The calculations of the costs in Equation (7.65) are given in the following paragraphs.

7.4.2.2.1 No Maintenance Action Carried Out after an Accurate Inspection

$W^A(k,x_k)$ is the expected cost if, after conducting an accurate inspection, no maintenance action is performed. It is computed as the expected cost of a failure occurring before period $k+1$, plus the expected future cost if the system survives until period $k+1$, given all the information available after conducting the accurate inspection. To compute the future expected cost starting from period $k+1$, we need to take an expectation over all possible expert judgements at period $k+1$, given all the information available, and the costs according to each value of the future expert judgement. Thus:

$$\begin{aligned} W^A(k,x_k) = &\left(C^F + \varphi V\left(k+1,\pi^0\right)\right)\left(1 - R(k,x_k,\Delta)\right) \\ &+ \varphi\left(\sum_{y_{k+1}} P\left(y_{k+1} \middle| H^k, x_k, y_k\right) V\left(k+1, \pi^{k+1}\left(y_{k+1}\right)\right)\right) R(k,x_k,\Delta), \end{aligned} \tag{7.66}$$

where $\left(C^F+\varphi V\left(k+1,\pi^0\right)\right)$ is the total expected discounted cost of a failure replacement; $R(k,x_k,\Delta)$ is the probability that the system is still working at period $k+1$, given that the system is in State x_k during the previous period k, and is given by Equation (7.50); $\left(1-R\left(k,x_k,\Delta\right)\right)$ is the probability of a failure occurring while the system is in State x_k at period k, before period $k+1$ is reached; and $\left(\sum_{y_{k+1}} P\left(y_{k+1}\middle|H^k,x_k,y_k\right)V\left(k+1,\pi^{k+1}\left(y_{k+1}\right)\right)\right)$ is the expected future cost at period $k+1$ provided that the system is still working at the beginning of period $k+1$. $P\left(y_{k+1}\middle|H^k,x_k,y_k\right)$ is the probability of the expert judgement at period $k+1$, once an accurate inspection is conducted at period k and is given as follows:

$$P\left(y_{k+1}\middle|H^k,x_k,y_k\right)=\sum_{x_{k+1}} P\left(y_{k+1},x_{k+1}\middle|H^k,x_k,y_k\right), \tag{7.67}$$

and using Equation (7.31) can be simplified to:

$$P\left(y_{k+1}=m\middle|H^k,x_k=i,y_k=l\right)=\sum_{j=1}^{N} q_{lm}^{j} p_{ij}^{(\Delta)},\ 1\le i,l,m\le N. \tag{7.68}$$

7.4.2.2.2 Imperfect Maintenance Action Carried Out after an Accurate Inspection

When an imperfect maintenance action is conducted after an accurate inspection, the conditional probability distribution of the system is updated to $\pi^k\left(AM\right)$, as defined in Equation (7.46). $M^A\left(k,x_k\right)$ in Equation (7.65) is the expected cost of conducting an imperfect maintenance action after an accurate inspection at period k and is calculated as:

$$\begin{aligned} M^A\left(k,x_k\right) &= C^M+\left(C^F+\varphi V\left(k+1,\pi^0\right)\right)\left(1-R\left(k,\pi^k\left(AM\right),\Delta\right)\right) \\ &\quad +\varphi\left(\sum_{y_{k+1}} P\left(y_{k+1}\middle|H^k,y_k,x_k,y_k'\right)V\left(k+1,\pi^{k+1}\left(y_{k+1}\right)\right)\right)R\left(k,\pi^k\left(AM\right),\Delta\right), \end{aligned} \tag{7.69}$$

where C^M is the immediate cost of performing an imperfect maintenance; $\left(C^F + \varphi V\left(k+1, \pi^0\right)\right)$ is the expected discounted cost of a failure occurring at period k, given that an imperfect maintenance action has been carried out after an accurate inspection at period k; and

$\left(\sum_{y_{k+1}} P\left(y_{k+1} \middle| H^k, y_k, x_k, y'_k\right) V\left(k+1, \pi^{k+1}\left(y_{k+1}\right)\right)\right)$ is the expected future cost at period $k+1$ given that the system is still working at the beginning of period $k+1$. $P\left(y_{k+1} \middle| H^k, y_k, x_k, y'_k\right)$ is the probability of the expert judgement at period $k+1$, given all the information available after an accurate inspection is conducted at period k followed by an imperfect maintenance action, and is given by:

$$P\left(y_{k+1} \middle| H^k, y_k, x_k, y'_k\right) = \sum_{x_{k+1}} P\left(y_{k+1}, x_{k+1} \middle| H^k, y_k, x_k, y'_k\right), \tag{7.70}$$

where y'_k is the updated expert judgement after an imperfect maintenance action is conducted at decision period k. Based on Equation (7.35) we have:

$$\begin{aligned} &P\left(y_{k+1} = m \middle| H^k, y_k, x_k = i, y'_k = l\right) \\ &\quad = \sum_{n=1}^{N} \sum_{j=1}^{N} P\left(y_{k+1} = m \middle| x_{k+1} = n, y'_k = l\right) P\left(x_{k+1} = n \middle| x'_k = j\right) P\left(x'_k = j \middle| x_k = i\right) \\ &\quad = \sum_{n=1}^{N} \sum_{j=1}^{N} q_{lm}^n p_{jn}^{(\Delta)} r_{ij}, \qquad 1 \le i, l, m \le N. \end{aligned} \tag{7.71}$$

The breakdown of the minimum expected total discounted cost at period k, according to the choices of inspection and maintenance action at the two decision steps, is depicted in Figure 7.4.

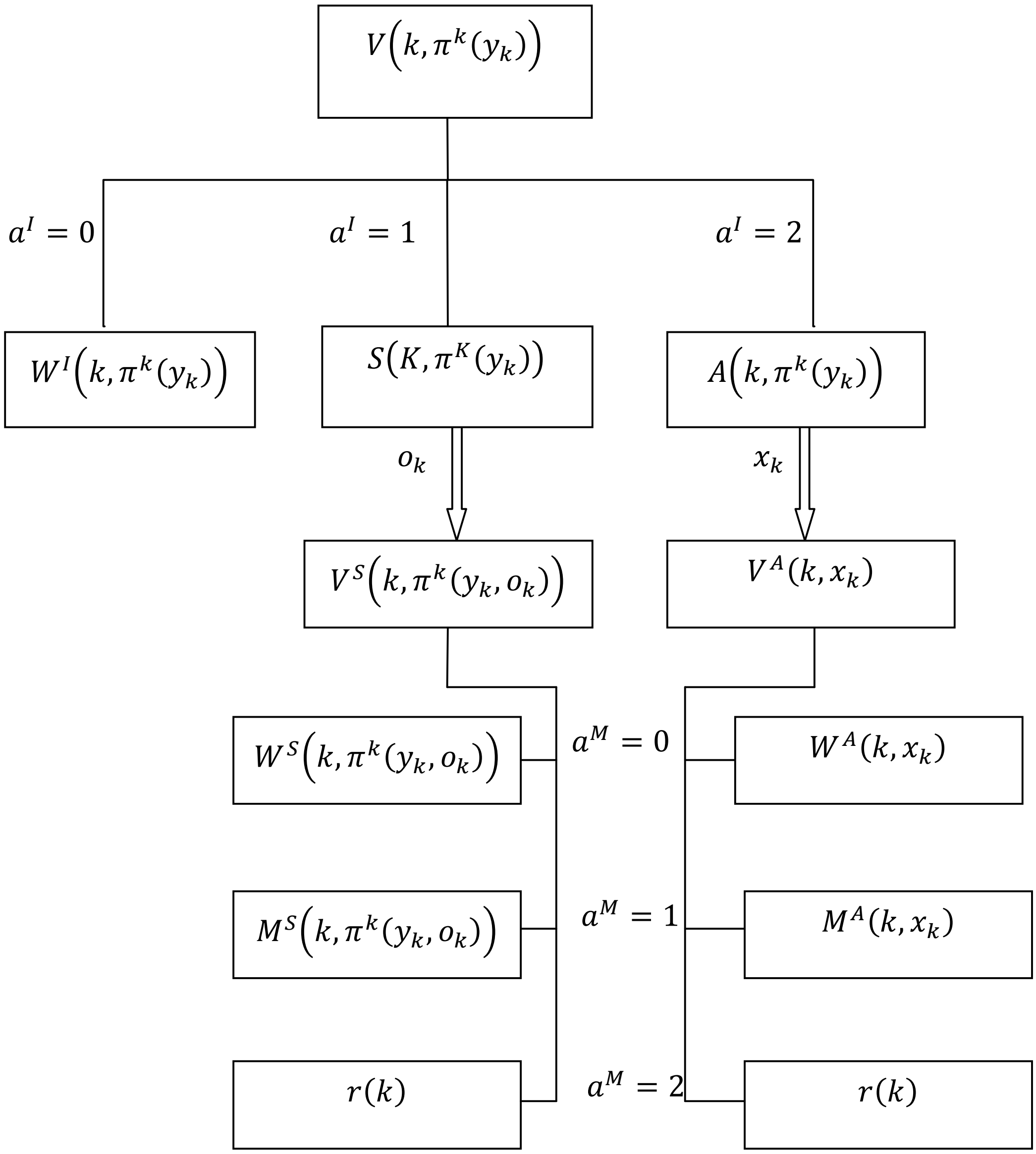

Figure 7.4: Breakdown of the minimum expected total cost at decision period *k* according to the choices of inspection at decision Step 1 and maintenance action at decision Step 2. The block arrows represent the flow of the information obtained through inspections.

7.5 Summary

A maintenance decision optimisation model, formulated as a Partially Observable Markov Decision Process (POMDP), was developed in this chapter. Since the true deterioration state of the maintained system is unknown, it is inferred from the history of all information available at discrete time decision epochs. We thus introduced the "belief state" that is the conditional probability distribution of the deterioration state given all past expert judgements, observations and maintenance actions. As illustrated in Figure 7.1, at each decision epoch, the decision regarding the maintenance policy is made in a maximum of two steps. Following each decision step, the belief state is updated so that the updated belief state at Step 1 will be the control factor for Step 2. According to the optimality principle in dynamic programming, we obtained the optimal cost as a function of the belief state for each decision step over a finite planning horizon. The breakdown of the optimal cost according to the choices of inspection at Step 1 and maintenance action at Step 2 of a decision epoch is illustrated in Figure 7.4. The optimal policy can be theoretically obtained by recursion from Equations (7.52)–(7.71). However, since the number of belief states is infinite, it is computationally unfeasible to update all of them given a choice of action. In Chapter 8, we develop an approximation method to solve the decision optimisation model.

8 Derivation and Sensitivity Analysis of the Optimal Policy

In this chapter we propose an approximation method to find the optimal policy for the two-step Partially Observable Markov Decision Process (POMDP) formulated in Chapter 7. Recall from Section 4.4 that POMDPs (Monahan, 1982; Lovejoy, 1991) are generalisations of Markov Decision Processes (MDPs) in which it is not assumed that the system state at each decision time epoch is precisely known. A policy for a Markov Decision Process (MDP) is a rule that specifies which action should be taken in each state. Solving an MDP means finding an optimal policy with respect to an objective function. An MDP over a finite planning horizon can be numerically solved by backward induction (Puterman, 1994), which is also called "value iteration" for solving MDPs (Sheskin, 2010). For the proposed POMDP the belief states are defined as the conditional probability distribution of the system's deterioration state. Since the number of these states is infinite, it is computationally not feasible to update all of them given a choice of action. We hence propose an approximation method to find the optimal policy, in which the prior probability distribution of the system states is discretised so that the computation of the optimal cost is only applied to specific belief states.

The approximation solution procedure is first described in Section 8.1. To illustrate the optimal policy and to explore its potential structural features, a numerical example is given in Section 8.2. The sensitivity of the optimal policy and the optimal cost to the parameter values is empirically evaluated in Section 8.3. Experimental sensitivity analyses are conducted in three sections investigating what influence (1) variation in cost parameters, (2) failure rate and (3) accuracy of the expert judgement, have on the optimal policy and the optimal cost.

8.1 Solution Procedure: Backward Induction Algorithm

To find the optimal policy minimising the expected total discounted cost over a finite horizon, we solve the POMDP using backward induction. Since the belief (state) space in our POMDP is the probability distribution of the system state and it is continuous, we first approximate the belief space by discretising it using regular grids. The optimal cost values are then computed and the optimal policy is found for the finite number of belief states in the grid. Let Ω' denote the discretised belief state. We assume that each grid point in Ω' represents a belief state at the beginning of a decision period. In other words, at the beginning of each decision period k, the prior conditional probability distribution of the system state given all information available before obtaining the expert judgement, π^k, is represented by the finite number of grids in Ω'.

As seen in Equations (7.52)–(7.71), the optimal cost at decision Step 1 of each period depends on the optimal cost at the next decision period as well as the costs associated with the maintenance actions at decision Step 2 in the same period. The breakdown of these costs is illustrated in Figure 7.4. To find the optimal grid-based policy over a finite planning horizon we step backward from the last decision period in the planning horizon, finding first the optimal maintenance action at decision Step 2 and then the optimal inspection choice at decision Step 1.

For each decision period k, $1 \le k \le K$, and $x_k \in \{1,2,...,N\}$ we first compute the expected total discounted cost associated with the preventive replacement performed after an accurate inspection, i.e. $r(k)$; the expected total discounted cost associated with imperfect maintenance action conducted after an accurate inspection, i.e. $M^A(k,x_k)$; and the expected total discounted cost incurred by taking no action after an accurate inspection, i.e. $W^A(k,x_k)$, using the minimum expected total discounted cost at decision period $k+1$, i.e. $V\left(k+1,\pi^{k+1}\left(y_{k+1}\right)\right)$. The optimal maintenance action according to the system state at this decision period is then found, as given by Equation (7.65). At this step the optimal maintenance action a_k^M is determined for each accurate inspection outcome, i.e. true state of the system. In other words, for each decision period in the planning horizon, first each true system state

$x_k \in \{1,2,...,N\}$ is mapped to an action $a_k^M \in \{0,1,2\}$. Then, for each simple inspection outcome $o_k \in \{1,2,...,Z\}$, the optimal cost functions $M^S\left(k,\pi^k\left(y_k,o_k\right)\right)$ and $W^S\left(k,\pi^k\left(y_k,o_k\right)\right)$ are computed for all the grid points in Ω'. The optimal policy for each point in the grid according to each simple inspection outcome is then found using Equation (7.57).

Having computed the optimal cost functions $V^A\left(k,x_k\right)$ and $V^S\left(k,\pi^k\left(y_k,o_k\right)\right)$ for decision period k, we then compute the cost functions $A\left(k,\pi^k\left(y_k\right)\right), S\left(k,\pi^k\left(y_k\right)\right)$, $W^I\left(k,\pi^k\left(y_k\right)\right)$ and finally find the optimal inspection choice for each point in Ω' using Equation (7.52).

The algorithm presented in the next page codifies the procedure of the backward induction explained above; it finds the two-step grid-based optimal policy for a planning horizon containing K decision intervals. $V\left(K+1,\pi^{K+1}\left(y_{K+1}\right)\right)$ denotes the optimal cost value at the beginning of the decision period $K+1$, i.e. at the end of the planning horizon. The value of $V\left(K+1,\pi^{K+1}\left(y_{K+1}\right)\right)$ can be assigned according to the application situation; for instance the salvage value of the maintained system might be taken into account. Setting $V\left(K+1,\pi^{K+1}\left(y_{K+1}\right)\right)=0,\ \forall\pi^{K+1}\left(y_{K+1}\right)$, implies that, when making optimal decisions, at the final decision period, K, only the costs incurred over this decision period are taken into account.

Backward Induction Algorithm: an Approximated Solution for the Two-Step POMDP

For a given value of the parameters, $P, F_i, r, B, Q(i), C^S, C^A, C^M, C^R, C^F$ and the terminal cost value $V\left(K+1, \pi^{K+1}\left(y_{K+1}\right)\right)$, do the following steps:

- For $k = K, K-1, \ldots, 1$ repeat:
  1. Compute the expected total discounted cost associated with preventive replacement, i.e. $r(k)$.
  2. For $x_k = 1, 2, \ldots, N$ compute $M^A(k, x_k)$ and $W^A(k, x_k)$, and then find the optimal cost $V^A\left(k, x_k\right)$ and the optimal maintenance action.
  3. For $y_k = 1, 2, \ldots, N$ and $\forall \pi^k\left(y_k\right) \in \Omega'$ repeat:
     a. For $o_k = 1, 2, \ldots, Z$ compute $W^S\left(k, \pi^k\left(y_k, o_k\right)\right)$ and $M^S\left(k, \pi^k\left(y_k, o_k\right)\right)$ and then find the optimal cost $V^S\left(k, \pi^k\left(y_k, o_k\right)\right)$ and the optimal maintenance action.
     b. Compute $W^I\left(k, \pi^k\left(y_k\right)\right)$, $S\left(k, \pi^k\left(y_k\right)\right)$ and $A\left(k, \pi^k\left(y_k\right)\right)$, and then find the optimal cost $V\left(k, \pi^k\left(y_k\right)\right)$ and the optimal inspection type.

