Modelling safety critical systems with ageing components, with application to underground railway risk and hazards

Susannah Naybour

Thesis submitted to the University of Nottingham for the Degree of Doctor of Philosophy

Department of Civil Engineering

November 2020

Acknowledgements

Firstly, I would like to thank my principal supervisor Professor John Andrews for giving me the opportunity to attempt this PhD, for providing consistent valuable feedback and for pushing me further than I thought possible. I would also like to thank Dr Manuel Chiachio-Ruano for his input, especially during the middle stages of this project. He encouraged me to become a more innovative thinker. In addition, I would like to thank Lloyds Register Foundation for sponsoring me throughout this PhD, I feel truly fortunate.

In addition, I would like to thank the University of Nottingham for the opportunities that have been available to me during my time as a student here, both academic and otherwise. I would especially like to thank the Sport Department, who provided me with ample chance for personal development outside of academic study.

Finally, I would like to thank all those who have supported me, including my family and friends and partner, Moe. It would have been much more stressful without your constant encouragement. Thank you to Victoria White, without whom I would have greatly struggled to get this work into a coherent document. Special thanks should also go to my fellow PhD candidate, Anni Bailey, who has been correcting my spelling since I was 12 years old, and I hope will continue to do so in future!

Papers

S. Naybour, J. Andrews and M. Chiachio-Ruano, “Efficient Risk Based Optimization of Large System Models using a Reduced Petri Net,” in *Proceedings of the 29th European Safety and Reliability Conference (ESREL)*, Hannover, Germany, 2019.

S. Naybour, M. Chiachio-Ruano and , J. Andrews, “A novel method for the reduction of Petri net complexity,” 2020- Journal Article, to be published.

Abstract

In this thesis methodologies for modelling risk on ageing systems are developed. In the first stages of the thesis, two systems on an underground railway are used to demonstrate the modelling approach. In the latter stages of this thesis the modelling approach is expanded further, presenting a method for optimisation of a phased maintenance strategy, an inclusion of uncertainty in model outputs and an approach to model size reduction.

Initially, a Petri net modelling approach is proposed to predict the derailment caused by component failures on a Switch and Crossing (S&C). A holistic methodology is adopted such that components of the system are divided into subsets of interconnected modules at a system level. Degradation within each module is idealized through a sequence of discrete states of wear until final failure occurs. Monte Carlo analysis is used to numerically evaluate the resulting Petri net. Through this methodology, different maintenance strategies, such as partial replacement, complete replacement, and opportunistic maintenance, are tested, to evaluate their influence on the final risk of derailment and predicted system state over time. This work includes a more in-depth modelling approach for S&C than that available in literature. This improves on the state of the art by removing assumptions of perfect maintenance and inspection. In addition, the approach includes modelling of dependencies between components, that are introduced through shared maintenance actions.

Secondly, a Petri net modelling approach is applied to an automatic fire protection system to assess the probability of system failure, throughout the system life. Components are modelled with individual Petri nets, which are connected by a phased asset management strategy. The model is solved numerically via Monte Carlo simulation and component failure probabilities are combined using logic developed through Fault Tree analysis. For each time period, this application gives the probability of detection, deluge and alarm system failure, along with the number of maintenance actions, system tests and false system activations. The key contributions from this work include a detailed model for the interlocking fire protection systems and the application of a phased asset management strategy. This phased strategy allows the modelling of different maintenance approaches that are applied at different times depending on the system age. This approach demonstrates an increased functionality in comparison to modelling approaches currently available for fire protection systems,

In addition, the modelling approach is extended further towards an optimal risk-based asset management decision making tool. The model for the fire protection systems is used as an application and is extended to give a measure of risk and whole-life cost. This extended model forms the basis of a two-stage optimisation approach within the framework of a phased asset management strategy. A Simulated Annealing algorithm is combined with a Genetic Algorithm to reduce system level risk and whole-life cost. A method for the incorporation of uncertainty in predicted model outputs is also presented. Novel aspects within this work include: the development of the optimisation approach for a phased asset management strategy and the developed algorithm for quantifying model output uncertainty given uncertain input parameters. The optimization of a phased system shows improvements on current model optimisation examples as it allows different strategies to be applied at different phases of the system lifecycle. It allows these phases to be determined in an automatic manner. The inclusion of uncertainty estimates on model outputs improves current Petri net modelling approaches, where uncertainty in input parameters is not included, as it allows decisions based on modelling outcomes to be more fully informed.

Finally, a method is presented that can be applied to large system level Petri net models to produce equivalent model at a reduced computational cost. The method consists of generating a reduced Petri net which approximates the behaviour of its larger counterpart with a shorter simulation time. Parameters in this reduced structure are updated following a combined Approximate Bayesian Computation and Subset Simulation framework. Novel contributions from this work include: the proposed reduction approach, a method for using this reduction approach to improve model optimisation efficiency and the exploration of the reduction approach to justify model structure selection. These improve on approaches for model reduction available in literature, which are commonly rule based and so less flexible. In addition, model choice is typically user defined without quantifiable evidence for the suitability of the selected model structure.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1: Background

Underground railways are currently present in 178 cities across the world and carry approximately 168 million passengers per day[1].

The Metropolitan line, which opened in London in 1863, was world's first underground railway. This was a sub-surface line created using a 'cut and cover' method, whereby a trench is dug just below the ground and then covered to leave an underground space through which a train can pass [2]. A steam engine was used to pull the carriages on the Metropolitan line, producing vast quantities of steam and smoke.

Electric trains were introduced to the London Underground network in 1890. Following this, there was a reduced need for ventilation, and a second type of underground line was employed, using a 'deep level tube' method of construction. Deep level tube tunnels, created by boring into the earth, reduce the need for surface digging [3].

Currently London Underground has 270 stations and 11 lines, 45% of the lines are underground in either cut and cover lines or deep level tube tunnels, with the remaining 55% above ground. The trains on the London Underground network are powered by an electrified third rail.

Table 1.1 gives a summary of the features of each of the London Underground lines including the usage, length of line and the number of delays in the 2016/17 Transport for London (TfL) reporting period. Data for this is taken from the TfL Performance data almanac [4]. In this recording period 1.38 billion passenger journeys were made on the Underground [5]. The table displays track failures and delays and cancelations of scheduled trains across all lines.

Worldwide there are several further examples of historic underground railways:

- The New York Subway was opened in 1904 and in 2018 had 665 miles of track, 472 miles of which are underground [6][7]. In 2018 there were 1.68 billion passenger journeys along the 21 routes that make up the network, the trains are driven manually and powered by a third rail.
- The Paris Metro was first opened in 1900. The Paris Metro has 14 lines, 127miles of track, and 303 stations and had 1.56 billion passenger journeys in 2018 [8][9]. Since 1999, a major renovation programme has been implemented to update and improve the Paris Metro.
- The Moscow Metro was opened in 1935; the network consists of 15 lines and is almost entirely underground. The network contains over 204 miles of track and has 269 stations [10]. There are deep tunnels that run under the river Moskva. In 2017 there were 2.37 billion passenger journeys on the Moscow Metro [1].
- The Tokyo Metro opened in 1927 and has 9 lines, both over ground and underground. In 2017 there were 3.46 billion passenger journeys on the Tokyo Metro [11][1].

Line	**Construction type**	**Usage (km operated)**	**Number of Stations**	**Length (km)**	**Number of trains delayed longer than 15 minutes**	**Number of track failures**	**Percentage of the schedule operated**
Bakerloo	Deep tube	3,703,207	25	23.6	91	50	96.7%
Central	Deep tube	13,057,296	50	73.3	180	131	96.2%
Circle	Sub-surface	3,907,744	36	26.5	59	47	93.2%
Hammersmith & city	Sub-surface		28	25.4			
District	Sub-surface	9,905,158	60	64.5	224	120	97.6%
Waterloo & City	Deep tube	353,841	2	2.4	26	6	97.6%
Jubilee	Deep level tube	10,830,995	27	37.2	64	28	98.6%
Metropolitan	Sub-surface	7,880,436	34	65.3	157	150	97.8%
Northern	Deep level tube	14,595,462	52	59.1	101	93	98.9%
Piccadilly	Deep level tube	11,836,864	52	65.6	179	151	93.4%
Victoria	Deep level tube	7,582,137	16	21.3	40	27	98.0%

Table 1.1: A table giving information about each of the lines on the London Underground network.

In addition, there are numerous underground railway networks utilised worldwide. Figure 1.1 shows the growth of the number of systems worldwide, by decade, broken down by continent [1]. In the most recent decade, the Asia-Pacific continent showed the largest growth in the number of networks. In addition, between the years of 2012 and 2017 the global number of passenger journeys on underground railway networks increased by 8,717 million, representing a 19.3% growth. Figure 1.2 gives the 10 networks worldwide that have the largest number of passenger journeys, as of the end of 2017. The growth was seen largely in the Middle East and North Africa region (MENA) with a 58% growth, Asia with a 28% growth and Latin America with a 20% growth.

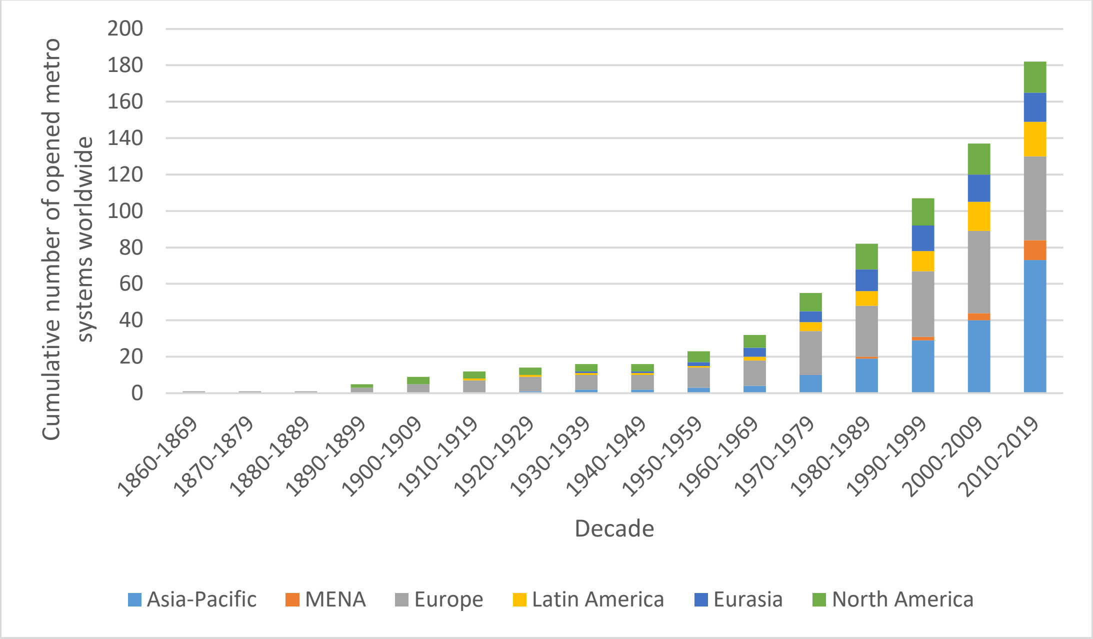


Figure 1.1: The number of underground railway systems worldwide, showing the growth in number with each decade

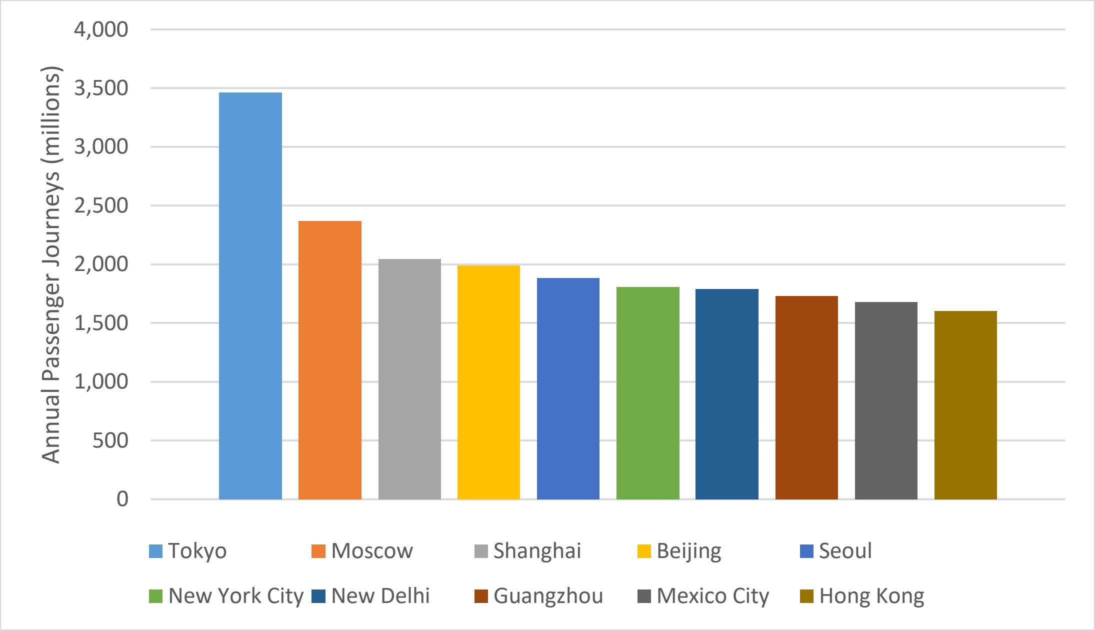


Figure 1.2: The 10 busiest underground railway networks worldwide, when compared by the annual passenger journeys in 2017

At the end of 2017 there were 642 lines installed worldwide, with a total combined length of 13,903km and a total of 11,084 stations across these lines. In addition, 1,901km of new infrastructure was put into service between the start of 2015 and the end of 2017. This represents a 15.8% increase in the total length of underground railway infrastructure worldwide. Figure 1.3 gives the 10 longest railways worldwide, as of the end of 2017.

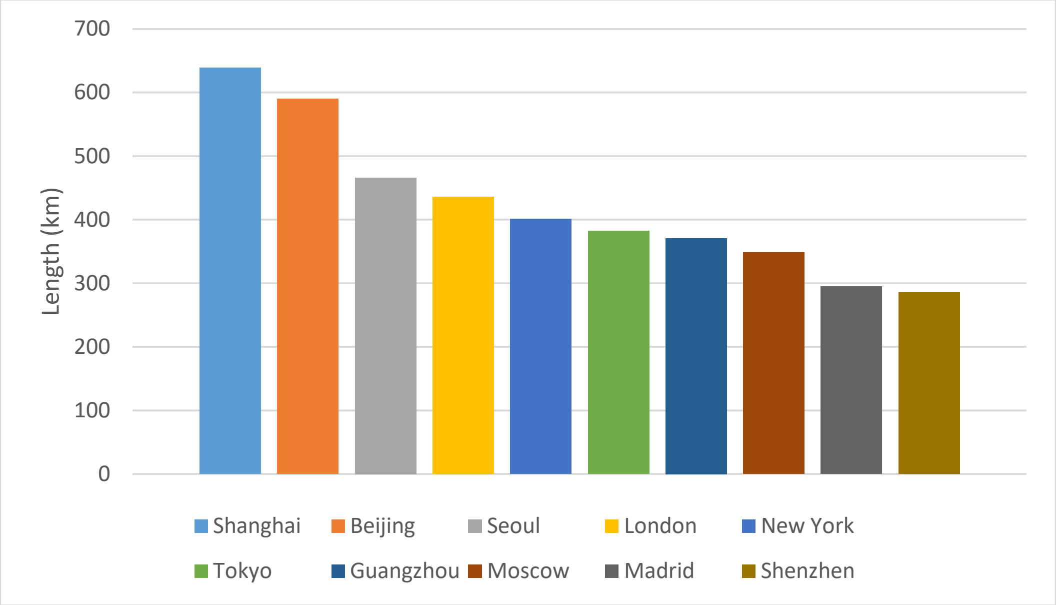


Figure 1.3: The 10 longest underground railway networks worldwide, as of data taken at the end of 2017

UITP, the International Association of Public Transport, predicts that the length of installed underground railway system networks will grow by over 50% before 2022, in comparison to the length of installed lines worldwide at the end of 2017. This comes with the expected construction of 200 new lines and extensions across most regions. There is also an expected increase in the number of fully automated lines worldwide.

The 'International Benchmarking report' by TfL compares the London Underground to other Underground railways worldwide. The report notes that the operational cost per car km is higher than average for London Underground. Despite a reduction in maintenance cost by 6% since 2010/2011, London Underground's maintenance cost is 19% higher than the average. This is attributed to high labour costs, asset condition and age. More specifically, maintenance costs for infrastructure and station facilities were higher than average [12]. In addition, customer risk of fatality was also higher than average. This was largely attributed to a relatively high number of fatalities due to suicides as opposed to accidents caused by failures within the London Underground network. This thesis explores potential methods for reducing the maintenance cost while considering the risk of systems in the network.

1.2: Historical Accidents on Underground Railways

The Kings Cross Fire occurred on the London Underground network resulting in multiple fatalities. The findings of the investigation into the fire had a significant impact on the risk assessment methods employed by London Underground. Developments were made from a largely reactive risk management approach to a predictive approach, and so, details of this accident are presented here, followed by examples of more recent accidents. The fire occurred on the 18th November 1987 and killed 31 people, injuring many more. The incident triggered an investigation into the accident to explain how the fire started, why there was a flashover and why there were fatalities. The report "Investigation into the King's Cross Underground Fire", by Desmond Fennel, details the findings [13].

The report investigated the start of the Kings Cross Fire. The report states that the fire started on a wooden escalator when a match fell between the skirting board and treads. Gaps were observed two weeks before the disaster, but no preventative action was taken. In addition, there was an accumulation of grease, dust, fibre and debris beneath the treads of the escalator. This should have been cleaned. The report states that the fire started around 19.25, and between 19.43 and 19.45 there was a flashover which caused to fire to erupt into the ticket hall.

The report also states that the fire was reported to a member of staff, by a passenger, at 19.30. The member of staff had received no fire training and did not inform the station manager or line controller. Also, there was no existing evacuation plan in place. Two police officers were, by chance, present and at 19.34 they made the decision to radio the fire brigade, with one police officer coming above ground so their radio would work and evacuate passengers through the ticket hall. There was no plan of the underground station available for the emergency services. At 19.43 the London Fire Brigade arrived in the ticket hall but were too late to prevent the flashover.

In addition, at 19.44, Piccadilly and Victoria line trains were ordered not to stop at the station. It was concluded that the continued movement of trains provided air flow to the fire. At 19.46 a Victoria line train was waved down and the passengers on the station were evacuated. The fire was not fully under control until 21.48.

There were several management issues identified in the report, including a lack of formal risk analysis process, a lack of evacuation procedure, inadequate staff training and poor maintenance of the escalator. Following the Kings Cross fire changes to the management of risk were implemented on the London Underground through the introduction of systematic evaluation of risks for several hazardous events. Currently the London Underground Quantitative Risk Assessment (LU QRA) is used to assess risks [14]. However, there have been several major accidents on the London Underground network since. These are detailed in Table 1.2.

Name	Category	Date	Consequences	Description
Chancery Lane derailment	Derailment	25/01/2003	32 injured	A motor from one of the train vehicles detached as the train approached the station. This caused the following vehicles to derail. The derailed vehicles collided with the tunnel walls. [15]
Hammersmith Derailment	Derailment	17/10/2003		The last carriage of a train derailed due to a broken rail. The passengers were evacuated via a second train. [16]
Camden town derailment	Derailment	19/10/2003	7 injuries	A train derailed as it passed over a Switch and Crossing due to wheel/rail interface issues. [16]
White city derailment	Derailment	11/05/2004		A train derailed as it passed over a Switch and Crossing. Controls were not implemented correctly, especially concerning the design of switch apparatus with curved approaches. [17]
Accident at Archway	Derailment	02/06/2006		A train derailed due to a broken switch rail, caused by a fatigue crack. Degraded timbers and loose fastenings were also

				present. [18]
Mile end derailment	Derailment	05/07/2007	21 Injured	A train derailed after striking a roll of fire resistant material lying on one of the running rails.[19]
Accident at Tooting Broadway	Platform/ train interface issue	01/11/2007	1 Injured	A passenger’s coat was trapped in the train door as it was leaving the station. The train was stopped by a passenger emergency alarm.[20]
Accident at Mile End	Persons struck by train equipment	17/11/2009	3 Injured	An inter-car barrier swung loose from the train as it departed and struck passengers on the platform.[21]
Accident at Gloucester Road-Earls Court	Derailment	12/05/2010	1 injured	An engineering train derailed due to a track gauge error. [22]
Passenger dragged at Holborn Station	Platform/ train interface issue	03/02/2014	1 injured	A passenger was dragged 10 metres after their scarf got caught in closing train doors.[23]
Passenger dragged at Clapham South Station	Platform/ train interface issue	12/03/2015	1 injured	A passenger fell beneath a train after being dragged by their coat, which was trapped in the closing doors of a train. [24]
Ealing Broadway Derailment	Derailment	02/03/2016		An incorrect position of the switch rails resulted in a derailment of a slow moving train as it passed over a Switch and Crossing. [25]

Table 1.2: A summary of the major accidents that have occurred on the London Underground since the Kings Cross fire.

The table demonstrates several derailments, resulting in passenger injuries. Of these, derailment commonly occurred at a Switch and Crossing. This suggests that further study in this area could be beneficial in order to attempt to reduce major accidents on the underground railway. In addition, the high number of fatalities caused by the Kings Cross Fire highlights the need for the consideration of fire safety in underground stations and tunnels.

In addition to these major incidents, passenger injuries and accidents occur much more frequently. In the 2016/17 period, 4,497 passenger injuries were reported of which 80 were classified as ‘major’, by London Underground. In the timeframe, there were also 15 train accidents on the London Underground and 3 Potentially Higher Risk Train Accidents, all of which were derailments [26].

The occurrence of major accidents on underground railways is not limited to London Underground. Table 1.3 gives several examples of major accidents that have occurred on underground railways worldwide, in the time period since the Kings Cross fire [27]. These accidents demonstrate the high number of fatalities, or injuries, which can be associated with major accidents on underground railways worldwide. These accidents demonstrate derailment hazards, fire hazards and collisions.

Name	**Date**	**City**	**Consequence**	**Description**
Nicoll Highway Collapse	20/4/2004	Singapore	4 fatalities, 3 injured	A retaining wall for a 'cut and cover' tunnel evacuation collapsed, causing a road to collapse. [28]
D.C Metro Red Line Crash	3/11/2004	Washington	20 injured	An empty train rolled backwards and collided with a stationary passenger train. The driver did not apply the brakes and there was no rollback protection in place. [29]
2009 Washington Metro Train Collision	22/6/2009	Washington	9 fatalities, 52 injured	A moving train collided with the back of a stationary train. Failures in the track circuit meant that the stationary train was undetected. [30]
Paris Metro Derailment	29/8/2010	Paris	24 injured	A derailed train fell onto a neighbouring track section as it approached a station. The suspected cause was over-speeding and a track, or wheel, defect. [31]
Union Square Crash	29/08/1991	New York	5 fatalities, 200 injured	A train derailed at a Switch and Crossing, due to a driver passing a red at high speed. The train was travelling too fast for the trip arms at work. [32]
Russsell Hill subway accident	11/08/1995	Toronto	3 fatalities, 140 injured	A commuter train collided with a stationary train after the driver ran through three red lights. The signalling system regularly displayed false red signals and the trip arms failed. [33]
Baku Fire	28/11/1995	Baku	337 fatalities, 200 injured	A train caught fire due to an electrical spark in the wiring under one of the cars. The trains were not made of flame resistant material. Many people died of carbon monoxide poisoning as

				the carriages caught fire. [34]
Nakameguro train disaster	08/03/2000	Tokyo	4 fatalities, 31 injured	The last carriage of a train derailed and collided with a train travelling in the opposite direction [35].
Brooklyn derailment	20/06/2000	New York	84 injured	Derailment of two of the train carriages as the train was pulling away from a station. [36]
Daegu	18/02/2003	Daegu	192 fatalities, 151 injured	An underground station fire spread between two trains that were stationary in a station. It was suspected that the fire started by arson. [34]

Table 1.3: A summary of some of the major accidents that have occurred worldwide since the Kings Cross fire

These accidents highlight the importance in preventing accidents on underground railways, in order to ensure the safety of passengers, staff and members of the public. Risk assessment can be used to prevent accidents such as these, by providing a framework to understand the systems across the network and identify areas where improvements are needed. Changes in design, operation and maintenance can be used to control risks to an acceptable level. Quantified Risk Assessment methods are commonly used to highlight areas of weakness so that preventative measures can be taken to reduce the likelihood of accidents occurring.

1.3: Basic Concepts

1.3.1: Risks and Hazards

For safety critical systems, such as an underground railway, it is important to minimise risk as accidents can lead to multiple fatalities. The Health and Safety authority (HAS) defines a hazard as [37]:

"A Hazard is a potential source of harm or adverse health effect on a person or persons"

They also define risk as:

"The likelihood that a person may be harmed or suffers adverse health effects if exposed to a hazard."

Risk, R, is defined quantitatively as the product of the consequences, C, of an undesired event and the frequency of its occurrence, f. This is given in Equation 1.1:

$$R = C \times f \tag{1.1}$$

The risk of a hazardous event can be reduced by either reducing the frequency of occurrence or by reducing the consequences, should the hazardous event occur.

1.3.2: Ageing Systems

For an ageing system with components operating up to and beyond their intended life, there is a cycle of replacement and repair to maintain operation. As components age their hazard rate commonly follows that of the 'Reliability Bath Tub Curve'. This curve features three phases as shown in Figure 1.4 [38]:

- Burn in: Initially there is a high hazard rate for the components. This is commonly due to manufacturing defects or poor installation of the component.
- Useful life: The component has a constant hazard rate due to random failures.
- Wear out: As the component reaches the end of its life its hazard rate increases due to processes such as corrosion, fatigue, or wear.

Testing of components after installation can be used to identify the expected high level of early life failures. As components enter the wear out phase, failures are more likely and components must be maintained, or replaced. Component failure can be revealed or unrevealed. In the unrevealed case, inspection or testing of the component is required to identify the failure. Risk assessment can be used to identify which combinations of failures can result in a hazardous event. As components in a system age, and failure becomes more likely, without effective maintenance there can be an increase in the risk associated with the system.

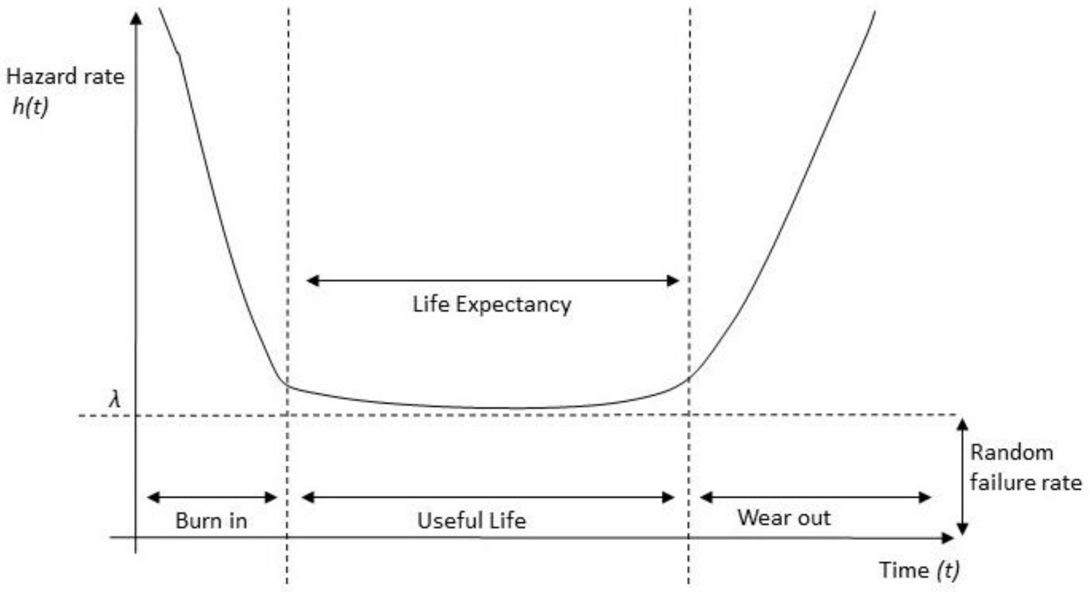


Figure 1.4: The Reliability Bath Tub Curve

1.4: Risk assessment on an underground railway

Risks to passengers, staff and members of the public on an underground railway can be lowered by either reducing the frequency of hazard occurrence, or reducing the severity of the consequences. By identifying the contributing factors to the occurrence of a hazard and the severity of potential consequences, areas of weakness can be identified. Improvements can be made in these areas.

Using London Underground as an example of the type of hazards present on an underground railway gives an insight into the sort of events that are currently considered in the risk assessments applied in the underground railway industry. In the LU QRA, major hazards are identified on each line and the risk of fatality of each is evaluated. Based on the similarity of the outcomes the major hazards are grouped. Each of these groups of major hazards is referred to as a ‘Top Event’. Hazards can be identified though methods such as checklists or a hazard and operability study (HAZOP).

London Underground has identified 18 Top Events which form the basis of their risk management strategies [39]. In practice these Top Events often contain distinct hazards that have been grouped under a single heading for presentation purposes. However, during quantitative analysis of the top event, the hazards are treated as distinct. The Top Events are:

Ventilation Hazard

In the ‘deep level tube’ tunnels ventilation is provided by ventilation shafts. On ‘cut and cover’ lines ventilation is provided by natural draughting. The movement of trains helps to move air through both types of tunnel. If a train is immobilised in a section of track, rising temperature and a lack of fresh air can pose a threat to passengers. There may also be incidents related to smoke or fumes. The ventilation hazard Top Event includes risks posed to passengers due to poor ventilation leading to heat, smoke or fumes in underground track sections.

Train Fires

Train fires can start under the floor of the train, in the saloon car or spread to the car from an external source. This Top Event includes any fires that occur on a train, including those started by malicious action.

Escalator Incidents

The escalator incidents Top Event includes any falls, or injuries, gained while using an escalator within a London Underground station. High levels of congestion can contribute to the severity of these incidents as, if one person falls, then it can cause others to fall. It is more likely that a fall may occur if the escalator stops suddenly. Large items of luggage can contribute to this group of incidents because they may topple and fall.

Flooding

There are a number of ways that flooding could occur. There may be flooding from the Thames due to a rupture in a tunnel running under the river, or flooding due to failure of the Thames barrier, leading to water entering at street level. Flooding may also be the result of broken pipes or sewers. In some areas of the tube water ingresses at a slow rate from the surrounding ground, pumps work to remove this water. Failure of these pumps could also lead to minor flooding.

Power Failure

The power failure Top Event includes loss of power from the National Grid, or loss of power from faults in the supply points. Total power failure to the network can result in stranded trains which can give rise to ventilation hazards, due to the fan failure and lack of train movement, or flooding hazards due electric pump failure. There is emergency lighting available in stations.

Derailment

The derailment Top Event includes all scenarios where a train leaves the track due to an unplanned event. This can occur for multiple reasons. Contributory factors include track related faults, Switch and Crossing failures, obstructions on the track or over speeding.

On train incidents

This Top Event includes the risks to passengers after they have boarded the train, such as unauthorised use of inter-car doors and train door opening between platforms.

Collision between trains

This Top Event includes collisions between passenger trains, or between a passenger train and non-passenger trains. These collisions can be end to end, side on, or ‘swipe’ collisions, which is where two trains moving in opposite directions graze past each other. For an end to end collision to occur there must be two trains in the same section of track and they must fail to brake in time. Signalling systems aim to prevent two trains being in the same section of track by displaying a red light if the previous section is occupied. However, if a signal shows red, ‘Trip and Proceed’ may be implemented. This is a process by which drivers may pass a red signal, at a low speed, if it is believed that the signal has failed. This could lead to a collision if there was actually a train in the following section.

Explosion

This Top Event considers explosions occurring from malicious action or from accidental build-up of flammable material or gasses. This can be especially dangerous in a tunnel, due to: confinement within the vicinity of the explosion, difficulties with evacuation and limited access for the emergency services.

Station Fires

This Top Event considers station fires occurring in both public and non-public station areas, and includes fires around lifts, escalators and machine rooms. There are automatic heat and smoke alarms to give advanced warning of a fire and in deep stations there are fire suppression systems fitted.

Collision Hazard

The collision hazard Top Event covers any event where a train impacts a fixed object. Examples of fixed objects that could be involved are: tunnel walls, platform edges, tunnel terminals and floodgates. Failure of the brakes and emergency brakes, on the train, can contribute to this Top Event.

Arcing

Arcing is a phenomenon by which a large amount of current passes between two conducting materials, through a non-conductive media, such as air. There is visible light emission and high temperature. The trains on the Underground are electrified with a live power rail and an electrical pickup on the train, hence, arcing can occur.

Structural Failures

The Structural Failure Top Event includes any collapse or failure of infrastructure on the network and covers bridges, stations, tunnels and shafts.

Lift incidents

The lift incident Top Event includes any event that occurs in or around the lift, such as passengers becoming stuck in the lift, falling down the lift shaft or uncontrolled lift movement.

Tunnel Fires

This Top Event includes any fire that occurs on track sections outside a station vicinity, either in a tunnel, or in an open section of track. The severity of the consequences of tunnel fires can be severe due to the smoke produced by the fire. Build-up of grease, dust and debris in tunnels can catch fire. An ignition source can come from electrical faults, arcing or deliberate action.

Stairs and Assaults

Stair hazards include any falls on stairs within stations. This Top Event also includes any assaults on customers in stations, or on trains. High levels of congestion can contribute to this Top Event.

Unauthorized Access to Track

This Top Event includes and hazardous situations arising due to the presence of unauthorised persons on or around the track. They may have come from a platform or another entry point. This is not only dangerous to the trespasser but could endanger passengers, for example, sudden braking may cause train passengers to fall.

Platform Train interface

The Platform Train interface Top Event includes a number of incidents that could occur and endanger a customer on the platform when a train is approaching, stationary or leaving the platform. The main scenarios include; a customer being struck by an approaching train while they are on the platform, a customer falling from a platform or between a train and the platform, a customer getting crushed in the train doors and a customer being dragged along the platform by being caught in a closed door.

Within the LU QRA, Fault Tree analysis is carried out to estimate the frequency of occurrence for the hazards grouped within these Top Events. Event Tree analysis is performed to estimate the risk, by considering potential consequences, of the hazards within each Top Event over a range of different outcomes. Following this introduction to current risk modelling methods and the hazards present on an underground railway, the next section presents the project aims and objectives.

1.5: Aims and Objectives

Changes in design, operation and maintenance can be used to control risks to an acceptable level. Risk assessment methods are commonly used to highlight areas of weakness so that preventative measures can be taken to reduce the likelihood of accidents occurring.

Therefore, the aim of this project is:

To develop a method that can be used to comprehensively model risk on an ageing and increasingly utilised underground railway.

This will be achieved through the following objectives:

1. Develop a modelling capability that incorporates the following features of systems in an underground railway:
   i) Increasing failure rates as a system ages;
   ii) Allow dependencies between different failure events, that can combine to result in a hazardous event;
   iii) Include complex asset management strategies, which incorporate phased and opportunistic approaches, to deliver maintenance to the parts of the system at the time in their life that they most need it;
2. Provide a framework for a risk-based asset management optimisation tool.
3. Include a measure of uncertainty in the predicted value of risk.
4. Validate the modelling capability by application to real systems and hazard scenarios.
5. Ensure that the computational efficiency enables the modelling capability to be incorporated effectively.

1.6: System Application

In this thesis, the proposed modelling approach is explored through application to two separate areas of study. The first area of study is train derailment due to a Switch and Crossing (S&C) failure and the second area of study is fire risk on underground stations. The application of S&C derailment was chosen due to the contribution it has made to past derailment accidents. Since the Kings Cross Fire in 1987, multiple major accidents have been caused by S&C failure, as detailed in Table 1.2. The fire risk on underground stations was selected for the second area of study due to the high number of fatalities demonstrated by the Kings Cross Fire and the fire at Daegu. A review of models available in literature for these applications is presented in Section 2.6 of this thesis. This review demonstrates areas for further research, and further justifies the selection of these applications.

1.7: Key Contributions

Chapter 2 presents a literature review and provides the justification of the research direction within this thesis. There are several contributions made within this thesis to the wider body of literature. Firstly, an approach to hazard or risk modelling that builds on existing methodologies applied in industry is presented in Chapter 3. This extends currently implemented methodology to allow more detailed modelling of components with complex degradation, maintenance and inspection strategies.

Two models have been created to demonstrate the proposed methodology. The first, which is applied to S&C derailment, goes into further depth than S&C models available in literature. The model improves on the state of the art as it removes assumptions of perfect maintenance and inspection and allows dependencies to be introduced through maintenance actions. In addition, the system state is predicted. This work is presented in Chapter 4. The second developed model considers a fire protection system and uses a combined modelling approach for deluge, detection, and alarm sub-systems. This model also includes areas of novelty, in comparison to models available in literature, for fire protection systems. Key areas of this include: the incorporation of a phased asset management strategy, modelling of probability of unrevealed failure, modelling false activation and the combination of all three sub-systems. The inclusion of phased asset management strategy modelling improves the state of the art as it allows exploration of strategies that can change throughout a system lifecycle. This work is presented in Chapter 5.

In addition, a novel approach for the optimization of a Petri net model, with a phased asset management strategy is presented. The approach is beneficial as it allows different strategies to be applied at different phases of the system lifecycle. A method for studying the convergence of the model is also applied; this improves on current convergence checks for Petri net models, where the convergence is plotted on a linear scale. In addition, a novel approach for estimating the uncertainty of the model outputs, given uncertain model inputs is presented. This improves on the state of the art for predictions, where the uncertainty in the output of the Petri net model is usually unstated, and uncertainty in model input parameters is not considered. A more informed decision can be made based on model outputs, if an estimate of uncertainty is provided for each model output. These methods can be found in Chapter 6.

Finally, in Chapter 7, a new Petri net reduction methodology is presented. This work includes research into comparison metrics that quantify the difference between outputs of separate Petri net models. The proposed reduction methodology is a more flexible approach than those currently available in literature, which are commonly rule based. Novel research exploring the proposed method is presented, including its use to improve current optimization methods using an approximate solutions space. Also, the use of the approach to justify the choice of reduced model structure is explored. This improves on the state of the art for model selection, where Petri net models are usually user defined, as the approach provides a quantitative measure to back up choice of model structure.

1.8: Summary

This chapter has provided a brief introduction to underground railway networks worldwide and provided background of several historical accidents on underground railways. There have been a number of occasions where accidents have occurred on the London Underground railway, and other underground rail systems across the world. These accidents can be attributed to component failures, design flaws or human error. In addition, controls that are designed to prevent accidents, such as trip arms, have been found to be insufficient. Accidents such as the Moorgate Disaster, Union Square Crash and the Kings Cross Fire, highlight the need for risks to be fully understood in order to identify areas where improvements are needed. These areas form the basis of applications used throughout this thesis, for demonstration of the methodology developed. Aims and objectives and an overview of the thesis contributions are also presented.

A structure of this thesis is as follows: Chapter 2 presents a review of methodologies for risk assessment, or asset management, applied in industry to underground and over ground railways, or proposed in literature. Chapter 3 presents the methodology proposed in this thesis. Chapter 4 focusses on modelling and testing complex asset management strategies for ageing systems, with an applied model presented for the derailment occurrence at a Switch and Crossing. Chapter 5 develops modelling of phased asset management strategies and is demonstrated with a model for fire protection system unavailability. In Chapter 6, methodologies are developed for risk-based optimisation of phased asset management strategies and for the consideration of convergence and uncertainty in the modelling approach. The developed methodologies are applied to the model presented in Chapter 5, for fire protection systems. Chapter 7 presents a new method for the reduction of model size via Bayesian Inference. Finally, the conclusion of the thesis is presented in Chapter 8.

Chapter 2 Literature Review

This chapter provides a review of current risk, hazard and asset management modelling methodologies in the railway industry, along with alternative methodologies reported in literature. Within each section of this chapter any gaps in existing approaches are discussed, in order to inform the research direction of the thesis. This forms the basis for the justification of the approaches proposed for application throughout this thesis.

In the first part of this chapter, a review of risk modelling methods currently implemented in the UK's underground and over ground railway is presented. Following this, in Section 2.2, a general review of system failure and risk modelling methods is given. A review of: Fault Tree analysis, Event Tree analysis, Petri net modelling, Markov modelling and Bayesian Networks is provided in context of modelling system failure, ageing or risk. In Section 2.3 a review is given of work surrounding the optimisation of asset management of a system, including methods that focus on risk-based optimisation. Section 2.4 presents a review of methods for incorporating uncertainty in model predictions. Section 2.5 gives a review of methods for reducing model complexity, with the view of improving the computational efficiency of large system models.

Within this thesis two system models are developed, the first considering derailment at a railway S&C, and the second considering underground fire protection systems, as discussed in Section 1.6 of this thesis. In this chapter a review of current S&C condition modelling methods is presented in Section 2.6. Likewise, a review of underground fire protection modelling is given in Section 2.6. These system specific reviews highlight any missing functionality in the models currently available for each system. Finally, the discussion and conclusion sections are given. These summarise any areas identified in this review for further development and give an outline of where these areas are addressed in this thesis.

2.1: Risk modelling in the UK railway industry

2.1.1: Underground Railway

The London Underground Quantitative Risk Assessment (LU QRA) uses a combined Fault Tree and Event Tree approach to quantify risk. First hazards are identified and divided into 18 categories identified as 'Top Events', which group similar hazards together [39]. Descriptions of these events can be found in Section 1.5 of Chapter 1 of this thesis.

A Fault Tree is constructed for each Top Event to identify any potential causes and predict the Top Event frequency of occurrence. Event Tree analysis is carried out to consider any consequences following the occurrence of the event. Here, different eventualities following an initiating event occurrence are evaluated, and any associated consequences for each eventuality are included. This evaluation includes any mitigating actions, a measure of passenger loading and the predicted severity. The predicted loss of life for each eventuality is then used as the measure of consequence, finally giving an estimate of the risk for each Top Event. A simple weighting factor is used to adjust for injuries. The risk of the whole London Underground network was estimated at 6.8 fatalities/year for the 2014 reporting period [39].

London Underground's model is focused on loss of life to members of the public and passengers[40]. The risks on each underground line are considered separately in the LU QRA, however, it is a line based model and so does not have a high level of resolution for consideration of different conditions present across each line, and the risks associated with them.

Hazardous events can occur which are related to human error, for instance, a driver ignoring a red signal. Currently HEART (Human Error And Reduction Technique) can be used, or expert judgement. HEART uses experimental evidence from studies such as nuclear power control room simulation studies; hence this may not be directly transferable to the railway industry [41]. The LU

QRA does not consider human factors and abnormal operations in much depth. However, the likelihood of detection and mitigating actions by staff, passengers or the public is included, allowing risk reducing methods to be modelled in some capacity.

The combined Fault Tree and Event Tree method used in the LU QRA has a graphical representation which can aid in the communication of the method to stakeholders. It is also an adequate method for dealing with events that are independent. The current method gives a pessimistic prediction for the overall risk on the underground network when compared to the accident statistics each year. This can be attributed to the contribution of rare events to the predictions made by the model. These rare events can have high fatality levels which increase the value of predicted risk each year. However, since they are infrequent in occurrence, the accident statistics may be lower year-on-year in comparison to the prediction.

Since the introduction of the LU QRA the predicted risk made by the model has been steadily decreasing. The initial value generated by the model was considered highly pessimistic and changes to the model to bring this value more in line with reality, along with risk reduction measures applied to the London Underground network, have led to this decrease.

The current model is based on historical data, consequence analysis and expert predictions. The data used may be incomplete due to poor recording, or outdated due to changes to the network. It is also possible for the model to consider events that have yet to occur, as there may be no data or current knowledge of the possibility of their occurrence. London Underground updates a few models in the LU QRA every year, this has the potential to distort the overall picture of the risk by introducing inconsistencies in relative risk values between the Top Events. In addition, there is no quantification of the uncertainty in the final predicted value of risk.

The combined Fault Tree and Event Tree approach used assumes a constant failure rate, which is acceptable for a system in its ‘useful life’. As a system ages and enters the ‘wear out’ phase the component failure rates may not remain constant and so the model can become increasingly inaccurate. With the current approach it is also difficult to consider complex maintenance strategies; only a constant repair rate is included. The combined Fault Tree and Event Tree method used assumes that events are independent. In reality events are often dependent on each other due to the operational strategy or maintenance strategy. The model also does not consider the time ordering of events [42]. This can be important where initiating and enabling events are concerned.

There are areas in which the Fault Tree method encounters difficulties, including those surrounding dependence between basic events, time dependence, sequencing of basic events and difficulty handling components with multiple degraded states. Although the Fault Tree method is not limited to components with a constant failure rate, the LU QRA assumes a constant failure rate.

2.1.2: Over Ground Railway

The Rail Safety and Standards Board (RSSB) uses the RBBS Safety Risk Model (SRM) to quantitatively analyse risk on the over ground railway. This model also uses a combined Fault Tree and Event Tree approach and is similar in many ways to the London Underground QRA. The first version of the RSSB SRM was released in 2001 and has been regularly updated and extended since then [14]. The purpose of the SRM is to give an overall estimate for risk and allow identification of areas that require improvement. The SRM quantifies risk at system level before breaking down this risk to route and operator level.

The SRM considers a larger number of hazards than the LU QRA. The SRM considers not only hazards that can result in fatalities but also hazards that can result in different severities of injury for passengers and public. The severities of injury considered are: major injury, minor injury, shock and trauma. Hazards that may affect the work force are also considered along with suicides that occur on the railway network. This provides a more detailed picture of the injuries sustained in an accident when compared to the LU QRA by considering high frequency low consequence events, such as non-fatal trips and falls, as well as low frequency high consequence events, such as derailments [43].

In the SRM, 131 hazardous events are considered which are grouped into accident categories including: train accidents, movement accidents, non-movement accidents and trespass [44]. Some examples of the hazardous events considered in the SRM are: road traffic accident, platform edge incident, assault and abuse, on-board injuries, and slips, trips and falls. The frequency of each event is estimated through Fault Tree analysis, following this Event Tree analysis is used to consider the consequences should the event occur. The results of this analysis are an estimate of the frequency of each event occurring along with a predicted number of casualties should each event occur.

In the SRM, the depth of the Fault Tree analysis stops where no more evidence is available, this means that the SRM has fewer lower Fault Tree levels and less reliance on expert opinion when compared to the LU QRA. The model is also updated fully every 18 months which ensures that all of the event predictions are proportionate to each other. This also allows the effects of any changes that have been made to be analysed. The model predicts a slightly pessimistic value for risk when compared to the accident statistics each year, however, this can be attributed to the contribution from rare events. This type of event may happen infrequently but can have severe consequences and so increases the predicted risk value.

The SRM has similar downfalls to the LU QRA. The data used in the model may be incomplete or inaccurately recorded and, for some cases, there is little data available and so there are difficulties in gaining a high level of confidence in the predictions. There is also the possibility that a rare event could be missed from the analysis. In addition, there are limited asset management strategies or time dependence incorporated into the model. Also, the failures are assumed to occur with a constant failure rate and therefore their times to occurrence follow an exponential distribution. This is explained further in Section 3.3 of this thesis. There is currently no measure of uncertainty in the model predictions and hazards in yards, depots and sidings are not included.

2.1.3: Summary

This section has given a review of methods currently implemented in underground and over ground railway systems to model risk. As this project aims to improve currently applied methods, this provides context to the work in the remainder of this thesis.

The current method used in the UK railway industry combines Fault Tree and Event Tree models to predict risk. It has several areas of weakness, especially in a situation where components are ageing and have a non-constant failure rate or there are dependencies between component failures. In addition, the approach does not model different asset management strategies and impact on the risk. Uncertainty is also not provided on the estimates gained from the analysis.

This review informs the research direction by highlighting the weaknesses in the approaches currently implemented in industry. There are areas for improvement surrounding risk modelling methods that can incorporate changing failure rates as a system ages and allows dependencies between failure events. In addition, modelling of the asset management of systems within this framework can be developed further. The next section of this chapter details some of the developments in literature surrounding different approaches to modelling risk and system failure.

2.2: Risk Modelling in Literature

There are developments and applications in literature for risk models or component failure models that move away from a combined Fault Tree and Event Tree approach currently employed in industry. Examples of these alternative approaches are Markov models, Petri nets models and Bayesian Networks. These approaches often aim to consider dependencies between events and time dependence and can have the ability to cope with non-constant failure rates. In addition, there are further developments to the Fault Tree and Event Tree methods available in literature, which extend their functionality. This section describes some methods proposed in literature and considers their suitability to modelling risk for an ageing system, namely modelling a changing failure rate as the components age, a selection of maintenance and inspection strategies and the incorporation of dependencies between components due to operational or maintenance strategies.

2.2.1: Fault Tree Analysis

A full explanation and an example of the Fault Tree methodology can be found in Section 3.1 of this thesis. However, there are several developments of the Fault Tree method which aim to solve some of the problems associated with Fault Tree based models. This section further describes some of the weaknesses identified in the Fault Tree method and some of the extensions that have been added to try and combat these weaknesses.

The Fault Tree Handbook, released in 1981, by the U.S Nuclear Regulatory Commission details the synthesis and analysis of Fault Trees [45]. The handbook also gives some of the problems and difficulties encountered with Fault Tree analysis:

- Since Fault Tree analysis only considers a top event resulting from complete failures it is not easy to use Fault Tree analysis to model situations that arise from incomplete failures, such as those where capacity is reduced.
- The method is not exhaustive because only a limited number of top events are considered.
- Fault Tree analysis is an expensive and time consuming method and it is difficult to make changes to a Fault Tree based model.
- In many Fault Trees parameters are considered as fixed values for ease of calculation, this may not be the case in real life, for instance failure rates may change with time.
- Quantification of a Fault Tree requires that the basic events are independent. This may not be the case due to a common causes leading to the occurrence of more than one basic event. The common causes can only be indicated by identifying minimal cut sets and manually looking for common causes within each cut set. For Fault Trees with a large number of minimal cut sets, approximations are made which ignore higher order minimal cut sets. However, if there is a common causes in the discounted minimal cut sets the probability of occurrence for the cut sets may still be significant, leading to an incorrect prediction for the top event when they are discounted.
- Measures of uncertainty and the effect of changing a variable can be carried out manually by changing the variable that is being tested and observing the effect on the top event. Monte-Carlo Simulation can also be used where multiple trials are carried out with a changing value for the variable, to measure the effect. These methods are time consuming for a large Fault Tree.
- Often uncertainty is not included in a Fault Tree model.