Note that in the proposed procedure the discrete approximation is only applied to the prior probability distribution. When computing the optimal cost, the posterior probabilities, i.e. probability of the system states updated based on the information obtained in a decision period, are not approximated. At the beginning of each decision period, the prior distribution probability π^k gives the probability of the system state as a grid in Ω'. At decision Step 1 the approximated π^k represented by a grid, together with the expert judgement y_k, indicate the optimal inspection choice. At decision Step 2, if a simple inspection has been conducted, the simple inspection outcome o_k, together with the approximated π^k and the expert judgement y_k, specify the optimal maintenance action. If an accurate inspection is conducted, the true deterioration state x_k is revealed which indicates the optimal maintenance action. This policy is demonstrated in Section 8.2.

8.2 Numerical Experiment

In this section we demonstrate the two-step optimal policy using a numerical example. The optimisation procedure mentioned above has been coded in MATLAB (presented in Appendix B). The MATLAB software package has also been used to illustrate the results graphically. The code written to illustrate the proposed two-step optimal policy has been also included in Appendix B.

8.2.1 Assumptions

To conduct the numerical experiment and the sensitivity analysis, we assign the parameter values based on the following assumptions. These assumptions are made to help the intuitive representation of a real scenario.

Assumption 1: $C^S \le C^A \le C^M \le C^R \le C^F$. This assumption states that: (1) conducting an accurate inspection is more costly than a simple inspection, $C^S \le C^A$. This is because of the more complicated, more expensive preparations needed for an accurate inspection. (2) Since conducting a maintenance action results in a longer period of downtime compared to an inspection, causing more revenue to be lost, the cost of conducting an accurate inspection is assumed to be less than carrying out an imperfect maintenance action, $C^A \le C^M$. (3) An imperfect maintenance action is less costly than a preventive replacement, $C^M \le C^R$. (4) A replacement caused by a failure is much more costly than a preventive replacement, $C^R \le C^F$.

Assumption 2: F_i, $1 \le i \le N$, i.e. the probability of the system failing in State i is non-decreasing in i, meaning that the system is more likely to fail in a higher indexed deterioration level reflecting a worse condition.

Assumption 3: p_i, $1 \le \text{i} < \text{N}$, i.e. the probability of self-transition when the system is in State i, is non-increasing in i. This means that, as the deterioration level becomes worse, the system is more likely to make a transition to a higher indexed deterioration level over one time unit.

Assumption 4: During a simple inspection, we are more likely to observe a higher indexed inspection outcome when the system is in a higher deterioration state. We interpret this assumption in the sense of likelihood ratio, that is, the ratio between the elements of the stochastic matrix that specifies the probabilistic relation between the deterioration state and the output of the simple inspection, i.e. $B = \left[b_i(o);\ 1 \le i \le N,\ 1 \le o \le Z \right]$,

$$\frac{b_j(\theta)}{b_j(o)} \ge \frac{b_i(\theta)}{b_i(o)} \quad \text{for } 1 \le i \le j \le N \ \text{ and } \ 1 \le o \le \theta \le Z,$$

where $b_i(o)$ is the probability of observing the simple inspection outcome o given that the system is in State i. This property of the stochastic matrices is referred to as their being "totally positive of order 2", or TP_2 in short, in the related literature (Rosenfield, 1976).

Assumption 5: The $N \times N$ transition probability matrix that reflects the efficiency of the imperfect maintenance action, i.e. $r = \left[r_{ij};\ 1 \le i, j \le N \right]$, is assumed to be a lower triangle matrix. This means that the imperfect maintenance action does not worsen the condition of the system. It is also assumed that as the deterioration level becomes higher it is less likely to recover the system to a lower deterioration level. We state this in the sense of likelihood ratio between the entries of the stochastic matrix r, as follows:

$$\frac{r_{jn}}{r_{jm}} \ge \frac{r_{in}}{r_{im}} \quad \text{for } 1 \le i \le j \le N \text{ and } 1 \le \mathrm{m} \le \mathrm{n} \le \mathrm{N},$$

where r_{ij} is the probability of restoring the system from State i to j by performing an imperfect maintenance action.

Assumption 6: When the expert's assessment is that the system is in a worse condition, $y_k = j,\ 1 \le l \le j \le N$, the ratio of the likelihood that the next expert judgement state is in a higher indexed state, to the likelihood that it is in a lower indexed state is greater than the reverse, that is

$$\frac{P\left(y_{k+1}=n \middle| y_k=k, x_{k+1}=i\right)}{P\left(y_{k+1}=m \middle| y_k=k, x_{k+1}=i\right)} \geq \frac{P\left(y_{k+1}=n \middle| y_k=l, x_{k+1}=i\right)}{P\left(y_{k+1}=m \middle| y_k=l, x_{k+1}=i\right)},$$

$1 \leq m \leq n \leq N$ and $1 \leq l \leq k \leq N$.

Let $Q(i)$ denote the transition probability matrix of the expert judgement states over one decision period, i.e. $Q(i)=\left[q_{lm}^i = p\left(y_{k+1}=m \middle| y_k=l, x_{k+1}=i\right),\ 1 \leq i,l,m \leq N\right]$. We can state this assumption in the sense of likelihood ratio between the elements of the matrix $Q(i)$, $1 \leq i \leq N$, that is

$$\frac{q_{kn}^i}{q_{km}^i} \geq \frac{q_{\ln}^i}{q_{lm}^i} \text{ for } 1 \leq l \leq k \leq N \text{ and } 1 \leq \mathrm{m} \leq \mathrm{n} \leq \mathrm{N}.$$

8.2.2 Parameter Values

Assuming the time unit t to be a week, we consider decision-making on a monthly basis, i.e. $\Delta = 4$. In order to illustrate the optimal policy in a schematic way and hence to observe the possible structural properties, we consider a system with three deterioration states, i.e. $N = 3$.

Based on the assumptions given earlier in this section, the following parameter values are considered for the numerical experiment.

The values considered for vectors P and F, assigned according to Assumptions 2 and 3, are given in Table 8.1.

Table 8.1: Values for the probability of failure and self-transition of the system states.

State	*Probability of remaining at the same state over a time unit*	*Probability of failure over a time unit*
i	p_i	F_i
1	0.99	0.001
2	0.98	0.009
3	1.0	0.1

Based on Equation (7.4) in Section 7.2.1, using the above parameter values, the monthly-based transition probability matrix P^{Δ}, is computed as:

$$P^{\Delta} = \begin{bmatrix} 0.9606 & 0.0382 & 0.0012 \\ 0 & 0.9224 & 0.0776 \\ 0 & 0 & 1.0000 \end{bmatrix}.$$

Also, as given in Sub-Section 7.3.4, since the transition probabilities are assumed to be stationary, the reliability of the system over a decision period will be stationary too, i.e. independent of the decision period. The parameter values in Table 8.1 yield the following reliability vector that is the probability of the system to not failing over a four week period:

$$R = \begin{bmatrix} 0.9739 & 0.6524 & 0.5905 \end{bmatrix}.$$

The values assigned to the stochastic matrices $B, Q(i)$ and r are as follows. Note that the values of these matrices are assigned so that the stochastic relationship between their elements stated in Assumptions 4, 5 and 6 in the sense of likelihood ratio, is satisfied.

$$B = \begin{bmatrix} 0.85 & 0.14 & 0.01 \\ 0.1 & 0.8 & 0.1 \\ 0.01 & 0.19 & 0.8 \end{bmatrix}, \quad r = \begin{bmatrix} 1 & 0 & 0 \\ 0.8 & 0.2 & 0 \\ 0.1 & 0.7 & 0.2 \end{bmatrix},$$

$$Q(1)=\begin{bmatrix} 0.85 & 0.14 & 0.01 \\ 0.2 & 0.7 & 0.1 \\ 0.1 & 0.4 & 0.5 \end{bmatrix},$$

$$Q(2)=\begin{bmatrix} 0.15 & 0.8 & 0.05 \\ 0.05 & 0.7 & 0.25 \\ 0.01 & 0.19 & 0.8 \end{bmatrix},\ Q(3)=\begin{bmatrix} 0.1 & 0.3 & 0.6 \\ 0.01 & 0.14 & 0.85 \\ 0.01 & 0.14 & 0.85 \end{bmatrix}.$$

The cost parameters are given by the following values (in £), according to Assumption 1:

$C^S = 1500,\ C^A = 3000,\ C^M = 5000,\ C^R = 20000$ and $C^F = 50000$.

The discount rate φ is given by $0 \le \varphi = \frac{1}{1+r} \le 1$ where r is the interest rate (Chiang, 1984). Since we consider decision-making on a monthly basis, that is $\Delta = 4$, we set the discount rate close to 1, $\varphi = 0.95$, implying a relatively low monthly interest rate, $r = 0.052$. This means that the costs associated with the next decision period, have present values almost equivalent to their values in the next month.

8.2.3 Results

Using the above parameter values, we can numerically solve the two-step POMDP over a two-year planning horizon, i.e. $K = 24$.

We set the terminal cost, i.e. the expected cost at the beginning of the decision period $K+1$, to zero, that is $V\left(K+1, \pi^{K+1}(y_{K+1})\right) = 0,\ \forall \pi^{K+1}(y_{K+1}) \in \Omega'$. To execute the proposed backward induction algorithm, the belief states π_1, π_2 and π_3 are discretised on a lattice with increments of 0.1. The optimal cost functions between the lattice points are approximated using bilinear interpolation (Press et al., 2007). The two-step grid-based optimal policy is obtained for the decision periods $k = 1, 2, ..., 24$, resulting in the total expected discounted cost of £32,846 over the two-year planning horizon.

Figure 8.1 illustrates the grid-based optimal policy, at decision Step 1, for the final decision period, $k = 24$. The X-axis denotes π_1, i.e. the conditional probability of the system being in State 1 and the Y-axis denotes π_2, i.e. the conditional probability of the system being in State 2. Since $\pi_1 + \pi_2 + \pi_3 = 1$, then the discretised belief space can be represented by the grids within the triangular area surrounded by the X-axis, the Y-axis and the line $\pi_1 + \pi_2 = 1$. Each point in the figure denotes an updated belief state, after the expert judgement is provided, that is the conditional probability of the system given the expert judgement y_k, and all other information available at period $k = 24$. To demonstrate the optimal decision rule, the grids have been marked according to their optimal policy.

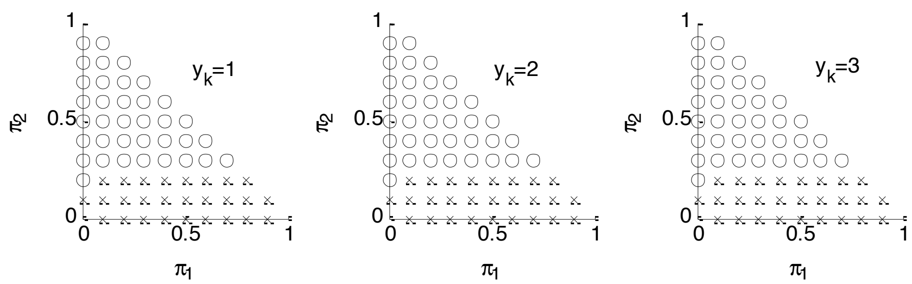

Figure 8.1: Optimal inspection rule at period k=24. × represents no inspection and ○ represents simple inspection.

Note that in the schematic optimal policy, the inspection policy is not shown for the belief states at $(1,0),(0,1)$ and $(0,0)$, which represent the situation when the system is in States $1, 2$ and 3 respectively, with probability of 1. With such certainty, there is no need to conduct an inspection to obtain information about the condition of the system. Recall from Chapter 7 that we assume that such certainty happens when the decision maker has access to the true state of the system, and this is only when an accurate inspection is conducted, or when a replacement is carried out, which transfers the system to State 1.

Based on this grid-based policy, at period $k = 24$, first the expert judgement y_k is obtained and then the corresponding optimal inspection choice is looked up in the grid-based optimal inspection policy in Figure 8.1 and the optimal inspection type, if any, is conducted accordingly. At this decision period the grid-based optimal policy is partitioned into two sub-regions for which the corresponding optimal action is to not conduct any inspection or to conduct simple inspection.

Once the optimal inspection type is conducted, if the optimal action at decision Step 1 is to conduct an inspection, the optimal maintenance action is carried out given the outcome of the inspection. Figure 8.2 illustrates the optimal decision rule at decision Step 2 for the final period, $k = 24$, corresponding to the outcome of the simple inspection, o_k. Note that the optimal action at decision Step 2 is only shown for the belief states for which the optimal action at decision Step 1 is to conduct a simple inspection.

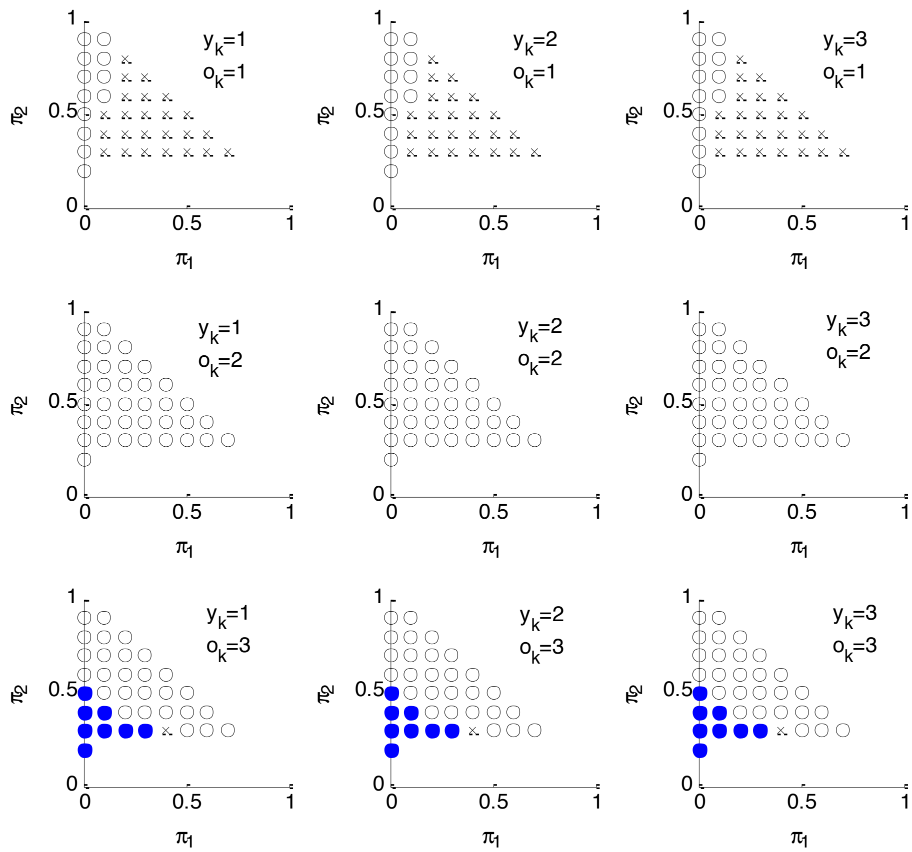

Figure 8.2: Optimal maintenance action rule, after a simple inspection, for k=24, with × representing no action, ○ representing imperfect maintenance action and ● representing replacement.

The optimal policy, at decision Step 2, according to the outcome of an accurate inspection, i.e. the true state of the system, is indicated in Table 8.2.

Table 8.2: Optimal maintenance action, given the outcome of the accurate inspection at k=1,2,…,24.

State	*Optimal maintenance action*
i	a_k^M
1	No action
2	Imperfect maintenance action
3	Replacement

Given the parameter values mentioned above, executing the backward induction algorithm yields a stationary maintenance policy for decision Step 2, according to the outcome of the accurate inspection. As shown in Table 8.2, the optimal policy indicates that the same decision will always be made according to the true state of the system, irrespective of the decision period, k. Based on this decision rule, in every decision epoch, when the system is at State 1, the optimal action is to not take any action and wait until the next decision epoch. The optimal maintenance action when the system is revealed to be in State 2 or State 3 is to carry out an imperfect maintenance action and replacement, respectively.

In this example, the optimal grid-based policy specifies the same decision for a given belief state in the grid, at decision periods $k = 1,2,...,18$. In other words, when there are more than five decision periods to go, given the assigned parameter values, the grid-based optimal policy is stationary. The optimal grid-based policy for decision Steps 1 and 2 for decision periods $k = 1,2,...,18$ are illustrated in Figure 8.3 and Figure 8.4, respectively.

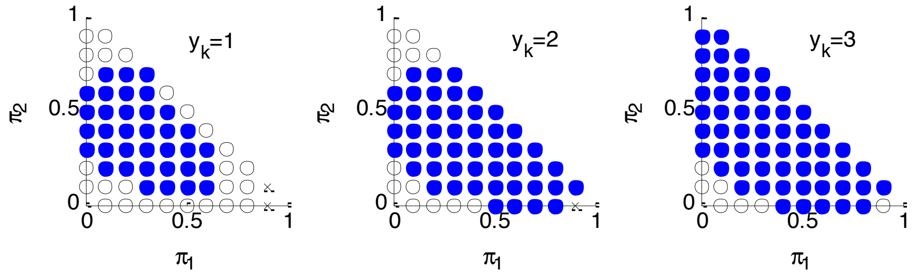

Figure 8.3: The optimal inspection rule for k=1,2,…,18, with × representing no inspection, ○ representing simple inspection and ● representing accurate inspection as the optimal policy.

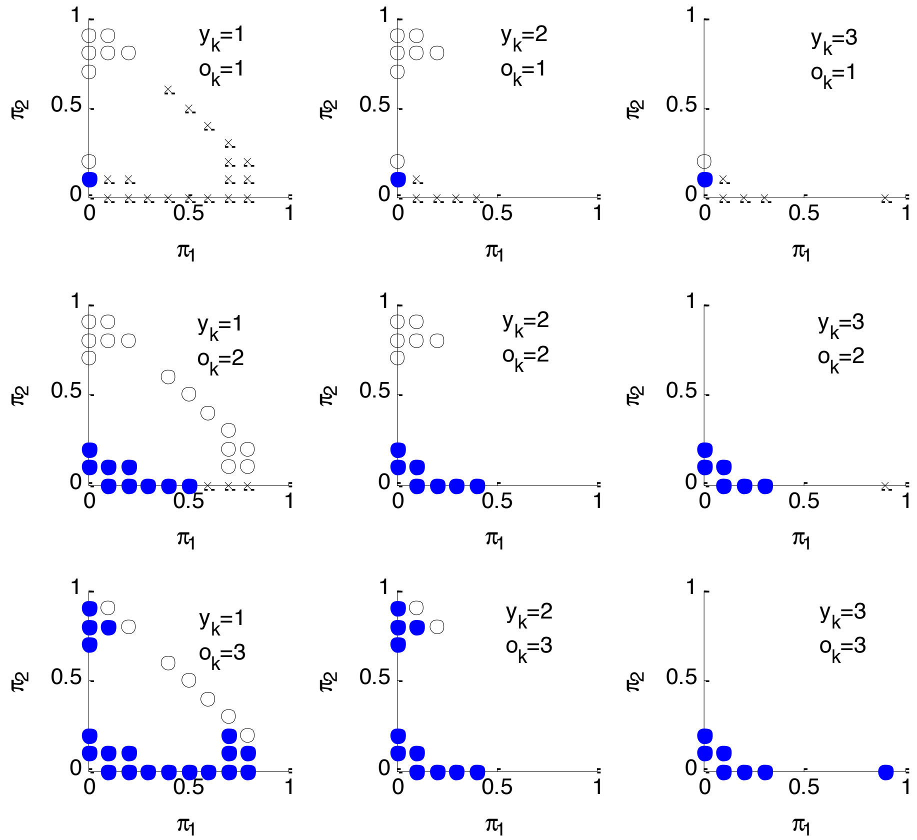

Figure 8.4: The optimal maintenance action rule, after a simple inspection, for k=1,2,...,18, with × representing no action, ○ representing imperfect maintenance action and ● representing replacement.

Figure 8.3 demonstrates that at any decision epoch $k \leq 18$, as the deterioration state ascertained by the expert judgement, y_k, increases, the optimal inspection policy recommends a more conservative decision for the belief states in the grid. In other words, the number of the belief states in the grid for which the optimal decision is to wait until the next decision epoch or to conduct a simple inspection decreases, while the number of the belief states for which accurate inspection is specified as the optimal decision increases.

The same feature applies to the optimal maintenance policy for decision Step 2. As seen in Figure 8.4, as the expert judgement y_k, and the observation o_k, increase, the number of the belief states for which the optimal action is to do nothing or to carry out an imperfect maintenance action decreases, while the number of the grids indicating a replacement increases.

8.3 Experimental Sensitivity Analysis of the Optimal Policy

In this section, the sensitivity of the optimal policy and the expected total discounted cost to the parameter values is evaluated empirically. This experimental sensitivity analysis is conducted in three sections which investigate what influence (1) variation in cost parameters, (2) failure rate and (3) the expert judgement transition probability matrix, have on the optimal policy and the optimal cost.

8.3.1 Sensitivity of the Optimal Policy to Cost Parameters

In this section, the effect of cost values on decision choices at decision Step 1 is explored. The objective of this sensitivity analysis is to find out how much the optimal policy is affected by the operational changes in inspection and maintenance action costs.

8.3.1.1 Simple Inspection Versus Accurate Inspection

First, the sensitivity of preference of the simple inspection over accurate inspection to the inspection costs C^S and C^A is explored. First we explore this through a numerical experiment. We obtain the grid-based optimal policy using the parameter values given in Sub-Section 8.2.2, decreasing the cost of conducting a simple inspection by 50%, i.e. $C^S = 750$. Note that this value is chosen, as an example, to illustrate the changes of the optimal policy caused by reducing the simple inspection cost through the schematic optimal policy. We will then, examine the sensitivity of the optimal cost and the optimal policy by varying the simple inspection cost between 0 and C^A. To compare the obtained optimal policy with that associated with the original parameter value $C^S = 1500$, we present the schematic stationary optimal policy, that is, the optimal policy for the decision period $k \leq 19$. The grid-based optimal policy for these decision periods is given in Figure 8.5 and Figure 8.6, illustrating the optimal inspection, and maintenance action rule after a simple inspection, respectively. The optimal maintenance action rule after an accurate inspection remains the same as the action rule given in Table 8.2. The expected total discounted cost associated with this optimal policy is £30,994 for the two-year planning horizon. This is approximately 5% less than the optimal cost computed based on the original parameter values.

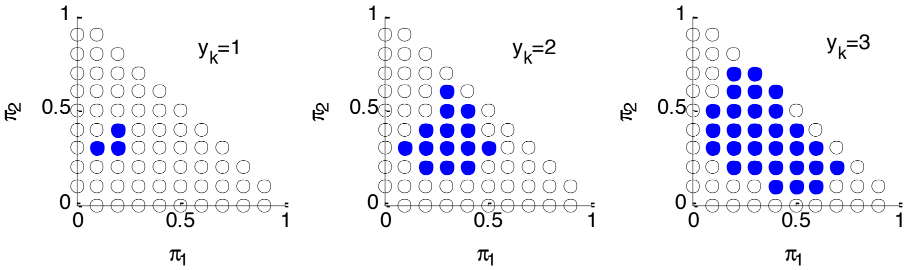

Figure 8.5: The optimal inspection rule for k=1,2,…,19 when $C^S = £750$, with × representing no inspection, ○ representing simple inspection and ● representing accurate inspection as the optimal policy.

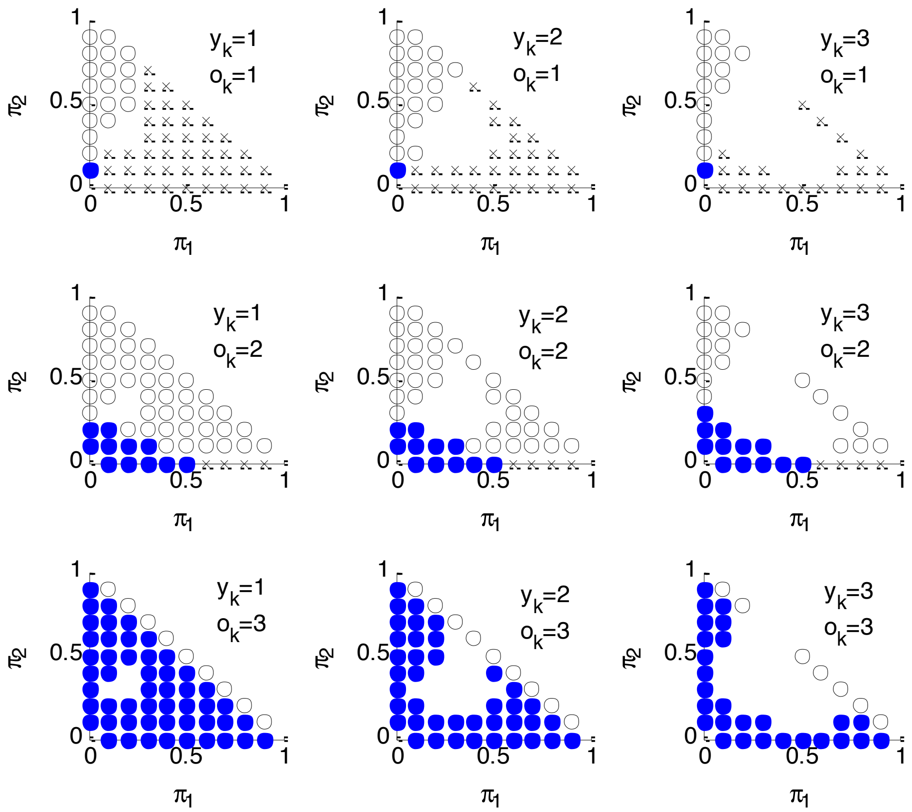

Figure 8.6: The optimal maintenance action rule, after a simple inspection, at k=1,2,...,19 when $C^S = £750$, with × representing no action, ○ representing imperfect maintenance action and ● representing replacement.

As expected, as the cost of conducting a simple inspection reduces, the number of belief states for which the simple inspection is recommended as the optimised inspection type increases. This can be observed by comparing the obtained inspection rule in Figure 8.5 with the one obtained for $C^S = 1500$ in Figure 8.3.

To explore the influence of variations of the value of C^S on the expected total discounted cost, we obtain the optimal policy varying the simple inspection cost within the boundaries stated by Assumption 1. According to Assumption 1, a realistic upper bound for the simple inspection cost would be the cost of accurate inspection, i.e. $0 \le C^S \le C^A$. Figure 8.7 plots the variation of the total expected discounted cost as a function of the ratio of simple inspection to accurate inspection cost, $\frac{C^S}{C^A}$.

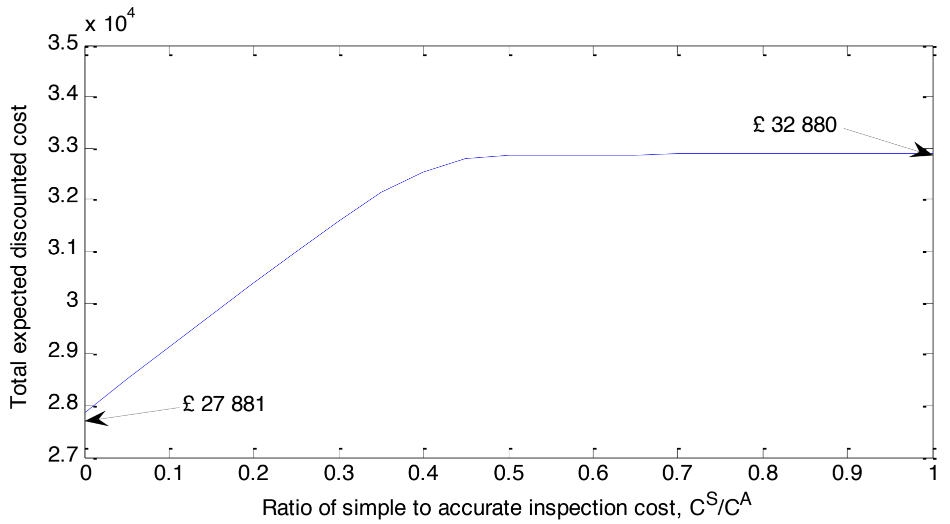

Figure 8.7: Variation of the optimal cost with ratio of simple inspection to accurate inspection cost. $C^A = £3000$ and C^S is varied between 0 and £3000, with increment of £100.

As seen in Figure 8.7, the optimal cost loses its sensitivity to the simple inspection: accurate inspection ratio when the simple inspection cost rises to approximately more than 65% of an accurate inspection cost, that is when $\frac{C^S}{C^A} \geq 0.65$ or $C^S \geq 1950$. This implies that the structure of the optimal policy will also lose its sensitivity to the value of simple inspection cost when $\frac{C^S}{C^A} \geq 0.65$. This means that, given the assigned parameter values, the simple inspection will not be chosen in the stationary optimal policy as the optimal inspection type when $\frac{C^S}{C^A} \geq 0.65$.

8.3.1.2 Imperfect Maintenance Action Versus Replacement

The sensitivity of the optimal policy to imperfect maintenance action cost C^M is explored next. We obtain the grid-based optimal policy using the parameter values given in Sub-Section 8.2.2, reducing the cost of conducting an imperfect maintenance action to $C^M = 3000$. To compare the obtained optimal policy with that associated with the original parameter values, i.e. $C^M = 5000$, we present the schematic stationary optimal policy that is the optimal policy for the decision period $k \leq 20$. The grid-based optimal policy for these decision periods is given in Figure 8.8 and Figure 8.9, illustrating the optimal policies for inspection and maintenance action after a simple inspection, respectively. The optimal maintenance action policy after an accurate inspection is given in Table 8.3. The expected total discounted cost associated with this optimal policy is £31,549 for the two-year planning horizon. This is approximately 4% less than the optimal cost computed based on the original parameter values.

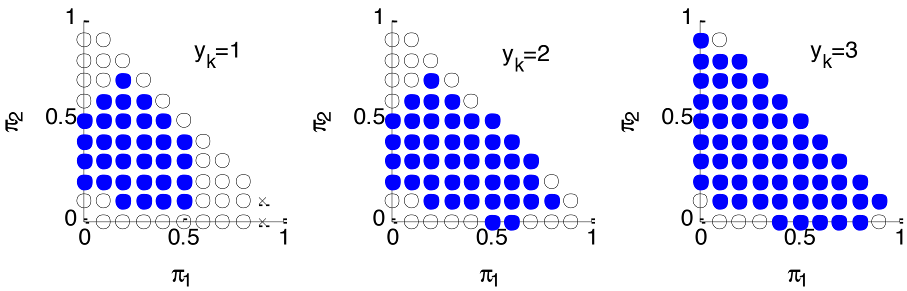

Figure 8.8: The optimal inspection rule for k=1,2,…,20 $C^M = £3000$, with $\times$ representing no inspection, $\circ$ representing simple inspection and $\bullet$ representing accurate inspection as the optimal policy.

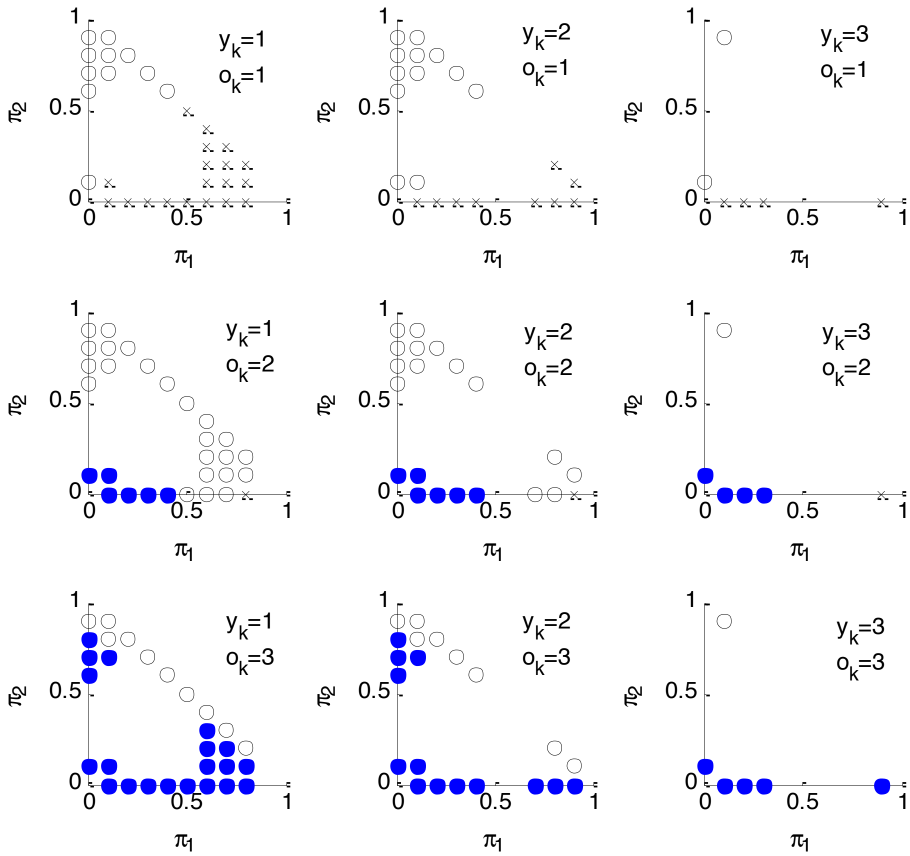

Figure 8.9: The optimal maintenance action rule, after conducting a simple inspection, at k=1,2,...,20 when $C^{M} = £3000$, with $\times$ representing no action, $\bigcirc$ representing imperfect maintenance action and $\bullet$ representing replacement.

Table 8.3: The optimal maintenance action rule, given the outcome of the accurate inspection.

State	*Optimal maintenance action*	
i	$k = 1, 2, ..., 23$	$k = 24$
1	No action	No action
2	Imperfect maintenance action	Imperfect maintenance action
3	Replacement	Imperfect maintenance action

As expected, as the cost of carrying out an imperfect maintenance action reduces, the number of belief states for which imperfect maintenance action is recommended as the optimal action increases. This can be seen by comparing the obtained maintenance action rule in Figure 8.9 with that obtained for $C^M = 5000$ in Figure 8.4. Also, as seen in Figure 8.8, the number of belief states for which the optimal inspection rule is to do nothing reduces as the cost of an imperfect maintenance action decreases. This is because for some belief states, for which the optimal inspection rule given the original parameter values was to do nothing, it becomes cost effective to carry out an imperfect maintenance action when the cost of carrying it out reduces. Since an imperfect maintenance action can be carried out only after conducting an inspection, for such belief states an inspection is recommended so that it can be followed by an imperfect maintenance action.