The paper, published by Dugan, Bavuso and Boyd, 1992, introduces dynamic Fault Trees that can model sequence dependent failures and use of components in standby [46]. Several gates are defined in this paper, these include:

- The ‘Functional-Dependency’ gate which contains a trigger event and events dependent upon it. When the trigger event occurs then the dependent events also occur and the fault propagates up the Fault Tree.
- The ‘Cold Spare’ gate which is used in situations where there is a primary operation backed up by other operations. For example, a back-up generator that is only used during a power

failure. A cold spare gate considers the degradation of the back-up operation while it is not in use.

- The ‘Priority AnD’ gate requires all of the input events to occur and they must occur in the given order.
- The ‘Sequence Enforcing’ gate can be used to model situations where the input events can only occur in a set order.

The dynamic Fault Tree is converted to a Markov chain for numerical analysis, for ease of formulation of a Markov model and to extend the functionality of a Fault Tree. The method makes several assumptions that are common with Fault Tree analysis; it is assumed that the basic events are random and independent, the failure rate is constant and the lengths of time are short so few failures will occur in the time interval. The assumption is also made that repairs cannot be made while the system is in use.

As the size of the dynamic Fault Tree increases, the size of the Markov model increases exponentially, hence it is costly and time consuming to analyse the dynamic Fault Tree quantitatively. A method described in the paper by Gulati and Dugan, 1997, [47] breaks a dynamic Fault Tree into independent subtrees which can be either dynamic or static, where static Fault Trees are those with traditional gates. To find a solution to the Fault Tree, each subtree is evaluated separately. Static Fault Trees are analysed by Binary Decision Diagrams and dynamic Fault Trees are analysed by the more time consuming method of conversion to a Markov chain. The top event in each subtree is replaced by a basic event representing the subtree, this process is repeated up the tree. This method makes it easier to evaluate the dynamic Fault Tree if only a small section has dynamic properties, as the more straight forward Binary Decision Diagram method can be used for a large portion of the dynamic Fault Tree.

Furthermore, an approach is presented by Magott and Skrobanek, 2012 [48], to extend the Fault Tree method to include time dependence. This approach aims to adapt the method further to consider measurements such as delay time between event cause and effect, hazard tolerance time and fault detection time. Two further gates are defined. These are casual gates which represent the delay times between cause and event, and generalisation gates which represent combinations of causes. The Fault Tree is constructed and then the time intervals for events and gates are calculated from timed state charts. Finding these time parameters can be difficult and so reduction methods for the timed state charts are used. However, there is currently no set of reductions that can be applied to every case. This makes the method time consuming and complex as there is no way of calculating the delay times automatically.

In order to incorporate durations that lead to critical events, failure sequences and repairable multi-states, a Fault Tree extension is proposed by Khanh Nguyen, Beugin and Marais, 2015[49]. The extended Fault Tree presented here is evaluated by considering the critical events of the Fault Tree which are then represented by a Petri Net. A Monte Carlo simulation is carried out to evaluate the Petri Nets. This method is applied to a satellite-based railway system, and gives similar results to a Petri Net simulation of a Fault Tree considering all events. However, the method is complex and requires approximation for the distribution function for each critical event. The method also only considers failures that occur with an exponential distribution with a constant failure rate.

The introduction of dynamic Fault Trees aimed to create a method that is *“flexible enough to capture the dynamic aspects of the system, but which is (almost) as easy to use as a fault tree”* [46]. A major strength in the Fault Tree model is that it is clear and easy to use to provide a framework for analysis of failure modes of the system. The method can be easily understood, explained and quickly evaluated. There has been a large amount of research into solving some of the limitations of Fault Tree analysis, including introducing time dependence, event dependencies and order to basic events. These methods are time consuming and difficult to apply and often do not incorporate non-constant failure rates of components. For an ageing system in particular, a method must be found that copes well with a non-constant failure rate.

2.2.2: Event Tree Analysis

This section gives a review of the Event Tree methodology. A full explanation and an example of the Event Tree methodology can be found in Section 3.2 of this thesis. Traditional Event Tree analysis is detailed in the "US Nuclear Regulatory Commission: Reactor Safety Study", published in 1975 [50]. Here, chains of events and their consequences, following an initiating event, are analysed with an Event Tree structure. Quantitative analysis can be undertaken by assigning probabilities at each branching point of the Event Tree and propagating these values through the Event Tree, under the assumption that the events are independent. Time dependence is not included in the model. In addition, there is no incorporated measure of uncertainty. Event Tree analysis struggles to represent how the state of the system, and the environment, influences the sequence of events, due to a lack of time dependence and limitations of event sequencing in the method.

In order to more represent an Event Tree structure as a matrix of probabilities, with the aim of improving the ease of analysis of the model, the paper by Kaplan, 1982 provides a methodology [51]. A probability matrix is created from the Event Tree model that represents the likelihood of moving between system states, within the Event Tree, which are a result of different event sequences. The method employs the combination of 'sub-event trees', which have a matrix representation and can be combined through matrix multiplication. The resulting matrix for the whole Event Tree relates the entry states of the system to the exit states of the system. Intermediate system states, which represent the system condition on the partial completion of a chain of events, can also be considered. For each intermediate system state present in the Event Tree, a set of triplets can be defined that represents the risk of the state. These triplets contain the possible exit states, the probability and a measure of the consequences, of each exit state. The paper presents an alternative method for the analysis of an Event Tree and for the representation of risk within partially completed event sequences. However, the method assumes both a fixed value for each event probability and for the consequences of each chain of events.

There are several weaknesses identified with Event Tree analysis surrounding time dependence, static event ordering, and dependencies between events. In addition, there are difficulties in finding accurate input values for analysis and incorporating uncertainty into the outputs of the model. A number of papers can be found that attempt to address these issues.

Further advances in Event Tree analysis are in the area of Dynamic Event Tree Analysis, which aims to allow the quantification of risk in dynamic event sequences. The report, "Dynamic Event Tree Analysis Method (DETAM) for Accident Sequence Analysis" presents an analysis method to this aim [52]. The methodology presented here simulates accident scenarios through dynamic branching of an Event Tree, governed by defined rules. There is a focus on dynamic responses of operators, and the system, during an accident. The method is suggested as an improvement to the static nature of traditional Event Tree analysis. The dynamic Event Tree method presented allows branching to occur at different points in time to create alternate Event Tree structures, depending on the conditions at that time point. The dynamic branching is governed by a set of branching rules and sequence expansion rules, and a set of variables included in a branching set and plant state. The branching set gives the variables that determine the new Event Tree sequences at any node in the Event Tree. The plant state is the set of variables that influence the frequency assigned to each branching. The branching rules determine when branching should take place and the sequence expansion rules limit the number of sequences possible. A quantitative tool is also defined, which can be used to compute state variables and branching frequencies. The approach presented in the paper accounts for ordering and timing of events and allows the human interactions with the system to be specifically modelled under different conditions. However, a dynamic Event Tree is more difficult to construct and analyse than a traditional Event Tree due to the number of extra system definitions required. For a large system this method could become extremely computationally expensive due to the multiple branching scenarios. The method also does not provide a framework for estimating the parameters governing the events and consequences in the model. A paper by Rutt et al., 2006, details work on a system software infrastructure for the analysis of Dynamic Event Trees, towards making a useable tool for industry [53]. However, it is stated that there is a large body of work to be completed before an end product is available.

This section has highlighted some of the limitations of Event Tree analysis, especially when modelling non-independent events, time dependent events or events where the sequence of occurrence impacts the outcome. The Dynamic Event Tree method may give interesting results as event sequences can be time dependent, however, requires further research and development to allow the method to be solved computationally for complex systems. Furthermore, several approaches have been proposed to propagate uncertainty through an Event Tree model. In addition, the Event Tree method is widely applied in the railway industry.

2.2.3: Bayesian Networks

Bayesian Networks have been proposed as a methodology to improve the RSSB SRM, by allowing the Event Trees used across the network to be generalised to a smaller collection of flexible models. Approaches with this objective are given in Marsh and Bearfield, 2008, and Bearfield and Marsh, 2005 [54][55]. The main aim of the work is to address the issue that repeat analysis of Event Trees is required, for each location with different attributes. In the work a Bayesian Network framework is proposed in order to extend the Event Tree method, used in the RSSB SRM. The proposed approach allows the same model to be used in multiple locations, despite the presence of different attributes at each location. The method identifies the factors and conditions that influence the events under consideration, and these are included within a Bayesian Network structure. An example application for an Event Tree for a derailment is presented. In this example, the location attributes identified include factors and conditions such as: track curvature, if the track is enclosed, train speed and rolling stock type. The Bayesian Network representation of the Event Tree, which includes these location specific attributes, is simulated for numerical analysis. In this method, with an increase in factors the Bayesian Network becomes increasingly large, escalating the computational cost and the quantity of data required for the model. Hence, the model can become complex with a heavy reliance on accurate data for the railway network. This method could be expanded to build a universal model for the risk analysis of the railway network, but this may result in an unfeasible level of model complexity.

There are several examples where the flexibility of Bayesian Networks is further demonstrated. The paper by Andrews and Fecarotti, 2015, incorporates Bayesian Networks into a modelling approach that considers maintenance of assets that are in use beyond their originally intended lifetime, in order to optimise the maintenance strategy used, to minimise whole life costs while maintaining the high level of safety [56]. Independent modules are identified in the system. A Petri Net approach is used to model independent parts of the system and a Bayesian Network is used to combine these subsystems to give a picture of risk for the whole system. The method is applied to an overpressure protection system on a wellhead platform. In this approach, failure rates change with time and dependencies between basic events are considered. The approach splits the life-time of components into discreet phases from working, through stages of degradation, and finally failure. The model is evaluated by Monte Carlo simulation, where delay times are taken from suitable statistical distributions. The approach allows analysis of diverse maintenance strategies which focus on different components at set times, as well as the impact of alternative system designs. Another benefit is that all of the potential failure modes of the system are included in a single model. However, for a large Petri Net with many dependencies the simulation of the model can be computationally expensive due to the nature of Monte Carlo simulation, where a large number of runs are needed for convergence of results.

In a contrasting approach, Bayesian Networks are proposed by Oukhellou, Côme, Bouillauta, and Aknina, 2008, for use in real time for the diagnosis of rail defects, which requires efficient computation to give current results without an observable time-lag. The method aims to find a solution to classify singularities detected by rail inspection [57]. Singularities can be detected in the rails for multiple reasons, including typical track structures introduced by installation or maintenance and rail defects. The model aims to distinguish between singularities that are a result of broken rails and singularities such as: fishplated joints, switch joints and welded joints. The state of the rail at a location is modelled by a Bayesian Network. Several Bayesian Network structures are considered in the paper, where different combinations of neighbourhood states, such as the state of the opposite rail or the surrounding rail section, can impact the rail state at any given point. Sensor data and a labelled track state database are used to train the model. The model outputs the probability that the current location is in each of the model states, where the states can include varying singularity conditions. The data used to train and test the model does not include broken rails, and so rail breaks are randomly inserted into the dataset. This introduces modelling bias and deviates from the natural rate of rail breaks. The paper states that if natural rail break occurrence is adhered to then there are too few broken rails to train the model. If this is common over any potential training dataset then it makes it difficult to implement the method, especially if a large quantity of high-quality data is not available. The model gives a good detection rate of rail breaks for the synthetic data used, but commonly classifies a fishplated joint as a broken rail. Improvements are suggested using hierarchical combinations of Bayesian Networks, resulting in a misclassification of 10.4% of the fishplated joints as broken rails.

Bayesian Networks can be used to infer relationships between underlying influences on an outcome. The paper by Wang, Xu, Tang, Yuan and Wang, 2017, uses a Bayesian Network to model the impact of weather conditions on S&C failure [58]. The Bayesian Network used in the approach is built from real data and expert opinion. In the model weather conditions are grouped by an assigned duration of one week, and classified as ‘snow’, ‘rain’, ‘thunder’ or ‘fine’. This is done under the assumption that only snow, rain and thunder impact weather related S&C failures. Thresholds can be assigned for the severity of the weather classification, based on the number of days of the weather type under consideration, for the week in question. For each of the ‘snow’, ‘rain’ and ‘thunder’ classifications there are two severity levels included in the model, all other weather is classified as ‘fine’. The optimal thresholds for each weather type level are discovered via an Entropy Minimization Based Discretization. Two levels of air temperature are also included in the model, classified as: high temperature and low temperature. The number of parameters required for the model is reduced using a casual noisy MAX model, and the parameters required for the model are derived through this process. Monte Carlo simulation of the Bayesian Network can be performed. This is done for a sample dataset and a comparison in made between predicted results to observed results. The model predictions show some agreement when compared to the observed values from the training dataset but less agreement when compared to a test dataset. A small dataset is used for the application of the model given in the paper, which can impact the accuracy of the model predictions.

Finally, an example of a Bayesian Network to incorporate the impact of human factors on accident occurrence, with system-based variables, is given by Castillo and Grande, 2016. Here, a Bayesian Network is implemented to analyse the probability of several incidents of different severities, given a number of variables [59]. These variables include: the type of driver assistance system, the infrastructure of the railway line, the rolling stock, the train speed, the signal state, the line terrain, any technical failures and driver factors, such as tiredness, attention and decisions. The driver’s tiredness is modelled deterministically with a dependence on time. The driver’s attention is modelled with a continuous Markov model. The modelling of human errors in risk analysis of railway lines is considered further by a paper by Castillo et at., 2016, via the same Bayesian Network methodology [60]. The remaining parameters used in the model are gained from expert opinion. Different items that may be present on a railway line are considered and a Bayesian Network is created, that is dependent on these items for each line. Line items included in the approach are: tunnels, S&Cs, signals, announcements and curves. The approach results in a large Bayesian Network, with a sub-net for each item encountered on the line in question. The resulting model cannot be easily solved due to its size, and so is partitioned into a sequence of sub-nets which are solved independently. The computational time increases linearly with the number of subnets. The method allows backward analysis to explain the causes of an incident; however, the method relies on expert opinion to give most model parameters instead of discovering parameters automatically from data.

In summary, Bayesian Networks have been demonstrated to have the ability of combining results from other models to give risk at a system level, this can be used to combine multiple Petri Net models for sub-systems. They can also be used to replace the Event Tree stage of analysis, such that one Bayesian Network can represent multiple Event Trees of a similar structure. In addition, they can be used for analysis of data to make predictions of failures or future trends. Similarly, to the Petri Net approach, as the model becomes large, simulation tools can be required, and these can be computationally expensive.

2.2.4: Petri net modelling

Petri Nets are proposed as an alternative approach for modelling failure analysis in the paper by Liu and Chiou, 1997, [61]. A full description of the Petri Net method, with an example, is given in Section 3.4 of this thesis. The paper demonstrates that Petri Nets have a strong ability to model dynamic behaviour, and so can be applied to systems with multi-state repairable components. The method presented in this paper uses a Fault Tree approach to model the failure modes of the system and then converts the Fault Tree into a Petri Net for analysis, changing each gate to an equivalent Petri Net structure. A matrix is used to represent the transitions available to a token in each place, and to evaluate the Petri Net to discover minimal cut sets. The use of the Petri Net allows sensors to be added to the model so that a failure will not propagate if it’s detected. Maintenance can also be incorporated into the model where a marked place can represent the system under repair after detection of failure by the sensor. There is also no incorporation of time dependence in the application of the method. The method also assumes that it is possible to solve the Petri net model analytically to give an exact solution; this may not be possible as the size of the model grows. A value for uncertainty is not included in the final estimated value.

Petri nets are also used to model dynamic behaviour across multiple subsystems in the paper by Ghazel, 2009 [62]. A method is proposed to model the risk at level crossings, with automatically controlled barriers, that arises when a train collides with a car at the level crossing. A scenario is considered where a traffic jam on one side of the level crossing means that cars move into the area in between the barriers before there is an escape route available. If a train approaches, the barriers come down and the car can be trapped in the path of the train. A model is created that is split into three subsystems: the road traffic system, the rail traffic system and the automatic control system. A stochastic Petri net is used to model each subsystem where firing times or probabilities are assigned to each of the transitions. Once the individual subsystems are modelled interaction between the subsystems are identified, for instance, a red light from the control system directs the traffic flow to stop. These interactions are used to connect the subsystems into a network that represent the whole system. The model aims to represent the position of the cars and the trains and evaluates the risk presented by them both being in the same section at the same time.

The firing times in the Petri net are determined in one of three ways. Firstly, they can be known set values, such as an instantaneous transition. Alternatively, a truncated normal distribution can be used if the value for transition time is known to fluctuate around an average, for example the time that the train arrives. Finally, the transition time can be modelled by a suitable statistical distribution that fits with data gathered, for instance the traffic flow. Monte Carlo simulation is carried out to evaluate the

Petri net. This predicts when the trains and cars will be in the same section at the same time. The assumption is made that the control system does not fail.

Coloured Petri Nets allow a more concise representation of model structure and are proposed to model the safety of signalling systems where more than one train is in the same track section, by She, Zhao and Yang, 2014 [63]. Here, there are 3 conditions for the system to be in a safe state and a Petri net model is built for each. The first condition is that the system must function correctly when there are no faults, the second condition is that the system must be safe in the case of random or systematic internal faults and the third is that the system must function safely under external influence. Coloured Petri nets follow the same principles as Petri nets but tokens also contain information, this enables a more concise model as different tokens can move through the same Petri net simultaneously and retain their individual meaning. In the model an extra place in introduced, known as a counter place. This place is linked to transitions that represent faults in the system, it limits the number of faults that the system can experience. The counter place can also represent the influence of an external condition leading to a fault. System states corresponding to hazards are represented by the makings of certain places in the coloured Petri net, this method finds the reachability of each of these system states under different conditions giving an analytical solution. There is no measure of uncertainty carried through the model.

In addition, Petri Nets have been applied for asset management modelling. The paper by Andrews, 2012 [64] considers the state of the track ballast and how this changes over time. Poor track geometry can lead to faster aging of the other assets, such as rails and sleepers, as well as a geometry failure resulting in a derailment. The model incorporates a non-constant rate of change between asset states. There are several methods for maintenance of track ballast included in the model, these include: tamping, stone blowing or manual intervention. This maintenance does not always return the ballast to its original state as some of the ballast can be destroyed in maintenance activity hence the condition of the ballast is dependent on the previous maintenance activity as well as its age. Due to this it is important to optimise the condition of the ballast so that the resources used for maintenance or renewal can be as effective as possible.

To model ballast degradation and maintenance, data is first gathered that represents the deterioration rate of the ballast and also the times at which different repair strategies are used. The data is split into phases that occur between each maintenance activity. The data is then analysed to show trends and a Weibull distribution is used to fit a curve to the data. A Petri net is then used to model the system which is solved by Monte Carlo simulation. There are three transitions introduced to the Petri net model for this system: the reset transition, the conditional transition and the convolution transition. The reset transition returns the Petri net to a specific state, for example after a maintenance activity. The conditional transition is dependent on the number of tokens in a different part of the network, for instance ballast degradation is dependent on the past maintenance activities. The convolution transition is used if the transition times are related to the same base condition. The firing times are sampled at random from the distributions that have been taken from the data and as more runs are completed the average results should start to converge. Whole life costs can be calculated and the maintenance strategy can be optimised to balance maintenance and renewal strategies with condition of the ballast. This method allows non-constant failure rates and multi-state components, therefore is applicable for ageing components with non-constant failure rates. However, this method requires custom made software to analyse each Petri net due to the custom transitions.

The literature reviewed in this section has demonstrated the flexibility of a Petri Net modelling approach, especially in application to complex degradation and maintenance processes on the component and sub-system level. There is also flexibility in the method surrounding the addition of model specific transition types. However, the size of the Petri Net model can quickly become large resulting in an inefficient and computationally expensive analysis by Monte Carlo simulation. For network level large systems, this may make the Petri Net method difficult to apply. The use of Coloured Petri Nets can reduce the size of large models with repeated modules. At the current time, there are limited software packages for solving Petri Net models, especially when a high level of customisation is required.

2.2.5: Markov Models

Markov models have been applied for the optimisation of inspection and maintenance procedures, with the consideration of system safety, in the paper by Podofillini, Zio and Vatn, 2006 [65]. The approach presented here considers crack formation in track rails that can result in rail breakage. To prevent rail breakage, track is inspected periodically by ultrasonic measurement cars to detect cracks in the rails. The aim of the method presented in the paper is to find an optimal strategy for use of the ultrasonic measurement cars, along with an accurate picture of risk and a method for testing different maintenance strategies. A crack in the rail is considered to have several states, where the maintenance activity required depends on the state. A crack will not be immediately detectable but as the crack worsens the likelihood of detection increases. In the model, there is the option for opportunistic maintenance where a crack in a non-critical state, where it is monitored for a period of time with the expectation that other similar cracks will develop. In this paper a non-homogeneous Markov model is used. The inspection is considered periodic, at a set time interval. The state of the crack is split into discreet phases and transition rates between the phases are assigned, to fit with data. In this model, in order for a crack to develop into a rail break, it must be undetected or have falsely identified severity. The probability of non-detection is dependent on the state of the crack and systematic failures. The probability of misidentification of the severity is taken from expert opinion. Transition matrices are defined which describe the degradation, inspection and maintenance processes. Common cause failures are then identified, such as the systematic miscalibration of the ultrasonic measurement car. A Genetic Algorithm is used to optimise the model for both cost and safety to find a relationship between inspection intervals and maintenance waiting times, and the risk of derailment. Uncertainty is not considered in this model. The approach applied in this paper uses a fixed inspection interval which does not allow a time dependent maintenance strategy.

In addition, a prediction of the condition between maintenance actions can be made using a Markov-based model. The paper by Bai, Liu, Sun, Wang and Xu, 2015, models the deterioration of track maintenance units, of 200m in length, where irregularities in the track can arise between maintenance actions [66]. In the model the track quality index, which is the sum of standard deviations of local geometry parameters, is used to quantify track irregularity for each section under consideration. When the track quality index reaches a specific threshold, the condition is deemed unacceptable. In the paper the track is considered in 1km sections, with 5 track maintenance units in each. This allows track section level maintenance of either: no global action, planned global action or priority global action across the units in the section. A Markov model is used to model to deterioration process between inspection actions, where the state at each subsequent time is predicted from the previous state. The model assumes exponential track degradation at a gradual rate, however, heterogeneous factors are included in the modelling of track irregularity development, including: the structure coefficient (the proportion of curve in the section) and gross tonnage. The inclusion of heterogeneous factors results in a random and uncertain state change in the Markov model used to predict track irregularities. The log-likelihood function is used to estimate the Markov transition probabilities. The method provides the condition prediction between maintenance actions. A large base of data is required for this method.

In order to consider hazard states for a railway system with a Markov model, Restel and Zajac, 2015, presents a methodology [67]. Several system states are defined including various working states with disruptions and traffic, unavailable states including those due to maintenance, accidents or serious accidents, and hazard states where the system is in use despite failures. The model was trained with synthetic data created from theoretical timetables, expert opinion and operational data. The model outputs show a good agreement when compared to this synthetic data, but the model is not validated with a test set of data. The model solely considers state changes based on data available and so does not provide a framework for testing underlying contributions to the states within the model. Also, an exponential distribution is assumed for state transitions within the model.

In addition, a Markov model has been demonstrated to give predictions of future track condition, given current maintenance strategies. The paper by Prescott and Andrews, 2015, employs a Markov model for the track geometry condition, and maintenance, over time [68]. Within this model is the possibility for maintenance actions to impact the future degradation of the ballast. In the model it is assumed that track geometry is measured over a 1/8th mile track section, with varying maintenance actions over time for each track section. The degradation of the track geometry is modelled with revealed and unrevealed states in a Markov model module, with an individual module of this type created following every maintenance action. Here, the degradation rates used in each repeated module depend on the history of maintenance actions for the track section. There are four track geometry condition classifications included in the model: a good condition, a critical condition where maintenance is required, a condition where speed restrictions are required and a condition where line closure is required. Application of the proposed approach results in a set of differential equations which are solved numerically using the fourth order Runge-Kutta algorithm. For the application presented in the paper, 80 differential equations are generated. The track geometry degradation rates, used in the model, are taken from the reciprocal of the time taken to reach each degraded state, and so are assumed constant within each state. This may not be the case for aging systems. Different asset management strategies are tested in the application given in the paper. These strategies include: the level of degradation that triggers track maintenance, the mean time to perform normal maintenance, the inspection interval and the renewal period. The model can be extended to consider hazardous states by considering the time that the track is in a state where there are unknown speed restrictions, or line closures, required. The method results in a large model, especially if complex processes are required, due to the repeated structure following each maintenance action. This could lead to difficulties with analysis of the resulting model.

The generalisation of a Markov model application to a network level problem has also been demonstrated in the paper by Yang and Frangopol, 2018. A method is provided to rank the maintenance priority of bridges, based on the financial risk associated with their structural degradation [69]. The method combines a reliability analysis, in the form of a Markov model, with consequence modelling for bridge failures. The degradation of the bridges is modelled with a Markov chain approach. The transition probabilities for this are determined from historic data, which is updated using random field theory to consider the spacial correlation of bridge failures across the network. The consequences modelling includes individual bridge cost and incorporates network analysis to consider the indirect impact of a bridge failure on the whole system. This indirect network level impact includes the extra travel time, extra travel distance and increased travel cost, for all users across the network, due to a bridge failure. A Monte Carlo simulation method is used to predict the potential consequences of a bridge failure. The bridges in the network are ranked based on their financial risk of failure. The method allows the network impact of bridge failure to be considered, alongside special correlation of bridge failures due to potential underlying factors such as loading and weather impacts. However, the method relies on the accuracy of historic data for reasonable model predictions. Also, the impact of different maintenance actions on the network is not modelled. Hence, this method does not provide a framework for the optimisation of different asset management strategies.

Markov Models provide an efficient framework for modelling the state of components and associated inspection and maintenance strategies. However, the state transitions are usually limited to cases where there is a constant failure rate, such that an exponential distribution of their residence times is assumed. In addition, a resulting state is predicted from the previous state only, which can make the consideration of underlying dependencies difficult.

2.2.6: Summary

This section has given a review of methods applied in literature for modelling risk, or system failure. The review is not limited to railway systems. This gives relevant information on the benefits and drawbacks of different existing modelling approaches, in order to inform the methodology steps applied in the remainder of this thesis.

There are weaknesses within each of the different methodologies. With the Fault tree approach extensions to the traditional method have been proposed to address issues with sequencing of events, time dependencies and dependencies between events. However, a constant failure rate is assumed, and modelling the maintenance of the system is limited. Markov models also assume a constant failure rate, and struggle to model underlying dependencies. Dynamic Event Trees are proposed for considering time dependence, event sequencing and component dependencies but these models quickly grow in size making analysis difficult. Bayesian Networks can be applied to combine independent systems. Finally, Petri nets have flexibility when modelling different failure rates and complex asset management strategies. However, larger models can have a large computational cost for quantitative analysis if they are solved using Monte Carlo simulation. In addition, uncertainty is not included.

This review informs the choice of methods for the development of the combined modelling approach applied in the remainder of this thesis. This is discussed further in Section 3.5 of this thesis, where a combined Petri net, Fault Tree and Event Tree approach is proposed. This proposed approach is demonstrated with two models across Chapter 4, Chapter 5 and Chapter 6. In addition, this section informs areas for further research such as the inclusion of uncertainty, addressed in Chapter 6, and efficiency of model simulation, addressed in Chapter 7.

2.3: Asset Management Optimisation

This section provides a review of the work available in literature for the optimisation of maintenance of a system. There are examples in literature of optimisation approaches for asset management strategies, with a focus of cost, and examples where the optimisation also considers risk, condition or failures within the system. In this section, a review of methods where cost is the focus is given first, followed by a review of methods where risk, condition or failures within the system are included with cost considerations.

2.3.1: Optimisation for cost

The paper by Dekker, 1996, provides a review of early works on maintenance optimisation models and outlines a framework for future optimisation methods [70]. The framework suggested involves initial identification and definition of the system, followed by modelling of the system to provide a range of possible maintenance options to decision makers, alongside a representation of their suitability. The onus is then on the decision makers to choose the most appropriate strategy. The paper states that generic methods for modelling the system can fall into deterministic and stochastic categories. The optimisation frameworks reviewed assume that the objective function of the optimisation follows a set of equations that can be solved numerically. This approach may not be applicable to complex processes, for instance, systems with incomplete repair or sub-system dependencies, due to difficulties in representing the processes in a mathematical equation form. The approach also has a heavy reliance on historic data. In addition, decision making processes for maintenance optimisation are assumed post-analysis, as opposed to part of an integrated asset management decision making tool.

A qualitative approach to asset management optimisation, with some quantification of system degradation, is presented in the paper by Rausand, 1998. The methodology presented in the paper aims to reduce the maintenance cost of a system by focusing maintenance resources on key system areas, and by removing any unnecessary maintenance actions [71]. The approach considers a system in operation with the aim of reducing personnel injuries, environmental damage, production loss and material damage. Data is collected for the system and a distribution modelling the degradation of the system is defined. The Weibull distribution is stated as the preferred distribution in most cases. Analysis is completed using an FMECA to find the reliability of the system. Various maintenance actions are then selected based on their suitability, governed by a set of defined qualitative rules detailing their impact on the dominant failure modes found via the FMECA. Maintenance intervals are tuned while the system is in operation through a trial and error approach. The approach does not tackle complex system level maintenance strategies, such as opportunistic maintenance, and leaves much of the decision making to the user, with the rule based selection of different maintenance actions. The trial and error approach to maintenance intervals does not allow the testing of different strategies prior to implementation of the chosen strategy.

Optimisation of a system with a stochastic degradation is considered in the paper by Grall, Bérenguer and Dieulle, 2002 [72]. However, the models only consider the maintenance and inspection for a single unit system with a stochastic degradation governed by a gamma distribution. This degradation model is applicable to components in their useful life phase, where the deterioration rate between two consecutive times can be assumed constant. Maintenance is assumed to be condition-based such that the system is repaired on failure, or when it reaches a critical threshold, identified through inspection. The method aims to optimise the critical threshold and the inspection interval of the system in question. There are several assumptions made within the methodology: maintenance returns the component to the perfect state, inspection always successfully reveals the state and failures are immediately detected. However, the method does allow for non-periodic inspection. The system running costs are represented by mathematical equations that are minimised through numeric integration. This may not be applicable to multi-unit systems or those with complex maintenance and inspection processes, or to systems that have complex degradation processes, such as those with components operating past the end of their useful life.

Tabu Search is combined with a Genetic Algorithm in a methodology proposed by Di, Si and Ze, 2012, which is applied for optimization of a scheduling Petri net [73]. Here, each proposed solution is screened via a Tabu Search before inclusion into a Genetic Algorithm. The optimization methodology is applied to a simple Petri net to demonstrate the searching capability.

Simulated Annealing algorithms have also been applied to Petri net models. Zimmerman, Rodriguez and Silva, 2001, applied a Simulated Annealing algorithm to optimise a Petri net modelling a manufacturing system [74]. The optimisation is applied to maximise the profit by allocating resources in an optimal way. A two-phase optimization strategy is also applied in order to decrease the computational cost of the optimisation method. Here, the approximate solution is found by numerically calculating upper and lower bounds on the throughput of the transitions in the Petri net, along with the mean number of tokens in the steady state. This approximation is then used as the basis for a more thorough search with a Simulated Annealing algorithm. This two-stage approach saves computational time but the approximation method is currently intractable for the complex Petri net modelling, with additional extensions, used in this thesis.

Another application of a Simulated Annealing algorithm is given in the paper by Jain, Swarnkar and Tiwari, 2003 [75]. The authors present an optimisation method applied in conjunction with a stochastic Petri net model. In the approach, a Simulated Annealing algorithm is applied to optimise policies for fabrication. The mean cycle time and tardiness are used as performance measures in a scalar objective function for the optimisation. In this paper the Simulated Annealing algorithm and Petri net modelling approaches are applied to the problem separately, as opposed to combined in a tool to integrate the modelling and scheduling of the system. However, the results of Simulated

Annealing algorithm showed good performance in comparison to those gained by rule based scheduling.

2.3.2: Optimisation including risk, condition or failure

In addition, there are works available on the optimisation of safety risk, reliability or condition of a system. Safety risk based optimisation is explored in the paper by Apeland and Aven, 2000, who uses a Bayesian and semi-Bayesian approach to classify the risk of a critical component failure [76]. A semi-Bayesian approach combines a classical assumption of the existence of a true value with a Bayesian representation of model inputs. The method allows uncertainties in the data, used to make predictions, to be included in the analysis. This is suggested for situations where a lack of data makes a classical approach difficult to implement and potentially inaccurate. In the method the Bayesian, or semi-Bayesian, representation of the critical component failure is propagated through a risk analysis using a Fault Tree and Event Tree approach. The process is repeated with alternative sets of strategies for the critical component, to predict the risk of the system under each different strategy. The method allows the decision maker to choose the best strategy based on this analysis. The method incorporates uncertainty which can aid in situations where data is rare, and the risk analysis is based on expert opinion. However, the method only considers independent strategies for critical components, as opposed to system level maintenance strategies or strategy optimisation over multiple components.

Arunraj and Maiti, 2007, gives a review of risk based maintenance methods that aim to maximise availability and efficiency of a system, by controlling the rate of deterioration and minimising the total cost of operation [77]. Several risk analysis methodologies are presented including qualitative and quantitative methods, with a common difficulty identified across methods in incorporating uncertainty. There is no recommendation made as to the most suitable method for risk analysis. The risk based maintenance methods reviewed follow a framework whereby the system risk is calculated via a risk analysis approach, reverse Fault Tree analysis is completed to back-propagate the risk from the system level to each component level and then the component strategy is adjusted to give the required risk. This approach is difficult to implement for complex systems due to the inherent flexibility in the back-propagation of risk values that is included in the method. In addition, the strategies for each component are treated independently, with no modelling of maintenance strategies applied on a system level. The approach requires a reliable risk modelling methodology and the correct identification of the system level Fault Trees. In addition, the back propagation of the failure probabilities to a component level is open to interpretation of the user.

The maintenance of series system of non-homogeneous components is optimised by Faddoul, Raphael and Chateauneuf, 2018, with the consideration of reliability constraints [78]. In the modelling approach, all components must be connected in a linear manner such that any component failure can cause a system failure. A Lagrangian relaxation technique is used to split the system into smaller sub-modules. In the method, the assumption is made that the reliability of the system is equivalent to the product of each of the component reliabilities. This allows the logarithm of the system reliability to be taken, to express it in terms of the summation of the logarithms of each component reliability. Each component is modelled with a Markov chain, with several states for each component. It is assumed that there is a dependence between the component state and its reliability, but that no other dependencies exist. An objective function is minimised subject to: the cost of the system under a given strategy minus the weighted sum of the logarithm of each component reliability under the given strategy. In this way, a high failure probability penalises the objective function. The optimal strategy given the weightings of each reliability are found through dynamic programming. The optimisation allows for a choice between several intervention actions and allows the maintenance to be optimised within different time periods of the system. The series system assumption is core to the method as it allows the reliability to be expressed explicitly in terms of each component. For safety critical systems, components are often in standby and so this method is not directly applicable.

Yang, Remenyte-Prescott and Andrews, 2015, [79] use a multi objective Genetic Algorithm, NSGA-II, to find a maintenance and rehabilitation strategy to minimise cost and maximise road condition within a time period. The optimization technique is implemented in both a simple deterministic and probabilistic model; 80 different variables were optimized in the application demonstrated in the paper.

The thesis titled, ‘Modelling railway bridge asset management’ by Le, 2014, presents an optimisation framework using a Markov and Petri net bridge model [80]. The optimisation of the Petri net model is completed based on the results obtained by the optimisation of the Markov model, to reduce the search space. Several variables were optimised including: the inspection period, if opportunistic maintenance is enabled, the maintenance schedule, intervention options and servicing. In addition to this the minor repair, major repair and renewal delay time was included. A Genetic Algorithm is applied in this case for optimisation.

A further approach is given by Yianni, 2017, in the thesis titled ‘A Modelling Approach to Railway Bridge Asset Management‘. The work presents an optimisation approach whereby a Genetic Algorithm is applied to a Petri net model for railway bridge asset management [81]. A ‘surrogate model’ is introduced that is used to find the approximate region of interest followed by a fine tuning effect of optimization on the full model. This ‘surrogate model’ is found by converting transitions to a single type to speed up simulation via a GPGPU and by simplifying some of the structure in the full model, such as removing branching behaviour. This results in different model outputs that are similar enough to be used in a 2-level Genetic Algorithm approach. The inspection procedure for the model is optimised, where the inspection frequency can have one of 3 values depending on condition.

Su and De Schutter, 2018, present a method to optimise the maintenance scheduling for a network where there is one or more available maintenance teams [82]. The network is split into segment, and each segment is assigned a probability distribution to govern condition and a measure of importance, for example the number of trains that pass over the section or the total tonnage. The maintenance scheduling of the network, given these quantifications of the segments, is optimised. The objective function for the optimisation includes a penalty for operating unsafe sections, a penalty associated with the loss of use of components that are replaced while still in their useful life, the component maintenance cost and maintenance team travel cost. The paper uses an enhanced Genetic Algorithm with roulette wheel selection and 200 members in each population. The enhancements of the Genetic Algorithm include: an initial population that is uniformly distributed, a variation operator and an elitist strategy to ensure that the best solution is not discarded and evolution is always based on the best solutions. These enhancements improve the results given by the algorithm. The approach does not have the capability to model complex maintenance strategies or inspection frequencies of the network sections.

A Genetic Algorithm to optimise the design of a deluge system subject to system performance parameters such as unavailability, lifecycle costs and spurious trip occurrence in work by Andrews and Bartlett, 2003 [83]. A Fault Tree model is used to give the system unavailability, which is analysed with a Binary Decision Diagram. A House Event is constructed for each possible design alternative. The optimisation aims to reduce the unavailability of the system within defined constraints. Constraints are placed on the lifecycle cost, cost of system testing, and the cost of preventative maintenance or corrective maintenance. If these constraints are violated during the optimisation then a penalty is applied. In addition, a spurious trip penalty is included. The optimisation of the system design aims to find optimal solutions for the number of each component, the choice of component and materials, and the maintenance test interval, and for some components, the quantity of preventative maintenance. An initial example of the method is given in the paper ‘Optimal safety system performance’ [84]. In this earlier paper, the method is applied to a simple high pressure protection system.

Accident cost is considered in an approach by Podofillini, Zio and Vatn, 2006 [85]. The paper models defects of railway track in two stages. In the first stage the defect becomes detectable and in the second state failure occurs due to the defect. A non-homogeneous Markov model is used to predict the failure probability of a rail section, with different maintenance and inspection strategies, under the assumption of periodic inspection. The model is analysed through numerical integration. A multi-objective Genetic Algorithm is used to optimise the inspection interval for the track and the time delay for maintenance of non-critical detected track defects. The optimisation objective function is based on the predicted maintenance and accident costs of the model.

2.3.3 Summary

This section has given a review of system optimisation methods available in literature. The first examples give methods that consider the cost of the system operation. The second examples include the cost with some measure of the system state, such as risk, reliability or accident cost. One of the objectives of this thesis is to develop a risk based asset management tool.

One identified gap in the literature is an applied optimisation process that combines safety risk and life-cycle cost for a Petri net based model. Secondly, this project aims to model complex asset management strategies. This includes phased asset management strategies, whereby different strategies are applied to the system depending on its age. The optimisation of phased strategies has been identified as a novel area for research, based on this review.

This review also informs the choice of methods for application to a phased optimisation problem. Across the literature reviewed in the section Genetic Algorithms are demonstrated to show good optimisation results where there are numerous parameters to be optimised. They also demonstrate a high level of flexibility. The Simulated Annealing algorithm has also been used with some success and can be seen as a more efficient search tool where there are fewer parameters to be optimised. Based on this review of literature, a combined Simulated Annealing and Genetic Algorithm method is proposed for a phased optimisation of the system. This approach is presented in Chapter 6.

2.4: Modelling Uncertainty

This section gives a review of methods present in literature for the incorporation of uncertainty in model outputs. This review is presented to align with the objectives of the project, whereby an estimate of the uncertainty on the risk values output from the model should be explored.

2.4.1 Modelling Uncertainty Literature Review

Arunraj, Mandal and Maiti, 2013, consider stochastic uncertainty and subjective uncertainty when modelling risk [86]. Stochastic uncertainty is attributed to the random nature of failures. Subjective uncertainty is due to factors such as a lack of knowledge, measurement error and subjective judgement. The paper proposes a method whereby the likelihoods of failure are represented as fuzzy numbers. Monte Carlo analysis is used to generate fuzzy cumulative distribution functions for failure probability, using different α-cut values. The DSW algorithm is used to combine these functions with fuzzy consequences of failure to give a representation of the risk and the associated uncertainty. The method is not applied within a Petri net framework and instead considers the combination of several distributions.

A method to combine Monte Carlo simulation with fuzzy calculus to allow the use of both possibilistic and probabilistic measures of uncertainty in the same computation of risk, is proposed by Guyonnet et al., 2003 [87]. The method uses a double loop approach with the probabilistic measure sampled in the outer loop and the α-cut cuts taken in the inner loop. The smallest and largest risk values are assigned to the upper and lower limits, within each iteration. The paper suggests the use of a minimisation or maximisation algorithm within the inner loop, when the functions are complex or not monotonic, however no algorithm is suggested.

There have been several works that have aimed to consider the impact of uncertain input values on the output values of a Petri net. Fuzzy Set theory can be used to consider the impact of the uncertainty of the input parameters on the final uncertainty presented by the Petri net. An overview of applications of Fuzzy Set theory to Petri nets is given by Cardoso, Valette and Dubois, 1996, and Fuzzy logic has been applied to estimate the uncertainty by Sadeghi, Fayek and Pedrycz, 2010[88] [89].

In an attempt to address issues when applying Event Tree analysis with limited quantitative data, the paper by Kenarangui, 1991, proposes a method to represent qualitative possibilities within an Event Tree with a fuzzy set [90]. In the proposed method, assigned fuzzy possibilities are propagated through the Event Tree to give fuzzy possibilities of each outcome. The fuzzy sets are defined based on a choice of qualitative categories, for example, the probability of a component failure may be ranked as high, medium or low. The consequences of each outcome, in the Event Tree, can also be estimated to give fuzzy sets for the risk of each outcome, which can be combined to give the risk of the initiating event under consideration. The fuzzy risk predictions for each outcome can be ranked using a maximising set. The method presented in this paper allows detailed Event Tree analysis and risk ranking in situations where there is limited data available for quantitative analysis through traditional quantitative methods. This is particularly applicable in cases where the analysis is based on expert opinion. There is reliance on the user to quantify the ranges assigned to each qualitative category and on the definition of each fuzzy set. This framework incorporates a representation of uncertainty in the knowledge used to analyse the Event Tree and does not assume the fixed values required for traditional quantitative Event Tree analysis.

Furthermore, Baraldi and Zio, 2008, present an attempt to combine possibilistic uncertainty measures, such as those presented in fuzzy set theory, and probabilistic measures of uncertainty, that may co-exist within the set of input parameters of Event Tree model [91]. Probabilistic representations are suggested for cases where there is sufficient data to create a probability density function, to represent the probability of each event, such as events considering component failures. Possibilistic representations, incorporating a fuzzy set method, are suggested for cases where there is scarce quantitative data available and estimations are based on expert opinion, such as in cases where events are rare or due to human errors. The method uses a Monte Carlo simulation and the extension principle of fuzzy set theory for analysis of the Event Tree. Here, a sample from the probabilistic representation is taken, alpha-cuts are used to select from the possibility distributions and these are combined through fuzzy interval analysis. This process is repeated for each alpha cut, before returning to the first step and repeating for a new sample from the probabilistic representation, hence, forming a double loop. The process returns a probability weighted average of the possibility measures associated with each fuzzy output interval. The process allows the likelihood of an Event Tree output value passing a certain threshold to be calculated. However, there is some difficulty in accurately estimating the confidence intervals of the Event Tree outputs. In addition, the method relies on a double loop analysis which could be computationally expensive for a large Event Tree. In this method, it is assumed that there is an independence between events represented by probabilistic measures. It is also assumed that the possibilistic representations all have the same confidence level, representing a strong dependence between information sources for these variables.

2.4.2: Summary

This review has presented methods in literature for modelling the uncertainty on a Petri net model output, or an Event Tree model output. A review of these specific methods has been completed because these methods are the ones implemented in this research. The Fault Tree method is also used in this thesis; however this is in the capacity of informing the logic of the system as an initial step, as opposed to modelling the system risk.

The modelling of the systems in this thesis is largely completed within a Petri net framework. For considering uncertain model inputs, this results in a looping behaviour sampling different values for the model inputs, in order to estimate the uncertainty on the outputs. One gap identified by this review is the use of a suitable algorithm to improve the efficiency of this approach, for complex systems.

This informs the research, presented in Chapter 6, where a novel approach for approximating uncertainty on model outputs is proposed. Here, a Simulated Annealing algorithm is implemented to approximate the uncertainty on model outputs due to Monte Carlo simulation of the model, and due to uncertain model inputs.

2.5 Model Reduction

This section gives a review of methods for reducing Petri net model complexity, in order to improve the efficiency of simulating a large model structure. The challenge of computational efficiency for model simulation was identified in Section 2.2.4 of this literature review. This review of model reduction is included here as it informs the work presented in Chapter 7 of this thesis, where an approach is proposed for the reduction of model size.

2.5.1 Model Reduction Literature Review

There have been a number of studies into the reduction of Petri net complexity. The methods presented in literature tend to define a set of rules to reduce specific sub-structures commonly found in Petri nets. Table 2.1 gives a summary of examples of alternative methodologies for the reduction of Petri net complexity. In the examples in this table several approaches have been used which fall into the following categories:

1) Comparison between Petri nets (or Petri nets and sequences) based on a sequence of events. These sequences of events are generated by the firing occurrence of highlighted transitions.
2) Comparison of signal outputs of Petri net structure with data signals, and using this to approximate parameters in the Petri net.
3) Folding of Petri nets whereby limited rules are specified for the stepwise reduction of Petri nets.
4) Replacement of self-contained sections within a Petri net to reduce the model size.
5) Decomposition of Petri nets by splitting the Petri net into sub-Petri nets and then reducing, if possible, following set rules.

Notably the paper "To aggregate or to eliminate? Optimal model simplification for improved process performance prediction" gives a methodology whereby the reduced structure of the Petri net is selected via an optimization algorithm, that considers the error of each reduction and the improved computational cost [92]. Specific reduction rules, known as foldings, are developed and the time for the transitions replacing these foldings is pre-assigned. This approach has the benefit of automatically producing the structure of the reduced Petri net while defining the parameters governing the transition firing times.

Method	**Comparison or Reduction?**	**Summary**	**Benefits**	**Limitations**	**Papers**
Discovering Petri nets from event logs	Comparison of Stochastic Petri net with event log.	The method considers the firing of key transitions for comparison with event logs.	Can be used to focus on the key events, rather than all transitions. Allows for finding firing times for non-exponential distributions.	Considers a simple Petri net structure and relies on an event log that is 'linear'.	[93]
Identification of time parameters from event sequences	Comparison of a Stochastic Petri net with an event sequence.	Presents a method for obtaining the time parameters for two normal transitions in conflict to match a pre-known event sequence.	The optimization algorithm shows good convergence for simple event sequences.	Assumes a normal distribution for the distributions, limited application due to the need for an event sequence. A large event sequence leads to a low optimization efficiency.	[94]
Reconstructing Petri net model parameters from data	Comparison of data to Stochastic Petri nets.	Considers the output of an infrequent signal in comparison to a Petri net model output.	Presents an algorithm that allows the unknown parameters to be approximated.	Only allows for the fitting of unknown constant probabilities in simple models.	[95]
'Foldings' to reduce Petri net structure	Reduction of Petri nets following specified rules.	A set of rules are given whereby the Petri net can be reduced by grouping transitions together.	Allows for the easy reduction of some Petri nets as the reduced firing times are already defined.	There are limited applications due to the assumptions required, the method also does not quantify the error introduced through the reduction and the method is less reliable when applied to large Petri nets.	[96][97]
Optimal 'folding' to reduce Petri net structure	Optimal reduction of Petri nets following specified rules.	A set of rules are given for the reduction of the Petri net by grouping transitions, an error and computational improvement is assigned to each reduction and these are optimised.	Provides a method for automatically reducing the structure of Petri net. A measure of the error of the new approximation is given.	Times for the transitions in the reduced Petri net are given empirically so may not reproduce the best result for the new structure. Petri nets can only be reduced via the presented 'foldings'.	[92]
Section removal and reduction of Petri nets	Reduction of Petri nets by removing self-contained sections.	Uses a semi-Markov method to reduce a section of a Petri net.	Presents a method for reducing the size of the Petri net.	This method relies on exponential transition firing times. There is no description on how to assign firing times or the increase in performance.	[98]
Decomposition of Petri nets by splitting	Reduction of the Petri net following splitting into sub-nets.	The Petri net is split into sub-nets. These are either reduced and solved in isolation or reduced and recombined before solving.	The splitting of the Petri nets leads to a faster solution, also if the sub-nets are recombined their parameters can be tuned.	The manual reduction of the sub-net structure is limited to established rules. There can be large dependencies between sub-nets resulting in a poor approximation due to the splitting of the Petri net.	[99] [100]

Table 2.1: A review of Petri net reduction and comparison methods

2.5.2: Summary

This review has presented methods that have been implemented in literature to reduce Petri net model structure, and hence improve the efficiency of simulating large models, as outlined as an objective of this project.

There are gaps present in the research surrounding this topic for cases where reduction is required but there are not specific sub-structures present in the Petri net. This is because reduction methods are commonly rule based. Alternative approaches suggest splitting the model into independent sub-structures. However, this removes any modelling of existing dependencies.

This review informs the research direction of this thesis by highlighting an area for further research, namely the development of a reduction method that is flexible for different model structures and governing distributions. The novel approach proposed in Chapter 7 defines the structure of the reduced Petri net and allows the parameters governing the transitions within it to be automatically discovered. This approach has been chosen as it does not limit the reduction of the Petri net to specific structure rules or distributions governing the transition delay times.

2.6: Specific System Modelling Reviews

As discussed in Section 1.6 of this thesis, two systems are used for novel model development, and demonstration of the methodology proposed in Section 3.5 of this thesis. The first model is applied to a railway S&C and the second model is applied to an underground fire protection system. This section provides a review of current modelling approaches and research for these systems, to highlight areas where the specific models developed in this thesis can make improvements on the state of the art for each system.

2.6.1: S&C Review

Liu, Saat and Barkan, 2012, perform a review of the major causes of train derailment from safety data available in the Track Safety Data Base [101]. Derailments were found to be the most common type of accident, of which derailments over S&C were a contributing factor. Also, an interaction was identified between derailment speed, and the frequency and consequences of the derailment. The paper recommends the development of an integrated framework to optimize cost and minimize risk.