To explore the influence of variations of the imperfect maintenance action cost on the expected total discounted cost, we obtain the optimal policy by varying the imperfect maintenance action cost from zero to its upper bound, that is the replacement cost, i.e. $0 \le C^M \le C^R$, as given in Assumption 1. Figure 8.10 plots the variation of the total expected discounted cost as a function of the ratio of imperfect maintenance action to replacement cost, i.e. $\frac{C^M}{C^R}$.

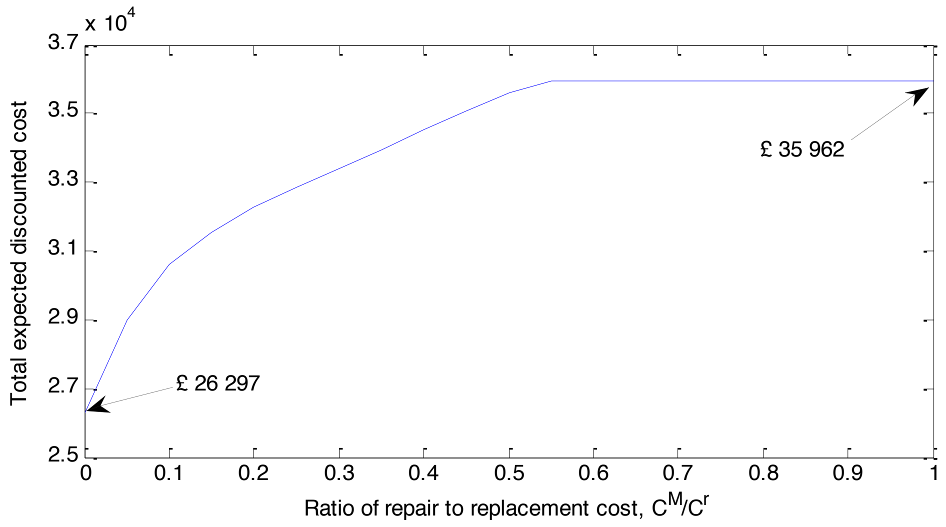

Figure 8.10: Variation of the optimal cost with the ratio of imperfect maintenance action to replacement cost. $C^R = £20000$ and C^M is varied between 0 and £20000 with increments of £100.

As seen in Figure 8.10, the optimal cost loses its sensitivity to the ratio of the imperfect maintenance action to replacement cost when the imperfect maintenance action costs rise to approximately more than 55% of the replacement cost, that is, when $\frac{C^M}{C^R} \geq 0.55$ or $C^M \geq 11000$. This implies that the structure of the optimal policy also loses its sensitivity to the imperfect maintenance action cost when $\frac{C^M}{C^R} \geq 0.55$. This means that, given the parameter values in Sub-Section 8.2.2, the imperfect maintenance action is not recommended by the stationary optimal policy when $\frac{C^M}{C^R} \geq 0.55$.

8.3.2 Sensitivity of the Optimal Policy to Failure Rate

To explore the changes in the optimal policy when the failure rate varies, we increase the failure rates given in Table 8.1 to the failure rates shown in Table 8.4.

Table 8.4: Increased failure rates from the original values.

State	*Original values of probability of failure over a time unit*	*Increased values of probability of failure over a time unit*
i	F_i	F_i
1	0.001	0.001
2	0.009	0.02
3	0.1	0.5

The increased failure rates yields the following reliability vector, as computed in Sub-Section 7.3.4:

$R = [0.9152 \quad 0.0658 \quad 0.0313]$.

As the failure rate increases, the reliability over a decision period is reduced, so we expect to have a more conservative policy. Figure 8.11 and Figure 8.12 illustrate the obtained optimal inspection and maintenance action rule after a simple inspection, respectively. The optimal maintenance action rule after an accurate inspection remains the same as the action rule given in Table 8.2. The expected total discounted cost associated with this optimal policy is £76,945.

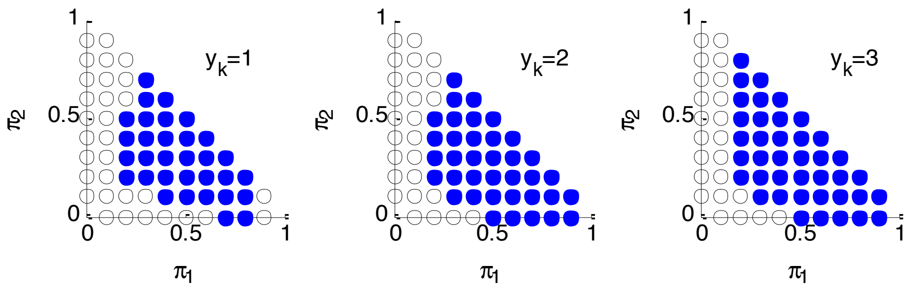

Figure 8.11: The optimal inspection rule for k=1,2,…,20 with increased failure rates, with × representing no action, ○ representing imperfect maintenance action and ● representing replacement.

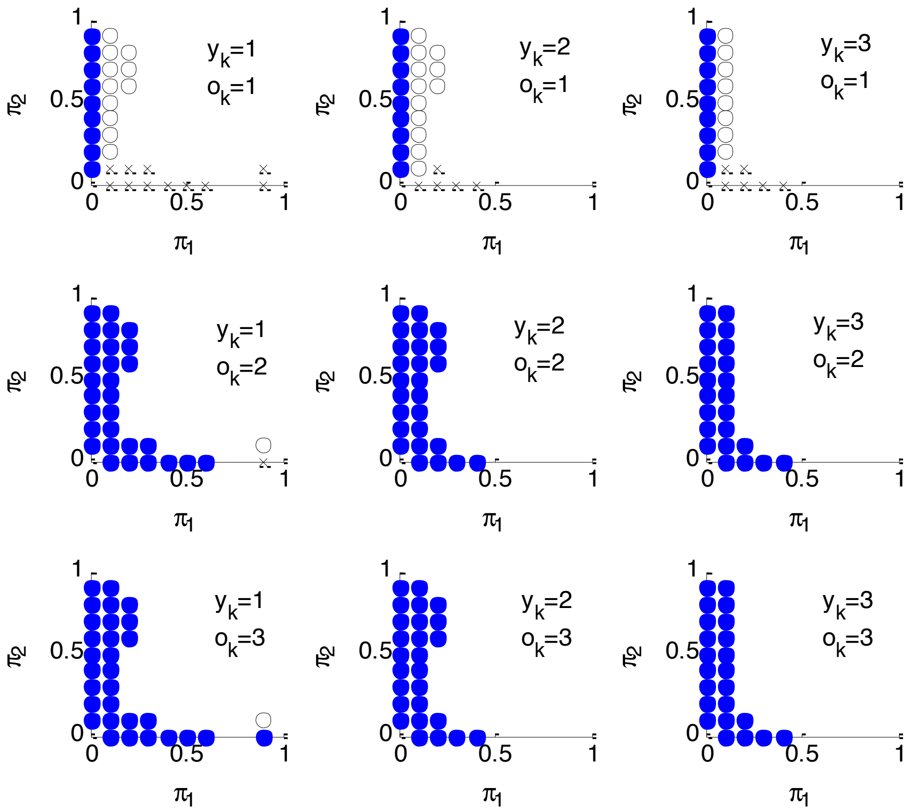

Figure 8.12: The optimal maintenance action rule, after a simple inspection, at k=1,2,...,20 with increased failure rates, with × representing no inspection, ○ representing simple inspection and ● representing accurate inspection as the optimal policy.

As shown in Figure 8.11 and Figure 8.12, increasing failure rates results in a much more costly, more conservative optimal policy. That is, the number of the belief states for which the optimal policy is recommended as doing nothing decreases, while the number of the belief states for which the optimal policy is to carry out an accurate inspection, or replacement at decision Step 2 increases.

8.3.3 Sensitivity of the Optimal Policy to the Accuracy of Expert Judgement

In this section we explore the influence of the accuracy of expert judgement on the optimal policy. Recalling Assumption 6, this accuracy is reflected by the stochastic matrix $Q(i)$, $1 \le i \le N$, where,

$$Q(i) = \left[q_{lm}^{i} = p\left(y_{k+1} = m \middle| y_k = l, x_{k+1} = i\right),\ 1 \le i, l, m \le N \right].$$

To increase the accuracy of expert judgement, we increase the transition probability of expert judgement to the true state of the system. Assuming that the current expert judgement is $y_k = l,\ 1 \le l \le N$, we change the entries of the stochastic matrix $Q(i)$, so that the probability of the expert judgement matching the true system state, i.e. $y_{k+1} = x_{k+1} = i$, increases. Denoting this probability with q_{li}^{i}, we have $q_{li}^{i} = p\left(y_{k+1} = i \middle| y_k = l, x_{k+1} = i\right),\ 1 \le i, l \le N.$

We change the original value of the expert judgement transition probability matrices to the values below, so that the accuracy of the expert judgement increases as explained.

$$Q(1) = \begin{bmatrix} 0.85 & 0.14 & 0.01 \\ 0.4 & 0.5 & 0.1 \\ 0.3 & 0.5 & 0.2 \end{bmatrix},$$

$$Q(2) = \begin{bmatrix} 0.15 & 0.8 & 0.05 \\ 0.05 & 0.7 & 0.25 \\ 0.01 & 0.7 & 0.29 \end{bmatrix},\ Q(3) = \begin{bmatrix} 0.1 & 0.3 & 0.6 \\ 0.01 & 0.14 & 0.85 \\ 0.01 & 0.14 & 0.85 \end{bmatrix}$$

We obtain the optimal policy over a two-year planning horizon. The optimal policy becomes stationary when there are more than three decision periods to go, that is for k=1,2,…,21. Figure 8.13 and Figure 8.14 illustrate the stationary optimal inspection, and maintenance action rule after a simple inspection, respectively. The optimal maintenance action rule after an accurate inspection remains the same as the action rule given in Table 8.2. The expected total discounted cost associated with this optimal policy is £32,777, which is less than the expected total discounted cost associated with the optimal policy obtained in Section 8.2, for the “less accurate” expert judgement. This makes sense as when the expert judgement is provided more accurately (e.g. by a very experienced senior engineer), the optimal policy is expected to cost less. However, using a more accurate expert judgement could cost more (e.g. higher level of payments to a senior engineer) and hence this cost should be taken into account when optimising the decision process in practice.

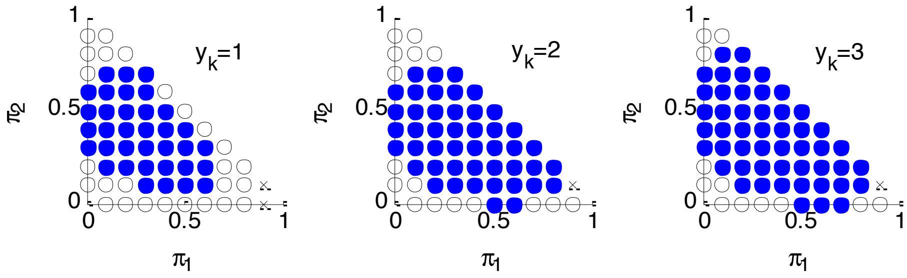

Figure 8.13: The optimal inspection rule for k=1,2,…,21 with the expert judgement transition matrix changed, with × representing no inspection, ○ representing simple inspection and ● representing accurate inspection as the optimal policy.

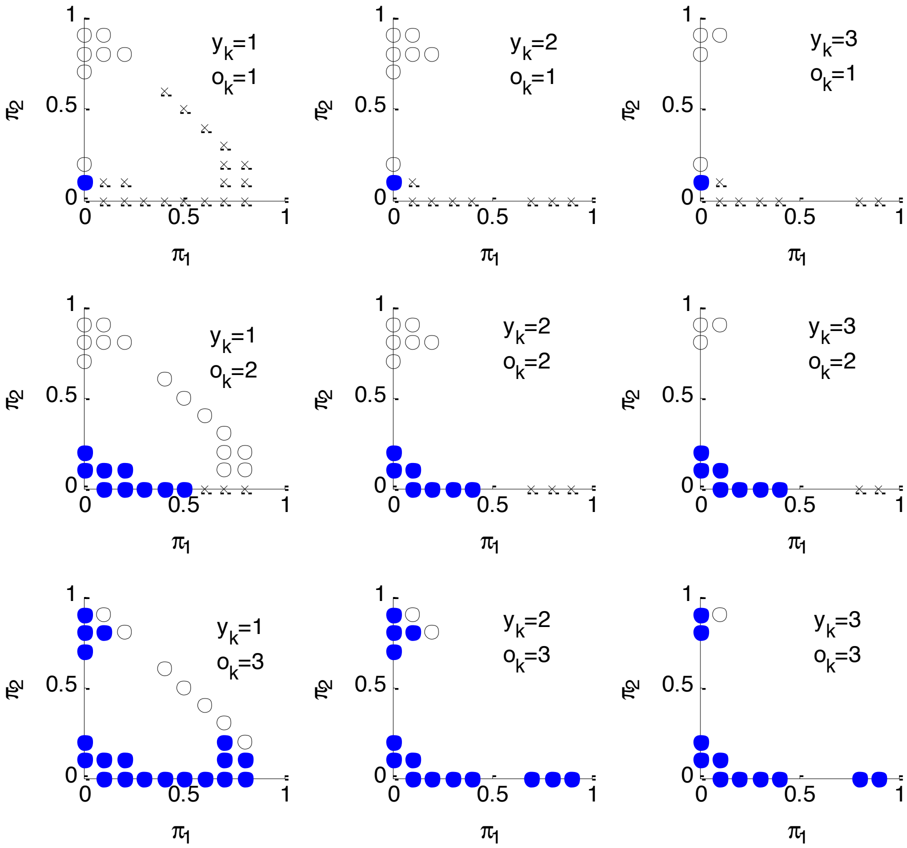

Figure 8.14: The optimal maintenance action rule, after a simple inspection, at k=1,2,...,21 with the expert judgement transition matrix changed, with × representing no action, ○ representing imperfect maintenance action and ● representing replacement.

As expected, when the expert judgement accuracy increases so that it is more likely to equate to the true state of the system, the number of the belief states for which an inspection is recommended by the optimal policy, decreases.

8.4 Summary and Conclusion

In this chapter we developed an approximate method to numerically solve the proposed two-step POMDP. In this method the prior probability distribution of the system state is discretised using regular grids, and thus the computation of the optimal policy is only applied to a specific number of belief states. The proposed two-step grid-based optimal policy was illustrated through numerical experiments, with parameter values assigned according to the assumptions given in Sub-Section 8.2.1 to help the intuitive representation of a real scenario.

Earlier, in Section 7.1, we described the two-step decision process and illustrated the underlying decision mechanism in Figure 7.1. Having formulated the decision optimisation model in Section 7.4 and provided a method for solving it in this chapter, we can now summarise the procedure regarding how to use the optimal policy in practice as depicted in Figure 8.15.

As seen in the schematic grid-based optimal policy depicted in Figure 8.1–Figure 8.9, the discretised belief state space is partitioned into sub-regions corresponding to different inspection choices at decision Step 1 and different maintenance actions at decision Step 2. The size of these regions can be denoted by the number of the grids marked according to inspection and maintenance action choices. At the beginning of each decision period k, first the expert judgement y_k is obtained and then the optimal inspection choice is looked up in the corresponding grid-based optimal inspection policy. The optimal inspection type is found by checking in which region the prior distribution probability of the system π^k falls. In situations when π^k falls on the borderlines, i.e. surrounded by grids marked according to different inspection choices, the inspection type of the grid $\pi^g = \left(\pi_1^g, \pi_2^g, \pi_3^g\right)$ that is the closest in terms of the distance between the coordinates $\left(\pi_1^k, \pi_2^k\right)$ and $\left(\pi_1^g, \pi_2^g\right)$ is selected. The same procedure can be followed when finding the optimal maintenance action in the grid-based optimal maintenance action policy.