Dindar and Kaewunruen, 2018, looked in more detail at immediate, casual and contributory factors to derailment at an S&C [102]. The study identifies several causes of derailments at S&C, these include infrastructure failures, interaction failures such as obstruction and flange climb, environmental factors, operational factors, malicious action and human causes such as over speeding and vandalism. Data was taken from the Rail Accident Investigation Branch data source. The data indicates that if there is a larger number of trains, or trains are traveling at a higher speed, then there can be a higher level of derailment occurrence and a greater severity of consequences. Derailments were the most common train accident type in the UK, with an S&C infrastructure failure contributing most to observed derailments, causing 39% of all derailments for the period of 2006-2016 in the UK. Of the S&C infrastructure failures, 56% were attributed to switch defects, 17% were attributed to stretcher bar failure, 11% were attributed to geometry failures, 11% were attributed to broken rails and 6% were attributed to poor ballast condition.

Contributory and casual factors were also analysed. Of the derailments, 20% were deemed to have had inadequate maintenance, or inspection, to the level where it was classified as a contributory factor and 32.6% had maintenance deficiencies to the level of classification as a casual factor. Casual factors are defined as an omission that if corrected would prevent the fatality, for example, false inspection. Contributory factors are defined as a behaviour that sets the stage for an accident, for example, poor maintenance culture.

For S&C, maintenance action priority is ranked in work by Ishak, Dindar and Kaewunruen, 2016, based on the operational risk of each associated failure mode [103]. In addition, the paper states that dynamic wheel-rail interaction and imperfect contact can cause detrimental impact loads, leading to a higher rate of deterioration. The paper also states that S&C derailments contribute to almost half of the track system failure fatalities and weighted injuries in the UK, for the years 2009-2014. The study introduces a risk-based maintenance approach to assess the rail degradation process. Factors influencing the degradation process were identified. Following this, hazard identification, consequence prediction, likelihood estimation, risk assessment, risk acceptance, residual risk monitoring, maintenance planning and recovery and contingency planning were carried out. There were several approaches used for these tasks, including: Event Tree analysis, Fault Tree analysis, reverse fault analysis, expert opinion and uncertainty analysis. A summary of the methods used can be found in the paper. The operational risk of each failure mode and the safety risk of each failure mode was given a ranking. Maintenance priority was assigned based on these rankings in a qualitative manner.

Zwanenburg, 2007, considered both the maintenance of S&C and conditions that can lead to a derailment [104]. The paper states that on plain track, a wheel can pass over a broken rail and not necessarily lead to a derailment, however, a broken switch rail will lead to a direct derailment. The study also suggests that degradation of S&C components depends on many properties. These include train properties such as: axle loads, train speed, total load and train condition. Additionally, track properties contribute to degradation. Track properties include: condition of the sub-base, materials used, quality of installation, track geometry and track condition. It is expected that track material and track geometry interact, with one in a poor condition likely to lead to the faster degradation of the other.

The paper also states that components are frequently replaced prior to reaching the end of their useful life. This is attributed to over-cautious inspection, timetabling restrictions and equipment or personal availability. Financial losses can be made through this early replacement. The most commonly replaced individual components identified in the study are the switch rails, crossing nose, check rails, intermediate rails, sleepers and ballast. Identified maintenance processes for the S&C include welding, along with rail grinding and tamping.

The thesis ‘Failure analysis of railway switches and crossings for the purpose of preventive maintenance’, by Hassankiadeh, 2011, collated the total number of failed S&C components in 2009 in the UK [105]. Switch rails accounted for 45% of the failed components and slide chairs accounted for 30% of the failed components. Ballast, Schiwag Roller, Stretcher bars and Stock rails made up 20.9% of failed components together. Crossing, Fishplate, Back drive, Sleepers and Spacer Blocks comprised less than 4% of failed components combined.

The circumstances of the failure modes of the S&C were also collated in the thesis. Obstructed switch corresponded to 40% of failures, dry chairs contributed to 17% of failures, cracks and breaks contributed to 9% of failures and poor ballast condition contributed to 7% of failures. Contamination, plastic deformation and adjustment each contributed to 5% of the failures, followed by wear and missing fastenings which each contributed to 3% of the failures. The thesis highlighted the impact of weather on S&C failure, with more failures observed in autumn and winter.

An FMEA was carried out to rank S&C failure causes by a risk priority number. The ranking resulted in the following in descending order of priority: switch obstruction, dry chairs, cracks and breaks, voiding ballast, adjustment, contamination (leaves), plastic deformation, wear, missing nuts, squat and rolling contact fatigue, creep, track gauge variation and finally wet bed. Depending on the risk priority number the components can be grouped with the suggestion of higher priority given to preventative maintenance of the most critical components. This research was qualitative but based on some quantitative evidence. There are no numeric methods presented for testing different maintenance or inspection options.

Further analysis of weather conditions on the failure of S&C is given in the paper by Dindar, Kaewunruen, An and Gigante-Barrera, 2017, which provides a high level Fault Tree for the derailment of a train at an S&C [106].

This literature review has highlighted the importance of considering S&C derailments and some of the issues surrounding the efficient management of the S&C assets. In the current literature, there are limited tools for the modelling of S&C asset management strategies, especially with the aim of considering their impact on derailment.

2.6.2: Fire Protection System Literature Review

The article entitled 'Tunnel safety, risk assessment and decision making' gives a summary of a project commissioned by the European Parliament on the assessment of tunnel fire safety on rail and road tunnels [107]. Recommendations are made, including the development and use of acceptable models that can aid tunnel fire safety design.

There are several approaches to fire safety assessment, proposed in literature, for underground stations or tunnels. Roh, Ryou, Park and Jang, 2009, combine fire simulation with evacuation simulation to assess the impact of platform screen doors on passenger safety, on an underground station [108]. The simulations detailed in the paper show that the time required for evacuation of the particular underground station under consideration, with platform screen doors, is approximately 6 minutes. This kind of analysis could be applied as part of the risk assessment of a specific station. In addition, the paper concludes that there is an improvement in ventilation with the addition of platform screen doors.

A conference paper by Taranda and King, 2009, discusses the use of passive fire protection methods in rail and road tunnels [109]. These passive fire protection methods include panels and coatings that can be applied to the tunnel, to reduce the structural damage if a fire should occur. The paper also recommends the use of water deluge systems in tunnels to control the spread of fire. The paper recommends a cost-benefit analysis backed up by quantitative risk assessment, to inform decisions on the use of active or passive fire suppression systems, however, no method for this is provided.

An Event Tree is proposed by Poon and Lau, 2007, to consider the risk posed by underground station and tunnel fires [110]**.** The paper states that a combination of sprinkler systems, hydrants, and water mist suppression systems are used in stations. From the Event Tree, critical chains of events are identified. A sensitivity analysis is carried out on the event parameters to test their impact on the end outcome. It is assumed that stations are fitted with suppression systems that are successful between 50% and 90% of the time, but tunnels are not fitted with active suppression systems. The analysis indicates that fires occurring in tunnel sections are more likely to lead to higher numbers of fatalities. A sensitivity analysis of 12 tunnel designs is presented. The design variable with the highest impact on human safety is found to be the lack of active suppression systems in tunnels. However, there is a lack of information about the source of the data, or estimates, used for quantitative analysis of each event or design.

Howarth, Kara-Zaitri and Chakib, 1999, present a scoring metric that is used to compare fire safety in a variety of passenger terminals, where a passenger terminal is designated as a building with two or more linked transport systems and facilities such as retail outlets [111]. These terminals included those with sub-surface stations. The metric introduced in the paper includes a representation of fire safety management, fire risk and passenger density. Here, fire safety management and fire risk are scored based on the analyst's opinion. A high-level review of the regulations in place reveals that there are more stringent guidelines for underground stations than the other passenger terminals considered in the paper. There are limited recommendations to improve fire safety management such as, reducing passenger density or improving fire management systems. The method provides a tool for the assessment of compliance with guidelines, but, there are few practical recommendations made in the paper for the improvement of the systems used to manage fire safety.

The conference paper by Pan, Lo, Liao and Cong, 2001, describes a test carried out on an underground station platform to assess the effectiveness of water mist systems [112]. It was found that the water mist system can reduce the consequences of fire by reducing temperature, smoke levels, carbon monoxide and carbon dioxide concentrations.

It may also be relevant to consider research which has addressed the failure of fire protection systems in settings outside of underground stations, or tunnel environments. The paper by Andrews and Bartlett, 2003, uses a Genetic Algorithm to find the optimal design of a fire water deluge system for an offshore platform [83]. Constraints are applied that limit the system design and maintenance actions according to financial cost. This includes the initial cost of each component and the cost of inspecting and maintaining the component. The cost of a false activation of the system is also included. A Genetic Algorithm is used to optimize the design, with respect to the unavailability plus an imposed penalty if the constraints are violated. The unavailability of the system is found via Fault Tree analysis, as is the false activation occurrence. A Fault Tree, using a house event to model design alternatives, is constructed and then converted to a Binary Decision Diagram for incorporation into the Genetic Algorithm.

A review of information and models available to study sprinkler system performance is given by Frank, Gravestock, Spearpoint and Fleischmann, 2013 [113]. Two approaches to model sprinkler effectiveness are discussed, the first focusing on component-based information and the second based on fire incident data. Comparisons are made between both modelling approaches. The majority of failures reported at a system level were due to the system being falsely shut off, which was deemed difficult to incorporate into modelling approaches at the component level. Current component models were also binary, with components only residing in the working or failed state, and with no dependencies between component states. In addition, due to the rare nature of fires, system-based studies were difficult to apply to specific systems in any detail. The paper highlights that historically there has been a lack of information or research into the reliability of fire suppression systems, and there is also a lack of data available. Previous methods for analysing fire risk include Event Tree analysis. However, it is difficult to determine the probabilities used for each event and how the probability should be modified based on changes to the system. Within this review, it is proposed that sprinkler performance may depend on: the sprinkler system design, age, deterioration, inspection, testing, maintenance and water supply, along with the building design and ventilation. The review does not consider the potential for sprinkler systems to fail when there is no fire present, leading to surplus activation.

The paper by Boyd and Locurto, 1986, uses a reliability block diagram to model a fire protection system for a high-rise building [114]. Reliability issues are discussed including the lack of redundancy in the system design. Namely, a water supply failure, a water pressure sensor failure, a power switch failure or a pump failure, resulting in a system failure. It is suggested that the introduction of component redundancies within the system structure can improve the reliability, for instance the addition of a back-up pump. It is also highlighted that the addition of redundancies to the system should be considered carefully to avoid common cause failures. The paper looks at the system that supplies electric power to the pump. This includes the diesel generator, circuit breaker, automatic transfer switch and utility line. A Fault Tree is created for this system and from this, the reliability of the system is calculated. There are weaknesses in this approach, including those of assumed independence between failures and constant component failure rates. The paper suggests the use of FMECA or Quality Assurance to test the impact of inspection, maintenance and testing on the failure rate of the components.

This literature review highlights the need for methods to assess the safety of underground stations with respect to a fire hazard, and to look for areas of improvement where fire protection systems can be used to reduce the risk of fatalities. There are deficiencies in the models present in literature, when concerning complex maintenance strategies, non-constant failure rates, dependencies between component failures and phased maintenance strategies.

2.6.3 Summary

This review has given examples of approaches available in literature for modelling the systems chosen for application in this thesis. These systems are railway S&C and underground fire protection systems. This is relevant to the thesis topic as it provides the background for the developed models and highlights areas where the developed models can improve on the state of the art. The S&C model is presented in Chapter 4 of this thesis. The fire protection system model is presented in Chapter 5 of this thesis.

This review has highlighted that for S&C modelling there is scope to develop models that consider the derailment at an S&C, and how the management of the S&C can contribute to this. The review also further justifies the selection of the S&C as an application due to the expected derailment should the S&C fail. This informs the research direction for the model presented in Chapter 4, by demonstrating areas for further research. These areas involve detailed modelling of the S&C condition, along with the contribution of casual and contributory factors, such as insufficient maintenance, failed inspection or unsuccessful maintenance such that the component is not returned to the 'as good as new' state.

This review has also highlighted the limited body of literature surrounding modelling of fire protection system condition. This is especially the case when attempting to model complex maintenance strategies, non-constant failure rates, dependencies between component failures and phased maintenance strategies. This informs the development of the model presented in Chapter 5, for the assessment of fire protection systems, which addresses these deficiencies to predict the system unavailability over time. Chapter 6 develops this model further to predict and optimise safety risk of the system.

2.7 Discussion

To fulfil the aim of this project a method should be developed that can model complex systems with a variety of degradation and asset management processes. The approach should be able to model multi-component systems, where there is inbuilt redundancy in the system. The method must also have the potential for incorporation into a risk-based asset management optimisation tool. This is likely to be simulation based due to the complex nature of the system, and hence the computational efficiency of the approach must be considered. In addition, consideration of uncertainty on predictions from the model should be explored. The review within this chapter has considered works in industry and literature surrounding these topics.

This review of literature has identified areas where further study can be completed. The method currently used in the UK underground and over ground railway industry has several areas of weakness, namely modelling non-constant failure rates, complex asset management strategies and uncertainty. Various methods have been reviewed for their suitability, with a Petri net approach selected for the modelling of the system condition and asset management. One identified gap in the literature is an applied optimisation process that combines safety risk and life-cycle cost for a Petri net based model. This is especially the case for complex asset management strategies, such as phased strategies, whereby different strategies are applied to the system depending on its age. Another area where further study can be completed is in the consideration of uncertain model inputs, and their impact on model outputs. Finally, for development of a Petri net based approach, there are gaps in literature for a flexible model reduction approach, in order to improve model simulation efficiency. These areas are identified in this review for further research, within this project.

In addition, the modelling available in literature for S&C and underground fire protection systems was presented. This review has highlighted gaps in S&C modelling surrounding the impact of factors such as imperfect maintenance or failed inspection on S&C condition, system state and derailment occurrence. This review has also highlighted the limited body of literature surrounding modelling of fire protection system condition. This is especially the case when attempting to model complex maintenance strategies, non-constant failure rates, dependencies between component failures and phased maintenance strategies. This part of the review has informed the modelling direction for the specific system models developed in this project.

The research presented in this chapter has also informed the choice of any existing methodologies for use in this project. The combined risk modelling approach proposed in this thesis should have several stages. The first stage of the methodology should develop the failure logic of the system to the component level. The second stage of the methodology should model the component condition and system level asset management strategies and give resulting system level metrics. The third stage of the methodology should take these system level metrics and provide a risk measure for the system, in terms of the predicted number of fatalities. As a result of this literature review, the following methods are suggested for use in development of a risk modelling methodology within this thesis:

- A Fault Tree approach is proposed to give system level failure logic in terms of component or sub-system level failure. This choice of approach can improve integration with current risk modelling methodologies in industry. The method can also be used to gain system metrics from component level failure, in a computationally efficient manner. The Fault Tree method is also chosen as an initial step to gain the system logic as it allows the user to easily explore combinations of events that can cause a system failure.
- A Petri net approach is suggested to model the component or sub-system state and failure, with component or system level asset management strategies included. This choice of approach gives a high level of flexibility and can model complex degradation and maintenance processes. The approach can also be used to gain system level logic. The Petri Net method is selected because it can be applied to systems with complex inspection and maintenance strategies and is flexible at modelling component failure rates. This allows different failure rates to be modelled for a component, including failure rates that change with increased maintenance actions.
- Monte Carlo simulation is suggested for numerical analysis of Petri Net models. Monte Carlo simulation is selected because it converges to the true solution and can be used where an exact analytical solution cannot be found, due to a complex model structure. The error associated with the convergence of the model can also be analysed, so that a sufficient number of runs are completed such that the final solution closely approximates the true solution.
- An Event Tree approach is suggested to gain risk estimates, in terms of the predicted number of fatalities, from the system metrics output from the Petri net model. Again, this gives an efficient method that can improve integration into current risk modelling methodologies in industry. Event Tree analysis is chosen since it is a clear graphical method to combine event frequency, with the probability of failure of enabling event and consequence analysis in order to predict risk.

The existing methods, detailed here, are combined to give a stepwise modelling approach that can be found in Section 3.5 of this thesis. Finally, a combined Genetic Algorithm and Simulated Annealing approach is suggested for the optimisation of a system. This is discussed further in Chapter 6.

Methods such as Bayesian Networks and Dynamic Event Trees could be valid alternatives to the Event Tree stage of the proposed methodology. However, since these approaches also come at a higher computational cost, integration with an optimisation approach could be intractable. Further study can be completed in this area but is outside the scope of this thesis.

2.8 Conclusion

The first part of this literature review has considered the risk modelling methodologies implemented in the UK railway industry. Several limitations of the method were identified. These limitations include: a lack of the modelling of ageing components with non-constant failure rates, modelling complex maintenance and inspection strategies, dependencies between failures and considering uncertainty in the values predicted by the models. There are also challenges faced in modelling risk across the railway network in a whole network model, due to the size and complexity of the network.

The second part of this chapter has reviewed risk modelling and asset management methodologies proposed in literature. Several methods have been reviewed including: Fault Trees, Event Trees, Petri nets, Markov models and Bayesian Networks. These methods were reviewed considering their application to complex ageing systems. This review informs the choice of methods for further development in this project and identifies areas of weakness with them. Notably, within a Petri net framework, areas that consider the uncertainty of the model predictions and the computational efficiency of simulating the model were highlighted as areas for further exploration.

In the third part of this chapter works available in literature for system optimisation are presented. This section informs the choice of optimisation methods for combination in Chapter 6, namely Genetic Algorithms and Simulated Annealing algorithms. This section also highlights where further study can be completed in this area, this includes the optimisation of a system level phased asset management strategy.

The fourth part of this chapter addresses the inclusion of uncertainty in model outputs, given uncertainty in model inputs. This review section highlights areas for further study, including the exploration of an algorithm to improve the costly analysis of varying model inputs. The fifth part of this chapter explores methods for model reduction, to address costly simulation times. This review section justifies exploration of a flexible approach to Petri net model reduction, which is not reliant on specific sub structures within the model and can handle different distributions governing firing times.

The final review in this chapter gives work available in literature for S&C and fire protection systems. This review highlights where new models can be developed to improve current modelling application. The review also further justifies the choice of model application in this project.

Following the review of literature presented in this chapter the following chapters address the areas identified for further study, in the risk and hazard modelling field, where improvements can be made:

- Chapter 3 gives a proposed methodology that can be applied in industry and allows: modelling of non-constant failure rates, complex maintenance and inspection strategies and dependencies between failures. This is given in Section 3.5 of the chapter. Prior to this, an introduction to the existing methodologies, that are implemented in the proposed approach, is presented in the early parts of the chapter.
- Chapter 4 gives a new model for a railway S&C. This model predicts derailment frequency and includes modelling of failed inspection, imperfect maintenance and opportunistic maintenance strategies. The model also predicts the overall state of the S&C, and includes any dependencies between this state and derailment occurrence. The model also outputs the number of times each maintenance action is completed.

- Chapter 5 gives a new model for a fire protection system. This includes modelling an interlinked deluge, detection and alarm system and the human interaction with the system. A phased asset management strategy is modelled here.
- Chapter 6 further expands the model in Chapter 5 to give a time-dependent risk estimate for the system and a life-cycle cost for the system. An optimisation method is proposed for a phased asset management strategy and this is applied to the model. Chapter 6 also presents a study into the convergence rate of the model and a novel method for estimating model output uncertainty given uncertain inputs.
- Chapter 7 presents a novel Petri net reduction method. The method is flexible for application to different model structures and governing distributions. The method is explored with four applications. The first two are simple examples to demonstrate the methodology. The third example includes the method within a two-stage optimisation procedure. The final example explores the use of the method in more depth, including the use of the method to justify reduced model structure selection.

Chapter 3 Proposed Methodology

Firstly, this chapter gives an overview of any existing methods that are applied in this thesis. This is intended as an aid to the reader. The methodologies for Fault Tree analysis, Event Tree analysis, Monte Carlo Simulation and Petri nets are presented. A discussion of each method is given at the end of the relevant section. Section 3.5 gives the methodology proposed in this thesis for risk and hazard modelling, which is implemented and expanded further in the remaining chapters of this thesis.

This chapter also gives a description of the software developed as part of this project, including any additions that were included in order to improve the computational efficiency of the software. This is included in Section 3.6 of this chapter.

3.1: Fault Tree analysis

Fault Tree analysis is used to determine system level logic and to provide justification for the risk models developed[115][116]. A Fault Tree is a deductive approach to risk assessment where a catastrophic event or failure, known as a top event, is the starting point. A top down analysis is used to break down the causes of the top event into various deeper intermediate events using Boolean operators [117]. This analysis stops when the limit of resolution is reached. The events at this point are classed as basic events.

3.1.1: Fault Tree symbols

A Fault Tree is constructed of a set of gates connected to events. An event below the gate is known as an input event and the event above the gate is known as the output event. The gates show the relationship between input events to cause an output event. There are several gates types that can be used in Fault Tree analysis, Figure 3.1 illustrates an OR gate and an AND gate and Table 3.1 gives further gate examples and an explanation of the logic they represent.

An AnD gate requires all the input events to occur for the corresponding output event to occur. An OR gate requires at least one of the input events to occur in order for the output event to occur.

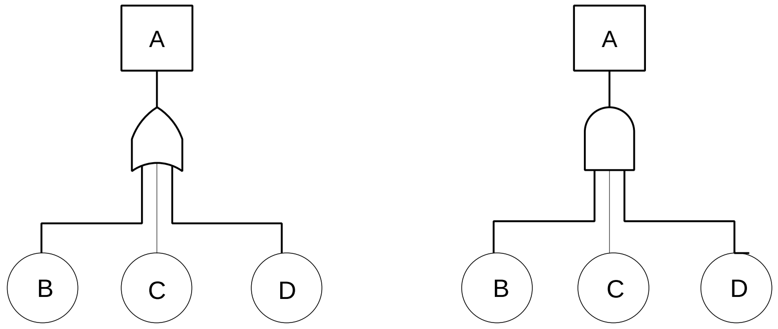


Figure 3.1: An example OR gate (left hand side) and AND gate (right hand side)

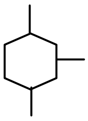
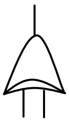
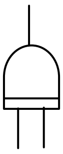
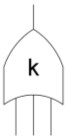
Name	**Symbol**	**Explanation**
Inhibit Gate		The output only occurs if the input occurs and a conditional event also exists.
Exclusive OR Gate		The output occurs if any of the input events occur but not if more than one input event occurs.
Priority AnD Gate		The output event occurs only if all of the input events occur in order from left to right
Voting Gate		If there are n input events to the gate, k of them must occur in order for the output event to occur.

Table 3.1: Example gates for Fault Tree analysis

Different symbols can also be introduced for different types of event within a Fault Tree. Table 3.2 gives these symbols.

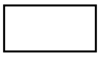
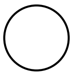
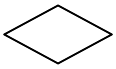
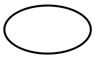
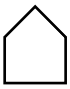
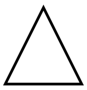
Name	**Symbol**	**Explanation**
Top Events/ Intermediate Events		An event that will be broken down further into either intermediate events or basic events.
Basic Events		An event that is considered a root cause within the scope of the model.
Undeveloped Event		An event that is not developed further but is not a basic event. It may be that there is no more information available.
Conditional Event		Used with an inhibit gate to give a condition.
House Event		An event that is considered either to be TRUE of FALSE.
Transfer Symbol		Used when the fault tree is large and the linked section is developed elsewhere.

Table 3.2: An example of the different event symbols used in Fault Tree analysis.

3.1.2: Qualitative analysis of Fault Trees

Qualitative analysis of the Fault Tree can be carried out to identify minimal cut sets. A cut set is a collection of basic events that, if they occur, will lead to failure of the system. A minimal cut set is a cut set that contains the smallest number of basic events such that if they occur then the Top Event will occur. A top-down or bottom-up approach can be used to find the minimal cut sets. Both methods require the idempotence, distributive and absorption laws of Boolean algebra to reduce the expressions generated. These laws are described below for basic events A, B and C [118].

Idempotence Law:

$$A \wedge A = A \tag{3.1}$$

$$A \vee A = A \tag{3.2}$$

Distributive Law:

$$A \wedge (B \vee C) = (A \wedge B) \vee (B \wedge C) \tag{3.3}$$

$$A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C) \tag{3.4}$$

Absorption Law:

$$A \wedge (A \vee B) = A \vee (A \wedge B) = A \tag{3.5}$$

Where $\wedge$ represents conjunction and $\vee$, represents disjunction.

In the top-down approach, the starting point is the top event. Each intermediate event at each level is then substituted into the expression using the logic described by the relevant gate until the expression for the top event contains only basic events. This expression is then reduced to the minimal sum-of-products or disjunctive normal form to give the combinations of minimal cut sets that cause the top event. In the bottom-up approach, basic events are combined using the Fault Tree logic and substituted into increasingly higher-level intermediate events, which are, in turn, combined using the Fault Tree logic until the Top Event is reached. Again, the expression is reduced to give minimal cut sets.

For large Fault Trees, computer programs can be used to generate the minimal cut sets. However, the number of minimal cut sets can increase exponentially with the number of gates [45]. Approximations can be made which disregard higher order cut sets. Here, minimal cut sets containing a number of events, above a certain threshold, are discarded as they are assumed to have a low likelihood of occurrence. An approximation such as this reduces the computational cost of analysis of large Fault Trees. However, this can influence the accuracy of the results gained from the Fault Tree, especially if there are undetected common cause failures for the basic events in the discarded minimal cut sets.

3.1.3: Quantifying Basic Events

In order to quantitatively analyse the Fault Tree, probabilities can be assigned to the basic events. This section gives methods for describing the performance of a component in terms of availability and reliability. Here, it is assumed that there are only two states for each component; it is either in a failed state or a working state.

The availability of a component has several different interpretations:

1) For components in standby the availability is the probability that the component works on demand. For instance, the availability of a back-up generator is the probability that it works when the main generator fails. Normally this sort of component fails in an unrevealed way, and hence, requires inspection to identify any potential failures.

2) For continuously operating component with revealed failures, the availability at a time is the probability that the component works at that time.
3) Where the productivity of a component over a period is required, the availability can be interpreted as the fraction of total time that a component can perform its required function.

Reliability is the measure of the probability of non-occurrence of a failure in a component or system over a defined interval. It is a useful measure when failure cannot be tolerated in a system such as in safety critical systems. The unreliability is the probability that the component will fail within an interval.

Equation 3.6 denotes the unreliability, which is the probability that a component has failed in the interval [0,t]:

$$F(t) = 1 - e^{-\int_0^t h(t')dt'} \qquad (3.6)$$

Different distributions for $h(t)$ can be substituted into this Equation to fit the data available.

For a constant failure rate, $h(t) = \lambda$ where λ is constant and independent of time. Substituting into Equation 3.6 gives the commonly used exponential distribution for unreliability. Equation 3.7 gives this distribution.

$$F(t) = 1 - e^{-\lambda t} \qquad (3.7)$$

For a repairable component with failures that are revealed and for constant failure rate, λ, and repair rate, ν, Equation 3.8 gives the unavailability.

$$Q(t) = \frac{\lambda}{\lambda+\nu}(1 - e^{-(\lambda+\nu)t}) \qquad (3.8)$$

For a repairable component, with unrevealed failures and a constant failure rate, λ, that is inspected periodically with a defined time interval, θ, Equation 3.9 gives the average unavailability.

$$Q_{av} = 1 - \frac{1}{\lambda\theta}(1 - e^{-\lambda\theta}) \qquad (3.9)$$

3.1.4: Quantitative analysis of Fault Trees

The Fault Tree can be analysed quantitatively to give a numerical value for the Top Event probability. The basic events must be independent for this analysis.

If there are no repeated events in the Fault Tree, then the top event probability can be calculated by propagating the probabilities of the basic events in the Fault Tree and combining the probabilities at each gate following the Boolean logic of the Fault Tree. This approach cannot be applied if there are repeated events. Instead, the minimal cut sets of the Fault Tree must be identified, and Top Event probability calculated using the Inclusion-Exclusion expansion.

The Inclusion-Exclusion expansion is defined in Equation 3.10, where C_i is each minimal cut set and N_c is the number of minimal cut sets. The elements of each minimal cut set are substituted into Equation 3.14 and simplified where possible.

$$P(TOP) = \sum_{i=1}^{N_c} P(C_i) - \sum_{i=2}^{N_c}\sum_{j=1}^{i-1} P(C_i \cap Cj) + \sum_{i=3}^{N_c}\sum_{j=2}^{i-1}\sum_{k=1}^{j-1} P(C_i \cap Cj \cap C_k) - \cdots + (-1)^{N_c+1}P(C_1 \cap C_2 \ldots \cap C_{N_c}) \qquad (3.10)$$

For a Fault Tree with many minimal cut sets, this calculation is computationally intensive. The Minimal Cut Set Upper Bound can be used to estimate the value of Top Event probability.

The Minimal Cut Set Upper Bound is defined in Equation 3.11. This can be used to find an estimate of the Top Event probability. The value for the Minimal Cut Set Upper Bound will always be greater or equal to the value calculated using the exclusion inclusion expansion.

$$Q_{SYS} = 1 - \prod_{i=1}^{N_c}(1 - P(C_i)) \qquad (3.11)$$

Top Event Frequency

The Top Event frequency can also be calculated from a Fault Tree. The Top Event frequency $(w_s(t))$ is the probability that the Top Event occurs, per unit time. To find this, consider a small interval of time, $[t, t + dt)$, the probability of an event occurring in this interval is the event frequency multiplied by the length of the interval $(w_i(t)dt)$. For the Top Event to occur in the interval, there must be the completion of one or more minimal cut sets in the interval. For this to happen, all but one event in a minimal cut set could occur before the interval and then the final event occurs in the interval, leading to system failure. Hence, the probability of Top Event occurrence in the interval, due to the final event in its minimal cut set, is the probability that the system is in a critical state for that event $(G_i(\boldsymbol{q}(t))$, multiplied by the probability of the event occurring $(w_i(t)dt)$. There may be multiple minimal cut sets with the same final event which means they would be completed at the same time. The sum is taken, over all basic events, to calculate the total probability of occurrence of the Top Event within a time interval $(w_s(t)dt)$. Equation 3.12 shows this relationship which can be simplified to Equation 3.13 to give the Top Event frequency. Here, $(G_i(\boldsymbol{q}(t))$ is Birnbaum's importance measure defined later in Equation 3.19.

$$w_s(t)dt = \sum_i G_i(\boldsymbol{q}(t)) \cdot w_i(t)dt \qquad (3.12)$$

$$w_s(t) = \sum_i G_i(\boldsymbol{q}(t)) \cdot w_i(t) \qquad (3.13)$$

The concept of initiating and enabling events can be introduced into a Fault Tree model to consider situations where the time order of events is important [38]. Initiating and enabling events are defined as follows:

Initiating events are events that perturb system variables and place a demand for a response from protection, control or safety systems.

Enabling events are inactive control or protection systems. By occurring they permit initiating event to cause the top event.

The initiating event must occur in a finite amount of time in which the enabling events have also occurred. If initiating and enabling events are not considered, then it can lead to an over estimation of Top Event probability, or frequency, as contributions that come from sequences of events that may not actually lead to the Top Event are included.

Importance Measures

Once a numerical quantification for the Top Event occurrence has been found, it is useful to perform further analysis of the Fault Tree to identify areas of weakness within the system in question. Importance measures can be used to assign a numerical value to the contribution of the basic events of minimal cut sets to the Top Event so comparisons can be made. The critical states of a system are needed to calculate importance measures. A critical system state, for a basic event, is a system state such that if the basic event occurs then the whole system will fail; there may be multiple states where this may occur for each system. Importance measures can either be deterministic or probabilistic.

A deterministic importance measure does not account for the probability of failure of the system and so can be useful if there is limited, or unreliable, data available. An example of this is the structural measure of importance (I) which is defined by Equation 3.14.

$$I = \frac{number\ of\ critical\ system\ states\ for\ basic\ event\ i}{total\ number\ of\ states\ for\ the\ remaining\ basic\ events} \tag{3.14}$$

Probabilistic measures of importance also consider how likely a component failure is to occur. Examples include Birnbaum's importance measure, criticality measure of importance and the Fussell-Vesely measure of importance. Birnbaum's importance measure, $(G_i(\boldsymbol{q}))$, for a basic event can be found from the partial differentiation of the expression for the system failure probability $(Q(\boldsymbol{q}))$ with respect to the failure probability of the occurrence of the basic event (q_i) as shown in Equation 3.15: [119]

$$G_i(\boldsymbol{q}) = \frac{\partial Q(\boldsymbol{q})}{\partial q_i} \tag{3.15}$$

The criticality measure of importance considers the probability that the system is in a critical state for a basic event and the probability that it has occurred. It is defined in Equation 3.16 and is weighted by the unavailability of the system.

$$I_i = \frac{G_i(\boldsymbol{q}(t))q_i(t)}{Q_{SYS}(\boldsymbol{q}(t))} \tag{3.16}$$

Similarly, the Fussell-Vesely measure of importance can be used to rank the basic events, by considering the contributions of minimal cut sets, containing the basic event of interest on the system failure. It is defined as the probability of the union of the minimal cut sets, containing the basic event in question, given the system has failed. The Fussell-Vesely measure of importance is defined in Equation 3.17.

$$I_i = \frac{P(\cup_{k|i\in k} C_k)}{Q_{SYS}(\boldsymbol{q}(t))} \tag{3.17}$$

Similarly, the minimal cut sets can be ranked using the Fussell-Vesely measure of minimal cut set importance. This is defined as the probability of occurrence of the minimal cut set in question, given the system has failed. This is given in Equation 3.18.

$$I_i = \frac{P(C_i)}{Q_{SYS}(\boldsymbol{q}(t))} \tag{3.18}$$

The Barlow-Proschan measure of initiator importance can be used to consider initiating events in a situation where the order of events is important. Equation 3.19 gives the Barlow-Proschan measure of importance where event i causes the failure of the system in a time interval $(0, t]$.

$$I_i = \frac{\int_0^t \{Q(1_i, \boldsymbol{q}(t)) - Q(0_i, \boldsymbol{q}(t))\} w_i(t) dt}{W(0,t)} \tag{3.19}$$

The sequential contributory measure of importance considers enabling events for system failure in an interval, given an initiating event. This is shown in Equation 3.20. The index j runs over initiating events, contained in a cut set, C_k, with the enabling event, i. This is an approximate expression.

$$I_i = \frac{\sum_{\substack{j \\ i\neq j \\ i,j\in C_k}} \int_0^t \left\{Q\left(1_i, 1_j, \boldsymbol{q}(t)\right) - Q\left(0_i, 0_j, \boldsymbol{q}(t)\right)\right\} q_i(t) w_i(t) dt}{W(0,t)} \tag{3.20}$$

Both measures of importance allow the events to be ranked by their contribution to Top Event occurrence, as either an initiating or enabling event.

3.1.5: Discussion

Fault Trees provide a framework for quantitative and qualitative analysis of risk by considering a set of Top Events. Basic events are combined using Boolean logic, which is demonstrated by the Fault Tree structure, to represent the combinations of events that result in the Top Event. There are several benefits to Fault Tree analysis, these include:

1. A Fault Tree based model allows focus on a top event and a systematic method for analysing potential causes.
2. A Fault Tree is a clear method for communicating risk to those outside the field as it is easily explained and interpreted.
3. Fault Trees facilitate both qualitative and quantitative analysis which can be used to make informed decisions.
4. Repairable, revealed and unrevealed component failure can be modelled.

On the other hand, there are several areas where Fault Tree analysis encounters difficulties:

1. The method is not exhaustive as only specific Top Events are considered.
2. For exact quantitative analysis of a Fault Tree, the basic events must be independent; this may not be the case due to maintenance strategies, dependencies between components or common cause failures.
3. It is difficult to incorporate non-constant failure rates into a Fault Tree based model.
4. It is difficult to model complex asset management strategies, unless the system is non-repairable.

In an underground railway there are repairable components that will fail at different rates depending on different factors such as the age, usage or quality. The rates of failure are also likely to change over time. In addition, there are continuous variables involved in the risks of an underground railway system, such as speed in derailment risk. There are also likely to be dependencies between different component failures, for example, due to opportunistic maintenance strategies. However, Fault Trees have a capacity to link component failure events to give a Top Event and this is the capacity with which they are implemented in this thesis.

3.2: Event Tree analysis

Event Trees are a method of analysis uses in system risk assessment. They are an inductive method that allows multiple sequences of enabling events, and their consequences, following an initiating event, to be considered [120]. Initiating events are those that can start an event sequence that can cause a hazard and place a demand on the system for a response. Enabling events are those that propagate and incident sequence and can reduce or escalate hazard consequences, given that an initiating event has occurred. Quantitative analysis of an Event Tree can be completed by assigning a probability of occurrence to each enabling event and combining these with an initiating event frequency, to give the frequency of each sequence of events. The consequences of these sequences of events can also be included to provide a framework for the calculation of the risk of the system in question. Event Tree analysis does not imply independence in events and so allows common cause failures or dependencies to be considered.

3.2.1: Event Tree structure

Figure 3.2 shows an illustrative Event Tree. An Event Tree analysis starts with the occurrence of an initiating event. In Figure 3.2, the initiating event, E_i, can be seen on the left hand side of the Event Tree.

The enabling events run from left to right, across the Event Tree. For the Event Tree Figure 3.2 there are five enabling events which can occur, in ascending order. Branching of the event tree occurs at specified points related to each of the enabling events. Often the branches are binary, meaning that there are only two branches created at each branching point that represent either occurrence, or non-occurrence, of the enabling event. However, partial failures can be considered as well as non-binary branching, if the branches cannot occur at the same time [121]. Figure 3.2 shows binary branching.

For each sequence of events in an Event Tree, further branching can be terminated if an event ends the incident sequence. Here, the branching in that sequence ends at the final logical point and this is represented by a horizontal line from the final event in the sequence to the end of the Event Tree. This can be seen in several cases for the Event Tree in Figure 3.2, for example for the sequence of events starting with Event 1, E_1, followed by Event 3, E_3, there is no further branching of that specific sequence of events. In addition, branching can be omitted if it is illogical, for example one enabling event ensures occurrence a subsequent enabling event. This is demonstrated in the Event Tree in Figure 3.2 for example for the sequence of events starting with Event 1, E_1, there is no branching corresponding to Event 2, E_2.

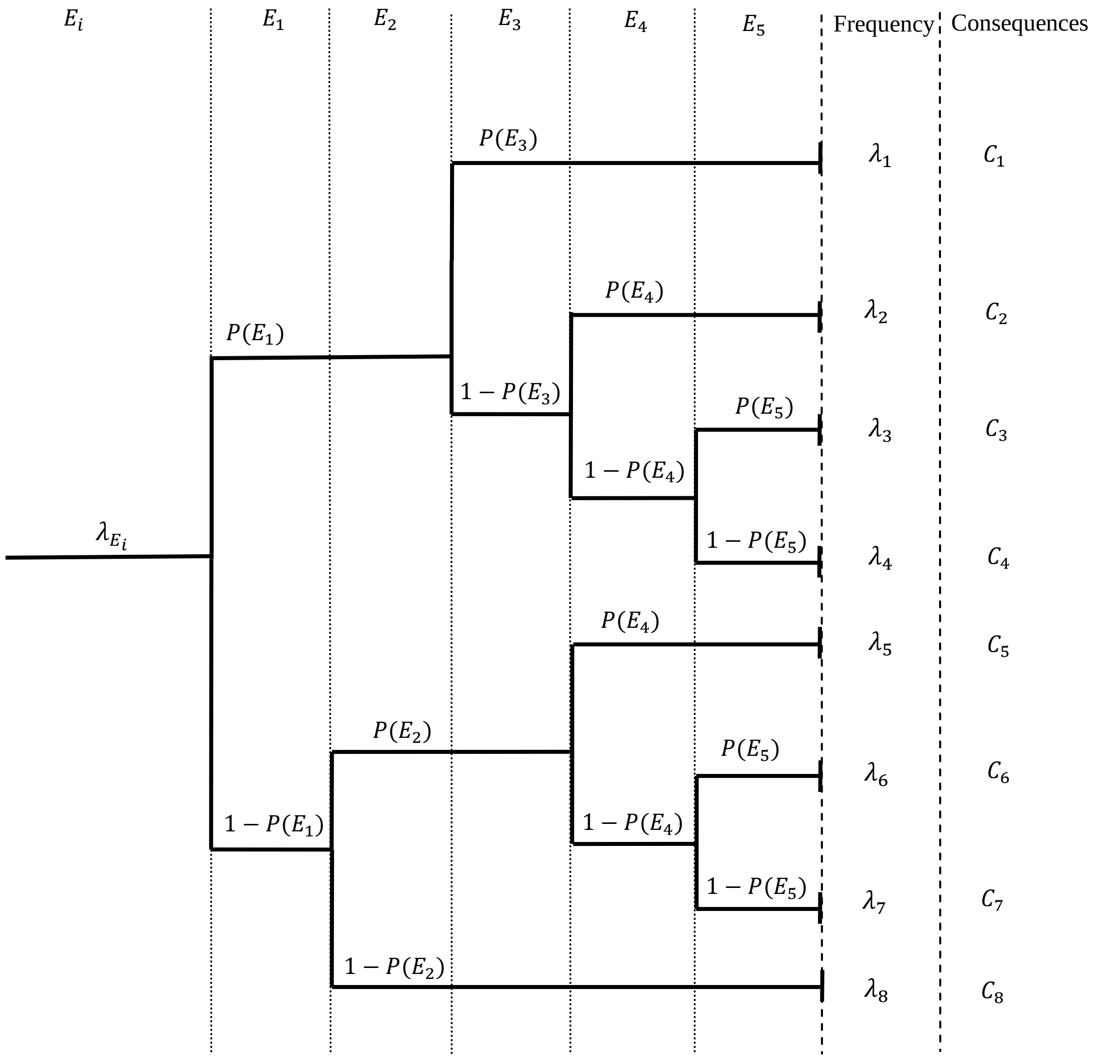


Figure 3.2: A sample Event Tree

3.2.2: Quantitative Event Tree analysis

Initiating event frequency, enabling event probabilities and consequences for each chain of events can be assigned to the Event Tree [120]. This is demonstrated in Figure 3.2. When events with the Event Tree are independent, the frequency of each chain of events can be found by multiplying the initiating event frequency with the probability of each event outcome in the chain of events. The risk for each chain of events can then be found by taking the product of the corresponding frequency and consequences. The total risk of the system can be calculated by summing the risk for each chain of events. Equation 3.21 shows this, where λ_i is the frequency of each sequence, C_i is the consequence of each sequence and N_E is the number of sequences.

$$Risk = \sum_{i=1}^{N_E} \lambda_i C_i \qquad (3.21)$$

The probability of occurrence for the enabling events that make up the branch points can be calculated via other methods such as Fault Tree analysis.

Dependencies may exist between events in an Event Tree. These can be total or partial. Where dependencies exist, the sequence probability cannot be found by simply multiplying the branch probabilities together, as you can with independent events.

If a total dependency exists between two events, the second event is either certain or impossible following the first event. These dependencies can be incorporated into the Event Tree structure. However, partial dependencies are those where enabling events, at branch points, have basic events in common. Hence, the same basic event may be in the minimal cut set for more than one of the enabling events. Two methods can be used to analyse Event Trees with partial dependencies are: extraction of the common factor and non-coherent analysis of the branch output events.

In the first method for analysis of an Event Tree with partial dependencies, the basic events in common, across the minimal cut sets of each enabling event, are identified [45]. Subsequently, the Event Tree is modified so that each of these identified basic events forms a separate branching point in the Event Tree structure. In addition, these basic events are removed from the minimal cut sets of each of the original enabling events, to avoid multiple inclusion in the analysis of the Event Tree. This results in an Event Tree structure with independent branching for quantitative analysis, but can lead to a large Event Tree structure if there are many partial dependencies between the enabling events.

The second method uses non-coherent analysis for quantification an Event Tree with partial dependencies. Here, the minimal cut sets for each enabling event are also required. These can be negated in order to gain the logical combinations of basic events, and their negations, for each output at each branching point of the Event Tree. In this method, prior to quantification of the Event Tree, each event sequence in the Event Tree is described in terms of these basic events and their negations. The inclusion-exclusion expansion can be used for this, followed by a simplification of the resulting expression through Boolean algebra. For quantitative analysis, the values for each basic event probability can be substituted into the final expression for each of the event sequences to give the probability of each of enabling event sequences. Finally, this can be combined with the initiating event frequency and the consequences of each chain of events to calculate the risk of each chain of events. This calculation can be lengthy if there are many partial dependencies in the Event Tree enabling events. To avoid this lengthy calculation the coherent Event Tree approximation can be made. Here, it is assumed that enabling events do not usually occur, and hence the probability of their non-occurrence is approximated to one. This removes many terms when implementing the inclusion-exclusion expansion, simplifying the method.

3.2.3: Discussion

There are several advantages to Event Tree analysis:

- Event Tree analysis can highlight areas of weakness in a system and allows the identification of where there may be a lack of controls.
- An Event Tree is a visual description that is clear and easily understandable which can aid in the communication of risk.
- Event Tree analysis allows the consequences for chains of events to be considered.
- Event Trees can be quantified by incorporating probability into each branch and combining this with the frequency of the Top Event and the consequences of each branch.

There are also several disadvantages to Event Tree analysis:

- Only one initiating event is considered in each Event Tree, this can lead to many Event Trees for a large system.
- There is no time dependence in the Event Tree and probabilities of branch events are considered as fixed discreet values.
- It is difficult to approximate the consequences of the event branches because there may be many unknown variables or unexpected factors that could influence the consequences.

Event Trees facilitate forward logic to consider what may happen following a major event or failure. For an underground railway they are useful as they allow the identification of areas of weakness as well as predictions of risk. Event Trees are used within this thesis to calculate the risk over a system life-cycle, given different system states predicted by a model. This risk measure can then be used as the basis of an optimization of the asset management strategy for the system, for instance by performing an optimisation to reduce the average yearly risk. This is discussed further in Chapter 6 of this thesis.

3.3: Monte Carlo simulation

Monte Carlo simulation is used in this thesis to gain a numerical result for models that cannot be solved analytically**.** Monte Carlo simulation refers to a range of algorithmic methods that can be applied to problems in order to gain a numeric result. Monte Carlo simulation can be seen as a statistical experiment where multiple runs are performed and each run is equivalent to an observation. After many runs the average of the outcome should converge.

Monte Carlo simulation can be used to model many different systems. To use Monte Carlo simulation to model risk, the system logic is required as well as failure and repair distributions. For each trial run, a value from these distributions is sampled at random. A further description of the Monte Carlo method including sampling methods for a range of distributions can be found in the book "Essentials of Monte Carlo Simulation: Statistical Methods for Building Simulation Models" [122].

3.3.1: Generating random numbers

Random numbers can be generated by simple experiments such as tossing dice, flipping a coin or drawing cards at random. This is difficult for situations where many random numbers are required. Pseudo-random numbers behave in a similar way to random numbers as they are independent and have a uniform distribution.

Recursion formulae can be used to generate pseudo-random numbers. Recursion formulae requires a 'seed' value to start the sequence. This seed value should be changed with every simulation for a variation in the pseudo-random numbers generated. However, keeping the same seed value is useful for identifying issues with a computer code, as each simulation will produce the same result if the seed is kept constant.

A commonly used pseudo-random number generator is the linear congruential generator. Equation 3.22 gives the recursion formula that is reliant on the previous value in the sequence. Equation 3.23 shows how each random number is generated following Equation 3.22. [3]

$$x_{n+1} = (ax_n + b) mod\, m \tag{3.22}$$

$$R_i = \frac{x_i}{m} \tag{3.23}$$

Here x_0 is the seed and a, b, m are integer constants, which are chosen to give a large cycle length and R_i is the generated pseudo-random number.

This is the pseudo-random number generator implemented in this thesis, through the inbuilt *'rand()'* function in C++ coding libraries, to generate random numbers to solve the models described. The seed value is initiated to the physical time of the simulation at the start of the running of any model. This avoids the same random number sequence being generated if there are two models running in series within the software. The cycle length for this random number generator is 4,294,967,296 which was deemed sufficient for the modelling completed in this thesis. Further random number generators can be found in the book "Random Number Generation and Monte Carlo Methods" [123].

3.3.2: Generating event times from distributions

Probability distributions can be assigned to component failure times. Various techniques can be used to sample a time at random, from these distributions. In this project, three probability distributions are assigned to components within the models developed. These are: the 2-Parameter Weibull distribution, the negative exponential distribution and the normal distribution. The techniques for sampling from these distributions are given in this section. Care must be taken when using the normal distribution due to the possibility of sampling a negative time value. Despite this, the normal distribution has been used throughout this project due to its common use in practical applications, but some restrictions have been applied. Firstly, if a negative time is sampled this value is approximated to a zero time. Secondly, when the normal distribution is implemented it must be in a scenario where there is a rare probability that a negative value will be sampled so as not to change the representation of the distribution in simulation, by this rounding up of negative values.

The inverse transform technique

The inverse transform technique can be used to sample a value, for the time to failure, from a probability density function. This requires the existence of the inverse of the cumulative distribution function. The cumulative distribution function is denoted by $F(t)$ and is the integral of the probability density function over the interval $[0, t)$. The cumulative distribution function is the probability that the failure has occurred in the interval $[0, t)$. Hence, over all time the cumulative distribution function is in the range $[0,1]$, a random number can be generated in the same interval.

The inverse transform technique can be used for a probability density function with a constant hazard rate, which yields the negative exponential distribution.

For a component, assume that the failure data follows an exponential probability density function with a constant hazard rate λ. [3] Such that the probability density function is given by Equation 3.24.

$$f(t) = \lambda e^{-\lambda t} \tag{3.24}$$

Figure 3.3 shows a graph of this probability density function with different values of λ.

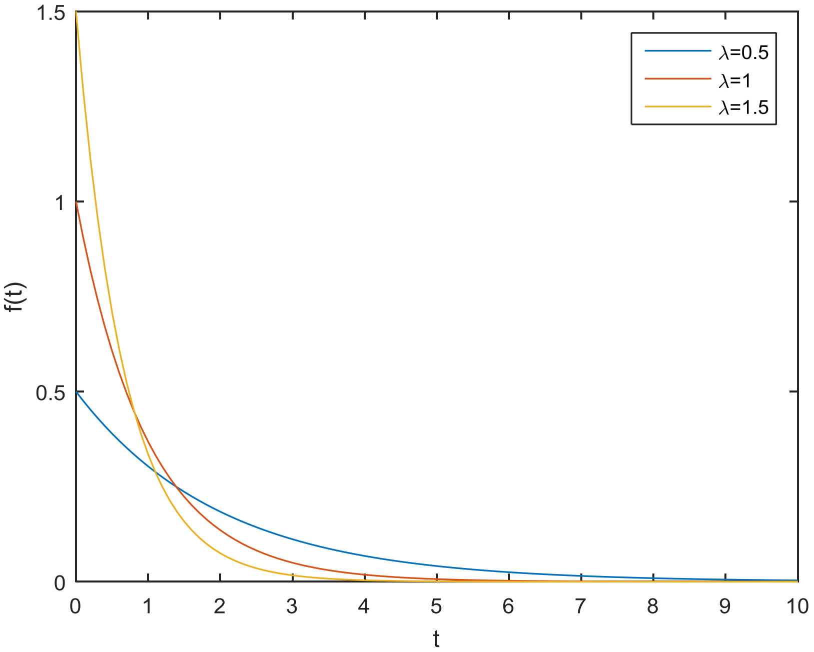


Figure 3.3: A graph showing the negative exponential probability density function.