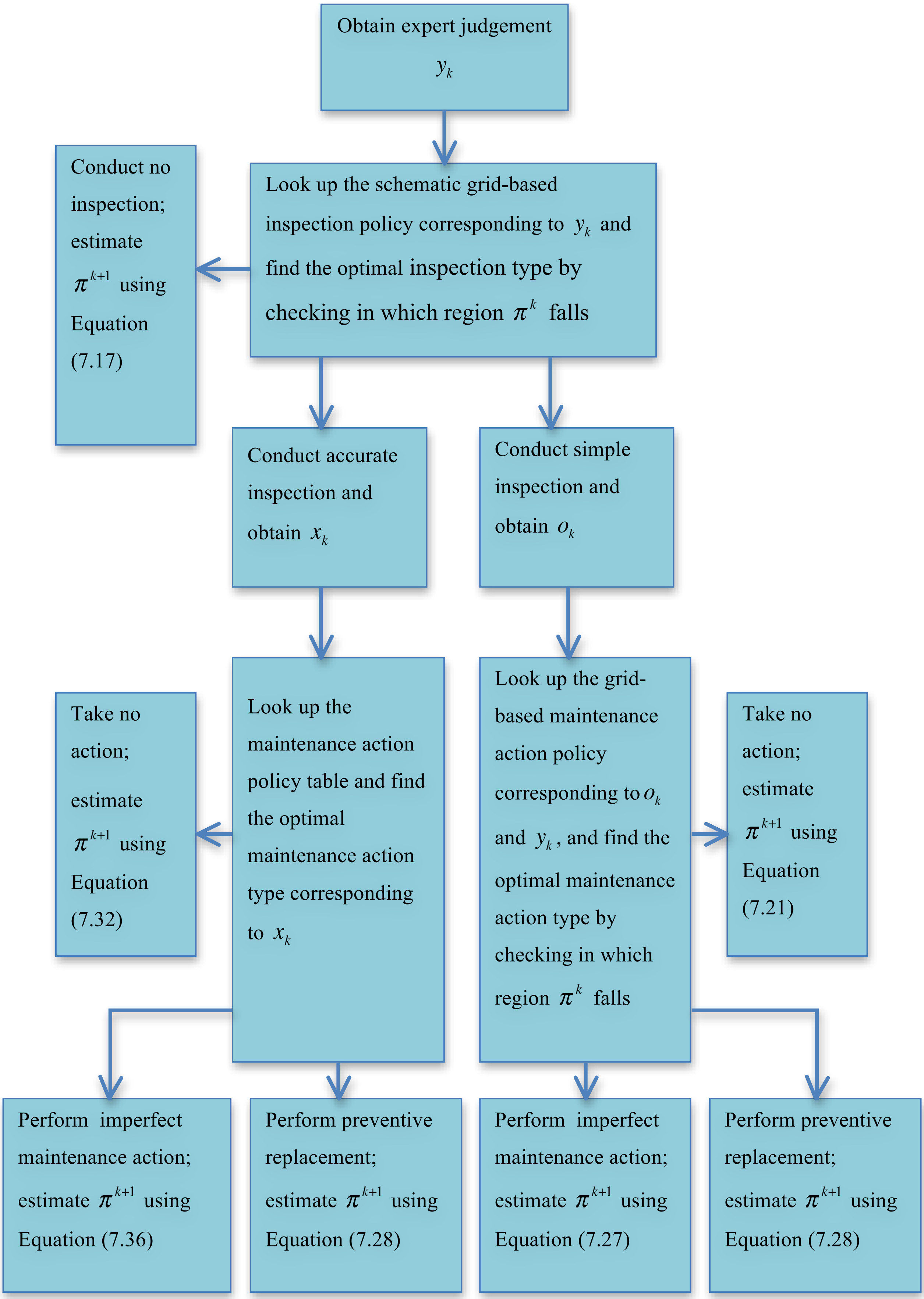

Figure 8.15: Procedure of using the proposed optimal policy at decision period k.

The numerical experiment results provided an insight into the structure of the optimal policy. In the paragraphs below we discuss some observed structural features that once established, can speed the computation and simplify the implementation of optimal policy.

As the deterioration state of the system, x_k, degrades, the optimal maintenance action becomes more conservative. For example, as shown in Table 8.2, when the system is in State 1 the maintenance action recommended by the optimal policy is to take no action. As the deterioration level degrades to State 2, the optimal action is to perform an imperfect repair, and it is a preventive replacement when the system is in the worst condition, State 3. This implies that the optimal policy has a “control-limit” or “threshold-based” structure with respect to the deterioration state of the system. In other words, the optimal maintenance action at decision period k, denoted by a_k^M, can be indicated by the maximum number of two thresholds $x^{(1)}$ and $x^{(2)}$ as follows:

$$a_k^M = \begin{cases} 0 \text{ if } x_k^{(0)} \le x_k \le x_k^{(1)}, \\ 1 \text{ if } x_k^{(1)} \le x_k \le x_k^{(2)}, \\ 2 \text{ if } x_k^{(2)}, \end{cases}$$

where $a^M = 0$ means to not take any maintenance action, $a^M = 1$ means to perform imperfect maintenance action, and $a^M = 2$ means to perform a replacement.

The same structural feature is also observed of the optimal inspection policy with respect to expert judgement y_k, and the optimal maintenance action policy with respect to both the simple inspection outcome o_k and expert judgement y_k. For example, as seen in Figure 8.3, as the deterioration level ascertained by the expert degrades, the inspection policy becomes more conservative. That is, as y_k increases, the sub-regions corresponding to “no inspection” and “simple inspection” become smaller while the sub-region corresponding to “accurate inspection” becomes larger. Also, as seen for example in Figure 8.9, as the expert judgement y_k increases across the simplexes from left to right, and the observation o_k increases across the simplexes from up to down, the optimal maintenance action becomes more conservative.

The suggestive "control-limit" feature can significantly speed the computation and increase the accuracy of the optimal policy in such a way that instead of finding an approximated optimal policy for each grid in a discretised belief state space, the optimal thresholds are calculated instead.

Another important observation is the stationary behaviour of the optimal policy. For example, for the numerical example conducted in Section 8.2 the optimal policy becomes stationary when there are more than five decision periods to go. In other words, for any decision period $k \leq 18$, the optimal policy is independent of k. This observation is suggestive of a stationary optimal policy when the expected total discounted cost is minimised over an infinite planning horizon. This means that in order to solve the POMDP over an infinite horizon we only need to find the stationary policy. Note that the convergence of the total expected discounted cost minimised over an infinite planning horizon is guaranteed by discounting the cost values using a discount rate $0 \leq \varphi < 1$.

The sensitivity analyses conducted in Section 8.3 showed that changing the model parameters affect the size of the regions corresponding to inspection and maintenance action choices in the grid-based optimal policy. For example, as seen in Figure 8.7, when a simple inspection costs more than 65% of the cost of an accurate inspection, $C^S \geq 0.65C^A$, for all decision periods $k \leq 20$ an accurate inspection is always selected over a simple inspection. In other words, under such conditions the sub-region corresponding to simple inspection is merged with the one related to an accurate inspection. Another interesting observation from the numerical experiments is that when the accuracy of the expert judgement compared to the accuracy of a simple inspection is reduced to some point, the expert judgement is no longer a control factor for selecting the optimal maintenance action. For example, in Figure 8.2 the grid-based inspection polices according to different expert judgements and same observation outcomes in each row are the same. This can significantly speed the computation of the optimal cost and the procedure of using it in practice.

The observations mentioned in the above paragraph imply that the relationship between the cost parameters and the probability parameters (e.g. the relationship between the elements of the expert judgement transition matrix in the sense of likelihood ratio as stated in Assumption 6) affect the maximum number of the regions that specify a “control-limit” optimal policy. The suggestive “stationary” and “control-limit” characteristics of the optimal policy observed through numerical experiments can be theoretically validated in future research under some conditions, such as the assumptions listed in Sub-Section 8.2.1.

While our results have used a particular discrete grid, similar results were obtained using a finer grid. This indicates that the results are robust to the choice of grid size. In particular, the sensitivity to changes in parameters is similar for different grid sizes.

9 Conclusions and Further Research

The main contribution of this thesis is to Condition Based Maintenance (CBM) modelling, through formally incorporating expert judgement into the decision support process. The inspiration for model developments was taken from a real CBM system within a large engineering company operating fans, as discussed in Chapter 2. To create manageable models, simplifications and assumptions have been made. Hence, one possible direction for expanding this research is to extend the developed models by removing some of the simplifications and relaxing some of the assumptions. The research scope could also be expanded according to the observations of the case study.

We discuss how each of the four research objectives, as stated in Chapter 1, has been addressed in Sections 9.1 to 9.4. As well as summarizing the contributions to knowledge for each objective, some suggestions for future research to extend the proposed models and algorithms are presented. In Section 9.5, we conclude by suggesting more fundamental directions to future research

9.1 Research Objective 1

Recall from Section 1.4 that the first research objective was to develop a stochastic model that captures the evolutionary relationship between expert judgement and the underlying deterioration condition of a system.

9.1.1 Contribution to Knowledge

A model was developed in Chapter 5 to capture this evolutionary relationship, to be used as a framework for diagnostics and prognostics. This model was formulated as a

Coupled Hidden Markov Model (CHMM) with discrete time and discrete states. The proposed CHMM enhances the capabilities of the standard Hidden Markov Models (HMMs) by introducing a new structure for the interaction among two hidden Markov processes and an observation process, while utilising the well-established methodologies of this modelling framework (e.g. forward-backward procedure). The new CHMM formulation has the advantage of reduced number of parameters compared to the CHMMs in literature (discussed in Section 4.3).

9.1.2 Suggestions for Future Research

In the proposed CHMM time is modelled discretely and hence the duration spent in system's states, i.e. the sojourn time, is characterised by a geometrically decaying function. The probability distribution function of sojourn time is explicitly modelled in Hidden Semi Markov Models (HSMMs) (as described in Sub-Section 3.5.5). The proposed CHMM can be further expanded based on a HSMM to allow the sojourn time follow other distribution functions. Therefore, an appropriate parametric distribution function can fit to real data when they are available in practice and this can improve the prognostic ability of the model.

9.2 Research Objective 2

The second objective of this research was to develop a parameter estimation method for the stochastic model, and to evaluate its performance with respect to potential application issues that might be faced in practice, such as the number of expert judgements.

9.2.1 Contribution to Knowledge

In Chapter 5, a training algorithm was developed to train the CHMM, by following steps analogous to the Baum–Welch algorithm. We defined re-estimation formulas for the CHMM parameters based on the concept of frequencies of event occurrence. This algorithm was demonstrated and evaluated by numerical experiments in Chapter 6. The experimental results empirically confirmed that the proposed training algorithm converges to local maxima; therefore it can be used as an efficient practical method for training the proposed CHMM. There remains an opportunity to prove that theoretically.

The effect of the number of training observation sequences and the initial parameter values on the performance of the training algorithm was investigated through experimental sensitivity analysis. Although the structure of the proposed CHMM has the advantage of a small parameter space and therefore the advantage of computation efficiency, the experimental results showed that possibility of overfitting still exists for parameters with relatively small values (e.g. $F_1 = 0.03$). When a small number of datasets (e.g. three) are used to train the model, the parameter values are adjusted to specific random features of the training data largely deviated from their true values; the result is an "overfitted" model, as opposed to a "generalised" model in which the parameter values are close to their true values.

9.2.2 Suggestions for Future Research

Recall from Sub-Section 5.3.4 that the proposed training algorithm iteratively updates the parameter values until the difference between the probabilities of the training data given the trained model, $P\left(U\middle|\hat{\lambda}\right)$, at consecutive iterations becomes less than a predefined threshold. The overfitting issue mentioned above could be managed by defining a generalisation (validation) error for the model, and using that as a condition to terminate the re-estimation procedure, which aims to fit the model to the training data.

Some methods have been developed to overcome overfitting in standard HMMs by using model entropy as a measure of generalising performance. The entropy of an observation sequence of length T produced by model λ is given by $H(\lambda,T) = -\sum_{\forall U \in \tilde{U}^T} P(U|\lambda)\log(U|\lambda)$, where $\tilde{U}^T$ is the set of all sequences of length T that can be produced by model λ (Walder, Kootsookos, and Lovell, 2003). An example of such methods is proposed by Walder et al. (2003). In their proposed method, an HMM model is trained by maximising a linear combination of model entropy and model likelihood in which a free parameter is used to balance the desired level of generality of the model.

Another method of dealing with overfitting would be to refine the re-estimation equations of our proposed algorithm, given in Sub-Section 5.3.5. These equations are defined based on the concept of frequencies of event occurrence. In the proposed training algorithm all of the sequences in the training data are used to compute the expected event counts, and then they are added together at the adjusting iterations, to re-estimate the model parameters.

Rabiner and Juang (1993) proposed a refinement for training standard HMMs using multi-sequence training. At updating iterations, the event occurrence counts are computed using each sequence. The parameters are then re-estimated using a weighted average of the expected event counts computed for each observation sequence. In this way, the contribution of each sequence to updating the model is proportional to the probability of observing this sequence by the current estimated model.

Another method of dealing with overfitting for multi-sequence training data is "Ensemble training" (Mackay, 1997), in which a separate model is trained for each training sequence and a weighted combination of an ensemble of trained parameters is computed. Later, Davis and Lovell (2004) proposed the Viterbi Path Counting method, in which the training sequences are used individually in turn and the model parameters are re-estimated by counting the states and transitions in the current observation sequence using the Viterbi algorithm (described in Paragraph 3.5.6.2). They compared this method with "Ensemble training" and Rabiner and Juang's multi-sequence training methods, and their experiments demonstrated that the choice of best training method depends on the structure of the HMM and the number and length of the training observation sequences.

Liu et al. (2004) also evaluated the three methods mentioned above, using hand gesture data. Their experiment showed that the Viterbi Path Counting (Davis and Lovell, 2004) performs better and has less dependency on the initial model, for training Left-to-Right models, than does Ensemble training (Mackay, 1997) and the method proposed by Rabiner and Juang (1993). Since the evolution of the deterioration of a system in our proposed CHMM follows the Left-to-Right Markov chain characteristics (as described in Sub-Section 5.2.1) it motivates us to consider applying the Viterbi Path Counting to the specific structure of the CHMM in future research.

9.3 Research Objective 3

The third research objective was to develop a model to support cost-effective decision-making based on a trade-off between the cost and benefit of alternative maintenance actions, using the information provided by the stochastic model.

9.3.1 Contribution to Knowledge

Motivated by the CBM industry application described in Chapter 2, a two-step decision optimisation model was developed in Chapter 7 within the framework of a POMDP. This model makes a contribution to the CBM optimisation literature by modelling both simple and accurate inspection and both imperfect and perfect (replacement) maintenance action types as the decision choices. The selections of inspection types and maintenance actions are made at two consecutive steps at decision periods, conditioned on all the information available, such that the expected total discounted cost is minimised over a finite horizon.

9.3.2 Suggestions for Future Research

In the proposed decision optimisation model we assumed that expert judgements are provided at predetermined regular intervals $t = \Delta, 2\Delta, ..., k\Delta, ...$. This model can be extended by considering the cost associated with expert judgement and modelling the times at which the expert judgements are provided, as decisions to be optimised. The cost-effectiveness of using expert judgements as additional information is particularly important when further assessment of condition monitoring data is contracted out.

9.4 Research Objectives 3 and 4

The final research objective was to examine the sensitivity of the optimal maintenance policies with respect to changes in parameter settings.

9.4.1 Contribution to Knowledge

In Chapter 8 we developed an approximate solution to numerically solve the proposed POMDP and obtain the optimal policy. In this method the prior probability distribution of the system state is discretised and thus the computation of the optimal cost is only applied to a specific number of prior belief states. The proposed two-step grid-based optimal policy was evaluated through numerical experiments, with parameter values assigned according to some assumptions to help the intuitive representation of a real scenario. The results of the numerical experiments led us to some important intuitive conclusions about the structural features of the optimal policy. As discussed in Section 8.4, the experiment results suggest that the optimal policy holds "stationary" and "control-limit" features under specific conditions. These features, when established, can speed the computation of the optimal policy and simplify utilising it in practice.

9.4.2 Suggestions for Future Research

The intuitive conclusions drawn from the numerical experiments can be theoretically validated in future research under some conditions such as those stated by the assumptions in Sub-Section 8.2.1.

9.5 Fundamental Direction for Future Research

Another possible direction for future research could be to expand the scope of this research based on an interesting observation obtained from the CBM system described in Chapter 2. It was mentioned in Chapter 2 that the large engineering company consists of four generating units and there are two fans in each unit. The failure of one fan will cause a substantial loss of revenue. Studying the maintenance event database of the generating units revealed that sometimes an action is carried out on both fans of a unit. In such situations, the transition of a fan to a worse deterioration condition sometimes makes a right truncation to the history of the other fan of the same unit. Therefore, the observation sequences related to the fans of one unit cannot be considered as independent sequences for training the model.

The dependency between two fans of a unit can be addressed by modelling the fans as two dependent systems. The interaction between their hidden deterioration states, in the simplest case, can be described as a pair of fully coupled HMMs (as described in Sub-Section 3.5.3) with the joint state (x_A, x_B) where x_A and x_B denote the deterioration states of fans A and B in a unit, respectively. The deterioration process of each fan at every discrete state is subject to censoring with a probability that is dependent on the other fan's probability of failure.

A modeling framework was proposed in this thesis to support diagnostics, prognostics and CBM decision-making, based on "expert judgement" on the state of the system. A goal was to align with industry needs so that future tools could be developed from this research. To ensure that the models proposed in this thesis are applicable to industrial practice, possible ways of dealing with model implementation challenges need to be explored. In particular, an issue that might be encountered in practice is the availability of expert judgement datasets.

In Chapter 2, the CBM system implemented in a large engineering company was described. This industry application was used as a motivation for formulating plausible models; however the original data from this industry application could not be used to populate the models. As discussed in Section 2.4, the expert judgements are only recorded in connection with maintenance interventions. Therefore, the expert judgement datasets, which could be extracted from the maintenance event database, do not contain expert judgements during normal operation of the fans. An important step that could significantly improve the usability of the models proposed in this thesis, is exploring possible ways of recovering the missing expert judgements.