Integrating the probability density function yields the cumulative distribution function as shown in Equation 3.25.

$$F(t) = \int_0^t f(t)dt = \int_0^t \lambda e^{-\lambda t} dt = 1 - e^{-\lambda t} \tag{3.25}$$

Since $F(t)$ is in the interval $[0,1]$ we can equate it to a random number, X in $[0,1]$, and rearrange the result to yield the time to failure given this random number. This method allows times to be sampled from the distribution. Equation 3.26 shows the random number equated to the cumulative distribution function.

$$X = F(t) = 1 - e^{-\lambda t} \tag{3.26}$$

Rearranging Equation 3.27 so that time is the subject, gives Equation 3.31.

$$t = -\frac{\ln(1-X)}{\lambda} \tag{3.27}$$

Since $(1 - X)$ is also a random number in [0,1] this expression can be simplified to give Equation 3.28.

$$t = -\frac{\ln(X)}{\lambda} \tag{2.28}$$

The time found in Equation 3.28 is a time randomly sampled from the negative exponential failure distribution.

The inverse transform technique can also be used if the probability density function follows a 2-Parameter Weibull distribution. The 2-Parameter Weibull distribution has a shape that is dependent on parameter values. The 2-Parameter Weibull distribution is particularly useful in situations where there may be changing hazard rates. The failure density function for this distribution is described in Equation 3.29 [3]. Figure 3.4 shows the 2-Parameter Weibull probability density function for different parameter values.

$$f(t) = \frac{\beta t^{\beta-1}}{\eta^{\beta}} e^{-\left(\frac{t}{\eta}\right)^{\beta}} \; where\; t \geq 0,\; \beta \geq 0,\; \eta \geq 0 \tag{3.29}$$

For the 2-Parameter Weibull distribution β is the shape parameter, it has the following interpretations:

$\beta < 1$: Increasing hazard rate

$\beta = 1$: Constant hazard rate

$\beta > 1$: Reducing hazard rate

η is the scale parameter which is the value for t where the probability of a component failure prior to this time is approximately 2/3.

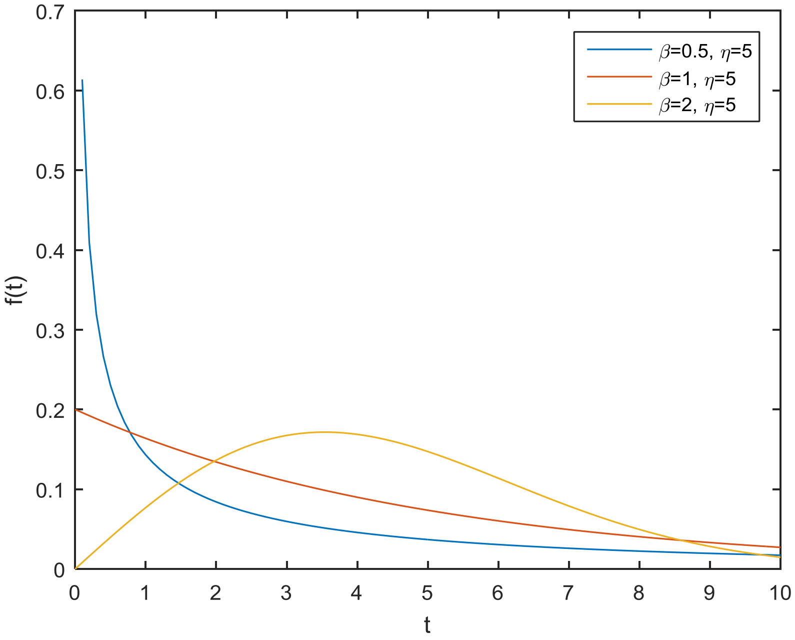


Figure 3.4: The 2- Parameter Weibull distribution for several different parameter values.

Integrating the probability density function yields the cumulative distribution function as shown in Equation 3.30:

$$F(t) = \int_0^t f(t) = \int_0^t \frac{\beta t^{\beta-1}}{\eta^\beta} e^{-\left(\frac{t}{\eta}\right)^\beta} dt = 1 - e^{-\left(\frac{t}{\eta}\right)^\beta} \qquad (3.30)$$

Setting this equal to a random number X, in [0,1], as before, rearranging and simplifying, gives a time randomly sampled from the 2-Parameter Weibull distribution. Equation 3.31 gives the formula for this.

$$t = \eta[-\ln(X)]^{\frac{1}{\beta}} \qquad (3.31)$$

In some instances, the cumulative distribution function cannot be analytically inverted, therefore this method is not valid. An example of where this method fails is with the normal distribution. The next section describes how the Central Limit Theorem can be used to sample a time to failure for a situation where the probability density function follows a normal distribution. This distribution is also applied to maintenance and inspection times in this thesis, which are sampled in the same manner.

Central Limit Theorem and Sampling from the Normal Distribution

The normal distribution is commonly used as it describes natural variance around a mean value. The probability density function for the normal distribution is given in Equation 3.32.

$$f(t) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(t-\mu)^2}{2\sigma^2}} \qquad (3.32)$$

Where the mean is μ and the standard deviation is σ**.** Figure 3.5 shows the normal distribution for several different values of the mean and the standard deviation.

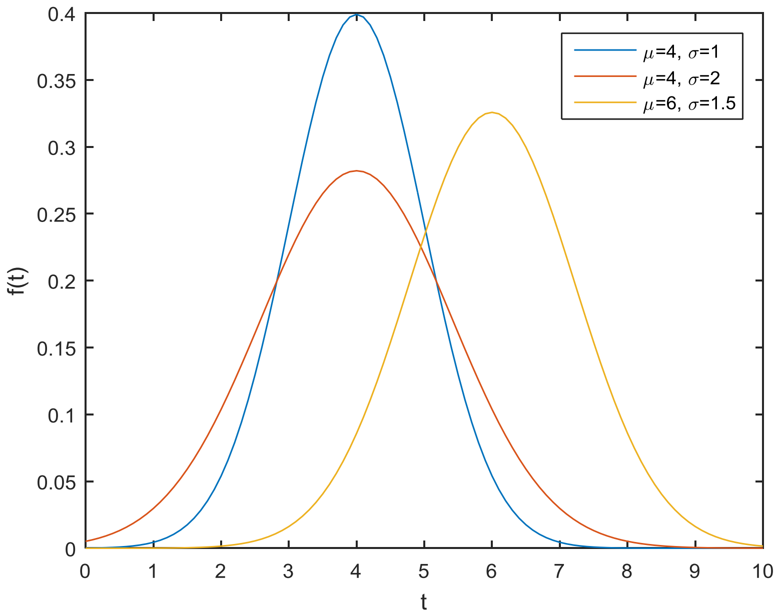


Figure 3.5: The Normal distribution for different values of the mean and standard deviation

Since the inverse transform technique cannot be applied to the normal distribution another method is needed to sample times from the normal distribution. The Central Limit Theorem can be used [3].

Consider a set of n independent random variables, $X_1, X_2, \dots, X_n$, that are identically distributed with a mean of μ and a variance of σ^2. Next consider the sum of these random variables, as given is Equation 3.33.

$$S_n = X_1 + X_2 + \cdots + X_n \qquad (3.33)$$

The Central Limit Theorem states that the difference between the sample average and the mean of the distribution, when multiplied by the root of the sample size, converges to the normal distribution with a mean of 0 and variance of σ^2.

Hence, the Central Limit Theorem gives the result in Expression 3.34 where, S_n/n is the mean if the random variables.

$$\sqrt{n}\left(\frac{S_n}{n} - \mu\right) \longrightarrow N(0, \sigma^2) \qquad (3.34)$$

Assume that n becomes sufficiently large that the limit is close to equality in expression 3.38. Transforming the normal distribution to have a variance of 1 and rearranging gives a random variable Y, that is asymptotically normally distributed with a mean of 0 and a standard deviation of 1, this is given in Equation 3.35.

$$Y_n = \frac{S_n - n\mu}{\sigma\sqrt{n}} \qquad (3.35)$$

A random number, X_i, can be taken from a uniform random distribution between [0,1], this distribution has a mean of 0.5. A finite n can be chosen so that the normal distribution can be approximated, $n = 12$ is a convenient choice. Hence, the uniform random distribution that the X_i values are taken from, has $\mu = 0.5$ and $\sigma^2 = \frac{1}{12}$. Substitution of these values into Equation 3.35 gives Equation 3.36.

$$Y_{12} = S_{12} - 6 = N(6,1) - 6 \qquad (3.36)$$

Let X be a random variable found by summing 12 values from a uniform random distribution. As shown in Equation 3.37, X is normally distributed with a mean of 6 and a standard deviation of 1.

$$X = \sum_{n=1}^{12} X_n = S_{12} = N(6,1) \qquad (3.37)$$

So $(X - 6)\sigma + \mu$ is approximately normally distributed with a mean of μ and a standard deviation of σ.

Hence, failure times with a normal distribution can be approximated by Equation 3.38.

$$t = (X - 6)\sigma + \mu \qquad (3.38)$$

3.3.3: Performing the simulation and convergence

In this thesis a distribution is assigned to each event in the model in question and the time to the event sampled via the methods described above, with a new event time sampled in each instance. Each run of the simulation is performed, and the outcomes are recorded. After many runs the average outcome should converge due to the law of large numbers and the Central Limit Theorem. A discussion of this can be found in the book “Exploring Monte Carlo Methods”[124]. It is at this point that no further simulations are required. A computer program can be written to run the simulation; however, this method can be time consuming if a large number of runs is required before the outcome converges. In this thesis a Monte Carlo simulation is used to obtain convergent numeric results for the models created. An example of a Monte Carlo simulation is given in the Petri net section of this chapter.

3.3.4: Discussion

There are several advantages to Monte Carlo simulation:

- Monte Carlo simulation does not assume independence between failures or constant failure rates.
- It is possible to model the system in any level of detail.
- Monte Carlo simulation is a flexible method that can be applied to many situations and produce many different outputs.
- It is possible where an analytical solution cannot be found.
- A model with statistical distributions can be solved via Monte Carlo simulation.
- Monte Carlo simulation can account for the random nature of failures.

On the other hand, there are some disadvantages of Monte Carlo simulation:

- Monte Carlo simulation of a model may require large computational power because many simulations may be needed before convergence is reached.
- A large quantity of random numbers must be generated.
- Solutions are an approximation.

Monte Carlo simulation is a powerful tool that can be applied in situations where there are non-constant failure rates or dependences between failures. This is applicable to an ageing underground railway where the failure rate may be changing with time or there are dependencies between component failures.

3.4: Petri nets

Petri nets are used in this thesis to model component ageing, failure, inspection and maintenance and the impact of this on system level failures. System level asset management strategies are also modelled with a Petri net approach.

A Petri net is a digraph with two types of node, known as places and transitions, and objects called tokens which can move in a certain way following the 'firing rule'. [117] The tokens in a Petri net have no assigned meaning, this means that Petri net modelling is flexible and can be applied to a large number of situations. Delay times can be assigned to the transitions in the model in what is known as a *Timed Petri net.* A probability model can also be associated with the transitions in the Petri net in what is known as a *Stochastic Petri net.* In this thesis Stochastic Petri nets are implemented. In a Stochastic Petri net any distribution of times to failure or repair events can be used. This section gives a basic description of the stochastic Petri net methodology as well as a simple application.

3.4.1: Petri net symbols

Transitions

Transitions represent an event or process and are drawn as a rectangle in a Petri net. They often have an associated delay time.

Places

Places represent conditions needed for the transition to occur such as the available resources or a state of the system. They are represented by a circle in a Petri net. The transitions are connected to places by arcs.

Figure 3.6 shows an example of a transition with three connected places, where the transition has a delay time associated with it. There are many different meanings that can be assigned to places and transitions. One application could be, input places representing the resources needed, the transition representing a task and the output place representing the products produced. In another application, the input places can represent the initial state of a component, the transition represent the ageing of the component and the output place represent the end state of the component [125]. This flexibility in the modelling approach allows it to be applied to a wide range of scenarios.

Tokens

Places can be marked by tokens which are denoted by a small black circle. When a place is marked it represents a truth in the condition. For instance, if a place represented a component state, when it is marked the component is in this state. Tokens can have different meanings depending on their location in the Petri net. For instance, in another location a token may represent a completed inspection or maintenance action. A transition is activated for firing when all the input places are marked by tokens. Figure 3.6 shows several places linked by a transition where the place P1 is marked by a token. Petri net models have an associated initial marking which determines the initial state and conditions of the system in question.

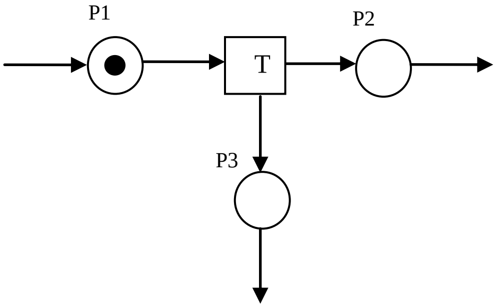


Figure 3.6: An example of a marked place

3.4.2: The Firing rule

The firing rule occurs at transitions and allows tokens to ‘move’ through a Petri net by creating and destroying tokens. For the firing rule to occur all the input places must be marked. In a Timed Petri net, or a Stochastic Petri net, this firing is not instantaneous and occurs after a delay time. Once the period of the delay time is complete all input places will lose a token and all output places gain a token. After firing, the delay time is reset.

The arcs may be weighted, in this case the number of tokens destroyed in the input places is equal to the weight of the arc. Similarly, the number of tokens created in the output place is equal to the weight of the arc.

Figure 3.7 shows a transition before and after the firing rule where the arcs are not weighted, and the transition has a delay time of T.

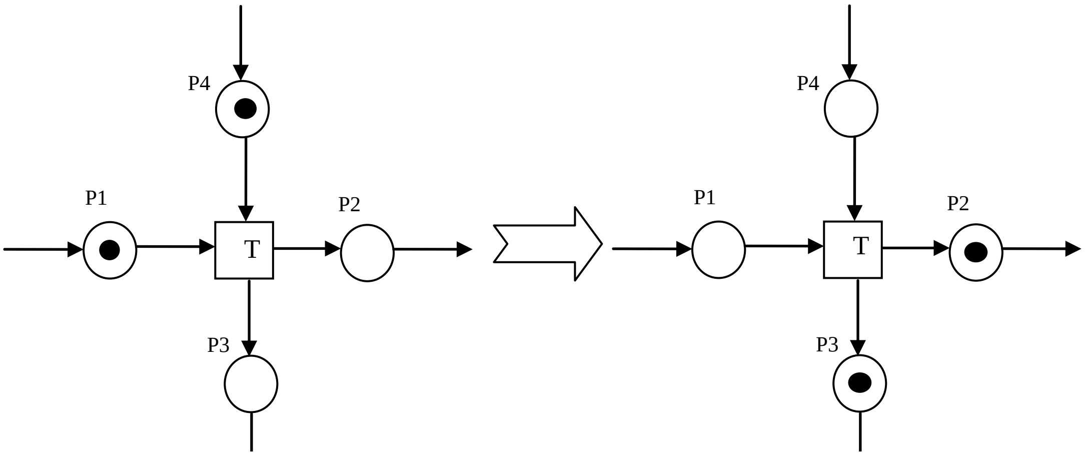


Figure 3.7: An example of the firing rule (before a time T on the left and after a time T on the right)

Inhibitor arcs

An inhibitor arc can be used to represent the situation where a transition cannot fire unless a condition is fulfilled. This is represented by a small circle and a line connecting the place to the transition; in this case the transition will not fire unless the place connected by the inhibitor arc is empty. Figure 3.8 shows an example of an inhibitor arc such that the transition cannot fire while place P3 is marked.

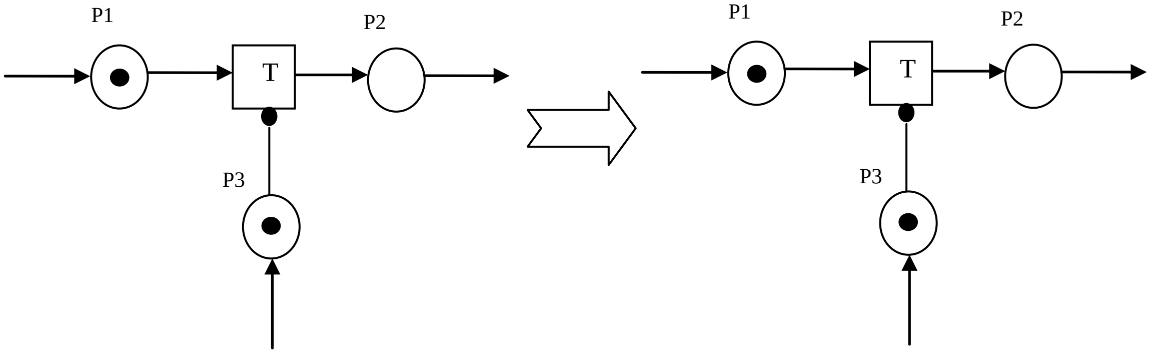


Figure 3.8: An example of an inhibitor arc (before time T on the left and after time T on the right)

A further discussion on Petri net synthesis can be found in the book "Petri nets Picture Book" [126].

There are also several extensions to the typical transitions used in a Stochastic Petri net, which are incorporated in the models presented in this thesis. These include Partial Reset Transitions, Full Reset Transitions, Probability Transitions and Conditional Transitions [64]:

- On the firing of a Partial Reset Transition, certain specified places are returned to their initial marking.
- On the firing of a Full Reset Transition, all places, except for those that are identified as counting actions or time, are returned to their initial marking.
- Probability Transitions represent situations where there can only be one result out of several. In these transitions there are several Output Places, each with an assigned probability to represent the likelihood of each situation occurring.
- Conditional Transitions have several distributions associated with them and a connected Place, known as a Conditional Place. Each time the Conditional Transition becomes active the distribution is chosen for each firing occurrence, based on the number of Tokens marking its Conditional Place.

These extensions are applied to the Stochastic Petri net models created in this thesis, an explanation of how they are implemented is given in in Chapter 4, Chapter 5 and Chapter 6, where they are used.

3.4.3: Analysis of Petri nets

Stochastic Petri nets are implemented in this thesis due to their ability to handle various probability distributions, which can be used to govern failure and repair times for components. There are various methods for the analysis of Petri nets such as the reachability graph method and the matrix-equation approach [127]. However, for large and complex Stochastic Petri nets these methods are intractable [128]. Monte Carlo Simulation of the Stochastic Petri net can be applied for quantitative analysis of the models developed in this thesis [64].

In Monte Carlo simulation of a Stochastic Petri net, the delay times are sampled randomly, from the probability distributions associated with the model, each time that firing of the transition in question is enabled. The tokens are then created and destroyed following the firing rule. The number of tokens arriving at significant places, or the duration that they remain in specific places, can be monitored. On many runs of the simulation, the average marking, or duration of marking, for these places will converge to an average value and this average value can be used to give information about the system. A computer program can be written to carry out this analysis.

Figure 3.9 shows a typical graph of Petri net convergence. This shows how the average metric converges with the increasing number of runs of the model.

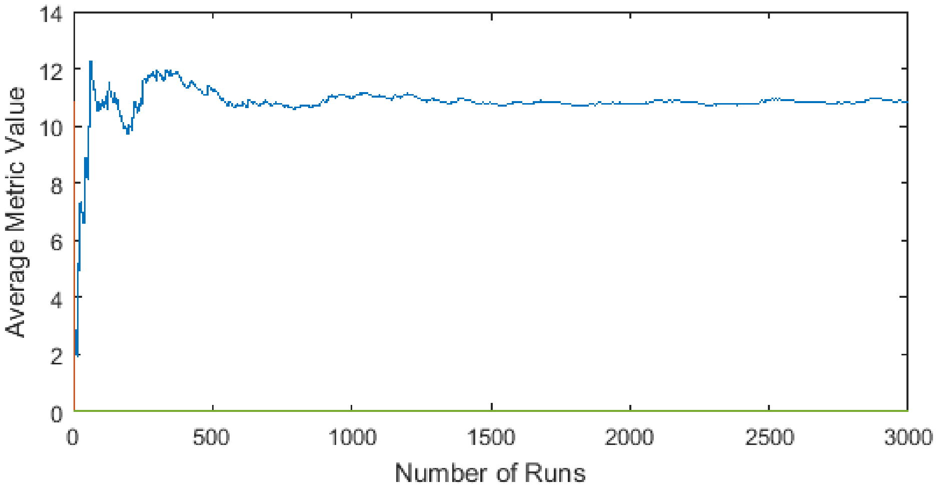


Figure 3.9: A graph showing how the average metric value converges with the number of runs.

Here, the average metric value is 10.55 units after 3000 simulations. The location where the model has reached a sufficient level of convergence can be found. At this point, the average mean value of the outputs only changes within a desired tolerance, with an increased number of runs. In this thesis, the reduction of the 95% confidence interval on the model outputs to a desired tolerance, is used to check for convergence. In addition, a method is presented to find the rate of convergence of the model. Here, the error on the mean value, with each run, is approximated as the difference between the upper and lower 95% confidence limit, and the rate of change in this error, with an increasing number of runs is found using a logarithmic approach. This is presented in Section 6.4 of this thesis. This method is used to allow a quantification of any uncertainty introduced into the model through its simulation, and to allow the user to calculate the required number of runs to reach a given tolerance.

3.4.4: Discussion

There are several advantages to a Petri net based modelling approach:

- A Petri net based model is often modular in nature. This allows different components to be grouped together.
- Ageing components can be modelled easily as any distribution can be assigned to the time to failure of the components.
- Petri nets have the potential to model complex maintenance and inspection strategies.
- Petri nets are a graphical description of the system and allow the analyst to logically consider how components interact.
- A Petri net based approach is flexible and not restricted to independent events.

There are also several disadvantages to a Petri net based modelling approach:

- Purpose build software may be needed to analyse the Petri net based model.
- The analysis of a large Petri net based model can be computationally heavy due to the large number of transitions and runs needed for a convergent answer.
- A Petri net based model is more abstract than a Fault Tree or Event Tree based model, making it a difficult tool to describe risk.

A Petri net based model is a flexible approach for modelling component failure, maintenance and inspection. It can overcome many of the difficulties encountered in a Fault Tree based model such as the incorporation of different failure rates. A Petri net based model is also useful for components that are maintained and inspected as it allows many different maintenance strategies to be applied. For an underground railway, with ageing components that have non-constant failure rates and complex inspection and maintenance strategies, this method of is suitable. In this thesis a Stochastic Petri net based modelling approach is used to model component and system level ageing, failure, inspection and maintenance.

3.5: Proposed Methodology

The existing methodologies presented in this chapter are combined to give a proposed methodology that is applied to model two systems in this thesis: a railway switch and crossing (S&C), in Chapter 4, and a fire protection system, in Chapter 5. The steps of the methodology proposed in this thesis are as follows:

1. Use a Fault Tree to analyse the failure modes of the system. This facilitates the decomposition of failure modes to basic events, which can be related to component state, informing more detailed modelling of the system at the component level. This method is selected as the initial step as it allows the user to explore Boolean logic expressions of events that can cause a system failure and allows the approach to build on current Fault Tree analysis methods, present in industry.
2. Split the system into modules, containing individual components, or groups of components if components are closely coupled. This gives a framework for the component level models, which can include any dependencies in component failures, such as those introduced through maintenance actions. This approach is suggested as it allows the model to be structured in a logical manner.
3. Split each component condition into several discreet states to allow the modelling of the component throughout its lifecycle. States should be chosen that reflect different real world actions associated with the component. For example, there should be different states where different maintenance actions, inspection actions or restrictions to component use are required. This method is used so that different phases of the component lifecycle can be modelled, as it ages.
4. Define inspection strategies at the component level. These strategies can include imperfect inspection and different inspection methods at different times. In addition, states where inspection identifies an existing failure can be included. This method is chosen to allow modelling of inspection actions at the component level and to enable activation of maintenance actions or restrictions on the system use, which are dependent on a discovered failure or degraded component condition.
5. Define the maintenance actions for each component state, either age-based, or condition-based. Where age-based strategies are incorporated, the component can be replaced after a defined time interval from the most recent maintenance action. Condition-based strategies can be implemented based on any revealed failures or degraded states found through component inspection. This step facilitates preventative maintenance actions and repair actions.
6. Build a Stochastic Petri net model for each module, incorporating component level ageing, failure, inspection and maintenance. This models the component condition, and allows the collection of results relating to life-cycle cost. The Petri Net methodology is selected for this stage as it can be applied to systems with complex inspection and maintenance strategies and is flexible at modelling component failure rates. This allows different failure rates to be modelled for a component, including failure rates that change with increased preventative maintenance actions.
7. Define the system level logic, such as combinations of failures leading to a system failure, and system level maintenance and inspection strategies. The system level failure logic is informed by the Fault Tree method in the initial step of this methodology. The maintenance and inspection strategies incorporate the interaction between multiple component inspection and maintenance methods, considering their impact at a system level.
8. Combine the component level modules using the system level logic. This step is implemented because it combines the component level modules across the system, to model scenarios

where multiple components fail at the same time, causing a system level failure. In addition, this step facilitates system lifecycle cost analysis, based on the lifecycle cost of components across the system.

9. Assign distributions to the transitions in the model to represent the system and component level ageing, failure, maintenance and inspection. These distributions can be based on available data or expert opinion. Any distribution can be applied at this stage, as long as it can be inverted, or there is a method to approximate it's inverse. This step allows the quantification of the model, and allows developed models to be adapted to specific systems. Different maintenance and inspection strategies can be tested by altering the distributions related to these within the model.
10. Simulate the Petri net model via Monte Carlo Simulation to give the probability or frequency of the failure modes of the system. In addition, this step gives information on the number and type of inspection and maintenance actions across the system, which can be used for life-cycle cost analysis. Monte Carlo Simulation is selected for this analysis as it converges to the true solution and can be used where an exact analytical solution cannot be found due to a complex model structure. The error associated with the convergence of the model can also be analysed, so that a sufficient number of runs are completed such that the final solution closely approximates the true solution.
11. Event Tree analysis is used to gain a measure of the system risk from these model outputs, by combining failure event frequency with estimated consequences. Event Trees are selected at this stage to tie in with existing analysis already present in industry. The method includes a clear representation of the logic used to combine event frequency, with the probability of failure of enabling events and consequence analysis in order to predict risk.

This methodology is not application specific. In order for this approach to be applied to a system the following characteristics of the system are required:

- Components within the system age and then fail; their condition does not improve with time.
- System failures are due to component failures and/or human operation failures. Failures caused by external factors such as natural disasters or deliberate malicious human actions are not included.
- Component condition within the system can be characterised into states of either the working state and failed state, or a number of discreet states from working through to failure. The time between these states can be characterised by a distribution.
- The maintenance and inspection of the components follows some definable logic, however consistent random inspection and maintenance can be modelled.
- For use of the model for life cycle analysis cost of inspection and maintenance actions must be quantifiable.

This methodology is implemented in this thesis to explore its capabilities. In Chapter 4, various system level maintenance strategies are applied to demonstrate the capacity of this methodology for providing a risk-based asset management decision making tool. In Chapter 5, a phased system level maintenance and inspection strategy is developed, such that the system level asset management strategy applied depends on the age of the system. In Chapter 6, the model developed in Chapter 5 is combined with an Event Tree analysis to provide the basis of a risk-based optimization of the phased asset management strategy. For clarity, the additional methodologies applied to these different chapters are described prior to their implementation. The logic of the models presented in this thesis was validated through expert opinion and the accuracy of the Petri Net simulation software was validated against expected results with test cases. The convergence of the models for sample results was also checked in each implementation. Sample data values are used for demonstration of the modelling methodology, with real world data, models developed through this approach can be validated further against expected outcomes.

3.6: Implementation

A generic software has been written to enable the quantitative analysis of the models developed in this thesis. Figure 3.10 gives a flowchart of the developed code. The software was validated with unit testing, including validation of the model state following firing of each transition type. Initially, the structure and parameters of the models are stored in spreadsheets in Excel. For each model there is a spreadsheet containing the data for the places in the model and a spreadsheet containing the data for the transitions in the model. The spreadsheet for the places contains information on: the individual identifier of the place, its initial marking, whether it is used to count outputs of the model and if it is returned to its initial marking on full reset of the model. The transition spreadsheet for the model contains information such as: the individual identifier of the transition, the type of transition, the input places, the output places and any places that inhibit the transition. The distributions, or probabilities, governing these transitions are also included. Distributions for the transitions that are incorporated into the software are: normal, lognormal, exponential or 2-parameter Weibull distributions. The software can be extended to include further distributions. There are several categories of transition included in the software:

1) Stochastic transitions with an associated probability distribution,
2) Timed transitions with a constant delay time, either zero or positive,
3) Probability transitions where there is a choice of outcomes, each with a probability of occurrence,
4) Partial reset transitions with an associated probability distribution that reset specified places within the Petri net on firing,
5) Full reset transitions, that reset all places within the Petri net, except those that are identified as counting relevant outputs of the Petri net,
6) Conditional transitions that have several associated probability distributions, where the probability distribution is chosen based on the marking of an associated conditional place, or places,
7) Global transitions with an associated place, one or more assigned periodic interval, and a conditional place that can be used to vary the assigned periodic interval.

For partial reset transitions, the spreadsheet also includes data on any places that are reset by each transition. For conditional transitions, the spreadsheet contains data on the number of conditional places and how the marking of these impacts the governing distribution.

Global transitions are included to decrease the simulation time of the model; they can be used to remove short periodic loops within the model which are computationally expensive, such as inspection loops. Here, when a linked place is marked, the global time within the simulation is extracted to find the firing time for the transition, as given in Equation 3.39. This can be used to find the time until the next inspection on the failure of a component. The periodic interval in these transitions can also vary depending on the marking of a conditional place, as with a conditional transition.

$$T = T_i - (T_{global}\ mod\ T_i) \qquad (3.39)$$

Where T is the delay time for the transition, T_{global} is the global time of the system and T_i is the inspection interval. A reference to the implementation of this can be found in Chapter 4 of the thesis “Rail Track Geometry Degradation and Maintenance Decision Making” [129].

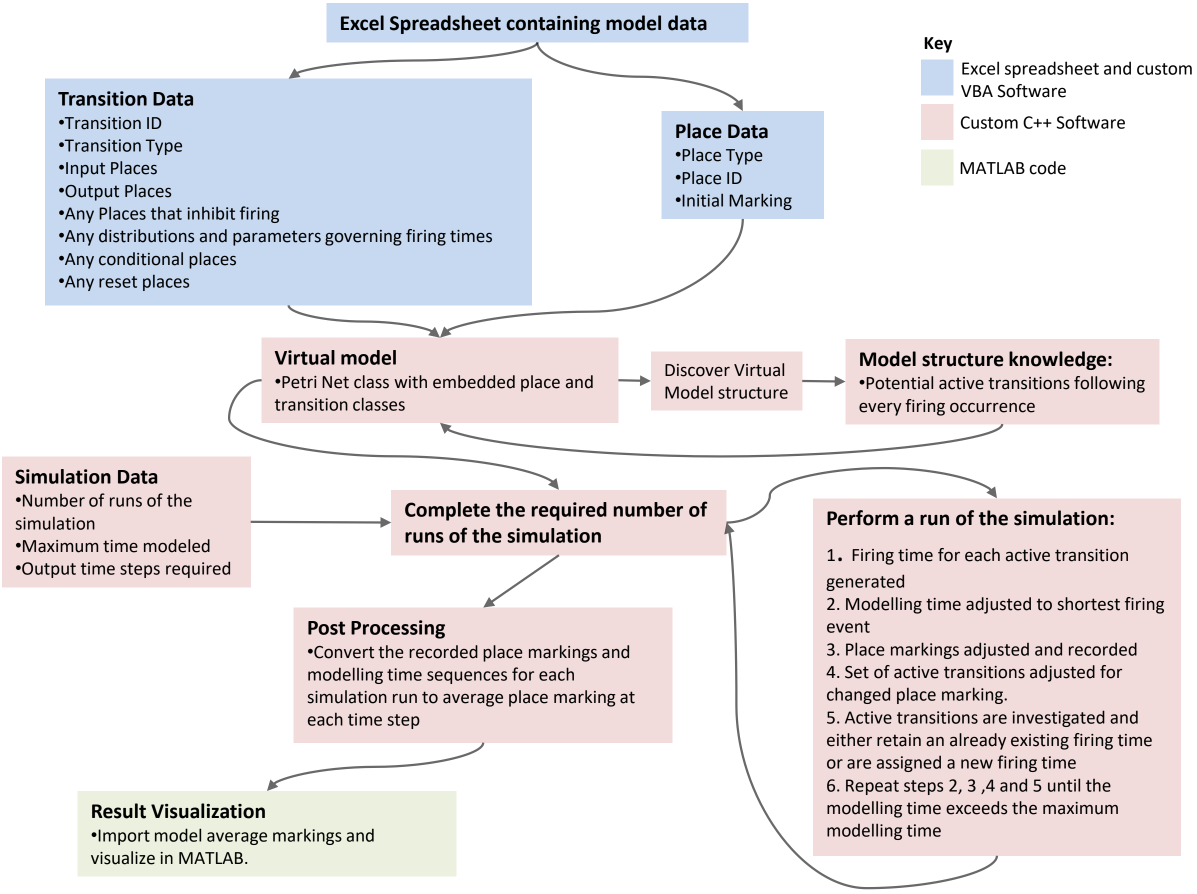

Excel Spreadsheet containing model data
Key
Excel spreadsheet and custom VBA Software
Custom C++ Software
MATLAB code
Transition Data
•Transition ID
•Transition Type
•Input Places
•Output Places
•Any Places that inhibit firing
•Any distributions and parameters governing firing times
•Any conditional places
•Any reset places
Place Data
•Place Type
•Place ID
•Initial Marking
Virtual model
•Petri Net class with embedded place and transition classes
Discover Virtual Model structure
Model structure knowledge:
•Potential active transitions following every firing occurrence
Simulation Data
•Number of runs of the simulation
•Maximum time modeled
•Output time steps required
Complete the required number of runs of the simulation
Perform a run of the simulation:
1. Firing time for each active transition generated
2. Modelling time adjusted to shortest firing event
3. Place markings adjusted and recorded
4. Set of active transitions adjusted for changed place marking.
5. Active transitions are investigated and either retain an already existing firing time or are assigned a new firing time
6. Repeat steps 2, 3 ,4 and 5 until the modelling time exceeds the maximum modelling time
Post Processing
•Convert the recorded place markings and modelling time sequences for each simulation run to average place marking at each time step
Result Visualization
•Import model average markings and visualize in MATLAB.

Figure 3.10: A flowchart of the developed code for model simulation

The data from the model spreadsheets is extracted from Excel and stored as '.txt' files which are in-turn read into a piece of software for the simulation of the Petri net, written in C++. This software requests the number of transitions included in the model, and the name assigned to the whole Petri net model, and from this reads in the Petri net model data. There are several classes within the software, including the different Petri net models which are each assigned an individual identifier. This enables multiple models to be simulated separately by the same code if required. Within each Petri net model are several other classes incorporating the logic of the Petri net such as the Transition class and Place class, and their associated rules. The software requests two further inputs from the user. Firstly, the total time required to be simulated for the system in question and the required resolution of the solution, such as a requirement to model the system for 40 years with a resolution of solutions to the nearest week. Secondly, the software requests the number of runs required for a Monte Carlo Simulation of the model.

There are two types of time that are associated with the simulation of a Petri net model. The first is the physical time associated with the model, for instance the period of interest for the system in question. The second type is the computational time that it takes for the simulation of the Petri net to be completed. Both are presented for each model within this thesis with a discussion of the computational time taken for the model simulation made throughout.

The Monte Carlo Simulation of the Petri net initiates by setting the marking of the places to the initial marking and setting the global system time to zero. This is repeated at the start of each run of the simulation. The Petri net is then analysed to find any transitions within the model that are active for firing. A delay time is assigned to these active transitions and the global time of the run of the simulation is increased, in the same quantity that the delay time of each of these transitions is reduced.

If, and when, any of these active transitions reach a point where a delay is less than, or equal, to zero, then the transition fires changing the marking of the Petri net. Following this firing, the Petri net is reanalysed to find any changes to the set of current active transitions within the model. Where current active transitions are unaffected by the most recent firing, the delay time is conserved for these transitions. In the case where there are multiple transitions that reach a delay time of less than, or equal, to zero, at the same time, then the transitions fire in a random order, which is altered at each occurrence. The outputs of the defined places are recorded at each time interval for each run of the simulation, along with the time stamp for each of the runs of the simulation. This is used to find the average outputs.

There are two techniques used within the code in order to decrease the computational time of simulation of Petri net models. Firstly, the global time of the simulation is increased in a non-linear manner, resulting in an individual time stamp for each of the runs of the simulation. The time step taken at each point is determined by the shortest delay time of the active transitions. This is implemented as there are no changes in the Petri net marking between these points. The second modification concerns the part of the software that searches the Petri net for any transitions that are active for firing. Prior to simulation, the software searches the Petri net to gather information about the structure of the model. Associated with each transition, there is a set of further transitions that may be impacted if the transition is to fire and these are gathered by the software. Following firing of a transition within a run of the simulation, the software checks this associated set of transitions to avoid the need to search all transitions in the model. A full active transition search is still completed on initiation of a run and on full reset of the Petri net model. This reduces the costly computational effort of repeated searching of the whole Petri net model for active transitions.

Outputs of the model are taken from the marking pattern of the places identified in the place spreadsheet. Due to the non-constant time stamp for each run of the simulation, some analysis is carried out on the outputs of the model at the end of the runs of the simulation. The marking of each place for each run is converted to the marking over the full time for the user defined time steps to enable the combination of the results, over different runs, to give an average solution. These outputs are stored as ‘.txt’ files, with one file for each average marking pattern.

Following this, this average marking pattern for each output can be analysed. This is completed in MATLAB for the models presented in this thesis.

3.7: Parameter assumptions

Since the parameters within the models in Chapter 4, Chapter 5 and Chapter 6 have been assumed, there are some model outcomes in each case that demonstrate a higher sensitivity to assumptions. For instance, when modelling rare events, a falsely assumed parameter can influence the outcome of the model with respect to the rare event. An example of this could be where a component failure directly causes a system failure; the system outcome can be highly sensitive to the parameters governing the rate of the component failure. This is discussed further in the summary section of each of the chapters, along with the contributions of each approach despite the assumptions. In addition, a discussion of the data required for application of each model, in order to remove these assumptions, is given, along with suggestions for how this might improve the presented model. An example of how data can be processed in order to determine the lifetime distributions can be found in Rama and Andrews, 2013 [130].

3.8: Contributions

Contributions of this chapter include a novel approach for hazard or risk modelling that can be applied to a railway industry. The approach combines existing risk analysis and asset management techniques, using a Petri net methodology. This method allows in depth modelling of components with changing degradation rates and complex asset management strategies. This is an improvement on current methods used in industry, which are more static in nature and struggle with dependencies introduced through maintenance actions and with modelling changing failure rates as components age. A bespoke research code for the analysis of such models was also developed, with the inclusion of numerous modelling options including both different transitions and distributions.

3.9: Conclusion

This chapter has introduced the methodology proposed in this thesis to model risk. The approach combines the Fault Tree and Event Tree methods with a Petri net modelling technique, solved by Monte Carlo simulation. This methodology has a focus on modelling ageing assets and considering maintenance and inspection strategies which are used to manage their operation. A description of the bespoke software, written for the analysis of the models in this thesis, has also been provided.

The methodology described in this chapter is applied to two example systems in this thesis to consider the impact of maintenance and inspection strategies on system state and risk. The beginning of each of the following two chapters contains a system definition and a description of modelling approaches applied. This methodology is then extended further in the latter stages of this thesis. Methods are given for the optimization and reduction of Petri net models.

Chapter 4 Modelling Derailments on an Underground Railway due to an S&C Failure

4.1: Introduction

In this chapter, a Petri net modelling approach is presented to predict the derailment occurrence caused by failure in a multi-component Switch and Crossing (S&C). A holistic methodology is adopted such that components from an S&C are divided into subsets of interconnected modules. Degradation within each module is idealized through a sequence of discrete states of wear, until the final failure occurs. Monte Carlo analysis is used to numerically evaluate the resulting Petri nets, thus obtaining the frequency of derailment using simulations. Through this methodology, different maintenance strategies, such as partial replacement, complete replacement, and opportunistic maintenance, can be tested to evaluate their influence on the frequency of derailment, as well as the whole life cost [131]. This chapter presents the Petri nets proposed for modelling derailment at an S&C. In addition, different asset management strategies are implemented for sample model inputs. The model outputs are presented for each sample case.

S&Cs are considered key assets within a railway system, as they enable flexible track operation. They allow different railway lines to be connected by guiding trains between different track sections. Figure 4.1 gives an example of a simple S&C [132]. The S&C can be mounted on an individual concrete block or on ballast with sleepers of either wood or concrete. This is placed on a flat layer of the subsurface of the track.

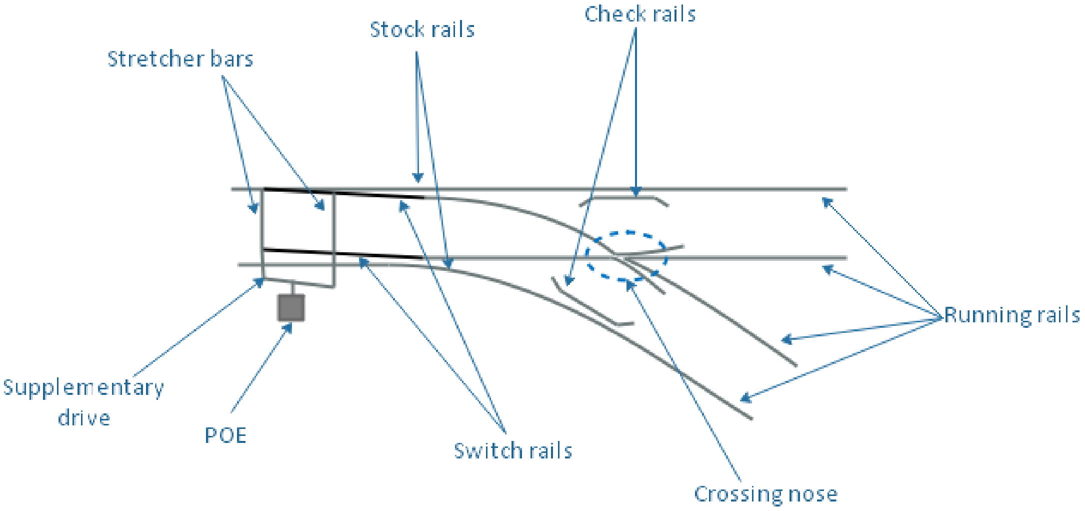


Figure 4.1: A diagram of an S&C

An S&C consists of several different components including moveable switch rails which allow the train to change lines [133]. There are two switch rails that are moved together by the Points Operating Equipment (POE), until one switch rail comes into contact with a fixed stock rail, allowing the wheels of the train to pass in the desired direction. There are stretcher bars that connect the two switch rails to ensure that they maintain the same gauge, these stretcher bars are connected to each other and to the POE by a supplementary drive. Slide chairs lie under the switch rails to allow them to move into position easily. Once the switch rails are in position they are locked via a locking device to ensure that there is no movement of the switch rails as the train passes.

In addition, intermediate and running rails are used to allow passage before, through and after the S&C. Intermediate rails run after the switch rails and inside the stock rails and running rails sit before and after the S&C. The crossing nose allows space for the wheel flange to pass at the point where the diverging rail paths cross. This S&C multi-component structure gives rise to a non-constant running surface and higher stiffness, in comparison to plain track. The non-constant running surface, and higher stiffness, can lead to the S&C experiencing higher rates of deterioration due to the resulting higher impact loads [134] [135]. In addition, there is an increased likelihood of derailment, due to the non-constant wheel rail interface through the S&C, if a failure is to occur [136].

Maintenance costs are high due to the need for specialist equipment. Hence, a clear understanding of the S&C derailment risk is required alongside a method for testing different maintenance strategies to allow safety standards to be maintained while minimizing the whole life costs.

The research presented in this chapter gives a new model that can be used to predict the frequency of derailment caused by a failure in an S&C. Within the model, different maintenance and inspection strategies can be tested to consider their effect on the derailment frequency predicted by the model. The results of this model can be used in conjunction with a traditional Fault Tree and Event Tree based approach, giving overall predictions of risk on an underground railway.

In order to discover the type of failures that have historically caused derailments at S&C, data was taken from the Federal Railroad Administration ‘Track Safety Data Base’, to give an analysis of the causes of derailments at S&C and the impact that these had [137]. All railway selections were included across all regions. The time frame was set from January 1975 until August 2018 and considered all derailments related to S&C or Track Appliances. There were 12,306 derailments in this category, following the removal of elements not related to S&C failure. In this period there were 290 non-fatal casualties recorded and 2 fatal casualties recorded. Although this data does not directly link to S&C failure on an underground railway, it can be used to identify common trends in the type of S&C failure that can result in a derailment and for that purpose it is included in this section of the chapter.

Figure 4.2 gives the proportion of derailments by cause. Figure 4.3 gives the proportion of financial cost of a derailment by cause. Figure 4.4 gives the number of non-fatal accidents by cause.

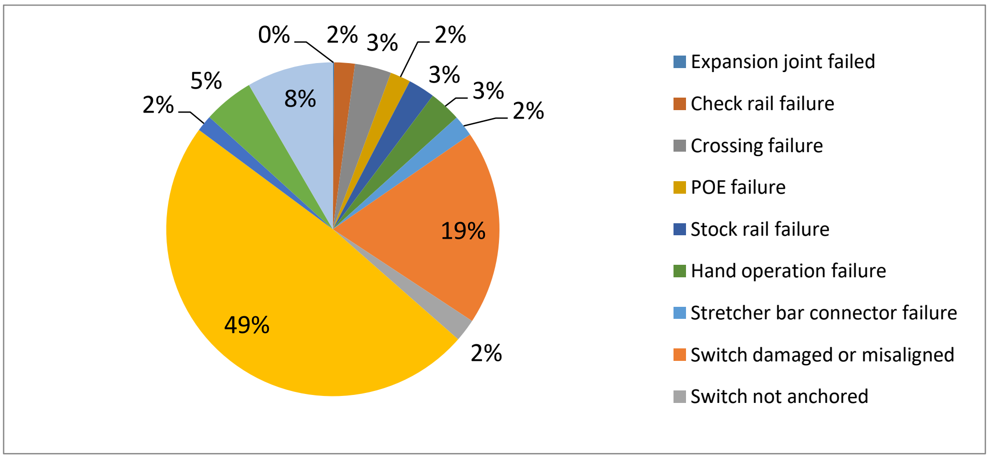


Figure 4.2: Causes of Derailments at S&C

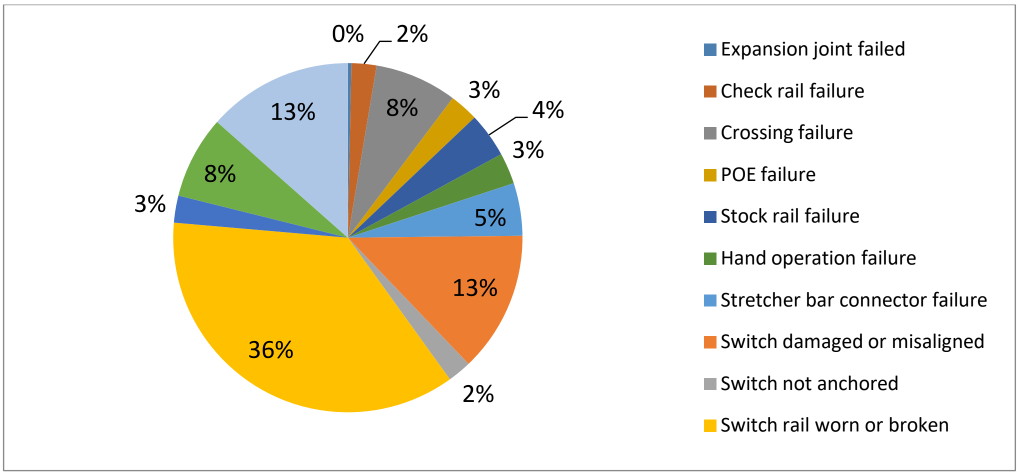


Figure 4.3: The reported financial cost of derailment by cause, at an S&C

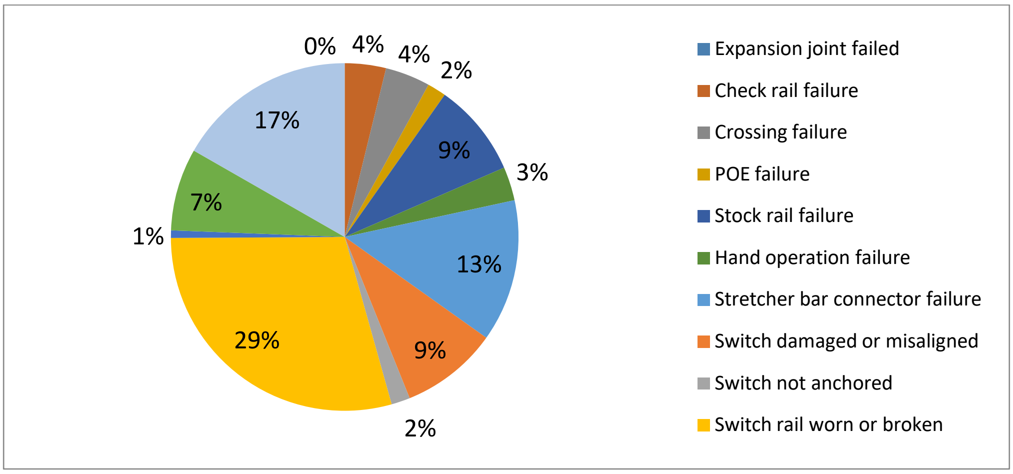


Figure 4.4: The number of non-fatal casualties by cause of a derailment at an S&C

For the data recorded there were two fatal casualties: one resulting from a worn or broken switch rail and one from a non-specific S&C defect. This data demonstrates that there are several component failures that can lead to a derailment. Firstly, failures related to the switch rails, including break or wear of the switch rail and damage or misalignment of the switch rail contribute largely to S&C derailment. Another large contributor is the stretcher bar and supplementary drive condition. Other contributions come from stock rail failure or crossing nose failure.

The model presented in this chapter has a focus on modelling complex degradation and asset management strategies. It explores the capability of the modelling approach proposed in this thesis to accommodate different degradation processes and associated maintenance actions for a component, along with multiple inspection methods that have a probability of failure. This chapter also focuses on incorporating imperfect maintenance options and the impact that this has on the system state. In addition, system level strategies are included such as opportunistic maintenance and full system renewal.

4.2: Method

4.2.1: Modelling Application

A modular Stochastic Petri net-based approach is proposed in this chapter to model the derailment occurrence caused by component failures. This chapter presents models for component condition, inspection and maintenance strategies. Following this, the models for the derailment occurrence are presented. The model can be applied to an S&C in both and underground and over ground system.

Figure 4.5 gives a Fault Tree for a derailment occurrence at an S&C, this Fault Tree analysis was completed as part of the work presented in this chapter. This analysis has been based on past studies and the failure data presented earlier in this chapter. Here, the derailment can occur through either over speeding or through a failure of the S&C system. In the latter case, for a derailment to occur, the train must cross the S&C with sufficient speed to cause a derailment and there must be a hazardous failure on the S&C. The S&C system failure states that can cause a derailment are split into four scenarios: the first is a poor geometry of the S&C, the second is wear on the S&C, causing the wheel to climb the rails, the third is a discontinuity in the rails on the S&C, and the fourth is an incorrect position of the switch rails. Each of these scenarios has contributing component failures, which are shown in the Fault Tree and expanded on later in this chapter.

There are two scenarios by which a train can pass over a failed S&C at a sufficient speed to cause a derailment. Firstly, the failure may not have been detected, hence the train passes over at full speed. Secondly, a speed restriction may be in place, but the speed restriction may be insufficient to prevent the derailment, due to further degradation of the system following the application of the restriction. For the models presented in this chapter, it is assumed that if there is a closure of the S&C then no train will pass through.

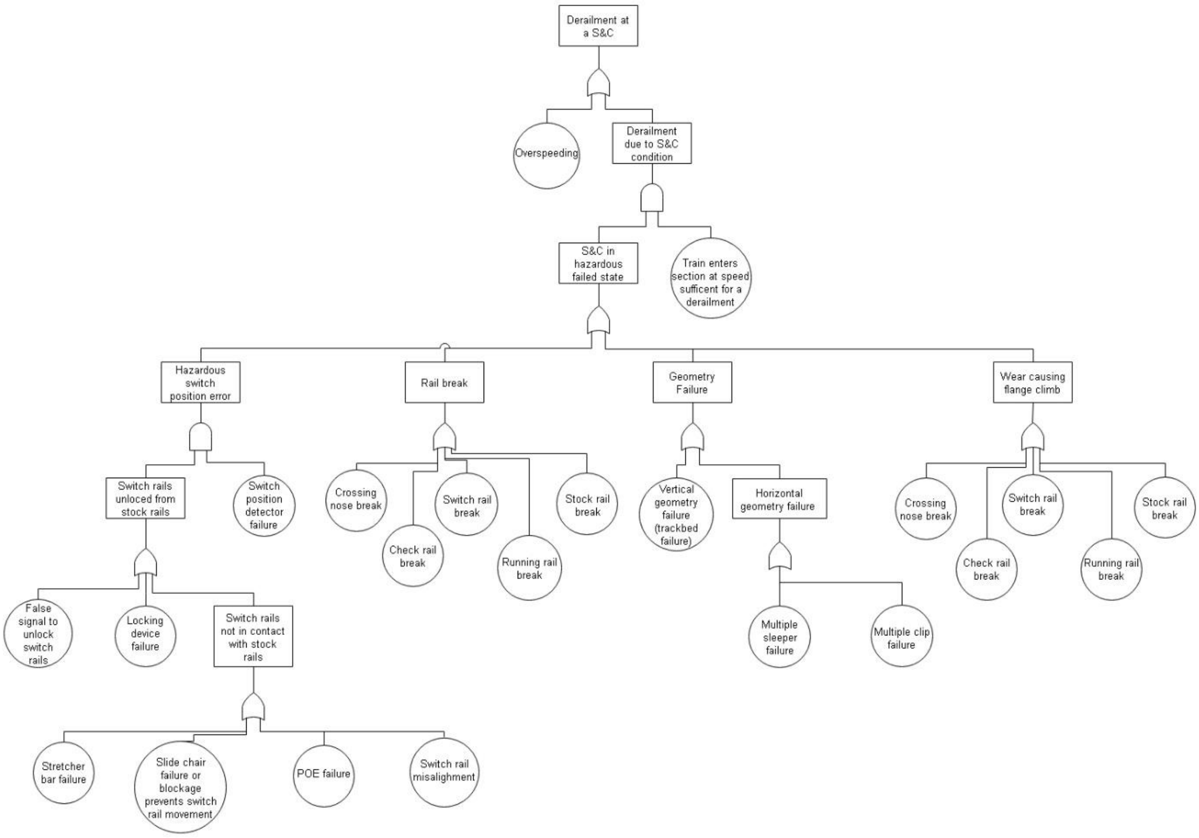

Derailment at a S&C
Overspeeding
Derailment due to S&C condition
S&C in hazardous failed state
Train enters section at speed sufficent for a derailment
Hazardous switch position error
Rail break
Geometry Failure
Wear causing flange climb
Switch rails unloced from stock rails
Switch position detector failure
Crossing nose break
Check rail break
Switch rail break
Running rail break
Stock rail break
Vertical geometry failure (trackbed failure)
Horizontal geometry failure
Crossing nose break
Check rail break
Switch rail break
Running rail break
Stock rail break
False signal to unlock switch rails
Locking device failure
Switch rails not in contact with stock rails
Multiple sleeper failure
Multiple clip failure
Stretcher bar failure
Slide chair failure or blockage prevents switch rail movement
POE failure
Switch rail misalighment

Figure 4.5: A Fault Tree for a derailment at an S&C

There are several methods by which a failure can be detected to prevent a train passing over the S&C either completely, or at full speed. Firstly, a failure may be detected through inspection and testing of the components of the S&C. Secondly, for some components the failure may be detected automatically. For instance, the switch position detector may reveal a failure in the switch position. Finally, in some cases, the failure can be detected by the driver, for example a complete rail break. These are included in the Petri net models presented in this chapter.

This Fault Tree is used as a starting point for the Petri net models presented in this chapter, which expand on this system level failure logic to model the S&C in a dynamic way. For the models developed in this chapter, the structure of this Fault Tree is used to inform the logic that combines component level failures, and the overall state of the system, to model events where a failed condition has occurred, and the train has crossed the S&C at a speed sufficient to cause a derailment. The failed condition events can be found in the third intermediate layer of the Fault Tree in Figure 4.5. The events that feed into these intermediate events are modeled within the Petri net framework for each of these failure events. Hence, the combination of the following risk scenarios with an OR logic, recreates the logic of the Fault Tree. Hence, the Petri net models developed give the following risk scenarios:

1. There is a hazardous switch position and the train has crossed the S&C at a speed sufficient to cause a derailment;
2. There is a rail break and the train has crossed the S&C at a speed sufficient to cause a derailment;
3. There is a geometry failure and the train has crossed the S&C at a speed sufficient to cause a derailment;
4. There is wear with the potential to cause wheel flange climb and the train has crossed the S&C at a speed sufficient to cause a derailment ;
5. Over speeding causes derailment through the S&C.

Each of the basic events in the Fault Tree is modelled. These events are combined within a Petri net structure, which follows the logic of the Fault Tree. Where these basic events are at the component failure level, the condition of the component is modelled with Petri nets, which include the degradation of the components involved, the inspection and maintenance, and the strategy for the application of any closures or speed restrictions, due to the component condition. In the combination of these lower level Petri nets, consideration is given to any applied restrictions or closures, to give the derailment occurrence for each of the risk scenarios detailed above. In this Chapter, Section 4.3.1 gives the component level Petri Net models. Section 4.3.2 gives the system level component inspection strategy model. Section 4.3.3 gives the system level maintenance scheduling model. Section 4.3.4 gives the models that implement the component level models, along with any restrictions applied to the S&C and a consideration of over-speeding, to recreate the risk scenarios detailed above.

In the models presented in this Chapter, the geometry failure intermediate failure event includes contributions from the condition of the ballast, sleepers and clips. The rail break intermediate event includes breaks in the switch rails, running rails, stock rails, crossing nose or check rails. The wear intermediate event includes wear on the switch rails, running rails, stock rails or crossing nose. The hazardous switch position intermediate event includes incorrect alignment, switch rail obstruction, signal failure, locking device failure, stretcher bar or supplementary drive failure, POE failure and dry slide chairs.

4.2.2: Non-traditional Petri Net Transitions used in the Models:

There are several extensions to the typical transitions used in a Stochastic Petri net, which are incorporated in the models presented in this chapter. These have also been cited in Chapter 3 of this thesis. These include Partial Reset Transitions, Full Reset Transitions, Probability Transitions and Conditional Transitions: [64]

- On the firing of a Partial Reset Transition, certain specified places are returned to their initial marking. These transitions are used to represent replacement or maintenance actions. For example, a partial reset transition, modelling the replacement of a rail, returns the marking of all places that model the condition of the rail to the initial marking. Where this initial marking corresponds to the ‘good as new’ state for the rail.

- On the firing of a Full Reset Transition, all places, except for those that are identified as counting actions or time, are returned to their initial marking. These transitions are used to represent full system replacement. For example, on the full replacement of the S&C, all components are assumed to return to the 'good as new' state. When the full reset transition fires, modelling this action, all places corresponding to the system state are returned to their initial marking. Where the initial marking corresponds to the 'good as new' state.
- Probability Transitions represent situations where there can only be one result out of several. In these transitions there are several output places, each with an assigned probability to represent the likelihood of each situation occurring. On the firing of a probability transition, only one output place is marked, and the choice is weighted by the assigned probability. This is used for situations where there is a probability that an action will either be a success or failure. For example, a probability can be assigned to the success or failure of an inspection action. A probability transition can model this such that either the place corresponding to the success, or the failure, is marked following firing of the transition.
- Conditional Transitions have several distributions associated with them and a connected place, known as a conditional place. Each time the conditional transition becomes active the distribution is chosen for each firing occurrence, based on the number of tokens marking its conditional place. These transitions are used where maintenance actions do not return a component to the 'good as new' state. For example, if repeated tamping actions impact the future degradation rate of the ballast, a conditional transition can be used to model the degradation, with a conditional place that counts the number of tamping actions between ballast replacements. The firing time of the conditional transition, that models the degradation, can then vary depending on the marking of the conditional place.
- Global Transitions can be used to replace inspection loops. The transition is assigned a periodic time interval, and if the transition is enabled then the firing time is determined by the remainder of the total simulation time when divided by multiples of the periodic time interval. These transitions are used in the sleeper and clip model, to model component inspection, where there are multiple repeated unit models, to improve the efficiency of the model.

4.2.3: Component Level Model Specification

In the models developed in this Chapter, the components of the S&C are modelled. The condition of the components across the system can impact the system state and combined failures can result in the occurrence of one of the intermediated failure events detailed in the previous section. This section gives a summary of the component models presented in this Chapter and details the process for deciding the number of states modelled for each component.

The components in each Petri net module were identified from past studies [133][138]. In the proposed model, the degradation of each component is represented by the discretization of the component condition into states ranging from the ''as good as new' state' to the failed state. Different components have different numbers of assigned states, based on the existence of measurable metrics to quantify degraded states of the component in question. The number of defined states is chosen based on the existence of measurable quantities to define these states and their associated maintenance actions. For instance, if a component can have the same maintenance action, but applied with two different condition dependant scheduling delays, then two degraded states are included. Likewise, if a component can have two different maintenance actions, depending on the type of degradation, then two states are included. Where intermediate degraded states exist for a component, the states are categorized by measures of each component condition. For example, the vertical track alignment can be used as a measure of ballast condition [139]. The measure used for the condition of each component can vary depending on the model application and data available, however, it must be consistent and provide a good representation of the condition of each component. Threshold values must be assigned to define the upper and lower limits of each discreet state, where they exist.

In the models presented in this chapter, where there are multiple degraded states for a component, the relevant maintenance actions are completed after a time delay which is specified for each detected degraded state. For example, more severely degraded states have a maintenance action scheduled following a shorter delay than less degraded states. The latter states can also trigger the application of speed restrictions or closures. Where condition monitoring of the component is not quantifiable, age-based maintenance, and application of restrictions such as system closure or speed restrictions, has been modelled.

A full description of each model and the associated Petri net is given in Section 4.3 of this thesis. Each component can have a number of states depending on the condition and the type of defect. Figure 4.6 details the options for the assigning of the states to each component defect. The figure shows the four different interventions that are modelled for each component defect: early preventative maintenance, routine preventative maintenance, priority maintenance with speed restrictions and priority maintenance with closure. There are two options for the number of modelled states of each component defect:

1) Option 1: component states can be quantified by a numerical measure for the defect, five states are modelled such that each intervention is triggered if the component defect reaches the corresponding state. Interventions link to the states as follows:
    i) If the component is in state s1, do nothing,
    ii) If the component is in state s2, early preventative maintenance can be applied if opportunistic strategies are implemented.
    iii) If the component is in state s3, apply routine preventative maintenance.
    iv) If the component is in state s4, apply priority maintenance and speed restrictions.
    v) If the component is in state s5, apply priority maintenance and closure.
2) Option 2: component states cannot be quantified by a numerical measure for the defect, only the working and failed states are modelled, but interventions are applied based on age-based intervals. Interventions link to states and age-based times as follows:
    i) If the component has been in use less than the first defined time, do nothing,
    ii) If the component has been in use for the first defined time, early preventative maintenance can be applied if opportunistic strategies are implemented.
    iii) If the component has been in use for the second defined time, apply routine preventative maintenance.
    iv) If the component has been in use for the third defined time, apply priority maintenance and speed restrictions.
    v) If the component is in state s5, apply priority maintenance and closure.

Inspection can be applied to the component in any state.

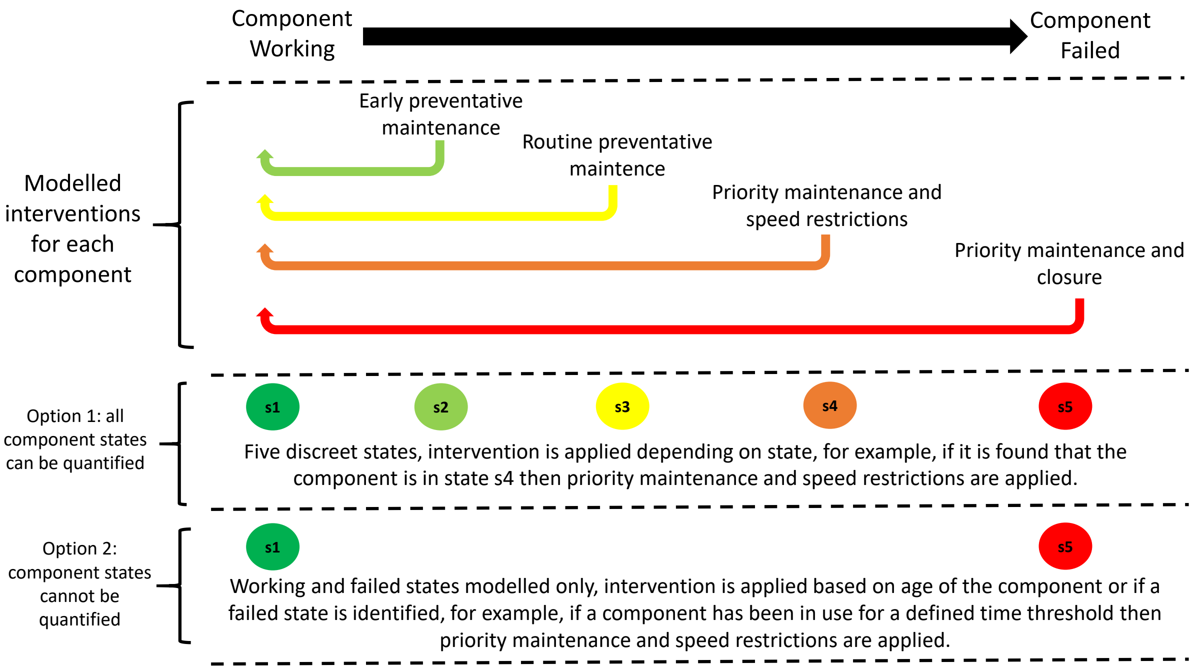

Component Working
Component Failed
Modelled interventions for each component
Early preventative maintenance
Routine preventative maintence
Priority maintenance and speed restrictions
Priority maintenance and closure
Option 1: all component states can be quantified
s1
s2
s3
s4
s5
Five discreet states, intervention is applied depending on state, for example, if it is found that the component is in state s4 then priority maintenance and speed restrictions are applied.
Option 2: component states cannot be quantified
s1
s5
Working and failed states modelled only, intervention is applied based on age of the component or if a failed state is identified, for example, if a component has been in use for a defined time threshold then priority maintenance and speed restrictions are applied.

Figure 4.6: A figure describing component state modelling options

In the models five states are preferred, if the condition of the defect in the component can be quantified, in order from working through failed. So that each state can correspond to a different system requirement as follows: no maintenance, opportunistic maintenance, routine maintenance, priority maintenance with speed restrictions and priority maintenance with closure.

The model also considers cases where a component can have different types of defect, where the maintenance action applied depends on the type of defect present for the component. In this case, each type of defect is modelled with either two or five states, depending on if the states can be quantified. Hence, the different maintenance actions are applied depending on the severity and type of defect. For instance, if a component had two measurable defects, where each defect had a different associated maintenance action, then five states would be used for each defect type. This allows the model to consider cases where a maintenance action improves one aspect of the component condition, but not another aspect of the component condition. Section 4.2.4 discusses the system states further, for cases where the states can be quantified by a numeric measure.

An overview of each component model, and the features for each, follows:

Ballast

It is assumed in this chapter that the ballast condition contributes to the vertical track geometry, but that the remaining elements of the track base remain in a good condition throughout. The model can be extended to incorporate the extra elements if required. The ballast is modelled with five discreet states of degradation, each of which can be detected by considering changes in the vertical track geometry. This is explained further in Section 4.2.4 of this thesis. Ballast tamping and undercutting actions are included in the model. It is assumed that ballast tamping has a negative impact on future degradation rates.

Sleepers and Clips

In the models presented in this chapter, each of the sleepers and clips are modelled individually. To demonstrate the methodology, ten sleepers with ten associated pairs of clips are modelled. This was deemed sufficient to demonstrate the methodology; the number should be extended to the true number in a physical system during application. Each of the sleepers or pairs of clips are modelled with two states: the working state and the failed state. Hence, there is a working sleeper state, a working clip state, a failed sleeper state and a failed clip state. A failed state can be discovered by considering changes in the horizontal track geometry. The behaviour of the population of sleepers and clips is also considered in this model. There are two failure modes assumed for the population of sleepers and clips. If there are two consecutive sleepers, or clips, in the failed state or three sleepers, or clips, in the failed state across the S&C, then the overall condition is deemed to be in the first failed state. It is assumed that in this state speed restrictions are required. If there are three consecutive sleepers, or clips, in the failed state or four sleepers, or clips, in the failed state across the S&C, then the overall condition is deemed to be in the second, more severe, failed state. It is assumed that in this state closure of the S&C is required. These thresholds for defining each failed state can be adjusted when applying the model to a specific S&C. In this chapter, individual sleeper or clip replacement is modelled, where the replacement of the sleepers also includes the replacement of the clips on the sleeper in question. Conversely, clips can be replaced in isolation. In addition, replacement of the population of sleepers and clips is modelled.

Rails

The fixed rails in the S&C, including the running rails and stock rails, are modelled in the same Petri net in this chapter. There are three degradation mechanisms modelled for the rails relating to: sub-surface cracking, loss of railhead material and surface cracking, rolling contact fatigue and wear. For each degradation pathway there are five discreet states, from the working state through to the failed state. These states are discussed further in Section 4.2.4. The categories are detailed further in Appendix 1, along with suggested state quantification methods. Depending on the discovered condition of the rails, including the level of degradation and the category of the defect, different maintenance actions are applied. In this chapter, rail grinding, or rail replacement is modelled. The chosen action depends on the category of the defect. For instance, rail grinding is applied for low level surface defects. From these rail models, two failure modes are gained, one relating to a rail break and one relating to a critical level of wear. The failure mode is dependent on the defect type that caused the failure. For instance, in this model, wear and subsurface cracking are both modelled, if the wear reaches a critical point then a wear failure event is triggered and if the cracking reaches a critical point then a rail break failure event is triggered.

The switch rails are modelled separately. The model for the switch rails also includes a pathway representing the alignment of the switch rails to the stock rails, and a corresponding maintenance action representing the adjustment of the switch rail alignment. Consequently, a third failure mode is gained from this model, which represents a failure in the switch rail alignment.

Crossing nose

Three degradation pathways are modelled in this chapter for the crossing nose. These are surface cracking, sub-surface cracking and deformation. Each of the degradation pathways has five discreet states and depending on the discovered condition of the crossing nose, including the severity and the category of defect, different maintenance actions are applied. In the model, replacement, grinding and welding actions are included. Two failure modes are gained from this part of the model, one representing a break in the crossing nose and the other representing a critical level of wear.

Check rails

In this chapter, two degradation pathways are modelled for the check rails. These are deformation of the check rail and lateral cracking. Five discreet states are modelled for each degradation pathway. Depending of the discovered state and category of any detected defects, grinding or replacement of the check rails is applied. Two failure modes are gained from the model, one representing a break in the check rail and one representing a critical level of deformity in the check rail.

Stretcher bar

The stretcher bar model has five discreet states from the working state to the failed state. These states represent the degradation of the stretcher bar at worsening levels. This degradation includes cracking, bending or corrosion of the stretcher bars. The replacement of the stretcher bars, with an urgency dependant on the severity of the state, is modelled in this chapter. One failure mode is gained from this model, representing a failure in the stretcher bar such that the switch rails do not maintain the correct gauge.

Slide chairs

In this chapter, two degradation pathways are modelled for the slide chairs, each with five discreet states. The first pathway represents cracking, wear or corrosion of the slide chairs. The second degradation pathway represents deterioration in the lubrication of the slide chairs. In addition, a transition is included that represents a blockage of the slide chairs by an external source. Two maintenance actions are included in the models. The first is replacement of the slide chairs and the second is clearing and lubrication of the slide chairs. These are applied depending on the severity and category of a discovered slide chair defect. One failure mode is gained from the model, representing a condition of the slide chairs such that the switch rails are prevented from moving correctly, due to either a blockage or dry slide chairs.

POE and locking device

Five discreet states of the POE are modelled in this chapter, from the working state through to the failed state. Replacement of the POE is included, depending on any detected degraded states. A failure mode is gained whereby the POE fails to move the switch rails into full contact with the stock rails.

Two states are modelled for the locking device: the working state and the failed state. Replacement of the locking device upon the discovery of a failure is modelled, along with age-based maintenance at three adjustable time intervals. A failure mode is gained whereby the locking device fails to lock the switch rails in place.

Switch position detector

Two states for the switch position detector are modelled: the working state and the failed state. Replacement of the switch position detector, on the discovery of a failure is modelled. In addition, three age-based maintenance actions, at adjustable intervals, are included. A failure mode is gained where the switch position detector is in the failed state and hence cannot reveal a failed switch position.

External signal failure

An external signal failure is also modelled in this chapter. There are two failure modes for the signal failure. The first represents a failure when the switch rail is falsely unlocked. The second represents a safe failure when the switch rail remains locked in place. This model can be expanded to model the signalling system in more detail; however, this is deemed outside the scope of this chapter.

The next section of this chapter gives a description of the quantification method for the different system states described in this section, where the states of the component can be identified by a measurable quantity.

4.2.4: Quantification of system states

For the Petri net models presented in this chapter, the condition of some of the components is discretized into five different states, as described in Section 4.2.3. In these cases, the condition of the component is classified by a measurable value (σ), which lies between threshold values that define boundaries of each state (σ_0, σ_1, σ_2, σ_3, σ_4, σ_5). Each component can have different maintenance actions, or maintenance scheduling delays, depending on the severity of the components' state. Five possible states are included in this model to enable five discreet action sequences on the component, depending on the component condition. These five resultant action sequences are described in Table 4.1. The model can be adjusted to include extra states if there are further associated action sequences, at the expense of increasing the model complexity.

	Action sequences modelled
State I	Regular inspection of component
State II	Optional opportunistic maintenance and regular inspection
State III	Maintenance and regular inspection
State IV	Maintenance, speed restrictions and regular inspection
State V	Maintenance and closure

Table 4.1: The component condition dependent action sequences for the model

This discretization into five states, governed by a measurable threshold, is applied in several areas of the model, with the following measures suggested for the value of σ. As an example of this quantification, the ballast states can be quantified by a measured difference between the track position and ideal position. Here σ is this measured distance, and the boundaries (σ_0, σ_1, σ_2, σ_3, σ_4, σ_5) are thresholds on this distance, such that different interventions are deemed necessary for the ballast. This discretization is also applied to the rail components in the model, Appendix 1 details quantification methods for different defects within the rails.

For some components in the model, for example those where maintenance is completed based on age instead of condition and hence the extra states hold no extra relevant information to the model, the component condition is discretized into fewer states to improve the efficiency of the model. The description in Section 4.2.3 details where this is applied.

For a successful application of the model, the measure of the component condition and the threshold values must accurately represent the component condition, the safety regulations, and the maintenance procedures. The states given in Table 4.1, for each relevant component, can be defined as:

- State I: The component is in the 'good as new' state ($\sigma_0 < \sigma \leq \sigma_1$). The condition of the component is such that it does not impact normal operation.
- State II: The component is in a 'used' state ($\sigma_1 < \sigma \leq \sigma_2$). The component condition is still within the threshold for safe operation and classified as within its useful life, however, it can be maintained, or replaced, through opportunistic maintenance or an early replacement strategy.
- State III: The component is in a 'worn' state ($\sigma_2 < \sigma \leq \sigma_3$). The component condition is still within the threshold for safe operation but has reached a point where routine replacement, or

corrective maintenance, is required due to the component approaching the end of its useful life.

- State IV: The component is in a 'degraded' state ($\sigma_3 < \sigma \leq \sigma_4$). The component condition is outside the threshold for safe operation. Speed restrictions need to be applied and priority replacement, or corrective maintenance, is required.
- State V: The component is in a 'dangerous failed' state ($\sigma_4 < \sigma$). The component condition can cause derailment if a train passes over the S&C. Closure of the S&C and emergency replacement, or corrective maintenance, is required.

Historic data can be collected for the S&C type under consideration and this can be compared to past studies, expert opinion, or data collected from similar S&Cs to determine the state threshold values and transition delays between states [130]. These distributions are used to govern the degradation of components in the model. Distributions can also be assigned for inspection intervals and the different delays for scheduling and completing maintenance activities. These delays can be found from historical records, based on the current asset management strategy, or in the case of testing different strategy effectiveness, can be chosen from a selection of possible asset management strategies. The quantification of each of the states for each component is dependent on the component in question. A suggestion for this quantification is given for each of the relevant components in Section 4.3.1 of this chapter.

In some cases, the corrective maintenance of a component does not return the state of the component to the 'good as new' state. This results in faster rates of degradation following the maintenance action; this is assumed in this model for the tamping of ballast or grinding of track rails [140] [141]. In these cases, the rate of ageing of the component is dependent on the number of previous interventions. To incorporate this, Conditional Transitions are used, where several distributions are assigned to each Conditional Transition and the distribution used each time the transition fires is chosen based on the number of previous maintenance actions [139].

4.3: Model Descriptions

In this section, the Petri net models for the components degradation, inspection and maintenance are given. The models presented in this section contain places representing the revealed and unrevealed degraded, or failed, states of each of the components. If a components state is unrevealed, then carrying out an inspection is necessary to reveal it [142]. In each case, there is a probability associated with the failure of the inspection to correctly identify the state of the component. Once the state is revealed, appropriate maintenance activities and protection measures are applied. At the end of this section, Petri net models are presented for the overall maintenance strategy of the S&C and derailment occurrence by cause.

For each component, or group of components, there is a degradation, inspection and maintenance model. Each model is split into two layers, with each layer split into two further sections. Figure 4.7 is an example of the first layer in a model. The first layer of each model contains the unrevealed states of the components in the top section, this section is shaded orange, as in Figure 4.7. The first layer of each model also contains the inspection strategy and the revealed states of the components in its bottom section, this section is shaded blue, as in Figure 4.7. Figure 4.8 is an example of a second layer of a model. The second layer of each model includes the maintenance activities in the bottom section, shaded green, as in Figure 4.8. The second layer also includes the implementation of speed restrictions or closure of the S&C, either for maintenance or due to a hazardous condition, in the top section, shaded in red as in Figure 4.8.

Delays due to maintenance activity are included and represent maintenance scheduling and the time taken for the maintenance to be completed. These scheduling delays are dependent on the severity of each state. For every component, its replacement is modelled. In addition, for some components, there are multiple maintenance options. Across the whole system, the following maintenance options are modelled:

- Component replacement
- Whole system replacement
- Ballast tamping
- Undercutting
- Rail grinding
- Rail re-alignment
- Welding
- Lubrication and clearing

Replacement of a component is assumed to return that component to the as 'good as new' state for all modelled defect types. Reset transitions are used for this. Other maintenance options are assumed to only impact certain aspects of the component condition and so only partially reset the condition of the relevant component's degradation. For example, rail grinding is assumed to improve the condition of the rails in terms of surface cracking and rolling contact fatigue, but it is also assumed that this action will not improve the condition of the rails in terms of sub-surface cracking.

In the following section, for each component model, first, the component degradation and maintenance models are presented. Following this the system level inspection and maintenance strategy models are presented and finally the risk scenario models are presented.

The overall state of the S&C is also modelled in this chapter. Five system level states are modelled: the open state, the state with speed restrictions, a closed state for maintenance, a closed state due to condition and a closed state following a derailment. The final state is assumed to last a short time only and is included to prevent repeated derailments at the same instance. If the system is in any of the closed states, a derailment cannot occur. If the system is in a speed restricted state, it is assumed that a derailment can occur, only if a component within the S&C reaches a hazardous failed state while the restriction is applied. These system states are applied depending on the condition of the components across the model. When certain places are marked, this alters the state of the system from the working state to another state as detailed in Table 4.2. These places allow the system state, over time, to be output from the model. The results for this can be seen in Section 4.4 of this chapter.

Places	**Interpretation**
X1	There are speed restrictions applied to the S&C due to condition
X2	The S&C is closed for maintenance
X3	The S&C is closed due to poor condition
X4	The S&C is closed due to a derailment

Table 4.2: The interpretation of each of the system state places used in the models in this chapter

There are also several common places used across the models to govern system level maintenance strategies. The places govern the system level activation of different maintenance actions. For instance, these ensure that if similar maintenance actions are required at the same time, then they are completed in the same visit. If opportunistic maintenance is enabled across the model, then when a team visits the S&C for a component that requires maintenance, opportunistic maintenance of any further components in the S&C, which are not in the working state and have similar maintenance actions, are also completed. There is a delay time associated with that time that this opportunistic maintenance is available to other components, which can be varied according to application. This is discussed further in Section 4.3.3 of this thesis. The opportunistic maintenance behaviour, and maintenance availability in general, is governed by the marking of the places in Table 4.2. These places are implemented further in Section 4.3.1 and Section 4.3.3 of this chapter, where models for maintenance activation across the system are presented. The models presented there link the component models through system level maintenance strategies. The places given in Table 4.2 are featured in each of the component models to enable maintenance to occur.

Places	Interpretation
Om1	Opportunistic maintenance is enabled
Ot1	Opportunistic ballast tamping is enabled
Ou1	Opportunistic ballast undercutting is enabled
Og1	Opportunistic rail grinding is enabled
Ym1	Maintenance is enabled
Yt1	Tamping is enabled
Yu1	Ballast undercutting is enabled
Yg1	Rail grinding is enabled

Table 4.3: A table of the system level maintenance places

4.3.1: Component degradation and maintenance models

In this section components are modelled in detail to allow:

- Unsuccessful inspection of components to be considered, so that hazards arising from undetected failures are modelled.
- Multiple maintenance actions to be applied to a component, where these actions may only improve certain aspects of the component condition, and hence fail to return the component to the ‘as good as new’ state.
- Maintenance actions to be modelled that impact the future degradation rate of components.
- Different maintenance scheduling delays to be modelled, where these delays are dependent on the severity of the component condition.
- System wide opportunistic maintenance strategies, where components can be replaced early if maintenance resources are already allocated to the S&C.
- The modelling of a population of identical components, to consider their interactions, as is applied in the sleeper and clip model.

This detailed modelling allows decision makers to consider different condition-based and system level maintenance decisions, and removes typical assumptions of perfect maintenance and inspection. The latter is of particular importance when modelling hazard occurrence, to prevent assumptions that the component is in a better state than reality. These assumptions could lead to insufficient prevention action and increase the likelihood of the hazard occurring.

Opportunistic maintenance is also included in the models. Here, components can be replaced early, based on either age or condition, if there are already maintenance resources allocated to the S&C. A penalty can be included into the model to account for the loss of useful life for early replaced components, however in this model this is deemed to be accounted for within the physical cost of more frequent component replacement over the life-cycle. For example, if a component is always replaced early, accounting for a higher number of replacement components over the lifetime, then there is an inherent increased life-cycle cost, attributed to the larger number of components required over the life-cycle. Each of the component models are presented in the following sections.

Condition of the track geometry due to ballast sleeper and clip condition

This section presents models for the ballast, sleepers and clips and models the impact that these can have on the track geometry, such that a derailment may occur. The first models presented concern the modelling of the ballast. The first layer of this model includes the condition of the ballast, where the ballast condition is modelled with five states, from working through to a failed state that causes sufficient risk of derailment. The state of the ballast can be quantified by the impact that the ballast condition has on the track geometry. The second layer of this model includes the maintenance and application of system level speed restrictions or closure.

Figure 4.7 gives the first layer of the model which includes modelling of the degradation of the ballast, along with inspection. Figure 4.8 gives the second layer of the model, this includes the maintenance of the ballast and the application of system level state changes. Table 4.4 gives a description of each of the places in these two layers.

In the model in Figure 4.7, the orange shaded region models the degradation of the ballast, with the good state on the left (State I), represented by place C1, through to the failed state on the right (State V), represented by place C6. The intermediate degraded states lie between these, represented by places P1, P2 and P3 and correspond to State II, State III and State IV for the ballast condition, respectively. These states, in the orange shaded region, are unrevealed. The transitions between these states, t1, t2, t3 and t4, are conditional on the number of previous ballast tamping actions since the most recent replacement or undercutting action. The place C7 counts the number of such actions, so that transitions t1, t2, t3, and t4 have distributions governing their firing times that are dependent on the number of tokens in place C7.

The blue shaded region, in Figure 4.7, models the inspection of the ballast. Manual inspection is enabled when Place P4 is marked. Automated measurement, such as that carried out with a track recording vehicle, is enabled when Place P7 is marked. The marking of Place P5 or Place P8 represents a failure in each inspection method respectively. The discovered states of the ballast are also shown here by places C2, C3, C4 and C5. Place C2 corresponds to a discovered degraded state corresponding to the unrevealed degraded state represented by place P1. Place C3 corresponds to a discovered degraded state corresponding to the unrevealed degraded state represented by place P2. Place C4 corresponds to a discovered degraded state corresponding to the unrevealed degraded state represented by place P3. Place C5 corresponds to a discovered failed state corresponding to the unrevealed failed state represented by place C6.

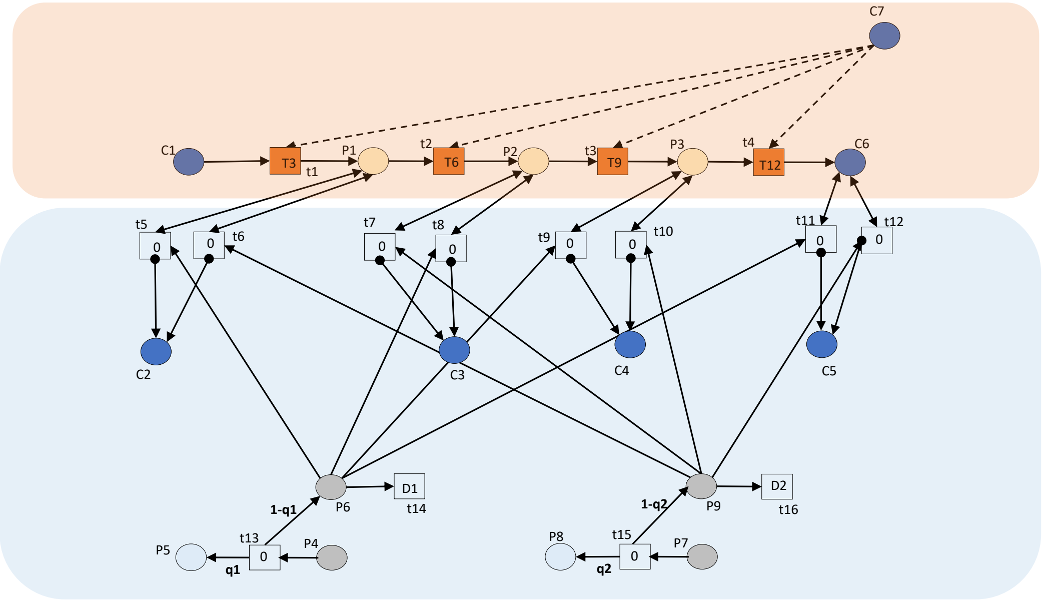


Figure 4.7: A Petri net model for degradation of the ballast.

After the identification of an imperfect state of the ballast, different maintenance actions can be used to improve the track geometry and component condition [143] [144] [145]. There are two maintenance actions included in this model. These are:

- Tamping of ballast to restore the vertical track geometry. This leads to a breakdown of ballast stones reducing the ballast condition on repeated tamping actions, which is incorporated through conditional Transitions (t1, t2, t3 and t4) in the model;
- Undercutting of the ballast, to improve vertical track geometry, where ballast condition is returned to State I, due to the replacement of the ballast.

Figure 4.8 gives the second layer of the ballast model. Here, the green shaded region models the maintenance of the ballast, with undercutting on the left and tamping on the right. It is assumed that a full replacement of the ballast does not happen in isolation, and so only occurs as part of a full system replacement. Depending on which maintenance action is enabled, one of tamping or undercutting of the ballast is completed. When the places, Yu1 and Yt1 are marked, then undercutting or tamping are enabled respectively. The activation of these processes is modelled in Section 4.3.3 of this chapter. Depending on the severity of the state, a different scheduling delay can be assigned to either maintenance action. For instance, when place P10 is marked the delay for scheduling and undercutting begins for a discovered State II defect. Similarly, when place P11 is marked the delay for scheduling and undercutting begins for a discovered State III defect. When place P12 is marked the delay for scheduling and undercutting begins for a discovered State IV defect. Finally, when place P13 is marked a delay for scheduling and undercutting begins for a discovered ballast failure (State V). This pattern is mirrored on the right hand side of the green shaded region for places P16, P17, P18 and P19, but tamping is scheduled instead of undercutting.

The red shaded region in Figure 4.8 ensures that the correct scheduling delay is assigned. This is done through transitions t35-t46. The red shaded region also applies any system level state changes, due to the ballast condition, through transitions t47, t48, t49 and t50. Here, place X2 represents a closure of the S&C due to a maintenance action, place X1 represents speed restrictions on the S&C, and place X3 represents a closure due to a hazardous failure.

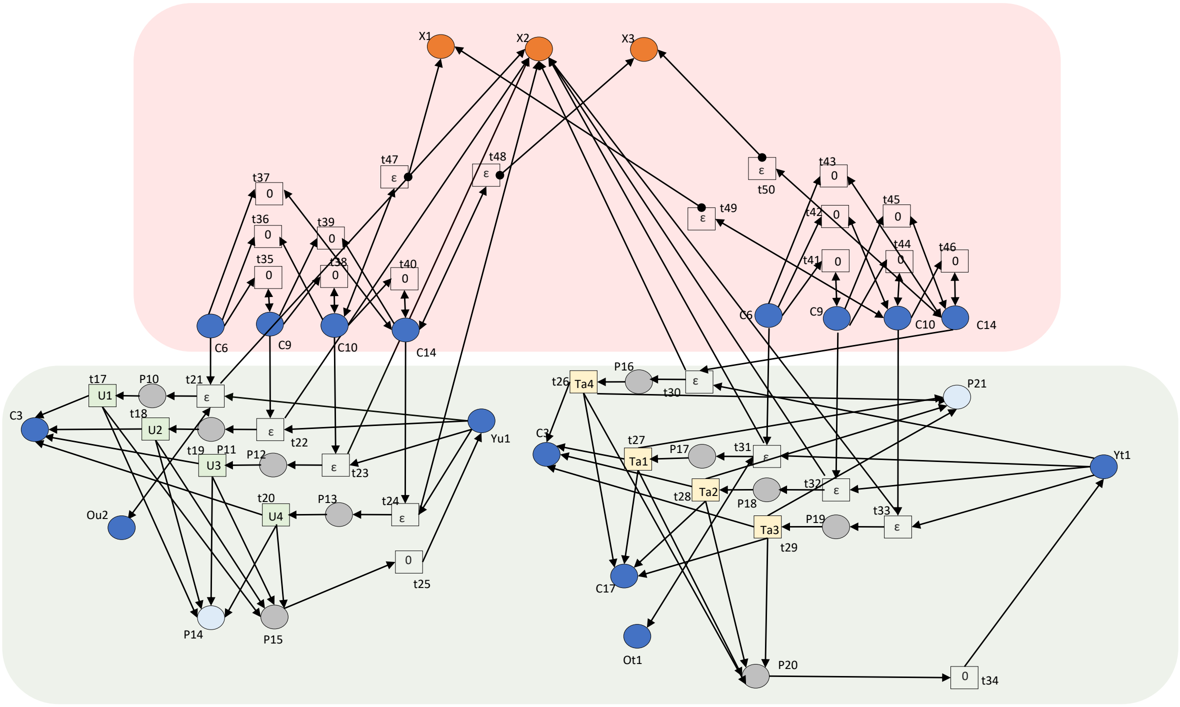


Figure 4.8: A Petri net model for ballast undercutting and ballast tamping.

Places	Interpretation
C1, P1, P2, P3, C6	The unrevealed condition states of the ballast in ascending order of state from State I to State V
C7	The number of tamping actions of the ballast since the most recent replacement
P6	Manual inspection is underway
P5	The number of failed manual inspections
P9	Automated geometry measurement is underway
P8	The number of failed automated geometry measurements
C2, C3, C4, C5	The revealed condition states of the ballast in ascending order of state from State II to State V
P16, P17, P18, P19	Ballast tamping is scheduled following a delay for each state in ascending order of severity
P10, P11, P12, P13	Ballast undercutting is scheduled following a delay for each state in ascending order of severity
P20, P15	A completed ballast tamp or ballast undercut, respectively
P21, P14	Counts the number of completed ballast tamp or ballast undercut, respectively

Table 4.4: A description of each of the places in the ballast module

The second set of models in this section concern the modelling of the sleepers and clips. These models do not follow the same structure as the other component models in this chapter, such that is described in the introduction of Section 4.3 of this thesis. This is because the sleeper and clip model differs as it considers each sleeper and clip unit separately, and combines these to model the population of sleepers and clips. Following this model, the remaining component models in this chapter follow the expected structure.

For illustration of the model it is assumed that there are 10 sleepers across the S&C, with a pair of clips attached to each sleeper. This can be extended, if required, to model a specific S&C, following the same logic. In this model, a module is used to represent the condition of each sleeper and pair of attached clips. Each of the sleepers or pair of clips is modelled with two states: the working and the failed state. It is assumed that multiple failures of the sleepers or clips can cause a derailment by impacting the track geometry. The behaviour of the population of sleepers, and associated clips, is used to determine the risk of the system and the application of any restrictions or closures. There are two failure modes modelled for the sleeper and clip units. The first failure mode is modelled when there are failures in two consecutive units, either due to the sleepers themselves or the attached clips, or there are three failed units across the S&C. In this case speed restrictions are applied. A derailment may still occur in this case, but a delay is introduced to model the derailment with a low likelihood. The second failure mode is modelled where there are three consecutive unit failures, or four or more failed units across the S&C. Here it is assumed that a derailment will occur if a train passes over the S&C.

The model for a unit containing a sleeper and a pair of clips is given in Figure 4.9. The orange shaded region models the failure of the sleeper or clips. Place SL1_n represents the working state of the sleeper and place CL1_n represents the working state of its attached clips. The failure of the sleeper is modelled by transition t1 and the failure of the clips in modelled by transition t4. Place SL2_n represents a failed state of the sleeper and Place CL2_n represents a failed state in either of the clips. Place MF_n is marked if the sleeper and clip unit is in the failed state due to the clips or the sleepers.

Inspection is governed by transition t2 for the sleeper and t5 for the clips. A global transition is used here to reduce the computational cost of simulating multiple models, under the assumption that a failure in the sleeper or clips is always detected on inspection.

In the model in Figure 4.9, place SCR is marked if there is a full replacement of all sleepers and clips underway this is marked by the model in Figure 4.11. The marking of this place prevents individual sleeper or clip replacement when a full replacement is already scheduled. If a failure is detected in the sleeper and clip unit, then replacement of the unit is scheduled and the green shaded regions model this. In this model if the sleepers are replaced, modelled by transition t3, then it is assumed that the clips on the sleeper are also replaced. However, if the clips are replaced, shown by transition t6, then the sleeper is not replaced. Places Ym1 and Om1 correspond to those used through the component models, such that if a sleeper or clip replacement is underway, other maintenance actions across the S&C are also enabled. The red shaded area in the model marks the place corresponding to closure of the S&C while maintenance is underway. Table 4.5 describes each of the places in this model.

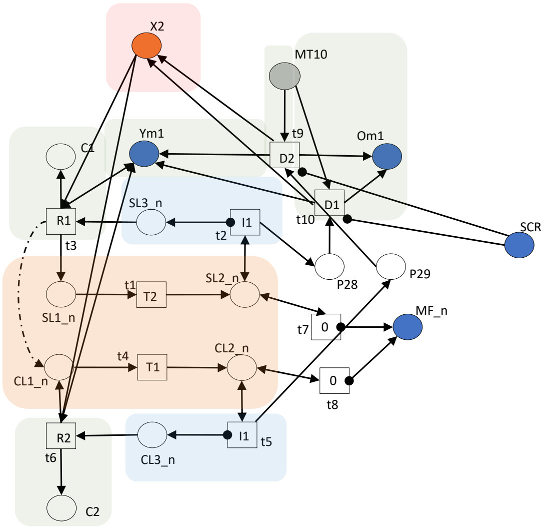


Figure 4.9: The module modelling a unit of one sleeper and its associated pair of clips

Places	Interpretation
SL1_n	The sleeper is in the working state
SL2_n	The sleeper is in the failed state
SL3_n	The sleeper is in a revealed failed state
CL1_n	The clips are in the working state
CL2_n	The clips are in the failed state
CL3_n	The clips are in the revealed failed state
C1	Counts the number of sleeper replacements
C1	Counts the number of clip replacements
P28	Triggers scheduling of maintenance due to a revealed failed sleeper condition
P29	Triggers scheduling of maintenance due to a revealed failed clip condition
MF_n	There is a failure in either the individual sleeper or its attached clips

MT10	Prevents duplicate maintenance activities for the same failure, the place is reset on maintenance to allow future maintenance actions.
SCR	There is full replacement of the sleepers and clips underway

Table 4.5: A description for the places in the sleeper and clip module

This unit module is repeated ten times to give places MF_n, for n in [1,10]. These individual unit failures are considered as a population in the model in Figure 4.10. In this model, each of these places, from the individual unit modules, can be found from left to right in the un-shaded section of the model. The top shaded section marks the place N_SCF with the total number of sleeper or clip units that are in the failed state. The middle, shaded section marks the place N_SC2 with the number of pairs of consecutive sleeper and clip units that are in the failed state. The bottom shaded section marks place N_SC3 with the number of triplets of consecutive sleeper and clip units that are in the failed state. These places summarise the behaviour of the population of sleepers and clips across the S&C. Table 4.6 gives the description of each of the places in the model.

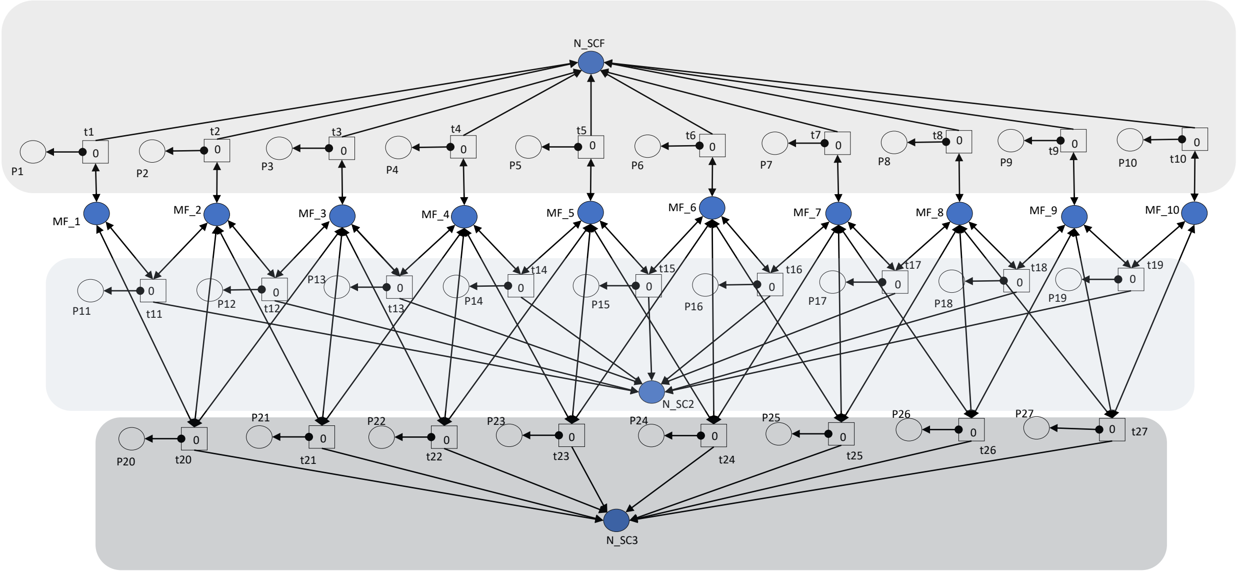


Figure 4.10: A model combining individual sleeper and clip units to give the population failure modes

Places	**Interpretation**
N_SCF	There is a failure in a sleeper and clip module
N_SC2	There is a failure in two consecutive sleeper and clip modules
N_SC3	There is a failure in three consecutive sleeper and clip modules
MF_1, MF_2, MF_3, MF_4, MF_5, MF_6, MF_7, MF_8, MF_9, MF_10	There is a failure in an individual sleeper and clip module, where these are found by the repeating model given in Figure 4.9.
P1, P2, P3, P4, P5, P6, P7, P8, P9, P10	Prevents double counting of individual sleeper and clip failures
P11, P12, P13, P14, P15, P16, P17, P18, P19	Prevents double counting of situations where there are failures in two sleeper and clip modules at the same time
P20, P21, P22, P23, P24, P25, P26, P27	Prevents double counting of situations where there are failures in three sleeper and clip modules at the same time

Table 4.6: Place description for the model for the combination of individual sleeper and clip failures

The Petri net in Figure 4.11 models the behaviour of the population of sleepers and clips to consider the application of system level speed restrictions or closure, due to the sleeper and clip condition. Place C8 corresponds to a failure of two consecutive sleeper and clip units, or three non-consecutive sleeper and clip unit failures across the S&C. Inspection, modelled by the global transition t32, can discover this failure. In this case speed restrictions are applied, modelled by transition t34, and each of the failed units are replaced individually. Place C9 corresponds to a failure of three consecutive sleeper and clip units, or four or more non-consecutive failed sleeper and clip units across the S&C. If this place is marked than a derailment can occur if a train passes over the S&C. If the state is revealed by inspection, then place P31 is marked. This triggers replacement of all sleeper and clip units over the S&C. This full replacement is modelled by the green shaded region, where transition t35 models the closure of the S&C due to poor condition of the sleepers and clips. This full replacement maintenance action resets all the sleeper and clip unit models to the 'as good as new' state. Table 4.7 gives the interpretation of the places in this model, where some places can be found in Table 4.5 and Table 4.6.