References

Ahmad, R. and Kamaruddin, S. (2012). An overview of Time Based and condition-based maintenance in industrial application. *Computers & Industrial Engineering*, 63(1): 135–149.

Baker, R. and Wang, W. (1993). Developing and testing the delay time model. *Journal of the Operational Research Society*, 44(3): 361–374.

Banjevic, D. and Jardine, A. K. S. (2006). Calculation of reliability function and remaining useful life for a Markov failure time process. *IMA Journal of Management Mathematics*, 17(2): 115–130.

Banjevic, D., Jardine, A. K. S., Makis, V. and Ennis, M. (2001). A control-limit policy and software for condition-based maintenance optimization. *INFOR*, 39(1): 32–51.

Barata, J., Soares, C. G., Marseguerra, M. and Zio, E. (2002). Simulation modelling of repairable multi-component deteriorating systems for "on condition" maintenance optimisation. *Reliability Engineering & System Safety*, 76: 255–264.

Barlow, R. E. and Proschan, F. (1965). *Mathematical Theory of Reliability*. New York: John Wiley.

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

Baum, L. E. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1): 164–171.

Bendell, A., Wightmann, D. W. and Walker, E. V. (1991). Applying proportional hazards modelling in reliability. *Reliability Engineering System Safety*, 34(1): 35–53.

Brand, M., Olivier, N. and Portland, A. (1997). Coupled hidden Markov models for complex action recognition. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition:* 994–999.

Brand, Matthew. (1999). Structure Learning in Conditional Probability Models via an Entropic Prior and Parameter Extinction. *Neural Computation*, 11(5): 1155–1182.

Brewer, N., Liu, N., Vel, O. and Caelli, T. (2006). Using Coupled Hidden Markov Models to Model Suspect Interactions in Digital Forensic Analysis. *2006 International Workshop on Integrating AI and Data Mining*, 58–64.

Brooks, A., Makareno, A., S., W. and Durrant-Whyte, H. (2006). Parametric POMDPs for planning in continuous state spaces. *Robotics and Autonomous Systems*, 54: 887–897.

Brown, M. K. and Turin, W. (1996). HMM based online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10): 1039–1045.

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.

Byon, E. and Ding, Y. (2010). Season-Dependent Condition-Based Maintenance for a Wind Turbine Using a Partially Observed Markov Decision Process. *IEEE Transactions on Power Systems*, 25(4): 1823–1834.

Cai, G., Chen, X., Lin, B., Chen, B. and He, Z. (2012). Operation Reliability Assessment for Cutting Tools by Applying a Proportional Covariate Model to Condition Monitoring Information. *Sensors (Basel),* 12(10): 12964–12987.

Camci, F. and Chinnam, R. B. (2005). Dynamic Bayesian Networks for Machine Diagnostics: Hierarchical hidden Markov models vs. competitive learning. *Proceedings of International Joint Conference on Neural Networks*: 1752 – 1757.

Campbell, J. D. and Jardine, A. K. S. (2001). *Maintenance Excellence: Optimizing Equipment Life-Cycle Decisions*. New York, USA: Marcel Dekker Inc.

Cappé, O., Moulines, E. and Ryden, T. (2005). *Inference in Hidden Markov Models* (p. 654). Springer.

Carr, M. J. and Wang, W. (2010). Modeling Failure Modes for Residual Life Prediction Using Stochastic Filtering Theory. 59 *IEEE Transactions On Reliability,* 162–169.

Chen, M., Madden, M. G. and Liu, Y. (2010). Refined learning of hidden Markov models with a modified Baum–Welch algorithm and informative components. *2010 IEEE International Conference on Progress in Informatics and Computing*, 165–169.

Chiang, A. C. (1984). *Fundamental Methods of Mathematical Economics.* Third Edition. McGraw Hill Book Company.

Chinnam, R. B. and Baruah, P. (2004). A neuro-fuzzy approach for estimating mean residual life in condition-based maintenance systems. *International Journal of Materials and Product Technology*, 20: 166–179.

Christer, A.H. and Waller, W.M. (1984). Reducing production downtime using delay-time analysis. *Journal of the Operational Research Society* 35: 499-512.

Christer, A. H., Wang, W. and Sharp, J. M. (1997). A state space condition monitoring model for furnace erosion prediction and replacement. *European Journal of Operational Research*, 101(1): 1–14.

Cooke, R. and Bedford, T. (2002). Reliability databases in perspective. *IEEE Transactions on Reliability*, 51(3): 294–310.

Cox, D. R. (1972). Regression models and life-tables (with discussion). *Journal of the Royal Statistical Society, Series B*, 34 (2): 187–220.

Davis, R. I. A. and Lovell, B. C. (2004). Comparing and Evaluating HMM Ensemble Training Algorithms Using Train and Test and Condition Number Criteria . *Pattern Analysis and Applications*, 6: 327–335.

Dekker, R. (1996). Applications of maintenance optimization models: a review and analysis. *Reliability Engineering & System Safety*, 51(3): 229–240.

Dekker, R., Wildeman, R. E. and Duyn Schouten, F. A. (1997). A review of multi-component maintenance models with economic dependence. *Mathematical Methods of Operations Research*, 45(3): 411–435.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39: 1–38.

Derman, C. (1962). On sequential decisions and Markov chains. *Management Science*, 9: 16–24.

Dieulle, L., Berenguer, C., Grall, A. and Roussignol, M. (2003). Sequential condition-based maintenance scheduling for a deteriorating system. *European Journal of Operational Research*, 150: 451–461.

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.

Dong, M., He, D., Banerjee, P. and Keller, J. (2006). Equipment health diagnosis and prognosis using hidden semi-Markov models. *The International Journal of Advanced Manufacturing Technology*, 30: 738–749.

Dong, M., and Yang, Z. (2008). Dynamic Bayesian network based prognosis in machining processes. *Journal of Shanghai Jiaotong University (Science)*, *13*(3): 318–322.

EN 13306:2001. Maintenance terminology. European Committee for Standardization, 2001

Forney, G. D. (2005). The Viterbi Algorithm: A Personal History. *Paper presented at the March 2005 Viterbi Conference, University of Southern California.*

Gai, J., Li, Y. and Stevenson, R. L. (2007). Coupled Hidden Markov Model for Robust EO / IR Target Tracking. *Proceedings of IEEE International Conference on Image Processing, 2007:* 41–44.

Ghasemi, A., Yacout, S. and Ouali, M. S. (2007). Optimal condition based maintenance with imperfect information and the proportional hazards model. *International Journal of Production Research*, 45(4): 989–1012.

Gorjian, N., Ma, L., Mittinty, M., Yarlagadda, P., Sun Y., (2009). A review on reliability models with covariates. *Proceedings of the 4th World Congress on Engineering Asset Management, Athens, Greece, September*.

Grosfeld-Nir, A. (1996). A Two-State Partially Observable Markov Decision Process with Uniformly Distributed Observations. *Operations Research*, 44(3): 458–463.

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

Horenbeek, A., Pintelon, L. and Muchiri, P. (2010). Maintenance optimization models and criteria. *International Journal of System Assurance Engineering and Management*, 1(3): 189–200.

Huynh, K. T., Barros, A., Bérenguer, C. and Castro, I. T. (2011). A periodic inspection and replacement policy for systems subject to competing failure modes due to degradation and traumatic events. *Reliability Engineering & System Safety*, 96(4): 497–508.

Ivy, J. S. and Pollock, S. M. (2005). Marginally monotonic maintenance policies for a Multi-State Deteriorating Machine With Probabilistic Monitoring , and Silent Failures. *IEEE Transactions on Reliability*, 54(3): 489–497.

Jardine, A. K. S., Anderson, P. M. and Mann, D. S. (1987). Application of the Weibull proportional hazards model to aircraft and marine engine failure data. *Quality and Reliability Engineering International*, 3(2): 77–82.

Jardine, A. K. S., Banjevic, D. and Makis, V. (1997). Optimal replacement policy and the structure of software for Condition-Based Maintenance. *Journal of Quality in Maintenance Engineering*, 3(2): 109–119.

Jardine, A. K. S., Joseph, T. and Banjevic, D. (1999). Optimizing condition-based maintenance decisions for equipment subject to vibration monitoring. *Quality*, 5(3): 192–202.

Jardine, A. K. S., 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.

Jardine, A. K. S., Makis, V., Banjevic, D., Braticevic, D. and Ennis, M. (1998). A decision optimization model for condition-based maintenance. *Journal of Quality in Maintenance Engineering*, 4(2): 115–121.

Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*. Springer New York.

Kacpryznski, G. J., Sarlashkar, A., Roemer, M. J., Hess, A. and Hardman, B. (2004). Predicting remaining life by fusing the physics of failure modelling with diagnostics. *Journal of the Minerals, Metals and Materials Society*, 56: 29–35.

Knezevic, J. (1987). Condition parameter based approach to calculation of reliability characteristics. *Reliability Engineering*, 19 (1): 29-39.

Kumar, A. (2010). Model based approach and algorithm for fault diagnosis and prognosis of coated gas turbine blades. *Proceedings of IEEE International Conference on Advanced Intelligent Mechatronics*, 800–805.

Kumar, A., Tseng, F. and Chinnam, R. B. (2011). Role of hidden-Markov models for autonomous diagnostics of cutting tools. *Proceedings of International Symposium on Innovations in Intelligent Systems and Applications* : 509–513.

Kumar, D. and Westberg, U. (1997). Maintenance scheduling under age replacement policy using proportional hazards model and TTT-plotting. *European Journal of Operational Research*, 99(3): 507–515.

Kwon, J. and Murphy, K. (2000). Modeling Freeway Traffic with Coupled HMMs. *University of Califronia at Berkeley.*

Levinson, S. E., Rabiner, L.R. and M.M. Sondhi (1983). An Introduction to the Application of the Theory of Probabilistic Functions of Markov Process to Automatic Speech Recognition. *Bell System Technical Journal*, 62(4): 1035-1074.

Lin, D., Wiseman, M., Banjevic, D. and Jardine, A. K. S. (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.

Liu, N., Davis, R. I. A., Lovell, B. C. and Kootsookos, P. J. (2004). Effect of Initial HMM Choices in Multiple Sequence Training for Gesture Recognition. *Proceedings of the International Conference on Information Technology: Coding and Computing*, 1: 608–613.

Lovejoy, W. S. (1991). A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research*, 28(1): 47–65.

Mackay, D. J. C. (1997). Ensemble learning for hidden Markov models. *Technical report, Cavendish Laboratory, University of Cambridge*.

Maillart, L. M. (2006). Maintenance policies for systems with condition monitoring and obvious failures. *IIE Transactions*, 38(6): 463–475.

Makis, V. and Jardine, A. K. S. (1992). Optimal Replacement in the Proportional Hazard Model. *INFOR*, 30(1): 172–183.

Marcel, S., Bernier, O., Viallet, J.-E. and Collobert, D. (1997). Hand gesture recognition using input-output hidden Markov models. *Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition* : 456–461.

Mazzuchi, T. (2008). A paired comparison experiment for gathering expert judgement for an aircraft wiring risk assessment. *Reliability Engineering System Safety*, 93(5): 722–731.

McCall, J. J. (1965). Maintenance policies for stochastically failing equipments: a survey. *Management Science*, 11.

McMillan, D. and Ault, G. W. (2008). Condition monitoring benefit for onshore wind turbines: Sensitivity to operational parameters. *IET Renewable Power Generation*, 2(1): 60–72.

McNaught, K. R. and Zagorecki, A. T. (2009). Using dynamic Bayesian networks for prognostic modelling to inform maintenance decision making. *Proceedings of IEEE International Conference on Industrial Engineering and Engineering Management*: 1155–1159.

Mcnaught, K. R., and Zagorecki, A. T. (2011). Complex Engineering Service Systems. In *Complex Engineering Service Systems:Concepts and Research* (pp. 277–296). London: Springer London.

Mohanty, S., Chattopadhyay, A., Peralta, P., Das, S. and Willhauck, C., (2008). Fatigue life prediction using multivariate Gaussian process. *Collection of Tech Papers AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Schaumburg, IL (USA), 7–10 April 2008.

Monahan, G. E. (1982). State of the Art – A Survey of Partially Observable Markov Decision Processes: Theory, Models, and Algorithms. *Management Science*, 28(1): 1–16.

Murphy, K. (2002). PhD Thesis, *Dynamic Bayesian networks: representation, inference and learning*. University of California, Berkeley.

Nakagawa, T. (2006). Statistical Models on Maintenance. *Springer Handbook of Engineering Statistics* (pp. 835–848). Springer, London.

Nicolai, R. P. and Dekker, R. (2008). Optimal maintenance of multi-component systems: A review. In: K. A. H. Kobbacy and D. N. P. Murthy (Eds.), *Complex System Maintenance Handbook* (pp. 263–286). Springer, London.

Nicolai, R. P., Dekker, R. and van Noortwijk, J. M. (2007). A comparison of models for measurable deterioration: An application to coatings on steel structures. *Reliability Engineering & System Safety*, 92(12): 1635–1650.

Ocak, H., Loparo, K. A. and Discenzo, F. M. (2007). Online tracking of bearing wear using wavelet packet decomposition and probabilistic modeling: A method for bearing prognostics. *Journal of Sound and Vibration*, 302(4-5): 951–961.

Ohnishi, M., Kawai, H. and Mine, H. (1986). An optimal inspection and replacement policy under incomplete state information. *European Journal of Operational Research*, 27(1): 117–128.

Ohnishi, M., Morioka, T. and Ibaraki, T. (1994). Optimal minimal-repair and replacement problem of discrete-time Markovian deterioration system under incomplete state information. *Computers & Industrial Engineering*, 27(1-4): 409–412.

Oppenheimer, C. and Loparo, K. A. (2002). Physically based diagnosis and prognosis for cracked rotor shafts. *Proceedings of SPIE Component and Systems Diagnostics, Prognostics and Health Management*, 122–132.

Pearl, J. (1997). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (2nd edition.). Morgan Kaufman.

Pecht, M. and Gu, J. (2009). Physics-of-failure-based prognostics for electronic products. *Transactions of the Institute of Measurement and Control,* 31(3,4): 309-322.

Peng, Y., Dong, M., and Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: a review. *The International Journal of Advanced Manufacturing Technology*, 50(1-4): 297–313.

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.

Percy D.F., Kobbacy K.A.H. and Ascher H.E. (1998). Using proportional-intensities models to schedule preventive- maintenance intervals. *IMA J Management Math*, 9(3): 289-302.

Percy, D.F., Alkali, B.M. (2007). Scheduling preventive maintenance for oil pumps using generalized proportional intensities models. *International Transactions in Operational Research* 14 (6): 547–563.

Pham, H. and Wang, H. (1996). Imperfect maintenance. *European Journal of Operational Research*, 94: 425–438.

Press, W. H., Teuolsky, S. A., Vetterling, W. T. and Flannery, B. P. (2007). 3.6 Interpolation on a Grid in Multidimensions. *Numerical Recipes, The Art of Scientific Computing* (pp. 133–134).

Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming (Wiley Series in Probability and Statistics)*. Wiley-Interscience.

Qiu, J., Seth, B. B., Liang, S. Y. and Zhang, C. (2002). Damage mechanics approach for bearing lifetime prognostics. *Mechanical Systems and Signal Processing*, 16(5): 817–829.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2): 257–286.

Rabiner, L. R. and Juang, B. H. (1993). *Fundamentals of Speech Recognition*. New Jersey, Prentice Hall.

Rammohan, R. and Taha, M. R. (2005). Exploratory Investigations for Intelligent Damage Prognosis using Hidden Markov Models. *IEEE International Conference on Systems, Man and Cybernetics*, 2: 1524–1529.

Rosenfield, D. (1976). Markovian deterioration with uncertain information. *Operations Research*, 24(1): 141–155.

Ross, S. M. (1971). Quality Control under Markovian Deterioration. *Management Science*, 17: 587–596.

Russel, M. and Moore, R. (1985). Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition. *Proceedings of Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'85:* 5–8.

Saranga, H., Knezevic, J. (2000). Reliability prediction for condition based maintained systems. *Reliability Engineering & System Safety*, 71: 219-224.

Scarf, P. A. (1997). On the application of mathematical models in maintenance. *European Journal of Operational Research*, 99(3): 493–506.

Sharma, A., Yadava, G. S. and Deshmukh, S. G. (2011). A literature review and future perspectives on maintenance optimization. *Journal of Quality in Maintenance Engineering*, 17(1): 5–25.

Sherwin, D. (2000). A review of overall models for maintenance management. *Journal of Quality in Maintenance Engineering*, 6(3): 138–164.

Sheskin, T. J. (2010). *Markov Chains and Decision Processes for Engineers and Managers*. CRC Press.

Shiryaev, A. (1968). An optimum methods in quickest detection problems. *Theory of Applied Probability*, 8: 22–46.

Si, X. S., Wang, W., Hu, C. H., 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.

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.

Tamura, N., Hayashi, K., Yuge, T. and Yanagi, S. (2010). Monotone Properties of Optimal Maintenance Policy for Two-State Partially Observable Markov Decision Process Model with Multiple Observations. *International Journal of Operations Research*, 7(3): 23–34.

Thurston, M. G. (2001). An Open Standard for Web-Based Condition Based Maintenance Systems. *IEEE System Readiness Technology Conference*. Valley Forge, PA., USA.

Tian, Z. and Liao, H. (2009). Optimization of maintenance policy using the proportional hazard model. *Reliability Engineering System Safety*, 94(5): 44–52.

Tobon-Mejia, D. A., Medjaher, K. and Zerhouni, N. (2012). CNC machine tool's wear diagnostic and prognostic by using dynamic Bayesian networks. *Mechanical Systems and Signal Processing*, 28: 167–182.

Tse, P., Atherton, D. (1999). Prediction of machine deterioration using vibration based fault trends and recurrent neural networks. *Transactions of the ASME: Journal of Vibration and Acoustics.* 121: 355–362.

Valdez, F. C. and Feldman, R. M. (1989). A survey of preventive maintenance models for stochastically deteriorating single unit systems. *Naval Research Logistics*, 36(4): 419–446.

Vlok, P. J., Coetzee, J. L., Banjevic, D., Jardine, A. K. S. and Makis, V. (2002). Optimal component replacement decisions using vibration monitoring and the proportional-hazards model. *Journal of the Operational Research Society*, 53(2): 193–202.

Wagner, H. M. (1970). *Principles of Operations Research with Applications to Managerial Decisions* (p. 937). Prentice Hall.

Walder, C. J., Kootsookos, P. J. and Lovell, B. C. (2003). Towards a Maximum Entropy Method for Estimating HMM Parameters. *Proceedings of the 2003 APRS Workshop on Digital Image Computing*, 45–49.

Wang, H. (2002). A survey of maintenance policies of deteriorating systems. *European Journal of Operational Research*, 139(3): 469–489.

Wang, W. (2000). A model to determine the optimal critical level and the monitoring intervals in condition-based maintenance. *International Journal of Production Research*, 38: 1425–1436.

Wang, W. (2002). A model to predict the residual life of rolling element bearings given monitored condition information to date. *IMA Journal of Management Mathematics*, 13(1): 3–16.

Wang, W. (2003). Modelling condition monitoring intervals: A hybrid of simulation and analytical approaches. *Journal of the Operational Research Society*, 54(3): 273–282.

Wang, W. (2007). A two-stage prognosis model in condition based maintenance. *European Journal of Operational Research*, 182(3): 1177–1187.

Wang, W. (2008). Condition-Based Maintenance Modelling. In: K. A. H. Kobbacy and D. N. P. Murthy (Eds.), *Complex System Maintenance Handbook* (pp. 111–131). Springer, London.

Wang, W., Banjevic, D. and Pecht, M. (2010). A multi-component and multi-failure mode inspection model based on the delay time concept. *Reliability Engineering System Safety*, 95(8): 912–920.

Wang, W. and Christer, A. H. (2000). Towards a general condition-based maintenance model for a stochastic dynamic system. *Journal of the Operational Research Society*, 51: 145–155.

Wang, W. and Zhang, W. (2008). An asset residual life prediction model based on expert judgements. *European Journal of Operational Research*, 188(2): 496–505.

Welch, L. (2003). Hidden Markov Models and the Baum–Welch Algorithm. *IEEE Information Theory Society Newsletter*, 53(4).

White, C. (1979). A Markov Quality Control Process Subject to Partial Observation. *Management Science*, 23: 843–852.

Wong, E. L., Jardine, A. K. S. and Banjevic, D (2011). Incorporating repair information into maintenance optimization models for repairable systems.

Proceedings of the 2011 IEEE Annual Reliability and Maintainability Symposium, 1–5.

Xiao, W. B., Chen, J., Dong, G. M., Zhou, Y. and Wang, Z. Y. (2011). A multichannel fusion approach based on coupled hidden Markov models for rolling element bearing fault diagnosis. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 226(1): 202–216.

Zhao, X., Fouladirad, M. and Berenguer, C. (2010). Residual-based inspection/ replacement policy for a deteriorating system with Markovian covariates. *2010 IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 636–640). IEEE.

Zhong, S. (2001). A new formulation of coupled hidden Markov models. *Technical report, Department of Electrical and Computer Engineering, University of Texas at Austin*.

Zhou, Y., Ma, L., Mathew, J., Kim, H. and Wolff, R. (2009). Asset life prediction using multiple degradation indicators and lifetime data : a Gamma-based state space model approach, (July), 20–24.

Zhou, Y., Ma, L., Mathew, J., Sun, Y. and Wolff, R. (2011). Maintenance strategy optimization using a continuous-state partially observable semi-Markov decision process. *Microelectronics Reliability*, 51(2): 300–309.

Zuashkiani, A., Banjevic, D. and Jardine, A. K. S. (2008). Estimating parameters of proportional hazards model based on expert knowledge and statistical data. *Journal of the Operational Research Society*, 60(12): 1621–1636.

Appendix A

MATLAB Computer Codes for Experimental Evaluation of the Coupled Hidden Markov Model Training Algorithm

A.1 Generating the Training Data

```
% Input: Number of deterioration states of a system
N=3;

% Input: Original parameter values for generating observation
%sequences
p=[0.1 0.7];
q=[0.7 0.4 0.3;0.7 0.4 0.3];
f=[0.1 0.3 0.5];

% Compute joint transition probability matrix
q=[q;1 1 1];
p=[p 1];

a=zeros(N,N,N,N);

for i=1:N
    for k=1:N
```

```
        for j=1:N
            for l=1:N
                if j==i&&l==k;a(i,k,j,l)=p(i)*q(k,j);end
                if j==i+1&&l==k;a(i,k,j,l)=(1-p(i))*q(k,j);end
                if j==i&&l==k+1;a(i,k,j,l)=p(i)*(1-q(k,j));end
                if j==i+1&&l==k+1;a(i,k,j,l)=(1-p(i))*(1-
q(k,j));end
            end
        end
    end
end

% Compute joint observation probability matrix
b= zeros(N,N,2*N);
b(1,1,1)=1-f(1);
b(1,1,2)=f(1);
b(1,2,3)=1-f(1);
b(1,2,4)=f(1);
b(1,3,5)=1-f(1);
b(1,3,6)=f(1);
b(2,1,1)=1-f(2);
b(2,1,2)=f(2);
b(2,2,3)=1-f(2);
b(2,2,4)=f(2);
b(2,3,5)=1-f(2);
b(2,3,6)=f(2);
b(3,1,1)=1-f(3);
b(3,1,2)=f(3);
b(3,2,3)=1-f(3);
b(3,2,4)=f(3);
b(3,3,5)=1-f(3);
b(3,3,6)=f(3);

%Input: Number of observation sequences
numex=10;

 %Generate training data with maximum length of 50
```

```
for i=1:numex
    seqs{i}=hmmgenerate(50,a,b);
end
```

A.2 Training Algorithm

```
%Re-estimation is repeated for maximum number of iteration or
%until threshold is reached

%maximum number of iteration
max_iter=100;
%Convergence threshold
thresh=1e-4;

% Input: Initial guess of parameter values
p=[0.1 0.7];
q=[0.7 0.4 0.3;0.7 0.4 0.3];
f=[0.1 0.3 0.5];

previous_loglik=-inf;
converged=0;
num_iter=1;

a=zeros(N,N,N,N);
b=zeros(N,N,2*N);

%Repeat re-estimation until convergence or maximum number of
iteration
while(num_iter<=max_iter)&& ~converged

%Compute joint transition matrix

a(3,3,3,3)=1;
a(1,1,1,1)=p(1)*q(1,1);
a(1,1,1,2)=p(1)*(1-q(1,1));
```

```
a(1,1,2,1)=(1-p(1))*q(1,2);
a(1,1,2,2)=(1-p(1))*(1-q(1,2));
a(1,2,1,2)=p(1)*q(2,1);
a(1,2,1,3)=p(1)*(1-q(2,1));
a(1,2,2,2)=(1-p(1))*q(2,2);
a(1,2,2,3)=(1-p(1))*(1-q(2,2));
a(1,3,1,3)=p(1);
a(1,3,2,3)=1-p(1);
a(2,1,2,1)=p(2)*q(1,2);
a(2,1,2,2)=p(2)*(1-q(1,2));
a(2,1,3,1)=(1-p(2))*q(1,3);
a(2,1,3,2)=(1-p(2))*(1-q(1,3));
a(2,2,2,2)=p(2)*q(2,2);
a(2,2,2,3)=p(2)*(1-q(2,2));
a(2,2,3,2)=(1-p(2))*q(2,3);
a(2,2,3,3)=(1-p(2))*(1-q(2,3));
a(2,3,2,3)=p(2);
a(2,3,3,3)=1-p(2);
a(3,1,3,1)=q(1,3);
a(3,1,3,2)=1-q(1,3);
a(3,2,3,2)=q(2,3);
a(3,2,3,3)=1-q(2,3);

% Compute joint observation matrix
b(1,1,1)=1-f(1);
b(1,1,2)=f(1);
b(1,2,3)=1-f(1);
b(1,2,4)=f(1);
b(1,3,5)=1-f(1);
b(1,3,6)=f(1);
b(2,1,1)=1-f(2);
b(2,1,2)=f(2);
b(2,2,3)=1-f(2);
b(2,2,4)=f(2);
b(2,3,5)=1-f(2);
b(2,3,6)=f(2);
b(3,1,1)=1-f(3);
b(3,1,2)=f(3);
```

```
b(3,2,3)=1-f(3);
b(3,2,4)=f(3);
b(3,3,5)=1-f(3);
b(3,3,6)=f(3);

 loglik=0;

%Define Denominator and numerator of parameter re-estimation
%formulas
num_p=zeros(1,N-1);
den_p=zeros(1,N-1);
num_f=zeros(1,N);
den_f=zeros(1,N);
num_q=zeros(N-1,N);
den_q=zeros(N-1,N);

%Repeat calculation for all sequences
for ex=1:numex

%Obtain length of observation sequences
obs1=seqs{ex};
T=length(obs1)+1;
obs=zeros(1,T);
obs(1)=1;

%Add 1 at the beginning of each sequence to denote joint State
%(1,1)
for t=1:T-1
obs(t+1)=obs1(t);
end

%Finish each sequence with failure, if there is any in the
%sequence
fail_time=T;
for t=1:T
if obs(t)==2 |obs(t)==4|obs(t)==6
```

```
fail_time=t;
break;
end
end
T=fail_time;

%Execute forward-backward procedure to compute posteriori
%probabilities

alfa=zeros(N,N,T);
beta=zeros(N,N,T);
gama=zeros(N,N,T-1);
alfbeta=zeros(N,N,N,N,T-1);
xi=zeros(N,N,N,N,T-1);
xii_tlk=zeros(1,N-1);
xikj_ti=zeros(N-1,N);
xi_til=zeros(N-1,N);
gama_tk=zeros(1,N-1);
gama_tkf=zeros(1,N);
num_i_visited=zeros(1,N);
xiik_sum=zeros(N,N);

%Make the entries of a (multidimensional) array sum to 1
scale=ones(1,T);
t=1;
alfa(1,1,t)=b(1,1,obs(t));
[alfa(:,:,t),scale(t)]=normalise(alfa(:,:,t));

%Forward variable
for t=2:T
    mm=zeros(N,N);
    for j=1:N
        for l=1:N
            for i=1:N
                for k=1:N
                    m=alfa(i,k,t-1)*a(i,k,j,l);
```

```
                    mm(j,l)=mm(j,l)+m;
                end
            end
            alfa(j,l,t)=mm(j,l)*b(j,l,obs(t));
        end
    end
    [alfa(:,:,t),scale(t)]=normalise(alfa(:,:,t));
end


if any(scale==0)
    ll=-inf;
else
   ll=sum(log(scale));
end

%Backward Variable
beta(:,:,T)=ones(N,N,1);
for t=T-1:-1:1
     for j=1:N
        for l=1:N
            bb=(beta(j,l,t+1)*b(j,l,obs(t+1))*a(:,:,j,l));
            beta(:,:,t)=beta(:,:,t)+bb(:,:);
        end
     end
    beta(:,:,t)=normalise(beta(:,:,t));
end

for t=1:T-1
    for i=1:N
        for k=1:N
            for j=1:N
                for l=1:N

alfbeta(i,k,j,l,t)=alfa(i,k,t)*a(i,k,j,l)*b(j,l,obs(t+1))*beta
(j,l,t+1);
```

```
                end
            end
        end
    end
  xi(:,:,:,:,t)=normalise(alfbeta(:,:,:,:,t));
end

for t=1:T-1
    for i=1:N
        for k=1:N
            xiikt=xi(i,k,:,:,t);
            gama(i,k,t)=sum(xiikt(:));
        end
    end
end



%Compute expected value of event counts to re-estimate
%parameters

% numerator and denominator of p
for i=1:N-1
    xii=xi(i,:,i,:,:);
    xii_tlk(i)=sum(xii(:));
    gamai=gama(i,:,:);
    gama_tk(i)=sum(gamai(:));
end

% numerator and denominator of q
for k=1:N-1
    for j=1:N
        xikj=xi(:,k,j,k,:);
        xikj_ti(k,j)=sum(xikj(:));
        xikj=xi(:,k,j,:,:);
        xi_til(k,j)=sum(xikj(:));
    end
```

```
end

% numerator and denominator of F
k=0;
if obs(T)==2;k=1;
elseif obs(T)==4;k=2;
elseif obs(T)==6;k=3;
end

fail_atstate=zeros(1,N);
if ~k==0
    for i=1:N
    xi_ikT=xi(:,:,i,k,T-1);
    fail_atstate(i)=sum(xi_ikT(:));
    end
end

 for i=1:N
    xiii=xi(:,:,i,:);
    num_i_visited(i)=sum(xiii(:));
end

loglik=loglik+ll;

%Re-estimate the parameters
num_p=num_p+xii_tlk;
den_p=den_p+gama_tk;
num_f=num_f+fail_atstate;
den_f=den_f+num_i_visited;
num_q=num_q+xikj_ti;
den_q=den_q+xi_til;
end

%Display log-likelihood and number of iteration
fprintf(1, 'iteration %d, loglik = %f\n', num_iter, loglik)
loglik_value(num_iter)=loglik;
```

```
num_iter_value(num_iter)=num_iter;
num_iter=num_iter+1;
converged=em_converged(loglik,previous_loglik,thresh);
previous_loglik=loglik;
p=num_p./den_p;
q=num_q./den_q;
f=num_f./den_f;
end

%Return number of iterations and estimated parameter values
num_iter=num_iter-1;
p
q
f

%Plot log-likelihood as a function of algorithm iteration
disp(num_iter)
plot(num_iter_value,loglik_value)
```

A.3 Functions Called in the Training Algorithm

A.3.1 normalise

```
%This function makes the entries of a (multidimensional) array
%sum to 1

function [M, z] = normalise(A, dim)
% [M, c] = normalise(A)
% c is the normalising constant
%
% [M, c] = normalise(A, dim)
% If dim is specified, normalise the specified dimension only,
otherwise normalise the whole array.

if nargin < 2
```

```
  z = sum(A(:));
  % Set any zeros to one before dividing
  s = z + (z==0);
  M = A / s;
elseif dim==1 % normalise each column
  z = sum(A);
  s = z + (z==0);
  %M = A ./ (d'*ones(1,size(A,1)))';
  M = A ./ repmatC(s, size(A,1), 1);
else
    z=sum(A,dim);
  s = z + (z==0);
  L=size(A,dim);
  d=length(size(A));
  v=ones(d,1);
  v(dim)=L;
  %c=repmat(s,v);
  c=repmat(s,v');
  M=A./c;
end
```

A.3.2 em_converged

```
%This function checks the convergence of log-likelihood
function [converged, decrease] = em_converged(loglik,
previous_loglik, thresh)
% EM_CONVERGED Has EM converged?
% [converged, decrease] = em_converged(loglik,
previous_loglik, %threshold)
%
% converged if
%   |f(t) - f(t-1)| / avg < threshold,
% where avg = (|f(t)| + |f(t-1)|)/2 and f is log lik.
% threshold defaults to 1e-4.
```

```
if nargin < 3
  thresh = 1e-4;
end

converged = 0;
decrease = 0;
 if loglik - previous_loglik < -1e-3 % allow for a little
imprecision
  fprintf(1, '******likelihood decreased from %6.4f to
%6.4f!\n', previous_loglik, loglik);
  decrease = 1;
end

% The following stopping criterion is in page 423 of (Press et
al., %2007)
delta_loglik = abs(loglik - previous_loglik);
avg_loglik = (abs(loglik) + abs(previous_loglik) + eps)/2;
if (delta_loglik / avg_loglik) < thresh, converged = 1; end
```

Appendix B

MATLAB Computer Codes for Solving the Partially Observable Markov Decision Process

B.1 Discretising the Belief States

```
% Discretise the belief state space of a system with N=3,
using %regular grids with steps of 0.1
N=3;
[pi1,pi2]=meshgrid(0:0.1:1);

%Change arrays to vector
pi1=pi1(:);pi2=pi2(:);
pi3=1-(pi1+pi2);
pi1=pi1(pi3>=0);
pi2=pi2(pi3>=0);
pi3=pi3(pi3>=0);

%Make the pi array which contains the discretised belief state
%space

pii=[pi1,pi2,pi3];
```

```
plot(pi1,pi2,'ks');
```