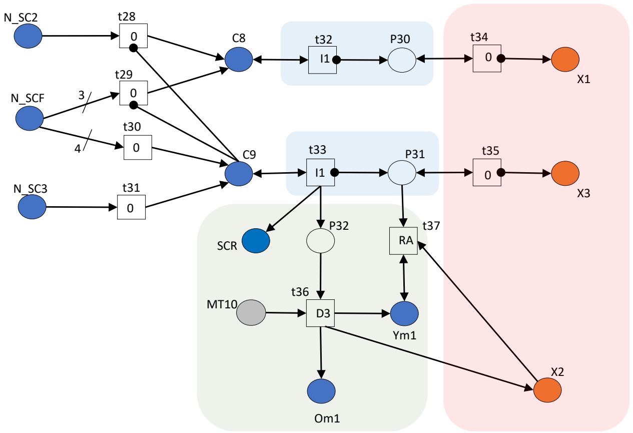


Figure 4.11: A model applying system level actions based on the population behaviour of the sleeper and clip units

Places	**Interpretation**
C8	There are two consecutive sleeper and clip unit failures or three non-consecutive sleeper and clip unit failures.
C9	There are three consecutive sleeper and clip unit failures or four non-consecutive sleeper and clip unit failures.
P30	Inspection finds two consecutive sleeper and clip unit failures or three non-consecutive sleeper and clip unit failures.
P31	Inspection finds three consecutive sleeper and clip unit failures or two non-consecutive sleeper and clip unit failures.
P32	Allows scheduling of full system replacement

Table 4.7: Interpretation of places for the model combining sleeper and clip units to give system level actions

Rail Components

This section models several different rail components, whose failure can lead to a derailment [146]. The components considered in this section are: the switch rails, the intermediate running rails, the stock rails, the crossing nose and the check rails. For each component model there is a first layer that represents the ageing and inspection of the component. For each component model there is also a second layer that describes the maintenance actions for the component in question and the application of system level speed restrictions or closure. For each component model in this section, visual inspection is used to reveal the condition of the rails along with ultra-sonic testing to check for weaknesses inside the rails [147]. In this section, the stock rails and intermediate running rails are grouped into one model, the fixed rail model, to reduce the complexity of the system model. However, if different rates of ageing are available for each of these fixed rails, this fixed rail model can be repeated several times for each of the rail types.

The first model presented in this section is the model for the stock rails and intermediate running rails. Figure 4.12 gives the first layer of this model and Figure 4.13 gives the second layer of this model. Table 4.8 gives a description of each of the places in this model.

Figure 4.12 gives the first layer of the Petri net model for the stock rails and intermediate running rails. There is a variety of different deterioration mechanisms leading to rail breakage or wear. In this model, these are grouped into three categories based on the maintenance and inspection activities associated with each, each group is referred to as a degradation pathway. Category 1 contains subsurface rail head defects or defects in the rail web or base, Category 2 contains rail head wear that results in an unrepairable loss of rail head material and Category 3 contains rail head defects that can be managed by rail grinding. The defects included in each category are given in Appendix 1, along with suggested methods of quantification of the degraded states of the rails.

In the first layer of the model presented here, the shaded orange region in Figure 4.12 models the degradation of the rails, with each degradation pathway stacked vertically. The states go from the working to failed state in the horizontal direction. These states are unrevealed.

In this orange shaded region, State I through to State IV for Category 1 defects are represented by places P1, P3, P6 and P9 respectively. For this category, replacement of the rails is required, and the condition of the rails can be inspected visually or by ultrasonic testing. State I through to State V for Category 2 defects are represented by places P2, P4, P7, P10 and A10 respectively. For this category, replacement of the rails is required, however the condition of the rails is only revealed by visual inspection [148]. State I through to State IV for Category 3 defects are represented by places A2, P5, P8 and P11 respectively. Rail defects in this category can be rectified or reduced by rail grinding. Here it is assumed that replacement of the rails occurs, instead of rail grinding, when the state is considered sufficient to cause a derailment. Rail defects in this category can be detected by both visual inspection and ultrasonic testing.

State V for both Category 1 and 3 defects corresponds to a rail break, which is represented by Place A9 in the Petri net model. State V for Category 2 defects corresponds to extreme wear on the rails, which is represented by Place A10 in the model. Place A1 represents the ‘as good as new’ state for the fixed rails, across all categories of defect.

The blue region in this layer models the inspection of the rails. The discovered states of the rails are also contained within the blue region. Places that are filled in dark blue are those that are present across multiple modules in the model. For visual inspection and ultrasonic testing there is a probability that the state of the rail will not be successfully identified. Place P13 corresponds to the activation of visual inspection and Place P21 to the activation of ultrasonic testing. The revealed states of the rail are represented by Places A3 through to A8, with Place A12 representing a revealed failure. The model also includes the probability of a failed rail being detected by a train driver; this is incorporated into Transition t45 of the model.

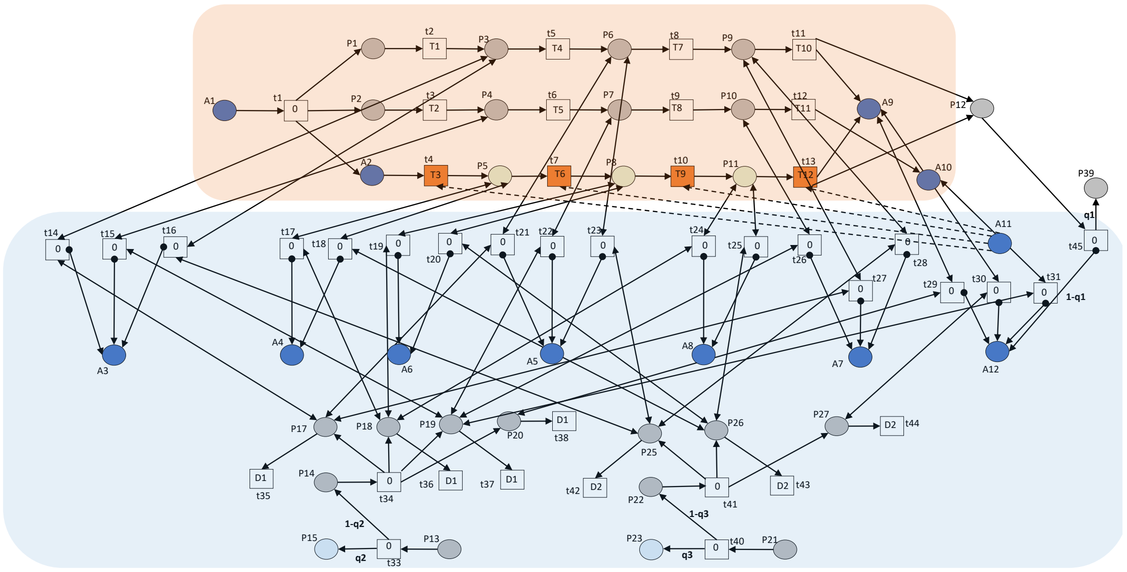


Figure 4.12: A Petri net model for the degradation of the fixed rails.

The green shaded region in Figure 4.13 models the maintenance of the rails, with replacement on the left-hand side and grinding on the right-hand side. Rail grinding is assumed to improve the condition of the surface of the rail only, and so this action only resets the places in the bottom degradation pathway. Replacement returns the whole degradation model to the ‘as good as new’ state. The blue filled places correspond to those in the previous figure. Maintenance actions are implemented following a scheduling delay that is dependent on the severity of the state. In this green shaded region, transitions t46 through to t49 represent replacement of the fixed rails and Transitions t55 through to t57 represent grinding of the fixed rails. For rails with Category 3 defects, it is assumed that repeated rail grinding does not return the rail to the ‘as good as new’ state and this is incorporated through conditional Transitions t4, t7, t10 and t14, in the Petri net in Figure 4.12.

The red shaded region in Figure 4.13 has two purposes: the first is to ensure that the correct maintenance action is applied, depending on any combined discovered defects and the most severe state. This is modelled by transitions t62-t73. The implementation of speed restrictions and S&C closure is also considered in this layer of the module with X1, X2 and X3 defined as in the models for the track geometry, transitions t74, t75 and t76 model this.

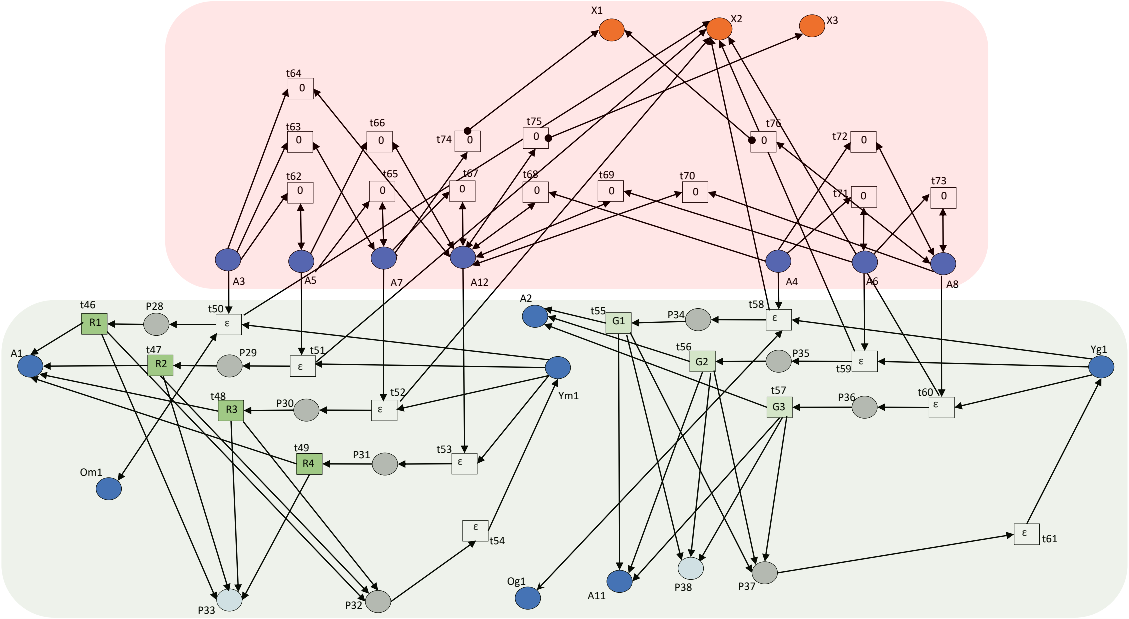


Figure 4.13: A Petri net model for the maintenance of the fixed rails.

Places	**Interpretation**
A1	The fixed rails are in State I
P1, P3, P6, P9	The unrevealed states of the fixed rails with Category 1 defects in ascending order of state from State I to State IV
P2, P4, P7, P10, A10	The unrevealed states of the fixed rails with Category 2 defects in ascending order of state from State I to State V
A2, P5, P3, P11	The unrevealed states of the fixed rails with Category 3 defects in ascending order of state from State I to State IV
A9	An unrevealed Category 1 or Category 3 State V defect (a rail break)
P12, P39	The potential to detect a rail break by a train driver and a failed detection of the rail break by a train driver, respectively
A11	The number of rail grinding actions between rail replacement
A3, A5, A7	The revealed states of the fixed rails with Category 1 or Category 3 defects in ascending order of state from State II to State IV
A4, A6, A8	The revealed states of the fixed rails with Category 2 defects in ascending order of state from State II to State IV
A12	The revealed state of the fixed rails with any category defect of severity State V.
P13	Visual inspection of the fixed rails is enabled
P15	Visual inspection of the fixed rails fails to identify the state
P14, P17, P18, P19, P20	Successful visual inspection of the fixed rails is underway

P21	Ultrasonic inspection of the fixed rails is enabled
P23	Ultrasonic inspection of the fixed rails fails to identify the state
P22, P25, P26, P27	Successful ultrasonic inspection of the fixed rails is underway
P28, P29, P30, P31	Replacement of the fixed rails is scheduled, with an associated delay, for each revealed state in ascending order of severity
P34, P35, P36	Grinding of the fixed rails is scheduled, with an associated delay, for each revealed state in ascending order of severity
P32, P37	A competed fixed rail replacement action and the number of completed such actions, respectively
P33, P38	A competed fixed rail grinding action and the number of completed such actions, respectively

Table 4.8: A description of each of the places in the fixed rail module

The second model presented in this section is the model for the switch rails. The switch rails are considered in a separate module from the fixed rails in this model, to allow additional modelling of their alignment within the S&C. The first layer of the model for the switch rails is given in Figure 4.14 and the second layer for the switch rail model is given in Figure 4.15. A description of each of the places in the switch rail model is given in Table 4.9.

Similarly to the model for the fixed rails the degradation of switch rails is modelled by the orange shaded region in Figure 4.14, with each degradation pathway stacked vertically and states of worsening condition from left to right. This region of the model has the same structure as the model for the fixed rails, except there is an additional row of states at the bottom of the orange shaded region. This additional row, which contains places M3, P47, P51 and P55, corresponds to the alignment of the switch rails to the stock rails. The remaining rows within this orange shaded region retain the same interpretation as the model for the fixed rails. As with the fixed rails, places in the upper pathway represent Category 1 defects, P42, P44, P48 and P52. Places in the second pathway represent Category 2 defects, P43, P45, P49 and P53. Places in the third pathway represent Category 3 defects, M2, P46, P50, P54. Place M1 represents the ‘as good as new’ state for the switch rails for all types of defect.

The inspection of the switch rails is modelled by the blue region in the first layer of this model. Similarly, to the model for the fixed rails, ultrasonic testing and visual inspection are incorporated into this model to reveal the state of the switch rails, along with detection of a break by the train driver. This part of the model has the same structure as the corresponding blue shaded region in the fixed rail model, except that there are four additional places, M8, M11, M12 and M14, corresponding to the detected states where there is a misaligned switch rail.

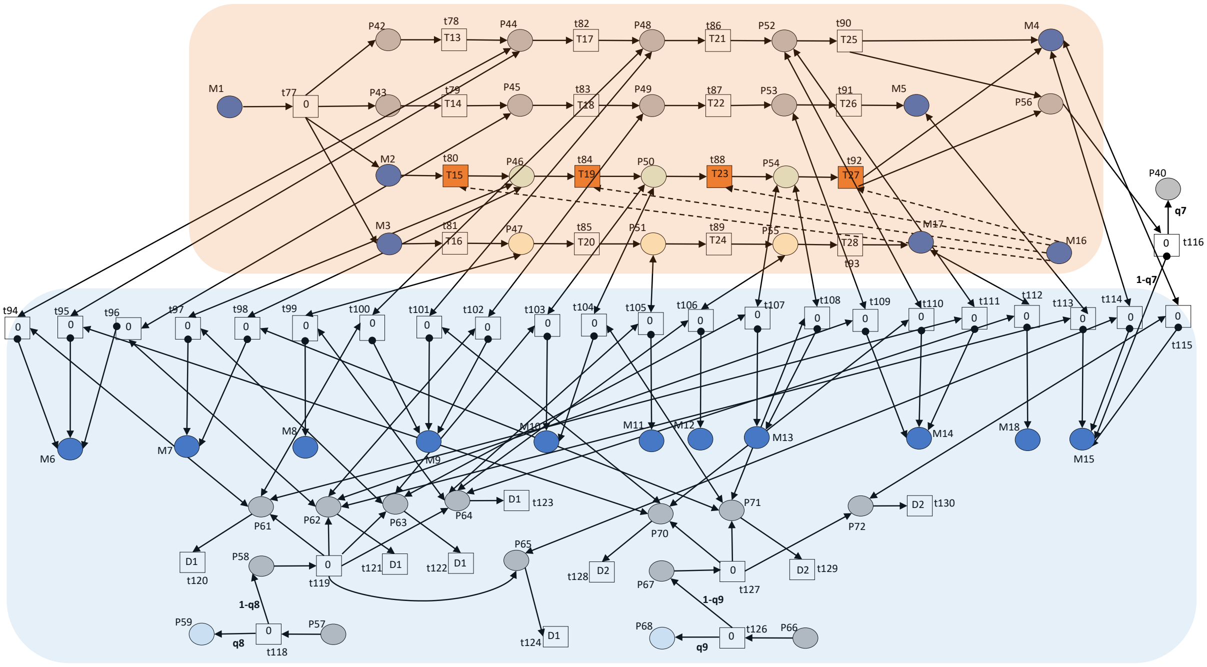


Figure 4.14: A Petri net model for the degradation of the switch rails

The second layer of the switch rail model is given in Figure 4.15. The region shaded green in Figure 4.15 models the maintenance of the switch rails, with the replacement modelled on the left hand side, grinding modelled in the centre and adjustment of the switch rail alignment modelled on the right hand side. This region of the model has the same structure as the green shaded region in the fixed rail model, except for the additional transitions and places on the right hand side of the region. These additional nodes model the switch rail adjustment, with transitions t147-t155. This adjustment of the rails only returns the pathway corresponding to the alignment of the switch rails, modelled in places M3, P47, P51 and P55 in the orange shaded region of the model in Figure 4.14, to the 'as good as new' state.

The red shaded region in Figure 4.15 applies system state changes with transitions t178, t179, t180, t181, t182. The red shaded region also ensures that the correct maintenance action is applied through transitions t156-t177.

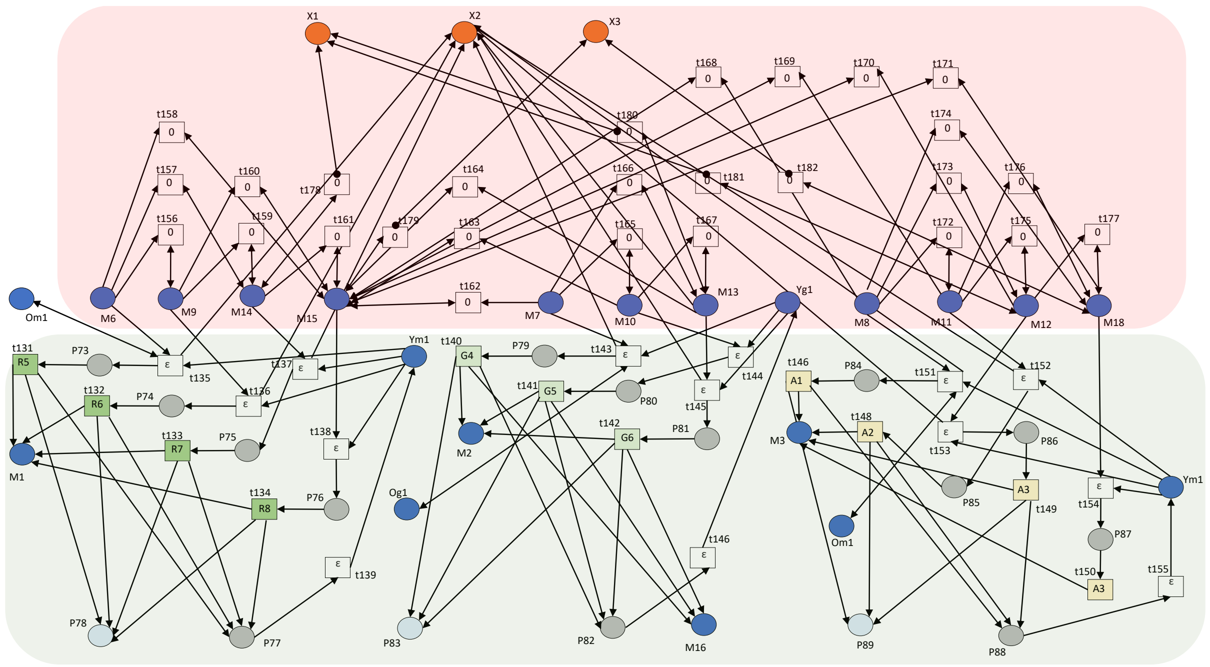


Figure 4.15: A Petri net model for the maintenance of the switch rails

Places	Interpretation
M1	The switch rails are in State I
P42, P44, P48, P52	The unrevealed states of the switch rails with Category 1 defects in ascending order of state from State I to State IV
P43, P45, P49, P53, M5	The unrevealed states of the switch rails with Category 2 defects in ascending order of state from State I to State V
M2, P46, P50, P54	The unrevealed states of the switch rails with Category 3 defects in ascending order of state from State I to State IV
M3, P47, P51, P55, M17	The unrevealed alignment states of the switch rails in ascending order of state from State I to State V
M4	An unrevealed Category 1 or Category 3 State V defect (a rail break)
P56, P40	The potential to detect a rail break by a train driver and a failed detection of the rail break by a train driver, respectively
M16	The number of rail grinding actions between rail replacement
M6, M9, M14	The revealed states of the switch rails with Category 1 or Category 3 defects in ascending order of state from State II to State IV
M7, M10, M13	The revealed states of the switch rails with Category 2 defects in ascending order of state from State II to State IV
M15	The revealed state of the switch rails with any category defect of severity State V.
M8, M11, M12, M18	The revealed alignment states of the switch rails in ascending order of state from State II to State V
P57	Visual inspection of the switch rails is enabled
P59	Visual inspection of the switch rails fails to identify the state

P58, P61, P62, P63, P64, P65	Successful visual inspection of the switch rails is underway
P66	Ultrasonic inspection of the switch rails is enabled
P68	Ultrasonic inspection of the switch rails fails to identify the state
P67, P70, P71, P72	Successful ultrasonic inspection of the switch rails is underway
P73, P74, P75, P76	Replacement of the switch rails is scheduled, with an associated delay, for each revealed state in ascending order of severity
P79, P80, P81	Grinding of the switch rails is scheduled, with an associated delay, for each revealed state in ascending order of severity
P84, P85, P86, P87	Adjustment of the switch rails is scheduled, with an associated delay, for each revealed state in ascending order of severity
P77, P78	A competed switch rail replacement action and the number of completed such actions, respectively
P82, P83	A competed switch rail grinding action and the number of completed such actions, respectively
P88, P89	A competed switch rail adjustment action and the number of completed such actions, respectively

Table 4.9: A description of each of the places in the switch rail module

The third model in this section considers the crossing nose condition. The first layer of this model, for the ageing and inspection of the crossing nose, is given in Figure 4.16. The second layer of this model, which considers the maintenance and system level speed restrictions or closure, is given in Figure 4.17. A description of each of the places in the model is given in Table 4.10.

The crossing nose is subject to high lateral forces as a train passes over the S&C. In this model there are three ageing mechanisms that can lead to a break or dangerous level of wear in the crossing nose [149]. The first mechanism modelled here is surface cracking of the crossing nose. This can be quantified by the number or depth of cracks. The second mechanism modelled here is sub-surface cracking, which can be quantified by the number or length of cracks. The third mechanism modelled here is deformation of the crossing nose, which can be quantified by the measured difference between the ideal crossing nose profile and the measured crossing nose profile.

The degradation processes of the crossing nose are modelled in the region shaded orange in the Petri net in Figure 4.16. Each of the ageing mechanisms are stacked vertically, with the states of each process arranged from left to right in increasing order of severity. In the orange shaded region of the model in Figure 4.16, the first process, represented by places R3, P97, P100 and P103, is surface cracking on the crossing nose. The second process, represented by places P96, P98, P101 and P104, is sub-surface cracking of the crossing nose. Place R15 corresponds to a State V crack that may have originated in either the surface, or sub-surface of the crossing nose. The third process, represented by places R4, P99, P102, P105 and R16, is deformation of the crossing nose leading to a failed rail profile that can result in wheel flange climb.

For the first and third process, relating to surface cracking and deformation, respectively, if the crossing nose is maintained instead of replaced, it is assumed that the condition does not return to the ‘as good as new’ state. Hence, future degradation can be faster following a repair action, for instance grinding of the crossing nose can remove material, eventually resulting in a break if done repeatedly. This is modelled by the conditional transitions, highlighted in orange, in this section of the model. The distributions governing the delay time for these conditional transitions are dependent on the marking of place R17, which counts the number of grinding actions, since the last replacement.

The inspection of the crossing nose is modelled by the blue shaded region. The crossing nose is inspected visually, when places P110-P113 are marked, and tested ultrasonically, when places P118-P120 are marked. This region also contains places that represent the states discovered through inspection of the crossing nose. For instance, place R5 is the corresponding discovered state for the unrevealed place P97. The pairing of these discovered and undiscovered states can be found by looking for the input and output places linked through transitions t196-t213 in this region of the model, or by referring to Table 4.10.

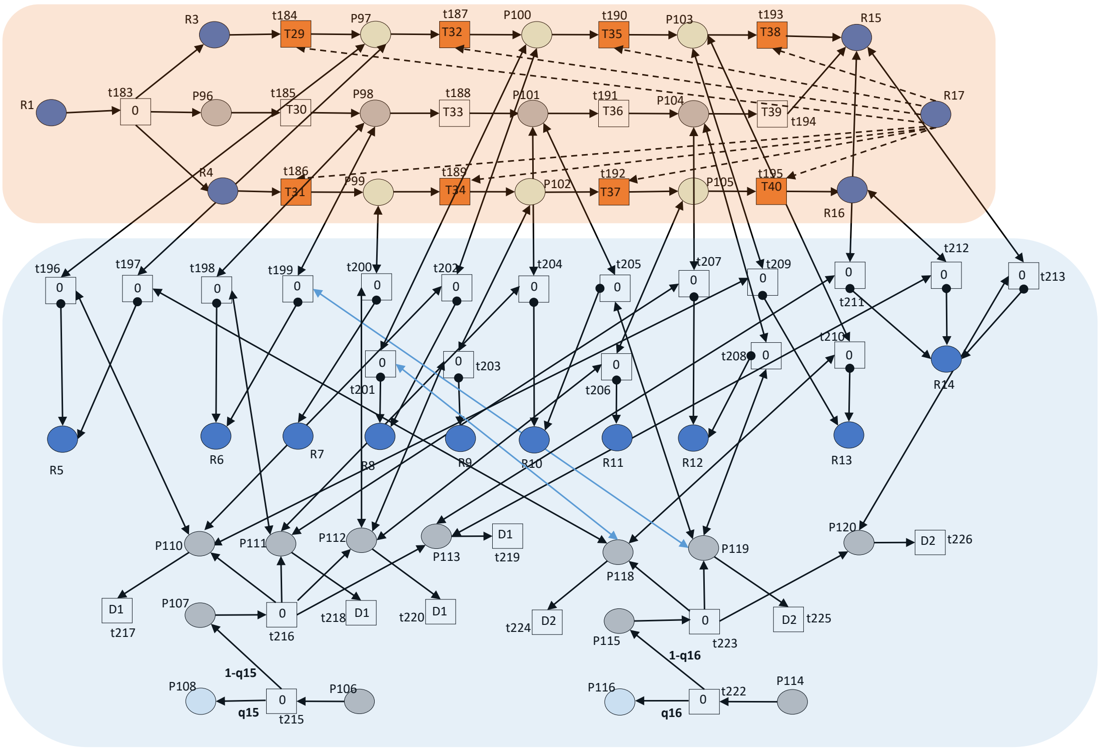


Figure 4.16: A Petri net model for the degradation of the crossing nose.

In Figure 4.17, the green shaded region models the maintenance actions for the crossing nose. The maintenance actions in this model for the crossing nose include replacement of the crossing nose, welding for low state deformation and grinding for low level surface cracking. These parts of the model can be seen within the green shaded region, with replacement on the left, grinding in the middle and welding on the right. In the left side of the region transitions t227, t228, t229 and t230 represent replacement of the crossing nose. In the middle of the region transitions t236, t237 and t238 represent corrective grinding of the crossing nose. On the right of the region transitions t250, t251 and t252 represent welding of the crossing nose. In this model, the grinding and welding of the crossing nose is assumed to only improve certain aspects of the crossing nose condition. Welding is assumed to improve the deformation in the crossing nose state and grinding is assumed to only improve any surface cracking present in the crossing nose [149]. Hence, these maintenance actions only return the marking of places corresponding to certain categories of defect in the orange shaded region to an improved state.

The red shaded region of this layer ensures the correct maintenance action is applied, depending on the state and the category of defect; this is done through transitions t250-t267. As with the other Petri nets presented in this chapter, implementation of protection measures is also included in this layer, following the same place definition as the earlier models. Transitions t268-t271 facilitates this.

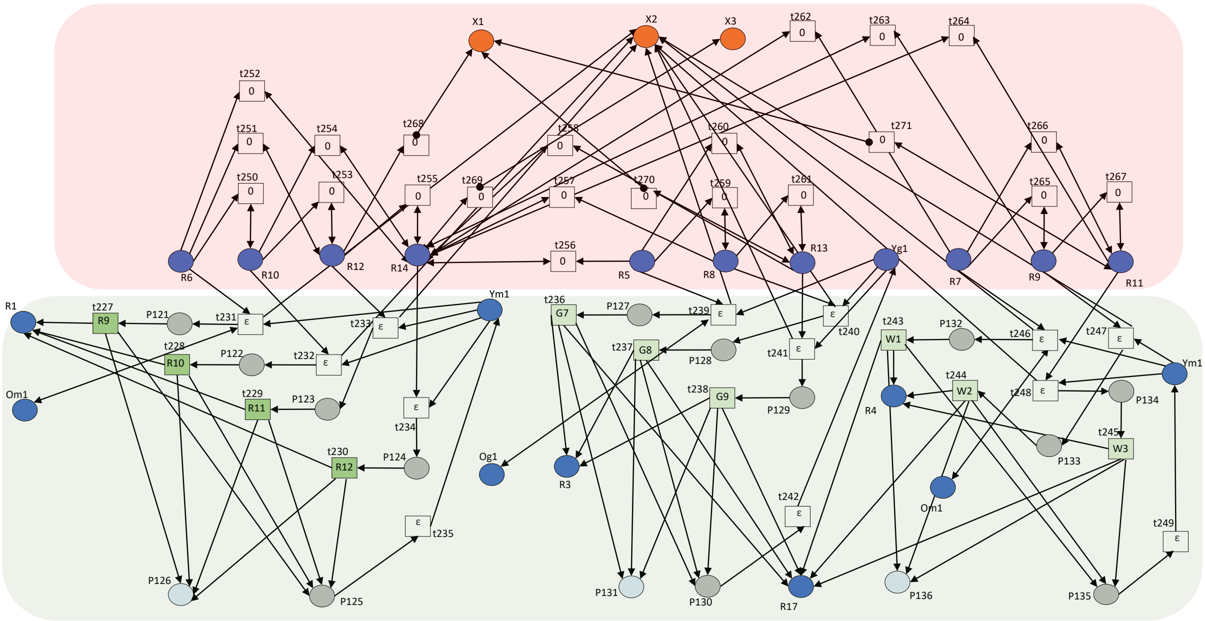


Figure 4.17: A Petri net model for the maintenance of the crossing nose.

Places	**Interpretation**
R1	The crossing nose is in State I
R3, P97, P100, P103	The unrevealed surface cracking states of the crossing nose in ascending order of state from State I to State IV
P96, P98, P101, P104	The unrevealed sub-surface cracking states of the crossing nose in ascending order of state from State I to State IV
R15	An unrevealed State V crack in the crossing nose
R4, P99, P102, P105, R16	The unrevealed deformation states of the crossing nose in ascending order of state from State I to State V
R6, R10, R12	The revealed surface cracking states of the crossing nose in ascending order of state from State II to State IV
R5, R8, R13	The revealed sub-surface cracking states of the crossing nose in ascending order of state from State II to State IV
R14	A revealed State V crack in the crossing nose
R7, R9, R11	The revealed deformation states of the crossing nose in ascending order of state from State II to State V
R17	Counts the number of grinding or welding operations on the crossing nose between replacement
P106	Visual inspection of the crossing nose is enabled
P108	Visual inspection of the crossing nose fails to identify the state
P107, P110, P111, P112, P113	Successful visual inspection of the crossing nose is underway
P114	Ultrasonic testing of the crossing nose is enabled

P116	Ultrasonic testing of the crossing nose fails to identify the state
P118, P119, P120	Successful ultrasonic testing of the crossing nose is underway
P121, P122, P123, P124	Replacement of the crossing nose is scheduled, following a delay, for each revealed state in ascending order
P127, P128, P129	Grinding of the crossing nose is scheduled, following a delay, for each revealed state in ascending order
P132, P133, P134	Welding of the crossing nose is scheduled, following a delay, for each revealed state in ascending order
P125, P126	A crossing nose replacement is completed, and the number of such actions is counted, respectively.
P130, P131	A crossing nose grinding action is completed, and the number of such actions is counted, respectively.
P135, P136	A crossing nose welding action is completed, and the number of such actions is counted, respectively.

Table 4.10: A description of each of the places for the crossing nose module

The final model given for rail components considers the check rails. The first layer of this model is given in Figure 4.18. The second layer of this model is given by the Petri net in Figure 4.19. A description of each of the places is given in Table 4.11.

As the train wheels pass over the crossing, the check rails ensure that the wheel stays on the correct path. Unlike the other rails in the model, the train wheels do not pass over the check rails, however the check rails are subject to high lateral forces as they guide the wheel through the crossing.

The first layer of the check rail model is given in Figure 4.18. In this layer the orange region represents the degradation model for the check rails. There are two ageing mechanisms included in the model for the check rails. Each of the ageing mechanisms are stacked vertically, with worsening states arranged from left to right. The first ageing mechanism is deformation or bending of the check rails which can be measured by the difference in rail head shape of the check rail from its initial position, represented by Places R18, P142, P144, P146 and R26. Conditional transitions are included here, where grinding of the check rails, if they are discovered to have low level deformation, is assumed to impact future degradation rates. The second ageing mechanism is lateral cracking of the check rails, represented by Places P141, P143, P145, P147 and R27, and this can be quantified by the number or depth of cracks in the check rails. The places in this region represent unrevealed states.

The blue shaded region in the first model layer considers the inspection of the check rails. There are two inspection methods included in this model. Firstly, visual inspection is enabled when places P152 and P153 are marked. Secondly, ultrasonic testing, in the case of lateral cracking of the check rails, is enabled when Place P155 is marked. Places R19-R25 represent discovered degraded or failed states of the check rails, where each has a corresponding unrevealed state. Table 4.11 describes these individually.

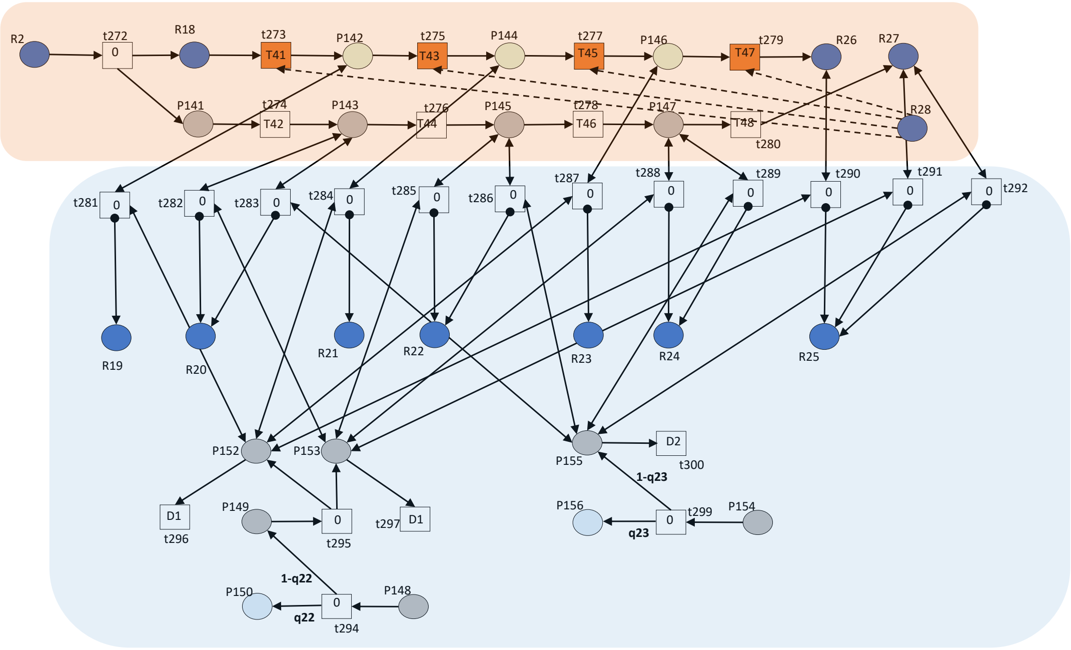


Figure 4.18: A Petri net model for the degradation of the check rails

The second layer of the check rail model is given in Figure 4.19. In this layer, the green shaded region models the maintenance of the check rails, with the replacement on the left-hand side and grinding on the right-hand side. Replacement resets the condition of the check rails to the ‘as good as new’ state. Grinding only improves any deformation and is assumed to not impact lateral cracking defects. Here it is assumed that, for deformation of the rail head in State IV or better, the rail can be ground to improve its condition. This is represented by transitions t321, t322 and t323. If the rail is in a worse state than State IV, or there is lateral cracking of the rail, then the check rails are replaced. This is represented by transitions t312, t313, t314 and t315.

The red shaded region, in this layer of the model, ensures that the correct maintenance action is applied. Transitions t317-t328 ensures this. As with the other Petri nets presented in this section, the implementation of system level speed restrictions and closures is also incorporated in this layer of the model, through transitions t317-t328.

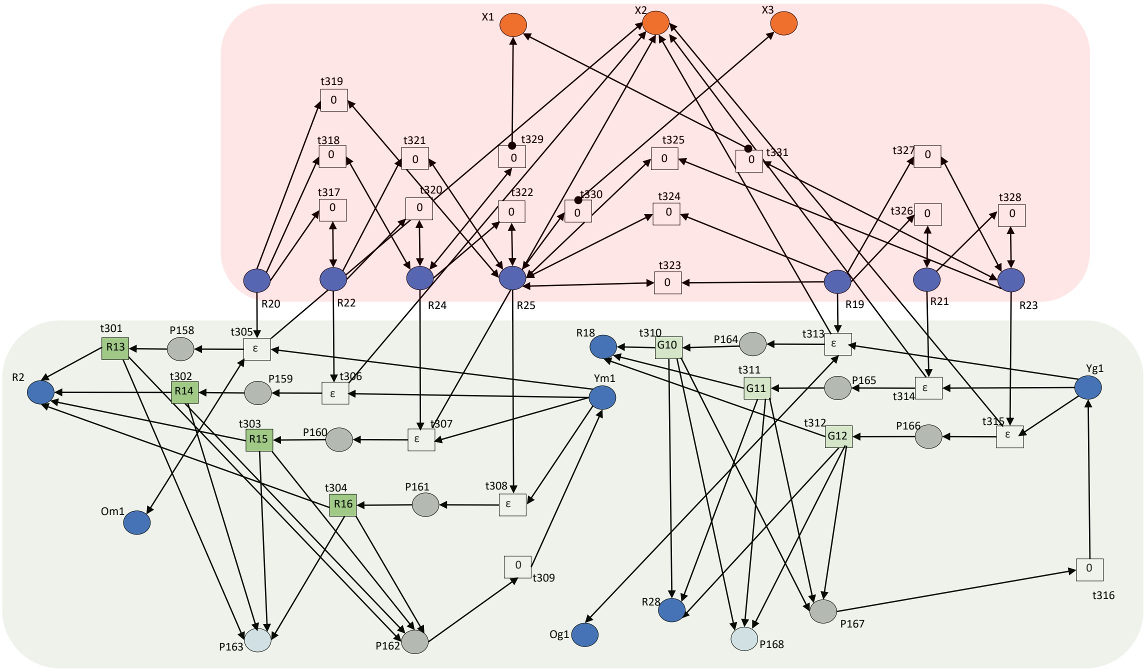


Figure 4.19: A Petri net model for the maintenance of the check rails

Places	**Interpretation**
R2	The check rails are in State I
R18, P142, P144, P146, R26	The unrevealed deformation states of the check rails in ascending order of state from State I to State V
P141, P143, P145, P147, R27	The unrevealed lateral cracking states of the check rails in ascending order of state from State I to State V
R28	The number of grinding actions of the check rails between replacement
R20, R22, R24, R25	The revealed deformation states of the check rails in ascending order of state from State II to State V
R19, R21, R23	The revealed lateral cracking states of the check rails in ascending order of state from State I to State V
P148	Visual inspection of the check rails is enabled
P150	Visual inspection of the check rails fails to identify the state
P149, P152, P153	Successful visual inspection of the check rails is underway
P154	Ultrasonic testing of the check rails is enabled
P156	Ultrasonic testing of the check rails fails to identify the state
P155	Successful ultrasonic testing of the check rails is underway
P158, P159, P160, P161	Replacement of the check rails is scheduled, after a delay, for each revealed state in ascending order of severity.
P164, P165, P166	Grinding of the check rails is scheduled, after a delay, for each revealed state in ascending order of severity.
P163, P163	A replacement of the check rails is completed, and the number of such replacement

	actions, respectively
P168, P167	A grinding action of the check rails is completed, and the number of such replacement actions, respectively

Table 4.11: A description of each of the places in the check rail module

Components impacting switch rail position

There are several different factors that can impact the position of the switch rails. If the switch rails are in a dangerous position, where they are not locked in contact with a stock rail, there can be a derailment at the S&C as the train passes over. The next section of this chapter considers various component failures that can result in a dangerous switch rail position.

In this section, models are presented for the condition of the stretcher bars and supplementary drive, POE, locking device, switch position detector and slide chairs. In addition, a model for an external signal failure is presented for a scenario where a false signal unlocks the switch rails on the passage of a train. A contribution to an incorrect switch position can also be made by the failure of the switch rail alignment, which is taken from the models for the switch rail condition given in Figure 4.14, earlier in this chapter.

The first model of this section gives a Petri net model for the cracking of the stretcher bars or supplementary drive and a resulting failure. The first layer of this model, which models the degradation of the components and their inspection, is given in Figure 4.20. The second layer of this model, which models the component maintenance and system level speed restrictions or closure, is given in Figure 4.21. A description of each of the places in this model is given in Table 4.12.

The stretcher bars connect the switch rails together and ensure that both switch rails move at the same time. There can be several stretcher bars along the switch rails, depending on the length of the switch, connected by a supplementary drive. A failure in the stretcher bars or supplementary drive can lead to a derailment as it can result in an incorrect positioning of the switch rails directing the train wheels in two separate directions. Alternatively, the train wheels may not make full contact with the stock rails or intermediate rails, on failure of the stretcher bars or supplementary drive. The condition of the stretcher bars can be quantified by the number of cracks present, a measure of deformation from the desired profile or a measure of the length over which corrosion is present.

In the Petri net in Figure 4.20, the degradation of the stretcher bars and supplementary drive is modelled by the section shaded orange. In the Petri net in Figure 4.20, places E1, P1, P2, P4 and E2 represent the states of the stretcher bars or supplementary drive due to the presence of cracking, bending or corrosion in worsening unrevealed states from State I to State V.

The inspection is modelled by the section shaded blue. In this model, it is assumed that the stretcher bars and supplementary drive are visually inspected in a periodic manner to classify their condition. When place P9 is marked then an inspection is underway. A measure of the condition may be the number or depth of cracks visible or the deviation of the shape of the stretcher bar from the ideal position. The blue region also contains the places that represent degraded or failed states that have been discovered through the inspection. Places E3, E4, E5 and E6 represent these discovered states.

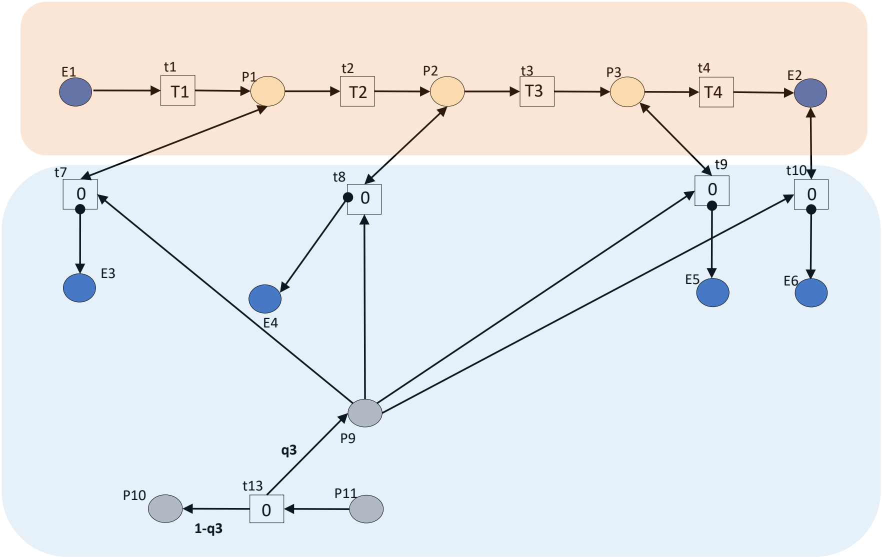


Figure 4.20: A Petri net model for stretcher bar and supplementary drive degradation

The replacement of the stretcher bars and supplementary drive is modelled by the green section in Figure 4.21, transitions t14, t15, t16 and t17 model this replacement. Replacement of the stretcher bars or supplementary drive occurs if they are found to be in State III or worse; this is governed by the Petri net in Figure 4.21. There is also the option to include opportunistic replacement strategies for stretcher bars, or a supplementary drive, found in State II.

The red section in Figure 4.21 ensures the correct maintenance delay is applied, through transitions t23-t28, along with any system level state changes, with transitions t29 and t30. Speed restrictions are applied if the stretcher bars or supplementary drive are in State IV and closure is applied if they are in State V or undergoing maintenance.

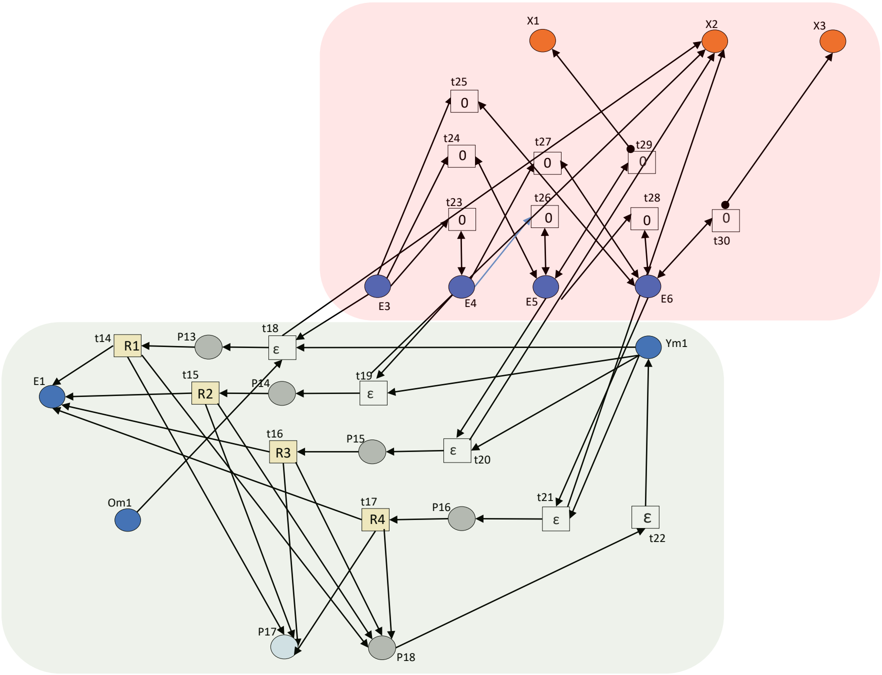


Figure 4.21: A Petri net model for stretcher bar maintenance

Places	Interpretation
E1, P1, P2, P3, E2	The unrevealed degradation states of the stretcher bars and supplementary drive, in ascending state from State I to State V
E3, E4, E5, E6	The revealed degradation states of the stretcher bars and supplementary drive, in ascending state from State II to State V
P11	Visual inspection of the stretcher bars and supplementary drive is enabled
P10	Visual inspection of the stretcher bars and supplementary drive fails to reveal the state
P9	Visual inspection of the stretcher bars and supplementary drive is underway
P13, P14, P15, P16	Replacement of the stretcher bars and supplementary drive is scheduled after a delay, for each revealed state in ascending order of severity
P18, P19	Replacement of the stretcher bars and supplementary dives is completed, and the number of such actions, respectively.

Table 4.12: A description of each of the places in the stretcher bar and supplementary drive module

The second model in this section considers the slide chair condition. The first layer of this model, which includes the degradation and inspection, is given in Figure 4.22. The second layer of this model, which includes the maintenance and application of system level speed restrictions or closure, is given in Figure 4.23. A description of each of the places in the slide chair model is given in Table 4.13. These models predict situations where the slide chair can fail thus preventing the correct movement of the switch rail leading to a derailment.

In the Petri net in Figure 4.22, the orange shaded region models the degradation of the slide chairs. Each ageing mechanism is stacked vertically with the condition of the slide chair material modelled with the top pathway and lubrication of the slide chairs modelled in the pathway below. The top pathway, which includes places P1, P2, P4, P6 and S3, gives the condition of the slide chairs whereby degradation can lead to blockage of the switch path by the failed slide chair material. States can be quantified by counting the number of cracks or corroded areas present. The pathway, which includes places S2, P3, P5, P7 and S4, gives the condition of the slide chairs in terms of lubrication or blockage from an external source. Here, transition t4 corresponds to blockage of the slide chairs from an external source. A blockage in the slide chairs can prevent the switch rails from moving into the correct position.

The inspection of the slide chairs is modelled by the blue region in Figure 4.22. The condition of the slide chairs is revealed through visual inspection, with a probability of successful state identification for each inspection. When places P8 and P9 are marked then inspection is underway. The blue shaded region also contains places that represent degraded or failed states that have been discovered by inspection, places S5-S12. Table 4.13 gives a description of each of these places.

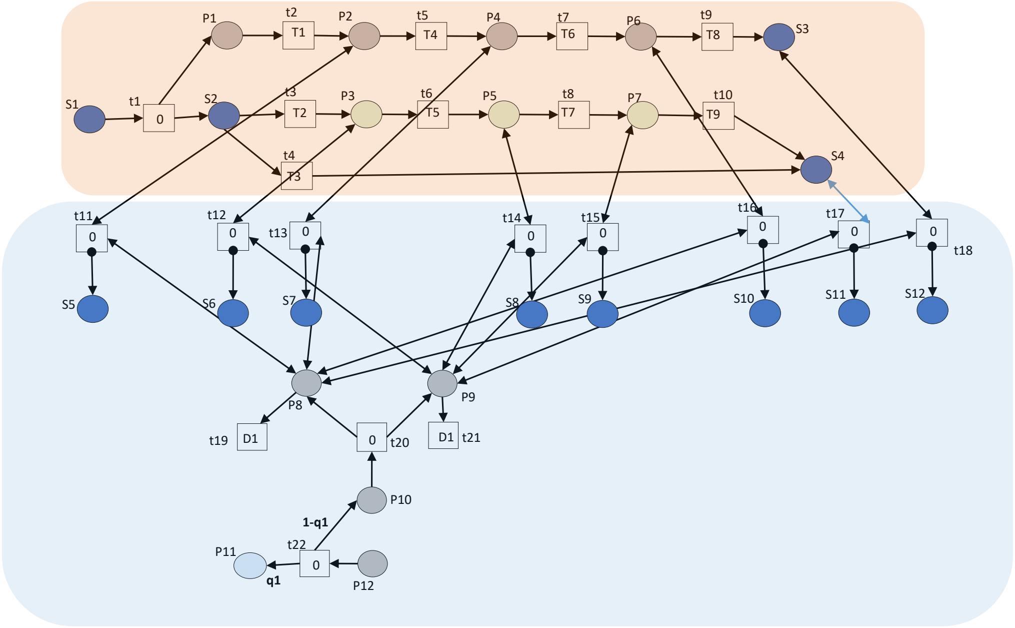


Figure 4.22: A Petri net for slide chair degradation and inspection

The green shaded region in Figure 4.23 models the maintenance of the slide chairs, with replacement on the left-hand side and clearing and lubrication on the right-hand side. The transitions with delay times R1, R2, R3 and R4 represent replacement of the slide chairs returning them to the 'as good as new' condition. The transitions with distributions, L1, L2, L3 and L4 represent cleaning, clearing and lubrication of the slide chairs. In both cases, manual intervention is required, corresponding to the marking of place Ym1 for routine maintenance and place Om1 for opportunistic maintenance. Replacement returns the whole degradation model to the 'as good as new' state. Clearing and lubrication does not impact the state of the slide chair material, and so any markings of places P1, P2, P4, P6, S3 and the places corresponding to any discovered failures of this type, places S5, S7, S10 and S12 is unaffected by this maintenance action.

The red shaded region in the layer ensures that the correct maintenance action is applied, through transitions t42-t53, and implements any system level state changes, through transitions t58-t61. Speed restrictions are implemented if it is found that the slide chairs are in a poor state. Closure is implemented if it is found that the slide chairs, or an external blockage of the slide chairs, is preventing the switch rails moving correctly.

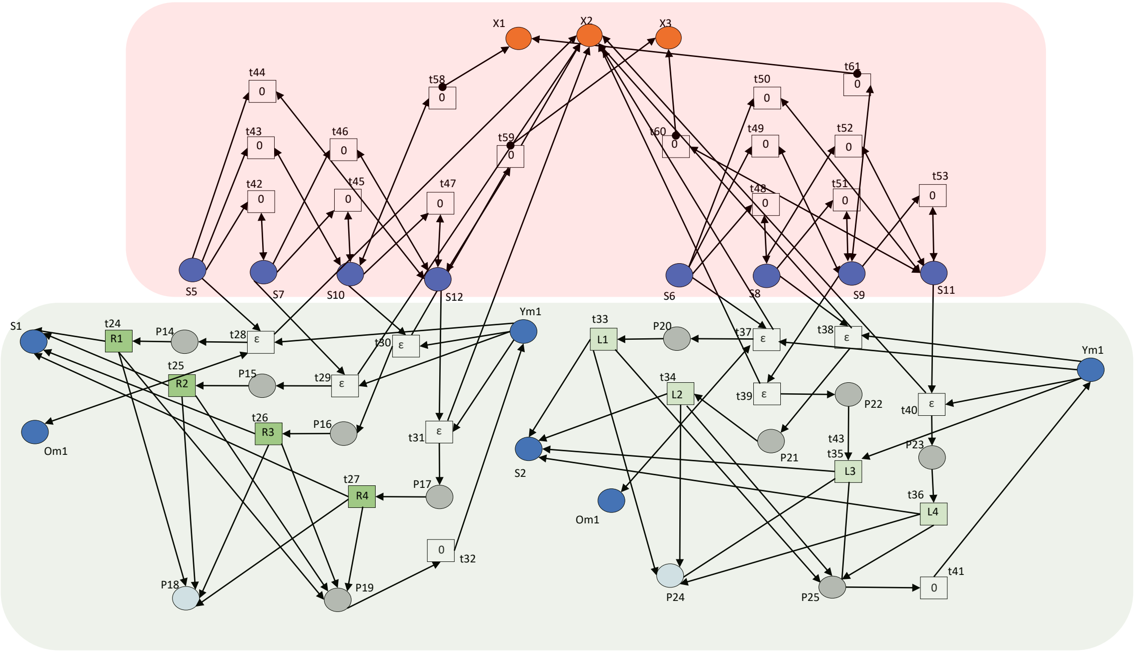


Figure 4.23: A Petri net for slide chair maintenance

Places	**Interpretation**
S1	The slide chairs are in State I
P1, P2, P4, P6, S3	The unrevealed degraded states of the slide chairs, with states in ascending order from State I to State V.
S2, P3, P5, P7, S4	The unrevealed condition of the slide chairs, with respect to blockage or lubrication, with states in ascending order from State I to State V
S5, S7, S10, S12	The revealed degraded states of the slide chairs, with states in ascending order from State I to State V.
S6, S8, S9, S11	The revealed condition of the slide chairs, with respect to blockage or lubrication, with states in ascending order from State I to State V
P12	Inspection of the slide chairs is underway
P11	Inspection of the slide chairs fails to reveal the state
P10, P8, P9	Successful inspection of the slide chairs is underway
P14, P25, P26, P17	Replacement of the slide chairs is scheduled, following a delay, for each revealed state in ascending order
P20, P21, P22, P23	Clearing and lubrication of the slide chairs is scheduled, following a delay, for each revealed state in ascending order
P19, P18	Replacement of the slide chairs is completed, and the number of such actions are counted, respectively.
P25, P24	Clearing and lubrication of the slide chairs is completed, and the number of such actions are counted, respectively.

Table 4.13: A description of each of the places in the slide chair module

The third model in this section considers the POE and the switch rail locking device. Figure 4.24 gives the first layer of the model, considering the degradation and inspection. Figure 4.25 gives the second layer of the model, considering maintenance and system level speed restrictions or closure. A description of each of the places in this model can be found in Table 4.14.

In the model layer given in Figure 4.24, the region shaded in orange models the ageing of the POE and locking device in worsening states from left to right. The top pathway, in this orange shaded region, considers the condition of the POE from the ‘as good as new’ state to failure. The ‘as good as new’ state corresponds to Place N1 and the failed state, corresponds to Place N3. The intermediate degraded states of the POE are given by places P2, P3 and P4 in order of increasing degradation level. These states could be quantified by measuring the time taken for the POE to fully move the switch rails. A failure of the POE can lead to the switch rails not being moved into the desired location. The lower pathway corresponds to the condition of the locking device with Place N2 corresponding to the ‘as good as new’ state, and Place N4 corresponding to the failed state. Failure of the locking device can cause the switch rails to become unlocked as a train passes over the S&C, leading to a derailment.

The blue shaded region models the inspection of the POE and locking device. It is assumed in this model that inspection of the POE can identify a degraded state. Places N5, N7 and N10 correspond to these revealed states in order of increasing degradation level. Places N12 and N11 correspond to a detected failure in the POE or locking device, respectively. Included in this model, are age-based replacement actions and the application of restrictions for the locking device. The user may specify a timeframe for the opportunistic replacement, routine replacement and priority replacement of the locking device, based on its expected lifetime. Places N6, N8 and N9 are marked when these defined time frames have elapsed, with speed restrictions applied if place N9 is marked.

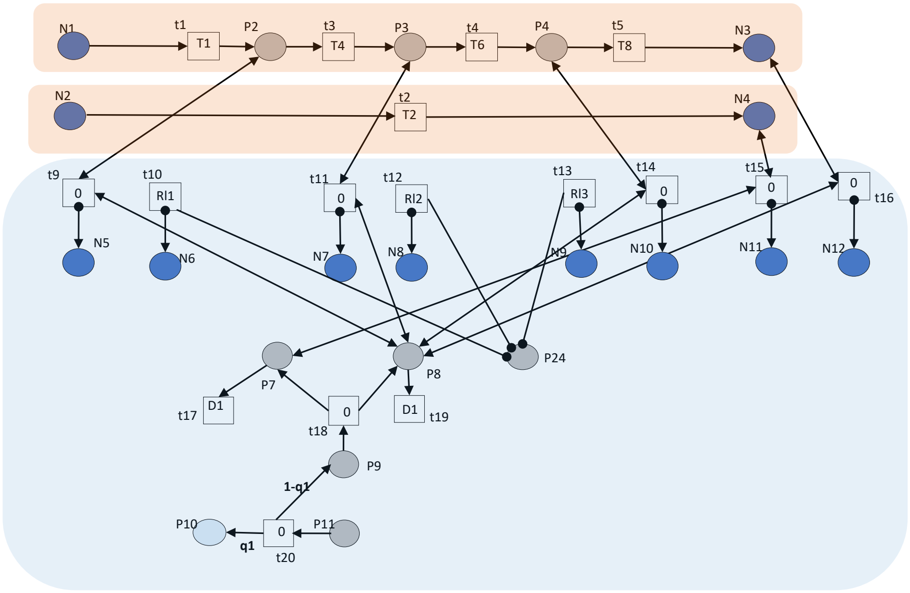


Figure 4.24: A Petri net for POE and points locking device degradation and inspection

For the model layer in Figure 4.25, the green shaded region models the maintenance of the POE or the locking device. Similarly to the previous models in this section, Place Ym1 enables manual intervention of the components and place Om1 enables opportunistic maintenance or replacement of the components. In the case of the POE and locking device, maintenance is assumed to return the component to the ‘as good as new’ state. The maintenance of the POE is modelled on the left-hand side and the maintenance of the locking device is modelled on the right-hand side. Transitions t22, t23, t24 and t25 correspond to replacement or repair of the POE. For the POE, maintenance is completed on a revealed degraded state in this model. Transitions t31, t32, t33 and t34 correspond to replacement or repair of the locking device. For the locking device, there is only a working and failed state modelled, based on the assumption that intermediate states cannot be quantified with a numeric measure. Age-based maintenance is included for the locking device, such that the age-based maintenance is completed if places N6, N8 or N9 are marked, this can happen through the logic given in the first layer of the model.

The red shaded region ensures that the correct scheduling delay is applied, depending on the severity of the state, through transitions t40-t51, and applies any system level state changes through transitions t52-t55. Speed restrictions are implemented if it is found that the POE or locking device are in the degraded state. Closure is implemented if it is found that the POE or locking devices are in the failed state. Closure is also implemented if maintenance of the components is underway.

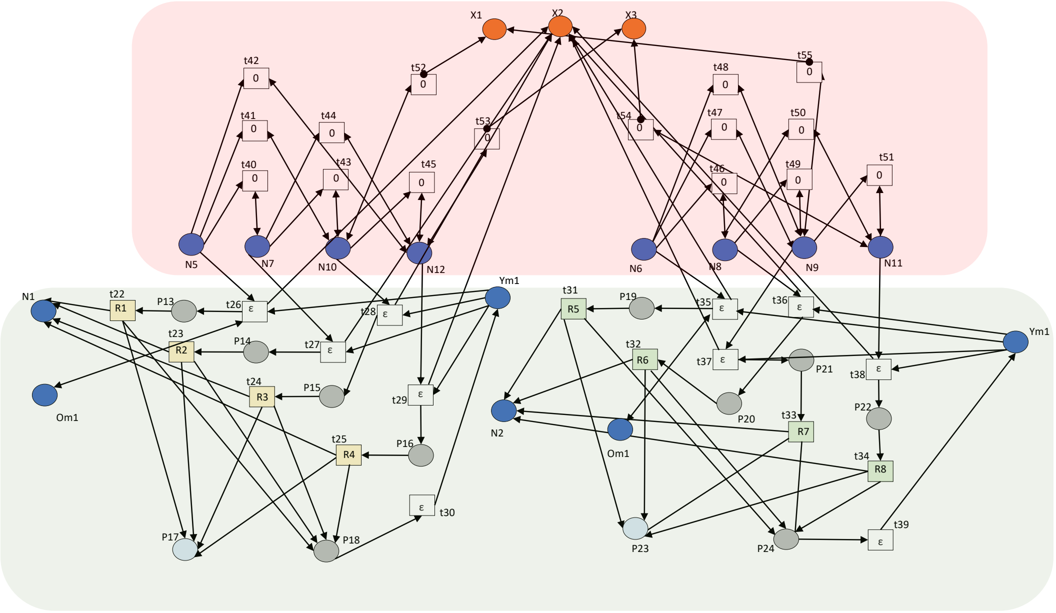


Figure 4.25: A Petri net for POE and points locking device maintenance

Places	Interpretation
N1, P2, P3, P4, N3	The unrevealed degraded states of the POE, in ascending order of state from State I to State V
N2, N4	The unrevealed state of the locking device, corresponding to State I and State V.
N5, N7, N10, N12	The revealed degraded states of the POE, in ascending order of state from State II to State V
N6, N8, N9	The age-based estimated states of the locking device corresponding to states from State II to State IV, in ascending order
N11	A revealed failure, State V, of the locking device
P11	Inspection is enabled for the POE and locking device
P10	Inspection fails to reveal the state of the POE and locking device
P9, P7, P8	Inspection of the POE and locking device is underway

P13, P14, P15, P16	Replacement is secluded for the POE, following a delay, for each revealed state of the POE
P19, P20, P21, P22	Replacement is secluded for the locking device, following a delay, for each revealed or estimated state of the locking device
P18, P17	A replacement of the POE is completed, and the number of such replacements is counted, respectively.
P24, P23	A replacement of the locking device is completed, and the number of such replacements is counted, respectively.

Table 4.14: A description of each of the places in the POE and locking device module

The fourth model in this section considers the switch position detector. Figure 4.26 gives the first layer of this model, which includes the condition of the switch position detector and inspection model. Figure 4.27 gives the second layer of the model which includes the maintenance model for the switch position detector, and the application of any system level restrictions or closures. A description of each of the places in the model can be found in Table 4.15.

In the model layer in Figure 4.26, the orange shaded region models the failure of the switch position detector, with the working state on the left-hand side, place W1, and the failed state on the right-hand side, place W2. Only two states are included here, under the assumption that intermediate degraded states cannot be quantified.

The blue shaded region models the inspection of the switch position detector, when place P4 is marked inspection is underway. This region also models the estimation of age-based states for the component. Since intermediate revealed degraded states are not included for this component they are instead estimated by the age of the component, transitions t5, t6 and t7 correspond to this age-based state estimation. This allows age-based maintenance to be included in the model. This gives three age-based states, represented by places W3, W4 and W5.

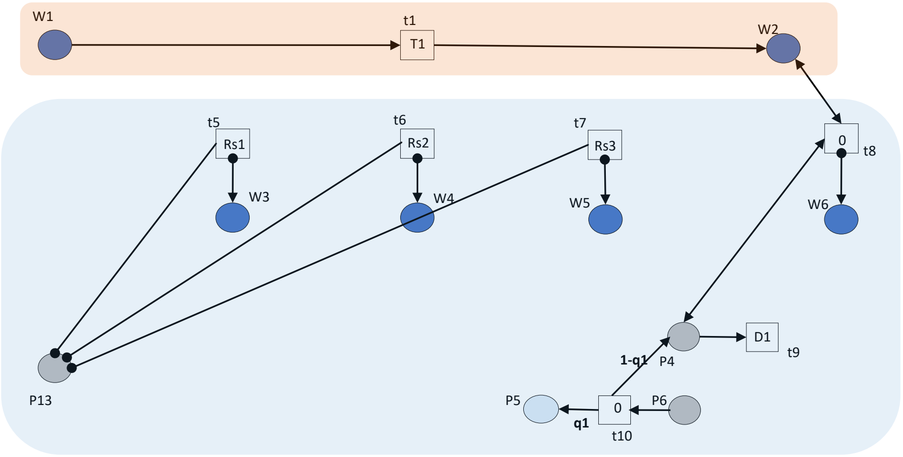


Figure 4.26: A Petri net for switch position detector degradation

The green shaded region in Figure 4.27 models the maintenance of the switch position detector. Transitions t12, t13, t14 and t15 model this maintenance, and are assumed to return the component to the ‘as good as new’ state. Here, the marking of the places, W3, W4 or W5, triggers opportunistic, routine or priority age-based replacement of the switch position detector.

The red shaded region ensures that replacement is carried out following the correct scheduling delay, through transitions t21-t26, and applies any system level state changes, through transitions t27 and t28. Speed restrictions are also applied based on age, if place W5 is marked.

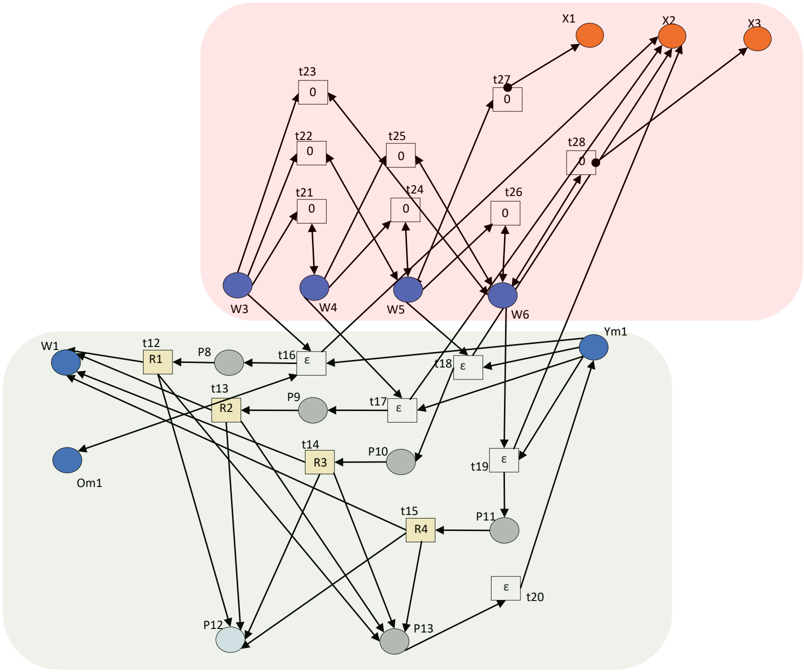


Figure 4.27: A Petri net for switch position detector maintenance

Places	**Interpretation**
W1, W2	The unrevealed degraded State I and State V of the switch position detector
W3, W4, W5	The age-based estimated states of the switch position detector corresponding to states from State II to State IV, in ascending order
W6	The revealed failure, State V, of the switch position detector
P6	Inspection of the switch position detector is enabled
P5	Inspection of the switch position detector fails to reveal the state
P4	Inspection of the switch position detector is underway
P8, P9, P10, P11	Replacement of the switch position detector is scheduled after a delay, based on the revealed or estimated state
P13, P12	Maintenance of the switch position detector is completed and the number of such maintenance actions is counted, respectively

Table 4.15: A description of each of the places for the switch position detector models

Figure 4.28 gives a model for an external signal failure. A description of each of the places can be found in Table 4.16. In this case, it is assumed to occur after a time governed by the distribution labelled with T1. Two failure modes are modelled. The first failure mode, represented by place Sf2, models a signal failure that causes the switch rails to become falsely unlocked. The second failure mode, represented by place P2, models a signal failure whereby the switch rails remain locked in a non-hazardous position. There is a probability associated with transition t2 that corresponds to either hazardous or non-hazardous external signal failure. It is also assumed that after a delay time, D1, the external signal failure will be resolved.

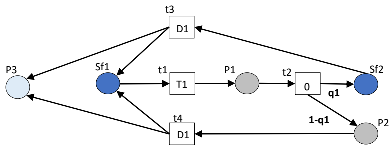


Figure 4.28: A Petri net for external signal failure

Places	Interpretation
Sf1	There is no external signal failure
P1	There is an external signal failure
Sf2	The external signal failure is hazardous
P1	The external signal failure is not hazardous, the system fails safe
P3	Counts the number of external signal failures

Table 4.16: A description of each of the places for the external signal failure model

4.3.2: System Level Component Inspection Strategies

The component models presented in Section 4.3.1 of this chapter are connection with system level inspection and maintenance models. When any of the transitions fire in the models presented in this section, the corresponding places in each of the component models are marked. To clarify, the models in this section enable component inspection across all of the component level models presented in this chapter. This ensures that components are inspected at the same time.

Three classification areas were chosen for the component inspection methods on a system level [150]. The first was visual inspection tasks. This included visual inspection of the rails and crossing, alongside the fastenings, slide chairs, stretcher bars and supplementary drive and ballast. The second inspection type included in the model were instrumental inspection tasks. These included ultrasonic testing of the rails and crossing and geometry measurements. The final inspection type includes a functional test of the POE movement, locking and detection.

An inspection interval was assigned to each maintenance inspection method, such that the inspection method is applied to the whole S&C at the same time. For example, a visual inspection is carried out of all the S&C components simultaneously. Figure 4.29 gives the Petri net model that governs the intervals for visual inspection, instrumental testing and functional testing of the POE, and switch rail locking and detection devices, in order from left to right. Place P1 counts the number of visual inspections of the S&C, Place P2 counts the number of instrumental testing actions of the S&C, and Place P3 counts the number of functional testing actions of the S&C.

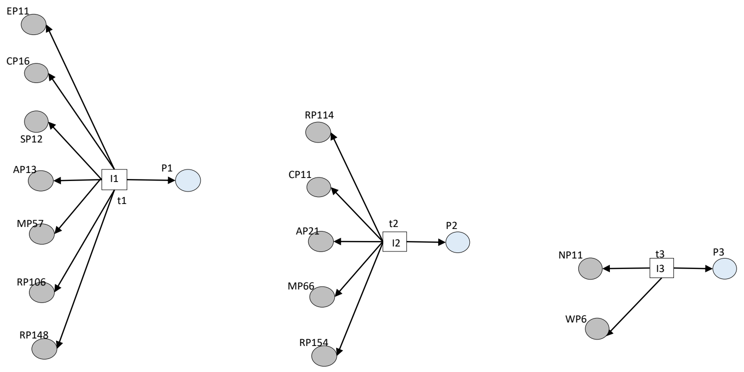


Figure 4.29: A Petri net model for the inspection interval for visual inspection, instrumented testing and POE, locking and position detector testing in order from left to right

For each component inspection method, different inspection intervals can be tested to consider the impact on the final derailment frequency and overall system state.

4.3.3: System Level Maintenance activation

In addition to the models for degradation, inspection and maintenance of the components of the S&C, models are included that govern the scheduling of each maintenance action across the component level models presented in Section 4.3.1 of this chapter. The models for this are presented in this section. The models presented here activate the maintenance actions across all the component level models. This is done by marking places that are shared across multiple component Petri net models. If maintenance is enabled for a specific component, this enables certain similar maintenance actions across all component level models.

In this model, if it is found that a component is residing in State III, then routine maintenance is scheduled following a delay. The length of this delay can be varied. If it is found that a component is residing in State IV, maintenance is scheduled as a priority. If it is found that a component is residing in State V, the failed state, it is assumed here that maintenance is scheduled immediately following a short delay. If a maintenance action is already being carried out on the S&C, and it is found that any component is residing in State II, and the maintenance action required for this second component is similar to that of the first, then the second component can be replaced at the same time to give an opportunistic strategy. Later in this chapter, the effectiveness of this strategy is tested.

In addition to this, for some components, the maintenance is completed based on the age of the component or on the discovery of a failure. As with the revealed failures, routine, priority or opportunistic maintenance can be scheduled, based on the components age.

Finally, complete replacement of the S&C is included in this model. This has been assumed to either occur following a set period, or after a derailment has occurred. The time at which this full replacement is carried out is also tested later in this chapter, to observe the effects on the system state and derailment occurrence.

In this model, the maintenance actions, with the exception of full replacement, are grouped into four categories: ballast tamping, ballast undercutting, manual intervention such as component replacement or repair, and rail grinding. If a maintenance action is required, due to a revealed state or estimated age-based state, then the corresponding maintenance activity is enabled after a delay. The delay is based on the expected availability of resources or a maintenance strategy for testing.

Figure 4.30 gives a Petri net for the full replacement of the S&C. Here, Transition t1 represents the scheduling delay of a periodic full replacement of S&C. Transition t3 represents the scheduling delay of a full replacement of the S&C following a derailment. Transition t2 represents the full replacement, and returns the marking of all places in the Petri net to that of the original marking, apart from those required to count actions throughout the Petri net model. While the full replacement is taking place, the S&C is closed, represented by the marking of Place X2. When Place P1 is marked, then full replacement of the S&C is enabled. Place P2 counts the number of full replacements.

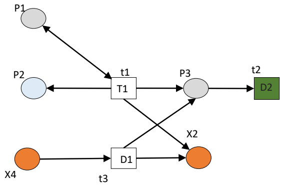


Figure 4.30: A Petri net for replacement of the full S&C

Figure 4.31 gives a Petri net for the scheduling of tamping or undercutting of the ballast. Here, Transition t8 represents the choice between each of these maintenance actions and there is an associated probability. The maintenance scheduling delay for ballast, residing in State III is given by distribution, W1, in Transition, t7. The maintenance scheduling delay for the ballast residing in State IV is given by distribution D3, in transition t5. The short maintenance scheduling delay for ballast residing in State V is given by distribution D4, in transition t6. Place Yu1 enables undercutting of the ballast, Place Ou1 enables opportunistic undercutting, Place Yt1 enables tamping of the ballast and Place Ot1 enables opportunistic tamping of the ballast. These relate to the places in the component maintenance modules. After a short delay, denoted by D6, the ballast maintenance is disabled in the Petri net to prevent repeated unnecessary maintenance actions.

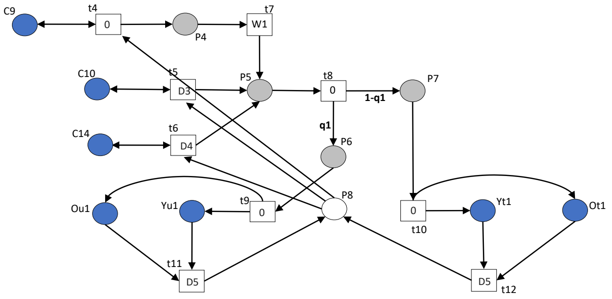


Figure 4.31: A Petri net for scheduling of tamping or undercutting ballast

Figure 4.32 gives a Petri net for the scheduling of manual maintenance interventions of components in the S&C. These interventions include: component replacement, component repair, component welding, component clearing, component cleaning and component lubrication. These actions have been grouped together, as they have less reliance on large equipment such as track grinders, tamping machines or undercutting machines, which may not always be readily available. Transitions t18 –t27 represent the scheduling of routine manual intervention for components in either revealed, or estimated, State III. Transitions t28-t42 represent the scheduling of priority maintenance for components in either revealed, or estimated, State IV. Transitions t43-t56 represent the scheduling of emergency maintenance for components in State V, following a short delay governed by distribution D7 in Transition t58. On the scheduling of a manual intervention, Place Ym1 is marked to enable the relevant maintenance actions in the component Petri net models. Additionally, place Om1 is marked, to enable any early manual intervention of components across the model that are residing in State II. Following a delay of D8 in transition t60, the maintenance is disabled to prevent unnecessary repeat maintenance actions.

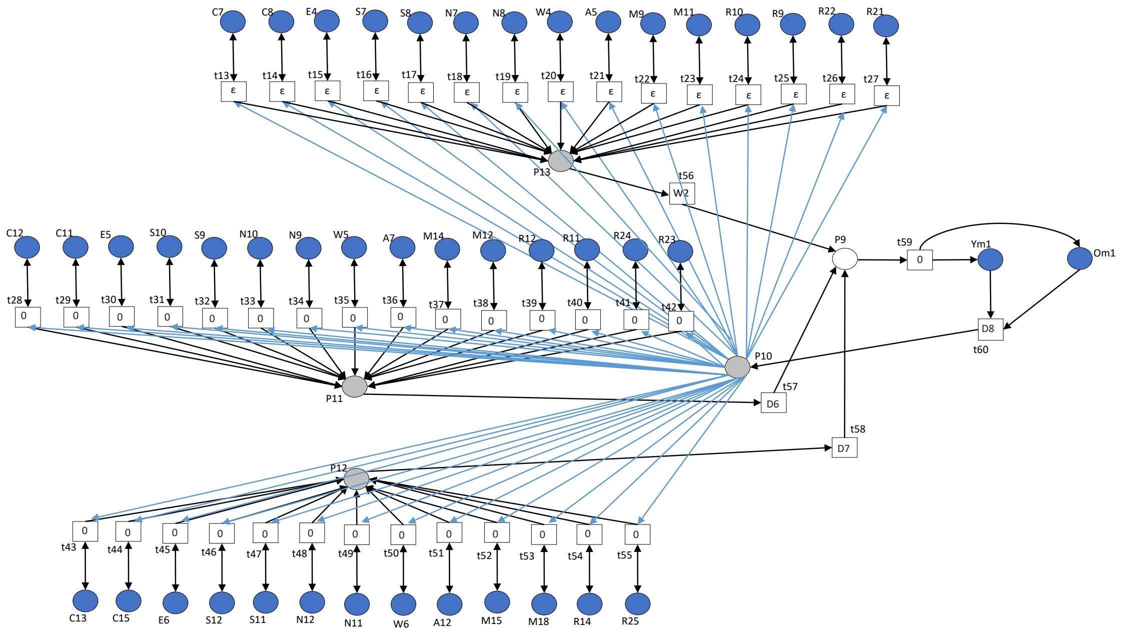


Figure 4.32: A Petri net for scheduling of component replacement or manual intervention

Figure 4.33 gives a Petri net for the scheduling of rail grinding for the components in the S&C. Transitions t61-t64, represent the scheduling of routine rail grinding for rails in the State III. Transitions t65-t68, represent the scheduling of priority rail grinding for rails in State IV. When the rails reach State V, the failed state, it is assumed in this model that they are replaced. When a rail grinding activity is scheduled, Place Yg1 is marked to enable the corresponding maintenance activities within the component Petri nets. Place Og1 is also marked to enable opportunistic grinding of any other rails in the S&C that are in state II. Following a short delay governed by distribution D10 in transition t72, the rail grinding maintenance actions are disabled across the Petri nets, to prevent repeat unnecessary maintenance actions.

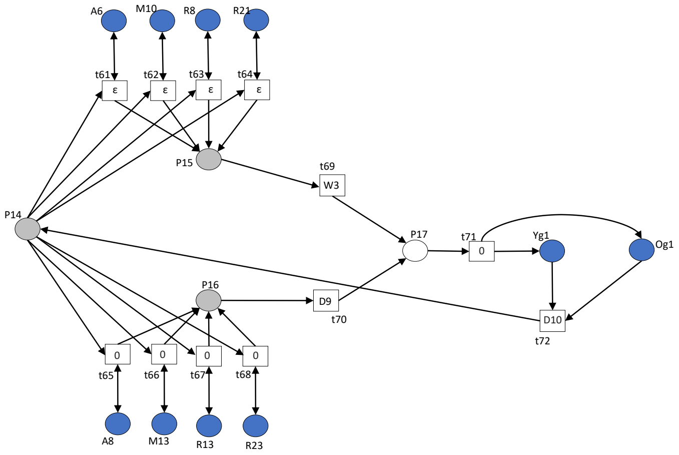


Figure 4.33: A Petri net for the scheduling of rail grinding

Varying strategies for system level maintenance scheduling can be tested, to consider the output on the system state and derailment occurrence.

4.3.4 Derailment Risk Scenarios

The risk scenarios, as given earlier in this chapter following the Fault Tree in Figure 4.5, are modelled by the Petri nets in this section of the chapter. Here, different types of component failure can cause a derailment. In each case, Place X1 corresponds to a speed restriction over the S&C, Place X2 corresponds to a closure of the S&C due to maintenance, and Place X3 corresponds to a closure of the S&C due to revealed failure. These places are marked by the discovered latter degraded states of the components across the model and can be seen in each component level Petri net model. Place X4 corresponds to the S&C state after a derailment has occurred. In this model, it is possible for more than one train to pass over a failed S&C in quick succession and derail. However, when Place X4 is marked and following a short delay, the S&C is closed for full replacement. This has been completed under the assumption that a derailment will quickly reveal the failure and extensive damage will be done to the S&C during the derailment, such that full replacement is required.

Figure 4.34 models the scenario wherein there is a hazardous switch position and a train passes over the S&C with sufficient speed to cause a derailment. In this Petri net, the switch position detector can reveal the failure in the switch position and prevent a derailment. If the switch position detector does not reveal the failure, then a derailment can occur. Place P1 corresponds to the system residing in the state where the switch is in an unrevealed hazardous position. A derailment may then occur, due to either passage of the train with no restrictions or closure on the S&C, or the passage of the train where there are restrictions in place on the S&C, but a further failure has occurred since the restrictions were applied. This is modelled by transitions t9 and t10, respectively. Place P2 corresponds to a derailment caused by a hazardous switch position. Place Dr1 counts the number of this category of derailment.

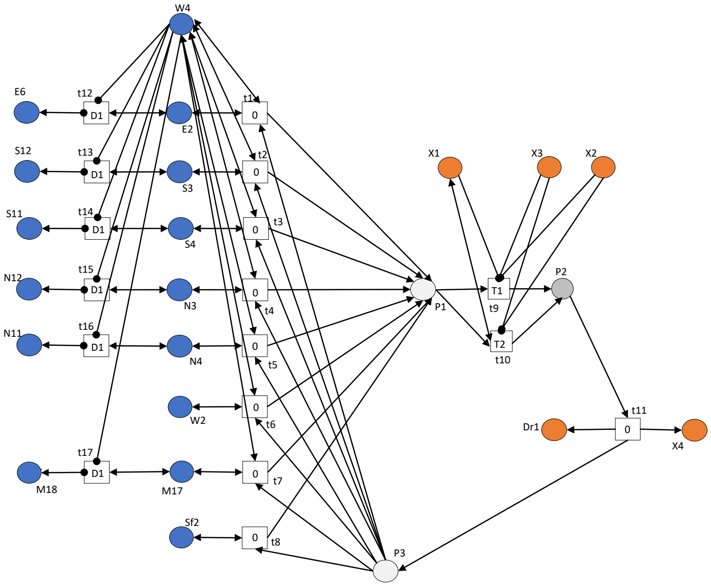


Figure 4.34: A Petri net modelling the passage and derailment of a train over an S&C with a hazardous switch rail position

Figure 4.35 gives a model for the scenario wherein there is a geometry failure and a train passes over the S&C with sufficient speed to cause a derailment. Place P4 corresponds to a state where there is a geometry failure and a derailment can occur if there is the passage of a train. As with the previous model, this may occur due to either a lack of speed restrictions or closure or the case where restrictions are applied but there is a further failure following the implementation of the restrictions, this is modelled by transition t21 and t22, respectively. Place P5 corresponds to a derailment due to a geometry failure. Place Dr2 counts the number of derailments due to this cause.

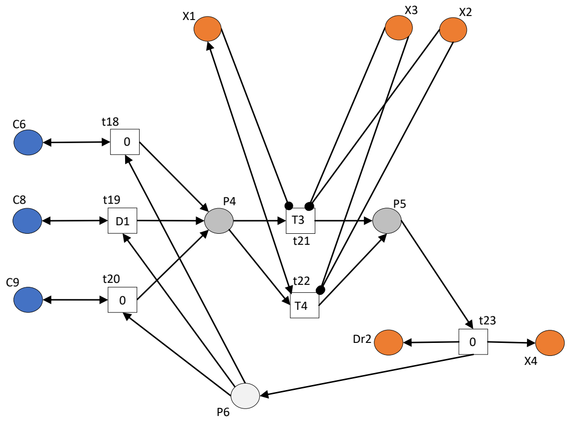


Figure 4.35: A Petri net modelling the passage and derailment of a train over an S&C with a geometry failure

Figure 4.36 models the scenario wherein there is wear causing wheel flange climb and a train passes over the S&C with sufficient speed to cause a derailment. Place P7 corresponds to a scenario where the S&C is in such a state that if a train passes over it a derailment can occur due to wear on the S&C components. This can either occur after a delay due to a lack of speed restrictions or closure, or the occurrence of a failure while speed restrictions are applied. Transitions t29 and t30 correspond to this respectively. Place P8 corresponds to a derailment due to this cause and Place Dr3 counts the number of this category of derailment.

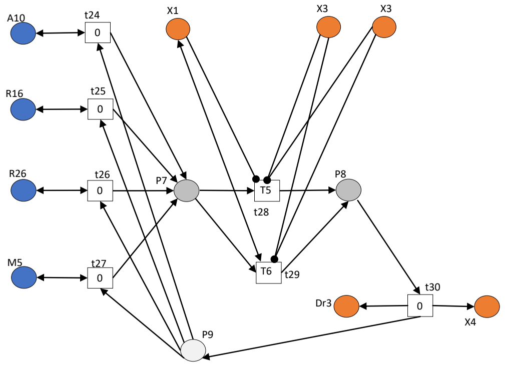


Figure 4.36: A Petri net modelling the passage and derailment, due to flange climb, of a train over an S&C with excessive rail wear

Figure 4.37 presents the scenario in which there is a rail break and a train passes over the S&C with sufficient speed to cause a derailment. In this Petri net, Place P10 corresponds to a situation where there is a rail break and a derailment can occur a train passes through the S&C. Transition t35 corresponds to a passage of a train, leading to a derailment under this condition, where there are no speed restrictions, or closures, implemented. Transition t36 corresponds to the passage of a train leading to a derailment under this condition, where speed restrictions are implemented but a further failure occurs while the restrictions are applied. Place P11 corresponds to a derailment due to the passage of the train under these conditions. Place Dr4 counts the number of such derailments.

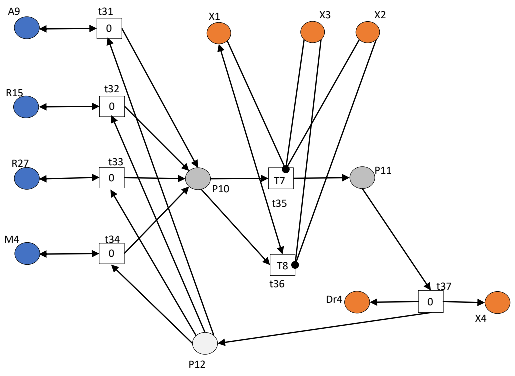


Figure 4.37: A Petri net modelling the passage and derailment of a train over an S&C with broken rail components

Finally, Figure 4.38 gives the Petri net model for a scenario where over-speeding causes derailment at the S&C. Here, Place P2 corresponds to the arrival of an over speeding train at the S&C. This arrival rate is conditional on any speed restrictions implemented on the S&C. In the application of this model, it is assumed that the arrival rate of an over-speeding train is more likely when speed restrictions are in place due to the driver ignoring the new speed restriction and travelling at a speed above this. This arrival rate is governed by distribution T1 and can be changed for application of the model. On arrival of an over-speeding train at the S&C, there is a probability that the derailment will occur. This is represented by Transition t2 in this Petri net. Place Dr5 counts the number of derailments due to over-speeding through the S&C.

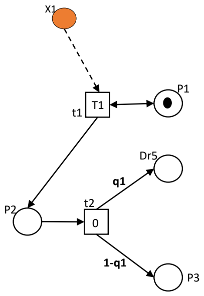


Figure 4.38: A Petri net modelling the over speeding of a train at an S&C with either a restricted or unrestricted speed control

4.4: Results for Sample Data Values

To demonstrate the capability of this modelling approach, sample values for the distributions and probabilities were assigned to Transitions within the models [151][152][153][154]. These sample values are given in Appendix 2. These values are assumed, using available estimates where possible. These sample values can be easily altered based on any available data, or expert opinion, for a specific S&C in question. Monte Carlo Simulation, with random sampling from the distributions and probabilities governing the system, can be used to generate quantitative results. The number of maintenance activities across the system and the probability of the system being in several states (working, restricted, closed for maintenance, closed due to a dangerous state or post-derailment) can be tracked. This can be done by recording the marking of Places X1, X2, X3 and X4 and the places across the model that count each individual component maintenance action.

Different maintenance strategies can be tested to investigate their impact on the state of the S&C. The models can also form a basis for optimization of the maintenance and inspection strategies to give an optimal solution to minimise both cost and derailment occurrence.

A Monte Carlo Simulations of the model was performed with 1000 runs, and the average number of maintenance actions and average system states across these runs were calculated. Two cases were applied, the first where opportunistic maintenance across the system was disabled, and the second where opportunistic maintenance was enabled across the system. Generalised result trends are discussed in Section 4.5 of this chapter.

4.4.1: Asset Management Strategy 1: No opportunistic maintenance

In the first simulation of the model, no opportunistic maintenance of the components was included. This represents a scenario where components remain in a partially degraded state until it is identified that routine maintenance is required. Figure 4.39 gives the probability that the S&C resides in each system state for each year over a 30-year period, the full replacement interval was set to 30 years. Figure 4.40 gives the average number of each maintenance actions for each year over a 30-year period. For 1000 runs of a Monte Carlo Simulation, the result was obtained after 170990.796 seconds.

It can be seen for this example that the S&C is in the working state for the majority of the 30-year time period. The S&C is rarely in the restricted state, but is more commonly closed due to a discovered poor condition. This can be attributed to the failure to detect components that are degraded. Prior to 10 years, there are limited closures due to the condition. Replacing the whole S&C system at the 10-year point would result in a repeat of the behaviour in the first 10 years, improving the system state over the 30 year period, due to the total reset transition applied during a full reset, which returns the system to the same state as the start of each run of the simulation.

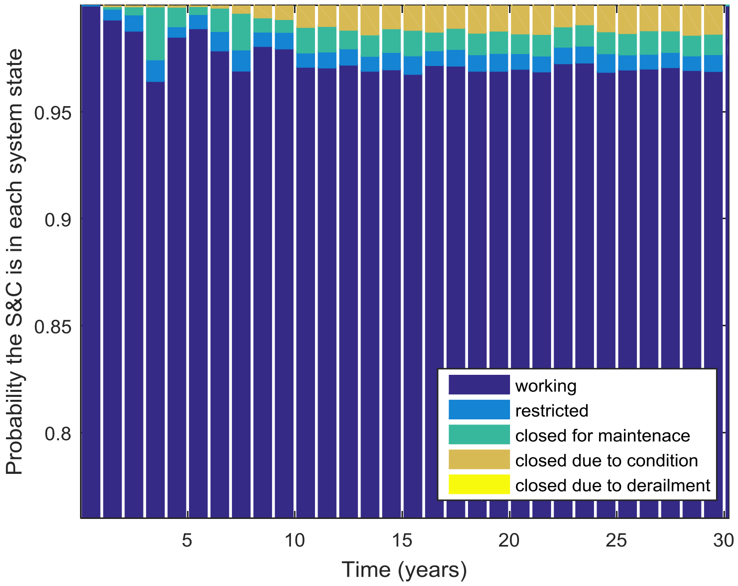


Figure 4.39: The S&C system state for the model without opportunistic maintenance

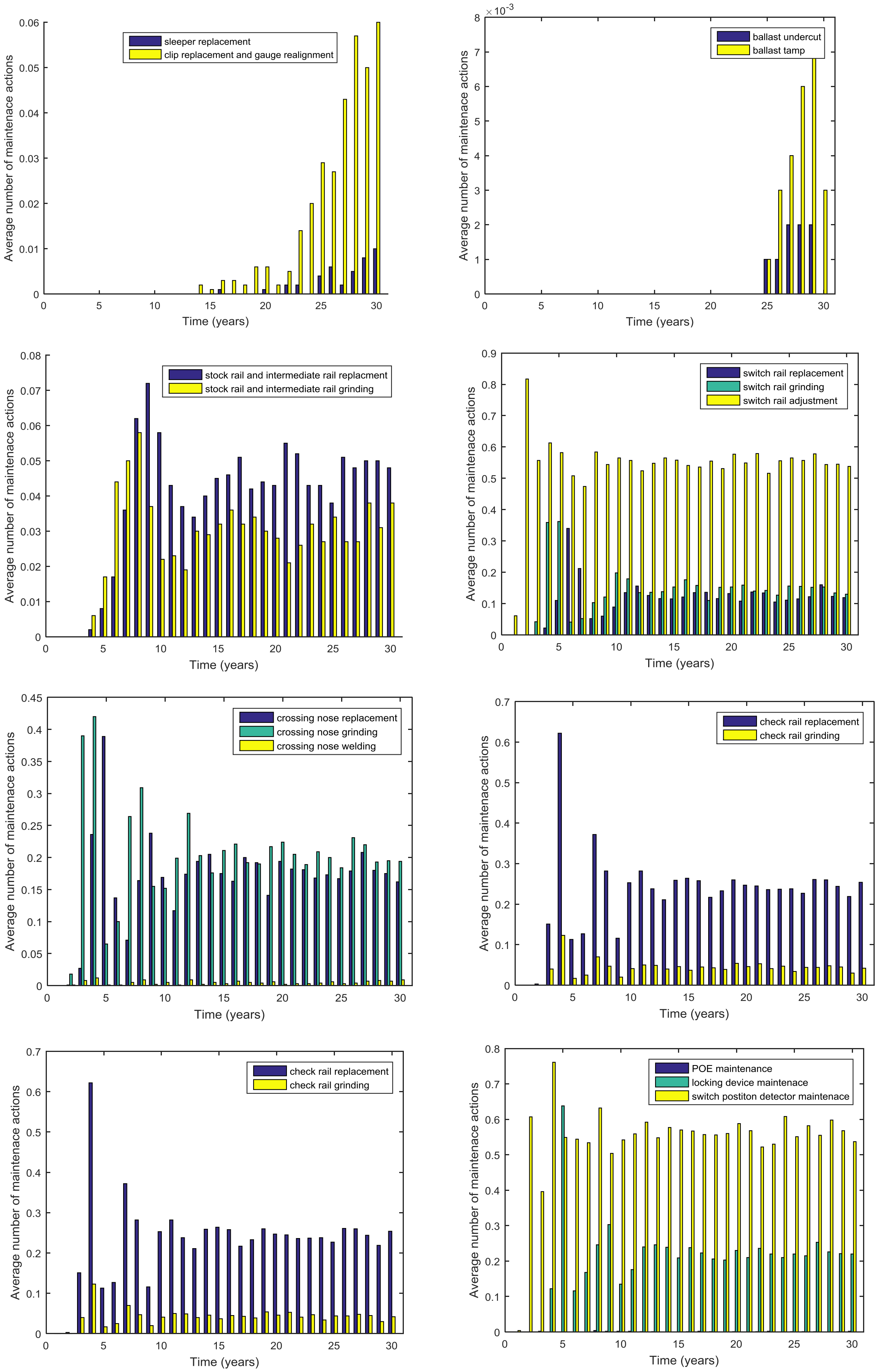


Figure 4.40: The number of each maintenance action for the model without opportunistic maintenance

The number of maintenance actions tends to follow two sorts of behaviour. Periodic behaviour in the number of maintenance actions in these results can be seen for some of the components. In some cases this behavior is more distinct, such as that for the stock rails, suggesting that these components have a more defined degradation time and are maintained periodically, often returning them to the 'as good as new' state. The second behaviour shown by some components is an approximately constant number of maintenance actions. This is seen in components with frequent maintenance actions such as the slide chair lubrication and clearing, as shown in Figure 4.40. It can also be noted that there is an increase in the number of maintenance actions at approximately the 5 year point, which corresponds to the first maintenance action for many of the components. Following the 5 year point the behaviour levels, this can be attributed to the stochastic nature of the model which can cause time-averaging of the results. The sleepers and clips show an increasing level of maintenance as the system ages. The long degradation times assigned to the ballast model give rise to maintenance which only takes places after 25 years.

The expected number of derailments, over the 30-year period for this case was 3.088, when the S&C is not replaced within the 30-year time period. Taking the number of derailments at each 5 year interval, and assuming that full replacement leads to a repeat in the behaviour of the model, gives rise to the predictions in Table 4.17.

		Full replacement time					
		5 years	*10 years*	*15 years*	*20 years*	*25 years*	*30 years*
Failure mode	*Switch position error*	1.872	1.986	1.984	2.0085	2.0604	2.043
	Geometry error	0	0	0	0.0015	0.0036	0.005
	Rail wear	0.012	0.342	0.426	0.468	0.4896	0.518
	Rail break	0.03	0.384	0.438	0.474	0.4932	0.498
	Over speeding	0	0.006	0.016	0.0195	0.0204	0.024
	Total	**1.914**	**2.718**	**2.864**	**2.9715**	**3.0672**	**3.088**

Table 4.17: A table of the expected number of derailments, with different system replacement times and no opportunistic maintenance

4.4.2: Asset Management Strategy 2: Opportunistic maintenance included

Secondly, opportunistic maintenance was included in the model, such that components discovered to be in State II are maintained if there is an alternative maintenance action scheduled. The rest of the model was kept consistent to the previous results to enable a comparison.

Figure 4.41 gives the probability that the S&C resides in each system state for each year over a 30-year period. Figure 4.42 gives the average number of each maintenance actions for each year over a 30-year period.

There are similarities between the results for the probability that the S&C is in each state in the cases with and without opportunistic maintenance. It can be seen that when opportunistic maintenance is added, the S&C spends less time in the restricted state and less time in the closed state due to poor condition.