B.2 Reliability Function over one Decision Period

```
% Transition probability , p(i)=probability of remaining in
%State i
N=3;

%del=duration of a decision period=4 weeks
del=4;

%weekly transition probability
p_i=[0.99;0.98;1];

%Monthly transition probability
p=zeros(N,N);

for i=1:N
    for j=1:N
        if i==j;p(i,j)=p_i(i);end
        if j==i+1;p(i,j)=1-p_i(i);end
    end
end
```

```
p=p^del

% R(k,i,Delta), probability of survival of the system when it
is % in state i at period k. Since the model is stationary and
we %know delta then we use R(i)instead of f(i,t) that is the
%probability of failure in t unit time when the system is in
State i,

%F(i) probability of failure at state i

F=[0.001 0.02 0.5];

%f(i,t) for t=1:del
f=zeros(N,del);

%Calculate f(N,t) for t=1:del
for t=1:del
    f(N,t)=F(N)*((1-F(N))^(t));
end

%Calculate f(i,t) for t=1
for i=1:N-1
    f(i,1)=(1-F(i))*p(i)*F(i)+(1-F(i))*(1-p(i))*F(i+1);
end

%Calculate f(i,t) for t=2:del
for i=N-1:-1:1
    for t=2:del
```

```
        a=0;
        for s=1:t-1
            a=a+(1-F(i))^(s)*p(i)^(s-1)*(1-p(i))*f(i+1,t-s);
        end
        f(i,t)=((1-F(i))*p(i))^t*F(i)+a+((1-F(i))^(t)*p(i)^(t-
1)*(1-p(i))*F(i+1));
    end
end

%Calculate R(i) for a known delta
R=zeros(N,1);

for i=1:N
   sum_f=0;
for t=1:del
    sum_f=sum_f+f(i,t);
end
R(i)=1-sum_f-F(i);
end
R
```

B.3 Updating the Belief States

```
%Expert judgement and the actual state of the system at k+1
%conditioned on decisions aI and aM at k
```

```
N=3;
Z=3;
q(1,:,:)=[0.85 0.14 0.01;0.4 0.5 0.1;0.3 0.4 0.3];
q(2,:,:)=[0.15 0.8 0.05;0.05 0.7 0.25;0.01 0.19 0.8];
q(3,:,:)=[0.1 0.3 0.6;0.01 0.14 0.85;0.01 0.14 0.85];
r=[1 0 0;0.8 0.2 0;0.1 0.7 0.2];
b=[0.85 0.14 0.01;0.1 0.8 0.1;0.01 0.19 0.8];
G=size(pii,1);

%y_m(N)=expert judgement after the repair=y'
[rmax,y_m]=max(r,[],2);

y_1=zeros(G,N,N);
x_1=zeros(G,N,N,N);
y_2_1=zeros(G,Z,N,N);
x_2_1=zeros(G,Z,N,N,N);
y_2_2=zeros(G,Z,N,N);
x_2_2=zeros(G,Z,N,N,N);
y_3_1=zeros(N,N);
x_3_1=zeros(N,N,N);
y_3_2=zeros(N,N);
x_3_2=zeros(N,N,N);
piio=zeros(G,Z,N);


%y_3_1(xk,yk+1) (i,m) Wait after an accurate inspection aI=3,
%aM=1
```

```
for m=1:N
    for i=1:N
        sum=0;
        for j=1:N
            sum=sum+p(i,j)*q(j,i,m);
        end
        y_3_1(i,m)=sum;
    end
end

%x_3_1(xk,yk+1,xk+1) (i,m,j)

for j=1:N
    for m=1:N

            for i=1:N
                der=y_3_1(i,m)+(y_3_1(i,m)==0);
                x_3_1(i,m,j)=(p(i,j)*q(j,i,m))/der;
            end

    end
end
%normalise x_3_1 and y_3_1 on the last dimension so that the
sum %of the conditional probabilities would be 1
x_3_1=normalise(x_3_1,3);y_3_1=normalise(y_3_1,2);
```

```
%y_3_2(xk,yk+1) (i,m) expert judgement at k+1 when aI=3 and
aM=2
 rp=r*p;
for i=1:N
    l=y_m(i);
    for m=1:N
        sum=0;
        for n=1:N
         sum=sum+rp(i,n)*q(n,l,m);
        end
        y_3_2(i,m)=sum;
     end
end

%x_3_2(xk,yk+1,xk+1) (i,m,n) x at k+1 when aI=3 and aM=2
for i=1:N
    l=y_m(i);
    for m=1:N
        for n=1:N
            der=y_3_2(i,m)+(y_3_2(i,m)==0);
            x_3_2(i,m,n)=(rp(i,n)*q(n,l,m))/der;
        end
    end
end

%Normalise x_3_2 and y_3_2
y_3_2=normalise(y_3_2,2);x_3_2=normalise(x_3_2,3);
```

```
%y_1 (xk,yk,yk+1) (g,l,m) y at k+1 when aI=1, since we do not
conduct an accurate inspection we have a belief state and
hence %G probabilities according to each grid

%piip(g,j)=pii(g,i)*p(i,j)
piip=pii*p;

for m=1:N
    y_1(:,:,m)=piip*q(:,:,m);
end

% x_1(xk,yk,yk+1,xk+1) (g,l,m,j) x at k+1 when aI=1
 for g=1:G
    for l=1:N
        for m=1:N
            for j=1:N
                der=y_1(g,l,m)+(y_1(g,l,m)==0);
                x_1(g,l,m,j)=(piip(g,j)*q(j,l,m))/der;
            end
        end
    end
end

%normalise x_1 and y_1
y_1=normalise(y_1,3);x_1=normalise(x_1,4);
```

```
%y_2_1(ok,yk,yk+1) (g,o,l,m) expert judgement at k+1 when aI=2
%and aM=1


%piib(g,o)=pii(g,i)*b(i,o)
piib=pii*b;


%piio(g,o,i)=p(xk|yk,ok) since there are same belief state for
%each j, j is not considered-pii(g,i) is the same for each j


for g=1:G
    for o=1:Z
        for i=1:N
            der=piib(g,o)+(piib(g,o)==0);
            piio(g,o,i)=(pii(g,i)*b(i,o))/der;
        end
    end
end


for g=1:G
    for o=1:Z
        for l=1:N
            for m=1:N
                sum=0;
                for j=1:N
                    for i=1:N
                        sum=sum+piio(g,o,i)*p(i,j)*q(j,l,m);
                    end
                end
```

```
                y_2_1(g,o,l,m)=sum;
            end
        end
    end
end

%x_2_1(ok,yk,yk+1,xk+1) (g,o,l,m,j) x at k+1 when aI=2 and
aM=1
 for g=1:G
    for o=1:Z
        for l=1:N
            for m=1:N
               for j=1:N
                   sum=0;
                    for i=1:N
                        sum=sum+piio(g,o,i)*p(i,j)*q(j,l,m);
                    end
                    der=y_2_1(g,o,l,m)+(y_2_1(g,o,l,m)==0);
                    x_2_1(g,o,l,m,j)=sum/der;
                end
             end
        end
    end
end

y_2_1=normalise(y_2_1,4);x_2_1=normalise(x_2_1,5);piio=normali
se(piio,3);
```

```
%y_2_2(g,ok,yk,yk+1) (g,o,l,m) y at k+1 when aI=2 and aM=2
  for g=1:G
    for o=1:Z
        for l=1:N
            ll=y_m(l);
            for m=1:N
                sum=0;
                for n=1:N
                    for j=1:N
                        for i=1:N

sum=sum+piio(g,o,i)*r(i,j)*p(j,n)*q(n,ll,m);
                        end
                    end
                end
                y_2_2(g,o,l,m)=sum;
            end
        end
    end
end

 %x_2_2(g,ok,yk,yk+1) (g,o,l,m,n) x at k+1 when aI=2 and aM=2
 for g=1:G
    for o=1:Z
        for l=1:N
            ll=y_m(l);
            for m=1:N
```

```
                for n=1:N
                    sum=0;
                    for j=1:N
                        for i=1:N

sum=sum+piio(g,o,i)*r(i,j)*p(j,n)*q(n,ll,m);
                        end
                    end
                    der=y_2_2(g,o,l,m)+(y_2_2(g,o,l,m)==0);
                    x_2_2(g,o,l,m,n)=sum/der;
                end
            end
        end
    end
end



y_2_2=normalise(y_2_2,4);x_2_2=normalise(x_2_2,5);
```

B.4 Backward Induction

```
%POMDP with regular discrete grids, finite horizon, using
%backward induction, ai=choices of inspection-1=no inspection-
%2=simple inspection-3=accurate inspection, am=choices of
%manitenance=1=no action-2=imperfect-3=replacement


c_s=1500 ;c_a=3000 ;
c_m=5000 ;c_r=20000;
```

```
c_f=50000;


%dr=discount Rate
dr=0.95;
%K=number of periods
K=24;


Rpii=pii*R;
%R(pii(AM)) Nx1 conditional reliability when aI=3, aM=2
rR=r*R;


%v=value function v(g,j,k) is the total expected cost at point
g %at period k when the
%current expert judgement is j, yk=j
v=zeros(G,N,K+1);
v_A=zeros(N,K+1);
policy=zeros(G,N,K+1);
policy_A=zeros(N,K+1);
v_S=zeros(G,Z,N,K+1);
policy_S=zeros(G,Z,N,K+1);
Q=zeros(G,N,3);
Q_A=zeros(N,3);
Q_S=zeros(G,Z,N,3);


cpu_time=cputime;


%Calculate the optimal total expected cost and the optimal
%policy at period k, for each point and expert judgement
```

```
for k=0:K-1
    f_cost=c_f+dr*v_A(1,K-k+1);
    r_cost=c_r+dr*v_A(1,K-k+1);
    v_interp1=TriScatteredInterp(pii(:,1),pii(:,2),v(:,1,K-
k+1));
    v_interp2=TriScatteredInterp(pii(:,1),pii(:,2),v(:,2,K-
k+1));
    v_interp3=TriScatteredInterp(pii(:,1),pii(:,2),v(:,3,K-
k+1));

    % Decision Step 2-Accurate Inspection V_A
           for i=1:N

%Q_A(i,1) W_A wait after accurate inspection

ex_cost=y_3_1(i,1)*v_interp1(x_3_1(i,1,1),x_3_1(i,1,2))+y_3_1(
i,2)*v_interp2(x_3_1(i,2,1),x_3_1(i,2,2))+y_3_1(i,3)*v_interp3
(x_3_1(i,3,1),x_3_1(i,3,2));
Q_A(i,1)=f_cost*(1-R(i))+dr*R(i)*ex_cost;

%Q_A(i,2) M_A imperfect repair after accurate %inspection

ex_cost=y_3_2(i,1)*v_interp1(x_3_2(i,1,1),x_3_2(i,1,2))+y_3_2(
i,2)*v_interp2(x_3_2(i,2,1),x_3_2(i,2,2))+y_3_2(i,3)*v_interp3
(x_3_2(i,3,1),x_3_2(i,3,2));
Q_A(i,2)=c_m+(f_cost*(1-rR(i)))+dr*rR(i)*ex_cost;

%Q_A(i,3) replacement after accurate inspection
```

```
 Q_A(i,3)=r_cost;
 [w,x]=min(Q_A(i,:),[],2);
 v_A(i,K-k)=w;
  policy_A(i,K-k)=x;
           end
    for l=1:N

        for g=1:G

           %Decision Step1 -Simple Inspection

               %V_S
               %sum v-S on ok
               sum_v_S(g,l)=0;
               for o=1:Z
%Q_S(g,o,l,1) W_S wait after simple inspection Ro(g,o)
R(pii(yk,ok)) Conditional reliability
                   sum=0;
                   for i=1:N
                       sum=sum+(piio(g,o,i)*R(i));
                   end
                   Ro(g,o)=sum;

ex_cost=y_2_1(g,o,l,1)*v_interp1(x_2_1(g,o,l,1,1),x_2_1(g,o,l,
1,2))+y_2_1(g,o,l,2)*v_interp2(x_2_1(g,o,l,2,1),x_2_1(g,o,l,2,
2))+y_2_1(g,o,l,3)*v_interp3(x_2_1(g,o,l,3,1),x_2_1(g,o,l,3,2)
);
Q_S(g,o,l,1)=f_cost*(1-Ro(g,o))+dr*Ro(g,o)*ex_cost;
```

```
%Q_S(g,o,l,2) M_S imperfect repair after simple inspection
%RSM(g,o) R(pii(SM)) Conditional reliability
                    sum=0;
                    for j=1:N
                        for i=1:N
                            sum=sum+(piio(g,o,i)*r(i,j)*R(j));
                        end
                    end
                    RSM(g,o)=sum;

ex_cost=y_2_2(g,o,l,1)*v_interp1(x_2_2(g,o,l,1,1),x_2_2(g,o,l,
1,2))+y_2_2(g,o,l,2)*v_interp2(x_2_2(g,o,l,2,1),x_2_2(g,o,l,2,
2))+y_2_2(g,o,l,3)*v_interp3(x_2_2(g,o,l,3,1),x_2_2(g,o,l,3,2)
);
Q_S(g,o,l,2)=c_m+f_cost*(1-RSM(g,o))+dr*RSM(g,o)*ex_cost;

%Q_S(g,o,l,3) replacement after simple inspection
                    Q_S(g,o,l,3)=r_cost;

                    [w,x]=min(Q_S(g,o,l,:),[],4);
                    v_S(g,o,l,K-k)=w;
                    policy_S(g,o,l,K-k)=x;

                    %sum_V_S(g,l) on ok

sum_v_S(g,l)=sum_v_S(g,l)+piib(g,o)*v_S(g,o,l,K-k);
```

```
               end

%Decision step 1

%Q(g,l,1) No inspection is conducted

ex_cost=y_1(g,l,1)*v_interp1(x_1(g,l,1,1),x_1(g,l,1,2))+y_1(g,
l,2)*v_interp2(x_1(g,l,2,1),x_1(g,l,2,2))+y_1(g,l,3)*v_interp3
(x_1(g,l,3,1),x_1(g,l,3,2));
               Q(g,l,1)=f_cost*(1-Rpii(g))+dr*Rpii(g)*ex_cost;

  %Q(g,l,2) Simple inspection is conducted
   Q(g,l,2)=c_s+sum_v_S(g,l);

  %Q(g,l,3) Accurate inspection is conducted
               sum_v_A=0;
               for i=1:N
                   sum_v_A=sum_v_A+pii(g,i)*v_A(i,K-k);
               end
              Q(g,l,3)=c_a+sum_v_A;



              [w,x]=min(Q(g,l,:),[],3);
              v(g,l,K-k)=w;
              policy(g,l,K-k)=x;
        end

    end
```

```
end

cpu_time=cputime-cpu_time

% The total expected cost over K decision periods, at time
k=0, %x0 &y0=1 and no actions is taken
v_interp1=TriScatteredInterp(pii(:,1),pii(:,2),v(:,1,1));
v_interp2=TriScatteredInterp(pii(:,1),pii(:,2),v(:,2,1));
v_interp3=TriScatteredInterp(pii(:,1),pii(:,2),v(:,3,1));

f_cost=c_f+dr*v_A(1,1);

ex_cost=y_3_1(1,1)*v_interp1(x_3_1(1,1,1),x_3_1(1,1,2))+y_3_1(
1,2)*v_interp2(x_3_1(1,2,1),x_3_1(1,2,2))+y_3_1(1,3)*v_interp3
(x_3_1(1,3,1),x_3_1(1,3,2));
Total_expected_cost=(f_cost*(1-R(1)))+(dr*R(1)*ex_cost)
```

B.5 Illustrating the grid-based optimal policy

```
%Input: decision period
k=20;

%belief states for which there is no uncertainty

%Figure(1)
%First decision step
%Plot the policy(g,l,k)at pi1 and pi2
```

```
figure(1)
for l=1:N
    subplot(1,N,l)

         for g=[2:10 12:65]

                if policy(g,l,k)==1
                   scatter(pi1(g),pi2(g),'kx');hold on;

                else if policy(g,l,k)==2
                       scatter(pi1(g),pi2(g),'ko');hold on;

                    else

scatter(pi1(g),pi2(g),'bo','filled');hold on;

                    end
                end
         end
        axis([0 1 0 1]);
        xlabel('\pi_{1}');
        ylabel('\pi_{2}');
        hold off;
end

  figure(2)
```

```
  for o=1:Z
      for l=1:N
          pp=l+3*(o-1);
          subplot(Z,N,pp);
              for g=[2:10 12:65]
                  if policy(g,l,k)==2
                  if policy_S(g,o,l,k)==1
                     scatter(pi1(g),pi2(g),'kx');hold on;
                  else if policy_S(g,o,l,k)==2
                          scatter(pi1(g),pi2(g),'ko');hold on;
                    else

scatter(pi1(g),pi2(g),'bo','filled');hold on;
                    end
                  end
                  end
              end
              hold off;
              axis([0 1 0 1]);
              xlabel('\pi_{1}');
              ylabel('\pi_{2}');
              hold off;
       end

  end
```