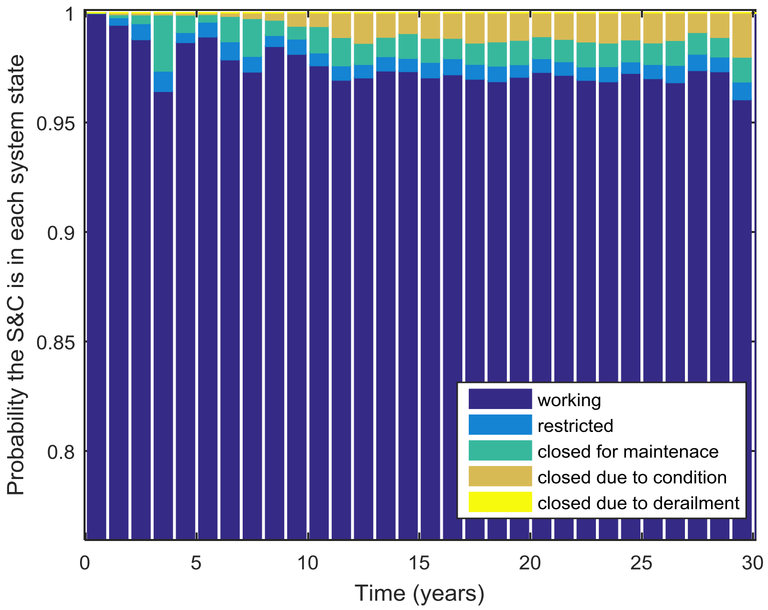


Figure 4.41: The S&C system state for the model with opportunistic maintenance

For the component maintenance actions, with opportunistic maintenance included, it can be seen that the number of maintenance actions at each time is higher for some components, such as the stock rail and POE. This suggests that in this case they may be being replaced prior to reaching the end of their useful life. For other components, the pattern of replacement is approximately the same, suggesting that they are maintained when they reach the end of their useful life in both cases.

The expected number of derailments over the 30 year period for the second maintenance strategy was 2.868, demonstrating that the addition of opportunistic maintenance to the model decreases derailment occurrence. Taking the number of derailments at each 5 year interval, and assuming that full replacement leads to a repeat in the behaviour of the model, gives rise to the predictions in Table 4.18. These results are dependent on the parameter values used, hence the analysis should be repeated when applying real data.

		Full replacement time					
		5 years	*10 years*	*15 years*	*20 years*	*25 years*	*30 years*
Failure mode	*Switch position error*	1.638	1.89	1.924	1.973	1.95	1.959
	Geometry error	0	0	0	0	0	0.002
	Rail wear	0.042	0.378	0.472	0.561	0.584	0.595
	Rail break	0.03	0.24	0.232	0.27	0.276	0.277
	Over speeding	0.006	0.009	0.02	0.024	0.0288	0.035
	Total	**1.716**	**2.571**	**2.648**	**2.828**	**2.839**	**2.868**

Table 4.18: The predicted number of derailments for different full replacement frequencies

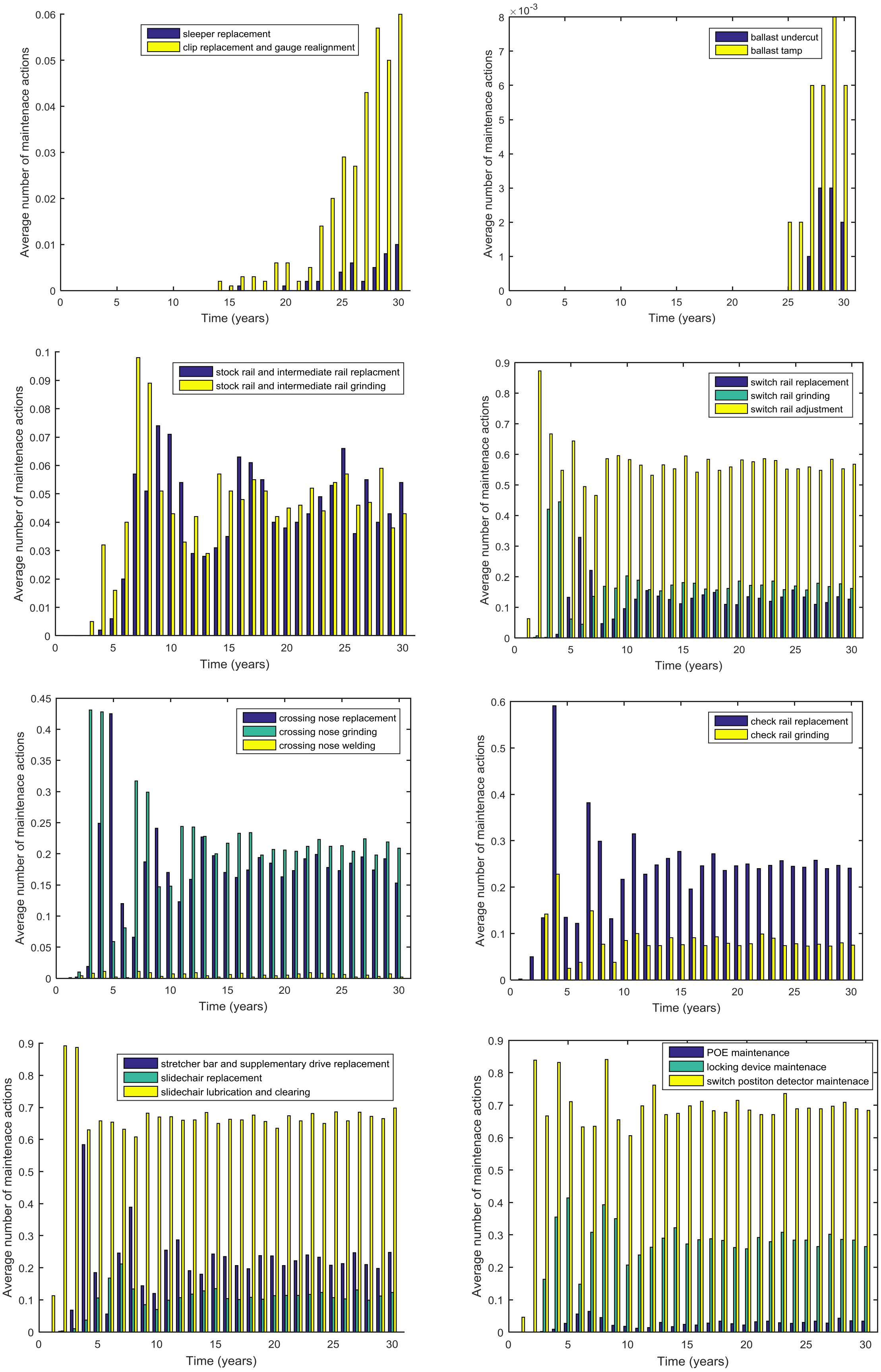


Figure 4.42: The number of each maintenance action for the model with opportunistic maintenance

4.4.3: Comparison of Derailment Results

Figure 4.43 gives the convergence of the average number of derailments over the 30-year period for each of the asset management strategies tested in this chapter. It can be seen that over the 30-year period the average number of derailments seems to converge. The rate of model convergence is discussed further in Chapter 6 of this thesis. To look at behaviour at specific time intervals, as opposed to over the whole time frame under consideration, will likely require a higher level of convergence since this removes averaging introduced intrinsically over time within the system model.

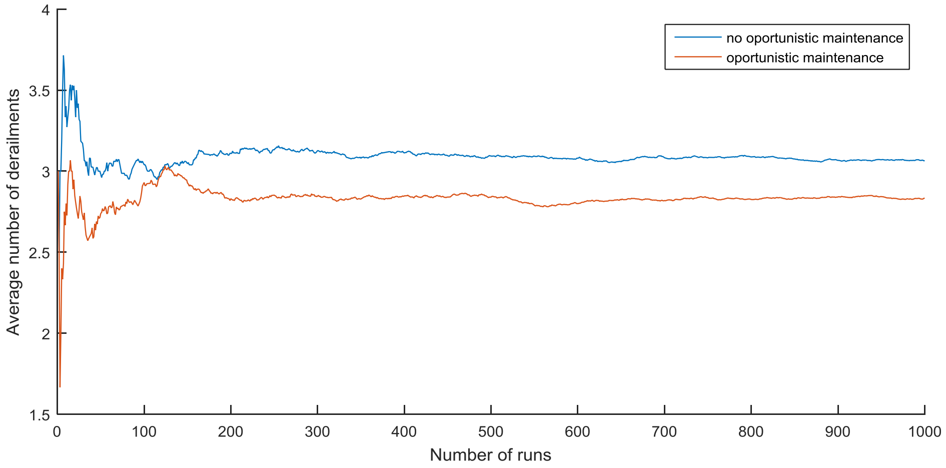


Figure 4.43: A figure showing the convergence of the average number of derailments, at 30 years, for each asset management strategy

The results in Table 4.17 and Table 4.18 show how the different derailment causes are impacted by the maintenance strategies implemented in this study. In general, it can be seen that decreasing the interval between system replacement reduces the expected number of derailments over the 30-year period, as does introducing opportunistic maintenance of components. The probability that there is a derailment due to over speeding is the similar for each case, because the over-speeding module is only dependent on any applied speed restrictions, which are similar for each strategy. From these results it can also be seen that the expected number of derailments due to a fault with the switch position is slightly influenced by the introduction of early opportunistic maintenance. The occurrence of a derailment due to a rail break is heavily influenced by the introduction of opportunistic maintenance. This can be attributed to the faster rates of ageing assigned to components within these modules. In contrast, the expected number of derailments due to the wear on rails in the S&C is not improved by the introduction of opportunistic maintenance, with approximately the same number of derailments for each of the strategies. Finally, derailment due to a geometry error was lower when opportunistic maintenance was included, but occurred rarely in both cases. This can be attributed to the slow degradation rates assigned to the ballast, sleepers and clips. Further strategies can be tested in the same way and a cost analysis can be carried out to numerically evaluate the benefits of each strategy. Different costs can be assigned to maintenance actions completed in isolation, compared to those completed in the same visit.

From these demonstrative examples it can be seen that, for the data used as input for the model, the addition of an opportunistic maintenance strategy and a shorter time between full replacements can be used to reduce the expected number of derailments over the 30-year period, but at an increased lifecycle cost.

These results have demonstrated some of the capabilities of this modeling approach to test the impact of different maintenance and inspection strategies. In the same way, the distributions used for scheduling delays for both inspection and maintenance can be varied to study the impact on the number of maintenance actions, system states and derailment occurrences.

4.4.4: Frequency of Derailments

To obtain the frequency of derailments, the expected number of derailments per year can be found. Figure 4.44 gives the frequency of derailments per year for the second maintenance strategy including opportunistic maintenance for the 30-year time period, without full replacement.

The number of derailments that occurred in this model over the 30 year period for the 1000 simulations is small, thus producing some unstable behaviour in the results. Fewer simulations result in more instability on the results. More simulations can be completed to reduce this instability in order to gain a more accurate value for the frequency of derailment at each time.

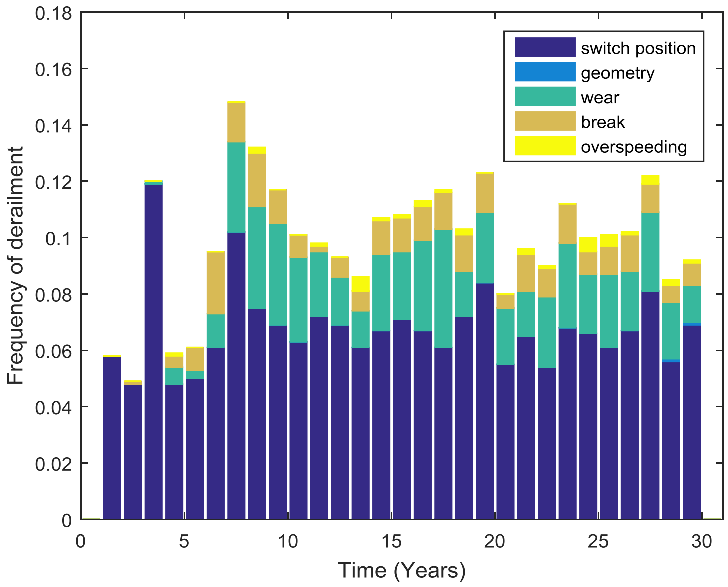


Figure 4.44: The frequency of derailments for the maintenance strategy including opportunistic maintenance

The frequency of derailment can be combined with a measure of consequence, in each case, in order to give the derailment risk of the S&C.

4.5 General Result Trends

A number of general trends in the results for this model can be observed, across different components and under different conditions. The results for the number of maintenance actions observed for each component tend to follow one of three trends:

1. Initially there is a low level of maintenance, followed by an increase. An example of this can be seen in the example on the left of Figure 4.45. This trend is expected for components that take a longer time to degrade and fail, such that limited maintenance actions are required, especially in the earlier stages of the system life.
2. The maintenance actions follow an oscillating pattern, with some damping such that there is initially a sharper narrower increase followed by a number of peaks with increasing width and decreasing height. An example of this can be seen in the central graph of Figure 4.45. This demonstrates a cyclic pattern of increased maintenance, then improved component condition hence decreased maintenance requirement, followed by decreased component condition and then increased maintenance requirement. The damping behavior is expected due to the accumulation of differences in the times sampled in each run of the model, with more accumulated differences as time progresses.

3. The maintenance actions show an initial increase followed by an approximately level rate. This can be seen for the switch rail adjustment in the example on the right hand side of Figure 4.45. This is expected for more frequent maintenance actions, such that the damping behavior discussed in the previous example occurs over a short time period to give an approximately constant rate.

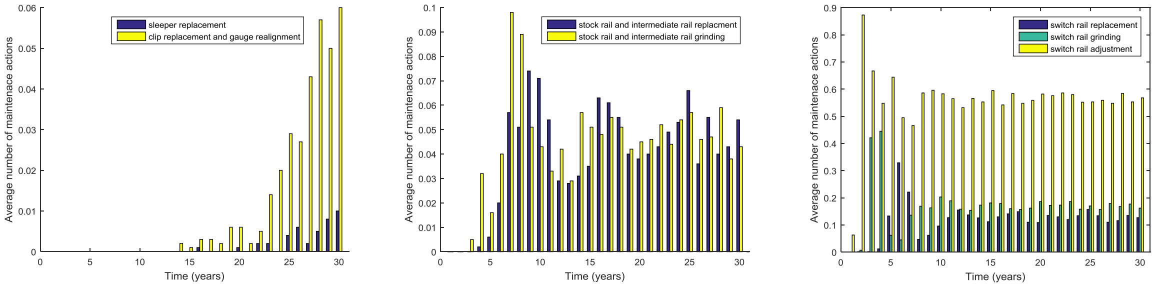


Figure 4.45: Examples of the different trends seen in the model results

Another observed trend is, when opportunistic maintenance is enabled in the model, some component types show a greater increase in the number of maintenance actions, whereas some components show minimal differences. This suggests that the components showing a minimal increase are the ones that most commonly trigger the maintenance actions and that the components that show the increase are those that are maintained early, in addition to the former. Hence, in these cases any observed decrease in derailment frequency can be mostly attributed to early opportunistic maintenance of some, but not all, of the components in the model.

For the application of full system restrictions there are two trends shown in the results. The first is an increase in the probability of the restricted or closed state, followed by a level behavior. This is shown for the state with speed restrictions and closure, due to the condition. This suggests an initial increase due to component ageing, followed by a leveling due to the maintenance actions controlling the condition. The second trend is seen for the closures due to maintenance, and is periodic with some damping behavior. The trend mimics the periodic pattern seen in some of the component maintenance numbers over time. The derailment frequency also follows this pattern, in the example presented here. Both these cases link to the interlocking cycle of component condition and maintenance described in the second part of the maintenance trend discussion.

4.6: Discussion

This chapter has introduced a Petri net model for the components in an S&C and considered how the failure of these can be combined with protection measures and train speed to predict the frequency of derailments by cause and the probability that the S&C is in different system states at each time. The model can be extended to include further components, such as the sub-base, and to offer different S&C configurations. In total, the model developed has 780 transitions and 616 places.

Two maintenance strategies have been tested for sample model inputs, to demonstrate the capacity of the method to provide a numeric tool for the analysis of different asset management strategies. These strategies look at potential benefits that can be gained by including opportunistic maintenance, where components can be replaced prior to reaching the end of their useful life. Further parameters in the model can be varied such as: inspection frequency, routine maintenance scheduling delays, full replacement scheduling and time between full replacement and opportunistic maintenance. This allows multiple strategies to be compared to make an informed decision on S&C management. In this analysis a penalty has not been assigned to the loss of useful life for a component. This is under the assumption that a repeated loss of useful life will require more new components across the system lifetime, hence inherently increasing the physical cost due to this loss of life.

These sorts of models can also form the basis of an optimization algorithm to automatically find solutions to give a lower risk within a constrained budget. This is discussed further in Chapter 6 and applied to the model developed in Chapter 5.

This method is reliant on a reliable source of data in order to give the degradation rates for the different components within the S&C and to validate the results of the model. The method is thus limited by the usefulness and reliability of the data available. However, expert opinion or data taken from extended life testing studies can be used to make predictions in the absence of historical condition monitoring data. There are also several assumptions made within the model as to the discretised degradation pathways of each component and the independence between each component degradation. The component degradation, inspection and maintenance models can be expanded and adjusted if there is data to support component degradation rate dependence, or if further inspection or maintenance actions are used. Similarly, additional states can be included for each component if required.

With an increase in model size there is an increase in the amount of time required to simulate the model in order to obtain a convergent answer. It is also difficult to quantify the accuracy of the large Petri net model for different model structures. Hence, assumptions are made based on the knowledge of the modeller. Again, this is discussed later in this thesis, in Chapter 7.

4.7: Parameter Assumptions and Use of Data

The examples in this chapter have been used to demonstrate the model, with assumed parameters. Some of these parameters can greatly impact the outcomes of the model, especially those parameters that can directly impact the derailment frequency. For example, test simulations with a decrease in the probability of a successful inspection can lead to a higher occurrence of derailments.

In this model the predicted number of derailments is most sensitive to assumed model parameters. This is due to the rare nature of this outcome. Hence, changes in component related parameters can strongly impact the number of derailments. This is especially true in cases where there are not backup systems in place. For instance, changing the parameters that govern the locking device such that it has a faster depredation rate, a slower inspection rate, a lower probability of successful inspection or a slower repair rate, such that it fails frequently in a dangerous way, greatly increases the frequency of derailments predicted by the model.

Two cases can arise due to incorrect assumptions about the model input data. Firstly, the model can over predict the derailment occurrence; this can lead to increased cost of maintaining and inspecting the system to try and prevent the predicted derailments. Conversely, care should be taken when using the predictions of the model in case there has been an underestimate of the frequency of derailment, due to assumptions that the component condition, maintenance and inspection are better than true. The maintenance cost and system state model outputs are less sensitive to assumed parameter values, as they are mostly controlled by multiple more common events such as maintenance actions across the model. Of course, if all parameters supplied across the model are unreliable then these outputs will also be unreliable.

However, despite the assumptions for the demonstration of the modelling approach, the model still shows how the logic of component maintenance, inspection and failure can be combined to give a system level model. The model allows the combination of parameters taken from expert opinion with parameters taken from data gathered in the field, or through extended life testing, and parameters that can be input by the user to test different maintenance and inspection strategies. Some generic result trends have also been extracted from the model, which can be interpreted for different scenarios, to consider over or under maintenance of components. The model can be quickly adapted to incorporate real data, and then used to make predictions and test different maintenance approaches.

Data should be collected for the degradation rate of each component; this can be from extended life testing of components or data gathered through condition monitoring. This forms the basis of the model and so should be of as high accuracy as possible. The data should be used to find the parameters and distributions for the transitions governing the degradation transitions within each component level model. This will improve the model as any assumptions about the number of states of the model or the effect of imperfect maintenance actions can be adjusted to fit the model more closely with the available data.

In addition, data can be collected for the maintenance and inspection strategies currently applied to the system. Such as: the time interval between identifying a failure in each component and the components repair, or the inspection interval of each component. This data can feed into the maintenance and inspection transitions within the model. This can improve the model by allowing the removal of assumptions about the maintenance actions applied to each of the components. Collecting this data will allow the model to be used to make an assessment of the current asset management strategy. However, the accuracy of these parameters is less crucial to the model success as they can be varied in order to test different strategies.

Data should also be collected for the rate of system level closures, restrictions and derailment occurrences. This data can be used to validate the current model predictions and make any required amendments to the model structure to bring the model more in line with reality. This can improve the model by making it more realistic, hence improving any future predictions.

4.8: Contributions

A novel model considering S&C condition, asset management and predicted derailment frequency is a key contribution of this chapter. The model improves on the state of the art as it goes into further depth and detail in comparison to models available in literature. This includes the modelling of individual sleepers and clips, and their combined impact on the system state. The model also includes imperfect maintenance actions, such as ballast tamping and rail grinding. The assumption of perfect inspection actions is also removed in this model. The approach also includes multiple maintenance types and allows for dependencies introduced through maintenance actions. Improvements with this approach can also be seen with the inclusion of preventative maintenance actions within a hazardous event framework. Hence, the model can be used to inform future maintenance decisions. System level restrictions and closures are also modelled, which demonstrates an improvement on current derailment risk assessment methods for S&C, which consider only derailment occurrence.

4.9: Conclusion

This chapter has focused on developing a modelling tool that can consider the impact of different maintenance strategies on the frequency of a derailment at an S&C. An S&C system has been defined for the demonstration of the method based on the review of current literature around S&C degradation and asset management, given in Chapter 2.

A Fault Tree for the derailment occurrence is presented. This forms the basis for the models given in the chapter. Discreet states are assigned to each component within the S&C system to model the condition from the perfect state to the failed state for each component.

A Stochastic Petri net model is used to model the interaction between the component conditions, the maintenance and inspection strategies and the passage of a train through the S&C with the potential to cause a derailment. This model is split into various interconnecting sub-modules which are tied together through system level inspection and maintenance strategies. A model for the over-speeding of the train under various conditions is also included. The risk scenarios, taken from the Fault Tree in the early stages of the chapter, are also modelled in this way, taking outputs from the individual component models.

Results are obtained from the model via Monte Carlo Simulation for sample input values. The impact on the derailment occurrence, S&C system state and maintenance actions of different maintenance strategies were tested to demonstrate the capacity of the method. For the input values used here, opportunistic maintenance reduces the derailment occurrence, as does a shorter full S&C replacement interval. The frequency of derailments by cause is also presented to demonstrate how this model can be combined with a consequence analysis to give a risk assessment for different maintenance strategies.

Finally, a discussion of the potential uses of a model such as this is presented, including the capacity to use a model such as this for cost analysis and optimization of maintenance strategies over the system lifecycle. The limitations of this modelling approach are also discussed, including the dependence on reliable input data, the computational cost of simulation of the model and the difficulty in quantifying the uncertainty on the outputs of the model.

In conclusion, this chapter has provided an application of a Stochastic Petri net approach to an S&C system to explore the benefits and challenges of such a method. The developed model includes imperfect inspection and maintenance, and system level inspection and maintenance strategies, including opportunistic maintenance strategies. The work presented here demonstrates that a Stochastic Petri net framework can be used to predict hazard frequency and model different asset management strategies. The model assumes a consistent maintenance strategy throughout the S&C system life, mostly based on the revealed component conditions. The next chapter of this thesis presents a Petri net model for the fire protection system on an underground station. This model focuses further on age-based preventative maintenance strategies and introduces different time phases of the system lifecycle with the maintenance strategy dependant on the system phase.

Chapter 5 Modelling Underground Fire Protection Systems

5.1: Introduction:

Fires in underground railway networks can have devastating consequences in terms of economic damage and loss of life. This has been demonstrated in accidents such as the King's Cross fire, the Baku fire and the Metro Station fire in Daegu [155].

The "Fire Precautions Sub-surface Railway Stations Regulations", 1989 [156], which were updated and amended in 2009 [157]**,** provide regulations for sub-surface railway stations in England. The regulations outline evacuation requirements in the event of a fire, including removal of obstructions or combustible material from escape routes and requirements for the use of fire-resistant materials and construction methods. Regulations are also given for the training of staff, maintenance of fire protection systems and additional precautions, such as, the banning of smoking on underground railway systems. In addition, regulations cover the means for fire fighting and fire detection and warning systems.

Within these regulations, in addition to fire hydrants, extinguishers and access for the fire brigade, an automatic suppression system must be installed in machine rooms, storage areas and any area used as a shop. The station must also have an automatic fire detection system, including the use of heat detectors where possible and smoke detectors otherwise. The detection system must be capable of operating via manual call points. A fire alarm and public address system must be installed.

A model is presented in this chapter to assess the performance of an automatic fire protection system to consider the probability that the system will fail to respond on demand, throughout the system life. This model can also be incorporated into a model to give an estimate of the system failure risk, this is demonstrated in Chapter 6.

5.2: System Definition

The models presented in this chapter cover the potential failures of automatic fire protection systems in an underground station. This includes an automatic deluge system, a fire detection system and an automatic alarm system. Included in this is the interaction of public or staff with the systems. A diagram of the fire protection system created for illustration in this chapter is given in Figure 5.1. The system modelled is a pressurized ringmain deluge system. Detection of a fire is possible by either a circuit containing heat detectors or a circuit containing smoke detectors. Notice of the fire is given by an alarm sounder circuit.

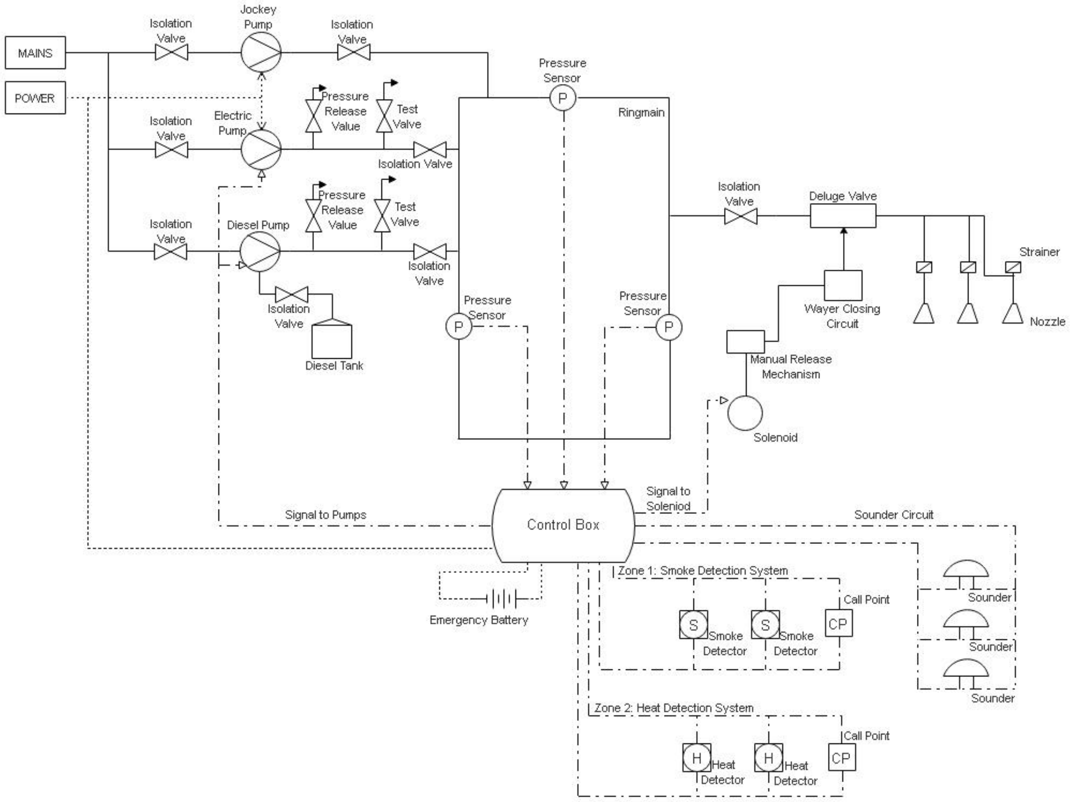


Figure 5.1: A diagram of the protection systems modelled in this chapter

The deluge system is comprised of a ringmain which is kept under pressure when the system is inactive. Water is pumped from the water mains into the ringmain to maintain the pressure via an electrically powered jockey pump. When there is no need for water flow, the deluge valve is in a closed position, preventing water from leaving the ringmain, passing though the pipework and out of the sprinkler head nozzles. The pressure of the system is monitored by pressure sensors.

The fire can be detected by one of two types of circuit. The first circuit consists of a set of smoke detectors and manual call points. This type of circuit can be installed in station areas where the temperature is regularly high, such as machine rooms. The second type of circuit consists of heat detectors and manual call points. In this model it is assumed that the smoke detectors, heat detectors and call points are sufficient in number and adequately placed such that if a fire occurs then failure of detection of the fire can only be caused by a component failure or human operating error, and not due to an incorrect location of the components. On detection of the fire by either type of circuit, a signal is sent to the control box. The control box is powered by mains electricity with a battery back-up. On detection of a fire, the control box sends a signal to the sounder circuit activating the alarm system. It is assumed that the alarm sounders are adequately arranged to inform passengers and staff of a fire.

The deluge system is triggered when the deluge valve is opened. The deluge valve is connected to a water closing circuit which maintains even pressure across the diaphragm of the deluge valve. When water leaves the water closing circuit, there is a pressure difference across the deluge valve that causes it to open. The water can leave the water closing circuit via either the manual release mechanism or following the opening of the solenoid valve, due to a signal from the control box [83].

Once the deluge valve opens, water flows from the ringmain and out of the sprinkler head nozzles. This causes a pressure drop in the ringmain which is detected via the pressure sensors. Following this pressure drop, there is a signal sent to the main pumps to compensate for the water loss in the system.

In the system modelled in this chapter there are two main pumps, both of which are on standby and individually have the capacity to provide enough water to the ringmain. The first pump is powered by the mains electricity and the second pump is diesel powered. Both pumps provide water from the mains water supply.

There are several isolation valves in the deluge system. These enable the water flow, or diesel flow, to be shut off in the case of maintenance testing or false activation of the system. There are also pressure release valves to avoid high pressure in the ringmain, and test valves that can be opened to mimic the behaviour of the system on activation.

This system has been chosen as an illustrative example of the methodology, based on its compliance with the regulations implemented for sub-surface railway stations in ‘The Fire Precautions (Sub-surface Railway Stations) Regulations’[156] [157]. In these regulations, there is no specific recommended deluge system. A pressurised ringmain system has been chosen for application in this chapter to follow the trend identified in the literature to install deluge systems in underground stations [109] [110] [112]. Furthermore, a ringmain-based system has been chosen over a frangible bulb sprinkler system due to its robustness and suitability to operate in high temperatures, common in areas of underground stations.

In this chapter, the failure modes of the system presented in Figure 5.1 are modelled in detail to illustrate the methodology developed in this thesis.

5.3: Method

The following methodology is implemented in order to predict the unavailability of a fire protection system on an underground station:

1. Identify the system failure modes
2. Perform Fault Tree analysis for each of the failure modes to identify the contributing component failures, combination of failures or human factors.
3. Model the probability of each component failure via a Stochastic Petri net approach, to give a distribution for the probability of each component failure with time.
4. Combine the distributions for the probability of each component failure, or human factor, at each time, via the Fault Tree structure obtained in Step 2, to give a distribution for the probability of each failure mode with time.

In Step 3, a Petri net model is built that incorporates the maintenance, testing and inspection of each component. The asset management strategies are applied to the system as a whole and hence impact components across the system, to represent system level intervention strategies. Through this, the method also provides a measure of the number of maintenance or inspection actions to enable life time cost analysis for the system.

In some cases, the failure of one component is not independent from that of a second, due to a dependency introduced through maintenance strategies. For instance, if one pressure sensor is replaced upon discovery of a failure, then all of the pressure sensors are replaced. When applying this method, where dependencies are identified the Petri net model must be designed to incorporate the dependencies. The outputs from the Petri net model, in this case, are then used as input to a corresponding intermediate event in the Fault Tree to incorporate the dependencies. In the application of this method it is assumed that, apart from in certain cases, although the failure of each component is dependent on an over-arching maintenance strategy, the individual component failures for any given strategy are independent of each other.

There are other options to avoid this assumption. A Petri net can be built to combine the occurrences of component failures to give the system failure modes. This increases the model size and hence the computational cost of simulating the model. This is implemented and discussed further in Chapter 6 of this thesis. Alternatively, a Bayesian Network can be used to combine the failure of the components to give a system level failure [158]. However, this is also costly in terms of computation. In review of these options, the combination via a Fault Tree structure was selected for efficiency and to fit well with the current Fault Tree and Event Tree methods used in industry. For this system, it was deemed reasonable to assume that component failures are independent of each other for any given maintenance strategy, due to a low level of interaction between most components in the system.

This method is applied to a deluge system, detection system and alarm system on an underground station. Periodic inspection of the components and testing of the system is included in the model. For each component it is assumed that failures can occur at random but that failures occur more frequently as the component reaches the end of its useful life. There are three types of maintenance included in the model. The first occurs when a component reaches a failed state, the second at a time interval that corresponds to the estimated end of its useful life, and the third before it is estimated that the component has reached the end of its useful life. These three maintenance options are included to allow a three-phase system level asset management strategy. The inspection frequency in this model can also vary through the system life for each component. In this model three sample system phases are defined to demonstrate the modelling capability. The phase entry times, each component inspection interval and the system testing interval can all be varied, for each system phase, to consider the impact on unavailability. These phase entry times allow the system level asset management strategy to be optimised. This is discussed further in Chapter 6. These different maintenance and inspection actions are enabled depending on the phase of the system, which are defined as:

1) Phase 1: When the system is first installed and for the time this it is expected that the components are within their useful life, assume that there are a limited number of failures and repair the components when a failure occurs. This phase should incorporate the random component failures. Components are inspected less frequently.
2) Phase 2: As the system ages and some components reach the end of their useful life, it can be expected that a number of failures will occur due to the age of the components. In this phase, activate replacement of each component when it is expected that it is in a degraded state at the end of its useful life, along with replacing the component upon failure. Components are inspected at an increased frequency, in comparison to the frequency of Phase 1 inspection.
3) Phase 3: When the system has reached the end of its expected useful life and is in operation past this point, it is assumed that a larger number of failures will occur due to the age of components, which can increase the probability of a system level failure. At this time, the early replacement of each component is activated, along with the repair of components upon failure. Early replacement means the components are replaced before the expected end of useful life. The components are inspected at an increased frequency in comparison to the frequency of Phase 2 inspection.

Throughout this chapter, each model is populated with estimated values to illustrate the modelling capability, on the analysis of system data these values can be adjusted to model the specific case under consideration. These sample values can be found in Appendix 3. It is trivial to alter these values, within the custom software developed for analysis of the model, for application of the model to a specific dataset. The rest of this chapter is structured as follows: firstly, the failure modes for the system are identified and, secondly, Fault Tree analysis is performed for each failure mode. Next, a sample maintenance strategy is described for the system. A Petri net for each component is then presented along with the results for the probability of failure of each component over time, for the given maintenance strategy. A Petri net is presented for the human factor elements of the system. Finally, the probability of each failure mode at each time for the given maintenance strategy is presented along with the results for the number of different maintenance actions, system tests and false system activations.

5.4: System failure modes

Three safety critical system failure modes were identified in this chapter for the fire protection system. In addition, spurious failure is also considered. The safety critical failure modes considered are:

1) Insufficient flow at the nozzle head for the deluge system, including a failure to start and a failure once active.
2) No sound made by the alarm system, due to unavailability of the alarm system.
3) No fire detection signal to the control box, due to unavailability of the detection system.

These failure modes are selected as they represent the total failures in the automatic suppression system, such that the risk of the fire spread is increased on their occurrence. Insufficient flow at the nozzle head of the deluge system includes scenarios where the deluge system will fail to control the fire for a sufficient length of time to allow for the safe evacuation of passengers and staff in the station, and the arrival of the fire brigade. It is desirable for the deluge system to completely extinguish a fire to prevent the loss of service time and reduce repairs required following a large fire. However, in this thesis accurately modelling the risk to human life is the primary concern and so a failure of the deluge system is defined as a scenario that may impact human life as opposed to infrastructure. The alarm system failure mode considers scenarios where the alarm and notification system does not function when required, to alert passengers and staff of the need to evacuate in a safe manner, before the fire has spread to a level where it can impact human life. The detection system failure mode includes any scenario whereby a fire is not detected by either heat or smoke detectors, or by a member of the public or staff who acts upon the discovery of the fire. This detection must occur before the fire is at a level that will impact the safety of human life.

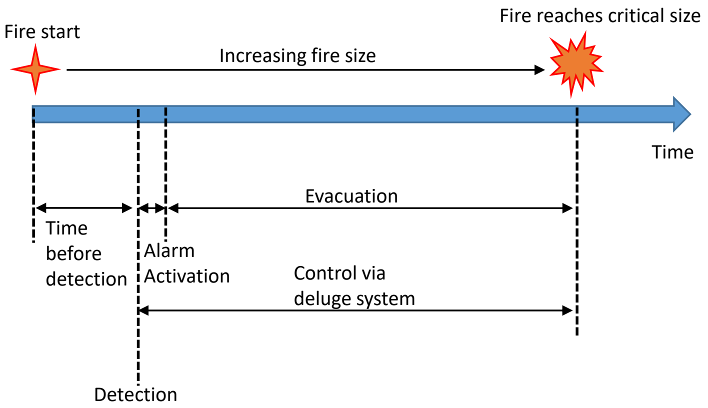


Figure 5.2: A diagram showing the times available for the fire control systems following a fire initiation

Figure 5.2 shows the actions of the system following the initiation of a fire that has potential to endanger human life, the time windows depend of the size of the fire. There must be enough time following the initiation of the fire for the public and staff within the station to be evacuated safely. In order for this to happen, the fire must be detected and the alarm activated. Following the detection of the fire, the deluge system can control the amount of time it takes for the fire to reach a critical size where human life is at risk. If the deluge system fails, the fire can reach a critical size faster, resulting in less time for the evacuation of passengers and staff.

In addition to this, the false activation of the deluge and alarm systems are included in this model. This enables the cost of such occurrences across the system's life cycle to be calculated from the model outputs. In this model, false activation of the system can occur via the following component failures [159]:

- The deluge valve fails to reach a closed position, causing water to flow from the ringmain and out of the sprinkler heads and triggering a reaction from the rest of the system as if there was a fire.
- The solenoid, or water closing circuit, fails in a way that lets water escape the water closing circuit. This results in a pressure difference across the deluge valve, causing it to open as if there was a fire.
- At least two of the three pressure sensors revealing a false drop in pressure leading to pump initiation and excessive pressure in the ringmain. This leads to a pressure difference across the deluge valve, causing it to open as if there was a fire.
- False detection by the smoke alarm circuit.
- False detection by the heat alarm circuit.
- False activation of a manual call point.

In this model, the number of false activations of the system are recorded. False activations can cause costly damage and station closures.

5.5: Fault Trees for the safety critical system failure modes

Following the method laid out in the book ‘Risk analysis for process plant, pipelines and transport’, a Fault Tree for each of the system failure modes was created [160]. When the size of the Fault Tree became large, transfer symbols were used to split the Fault Tree over several figures.

The Fault Trees were developed following several steps. Once the top event is identified it is analysed to give any contributing events. These subsequent events are then analysed for any deeper level contributing events, and this process is again repeated until component level events are reached. For instance, the event for a lack of water flow from the ringmain deluge system is decomposed to consider a lack of water flow through relevant components at increasing distances from the sprinkler head in the logic of the system, as the Fault Tree levels deepen.

Each of the basic events in the following Fault Trees has an associated Petri Net model, to give the probability that each of the basic events occur at each time. For example, the basic event probability corresponding to a smoke detector failure is modelled by a Petri Net that outputs the probability of the smoke detector failure at each time, and additionally considers the degradation, testing, inspection and maintenance of the smoke detector. This modelling process is applied for each of the basic events, resulting in a collection of models for components across the system. In addition, the Fault Trees provide the framework for combining the component failure models, to give system level failure probability.

5.5.1: No fire detection signal to the control box

The first safety critical system failure mode presented in this chapter is a detection system failure resulting in no signal from the detection system to the control box. The top event of this Fault Tree is also the transfer event, T5, for two Fault Trees given in subsequent sections of this chapter. Zone 1 corresponds to a non-public area fitted with smoke detectors and Zone 2 corresponds to a public area fitted with heat detectors. Here, it is assumed that the fire occurs in either Zone 1 or Zone 2 of the station.

The Fault Tree, for a lack of detection signal to the control box, is given in Figure 5.3. Here Analysis of this Fault Tree gives two second order minimal cut sets and four third order minimal cut sets.

Second order minimal cut sets:

1. {ZNfd1,WIRtf1} – Corresponding to a fire detectable in Zone 1 and a Zone 1 wiring failure
2. {ZNfd2,WIRtf2} – Corresponding to a fire detectable in Zone 2 and a Zone 2 wiring failure

Third order minimal cut sets:

1. {ZNfd1,SKDdf,CPcf1}- Corresponding to a fire detectable in Zone 1, an automatic smoke detection failure and a manual call point failure in Zone 1
2. {ZNfd1,SKDdf,CPof1) – Corresponding to a fire detectable in Zone 1, an automatic smoke detection failure and a human error by a staff member to operate a manual call point
3. {ZNfd2,HTDdf,CPcf2} – Corresponding to a fire detectable in Zone 2, an automatic heat detection failure and a call point failure in Zone 2
4. {ZNfd2,HTDdf,CPof2} – Corresponding to a fire detectable in Zone 2, an automatic heat detection failure and a failure of a member of public or staff to operate the manual call point in Zone 2

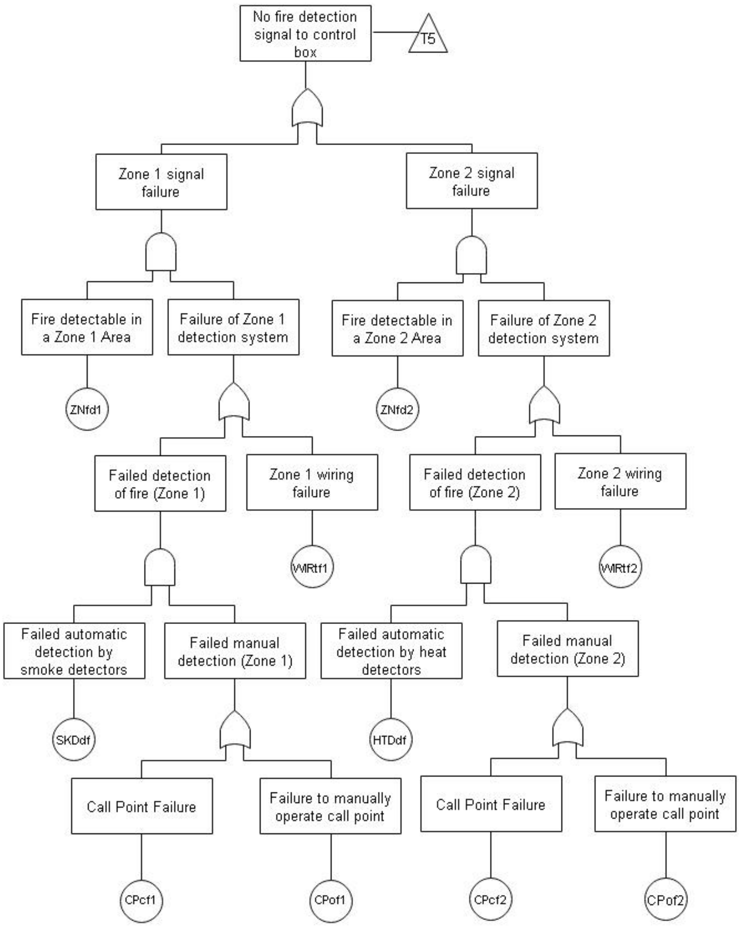

No fire detection signal to control box
T5
Zone 1 signal failure
Zone 2 signal failure
Fire detectable in a Zone 1 Area
Failure of Zone 1 detection system
Fire detectable in a Zone 2 Area
Failure of Zone 2 detection system
ZNfd1
ZNfd2
Failed detection of fire (Zone 1)
Zone 1 wiring failure
Failed detection of fire (Zone 2)
Zone 2 wiring failure
WIRtf1
WIRtf2
Failed automatic detection by smoke detectors
Failed manual detection (Zone 1)
Failed automatic detection by heat detectors
Failed manual detection (Zone 2)
SKDdf
HTDdf
Call Point Failure
Failure to manually operate call point
Call Point Failure
Failure to manually operate call point
CPcf1
CPof1
CPcf2
CPof2

Figure 5.3: A Fault Tree for no fire detection signal to the control box

5.5.2: No sound made by alarm system

The second failure mode of the system that is analysed via Fault Tree analysis is that no sound is made by the alarm system. The Fault Tree for this is given in Figure 5.4. The transfer event, T5, developed in the previous section, is included in this Fault Tree model.

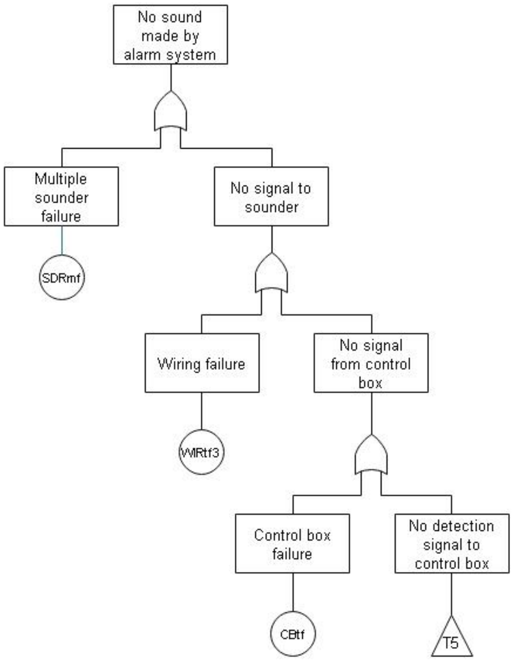


Figure 5.4: A Fault Tree for no sound made by the alarm system

Analysis for this Fault Tree, including the incorporation of the transfer event T5, gives three first order minimal cut sets, two second order minimal cut sets and three third order minimal cut sets.

First order minimal cut sets:

1. {SDRmf} - Corresponding to a multiple sounder failure
2. {WIRtf3} – Corresponding to a wiring failure
3. {CBtf} – Corresponding to a control box failure

Second minimal cut sets:

1. {ZNfd1,WIRtf1} – Corresponding to a fire detectable in Zone 1 and a Zone 1 wiring failure
2. {ZNfd2,WIRtf2} – Corresponding to a fire detectable in Zone 2 and a Zone 2 wiring failure

Third order minimal cut sets:

1. {ZNfd1,SKDdf,CPcf1}- Corresponding to a fire detectable in Zone 1, an automatic smoke detection failure and a manual call point failure in Zone 1
2. {ZNfd1,SKDdf,CPof1) – Corresponding to a fire detectable in Zone 1, an automatic smoke detection failure and a human error by a staff member to operate a manual call point
3. {ZNfd2,HTDdf,CPcf2} – Corresponding to a fire detectable in Zone 2, an automatic heat detection failure and a call point failure in Zone 2
4. {ZNfd2,HTDdf,CPcf2} – Corresponding to a fire detectable in Zone 2, an automatic heat detection failure and and a human error by a staff member to operate a manual call point

5.5.3: Insufficient flow at the nozzle heads of the deluge system

The Fault Tree for insufficent flow at the nozzle heads of the deluge system is given in Figure 5.5. The Fault Tree can be analysed to give the minimal cut sets that contribute to this failure mode. There are several further transfer events in this model. These are presented in the following figures, prior to a summary of the minimal cut sets for this top event.

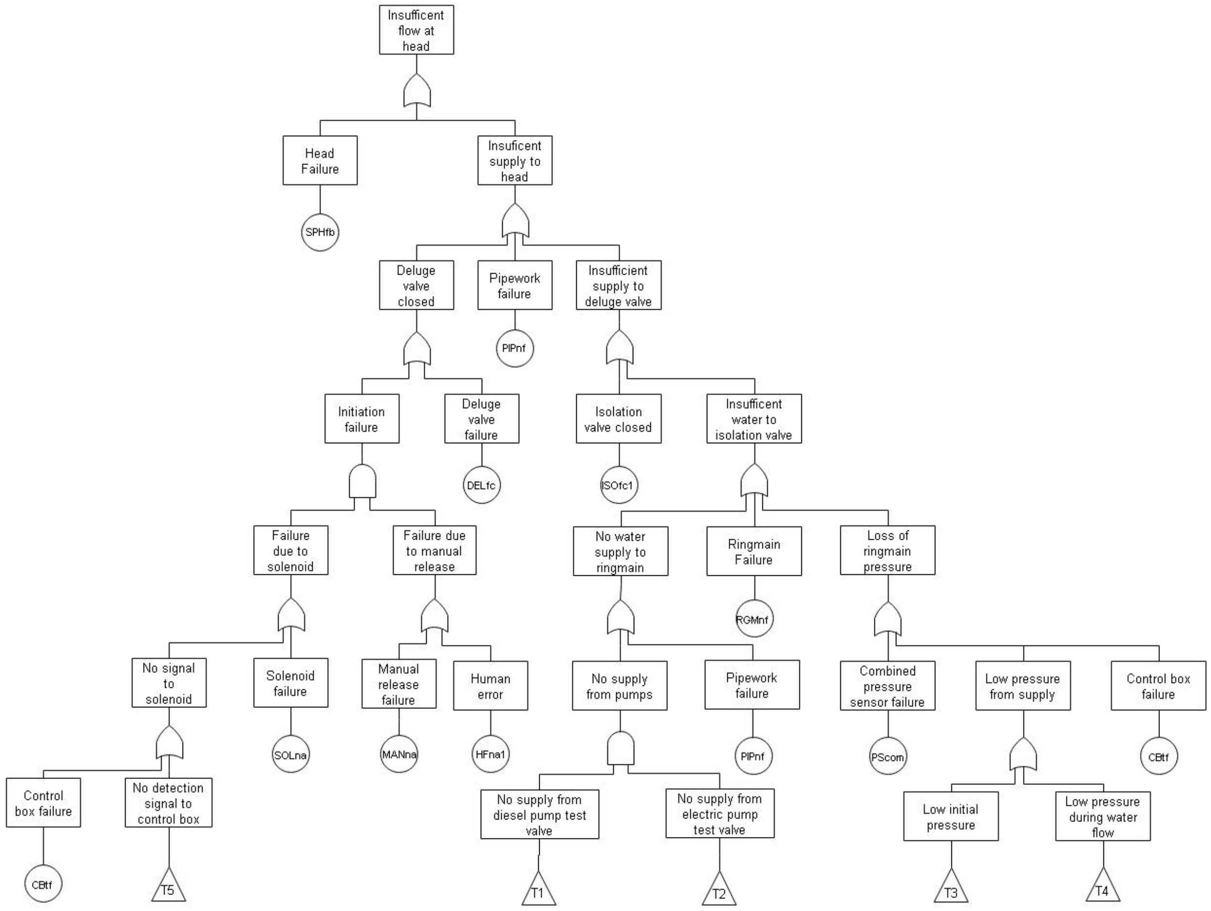


Figure 5.5: A Fault Tree for insufficient flow at the nozzle heads of the deluge system

The Fault Tree for Transfer Event T1, representing no supply from the diesel pump test valve, is given in Figure 5.6.

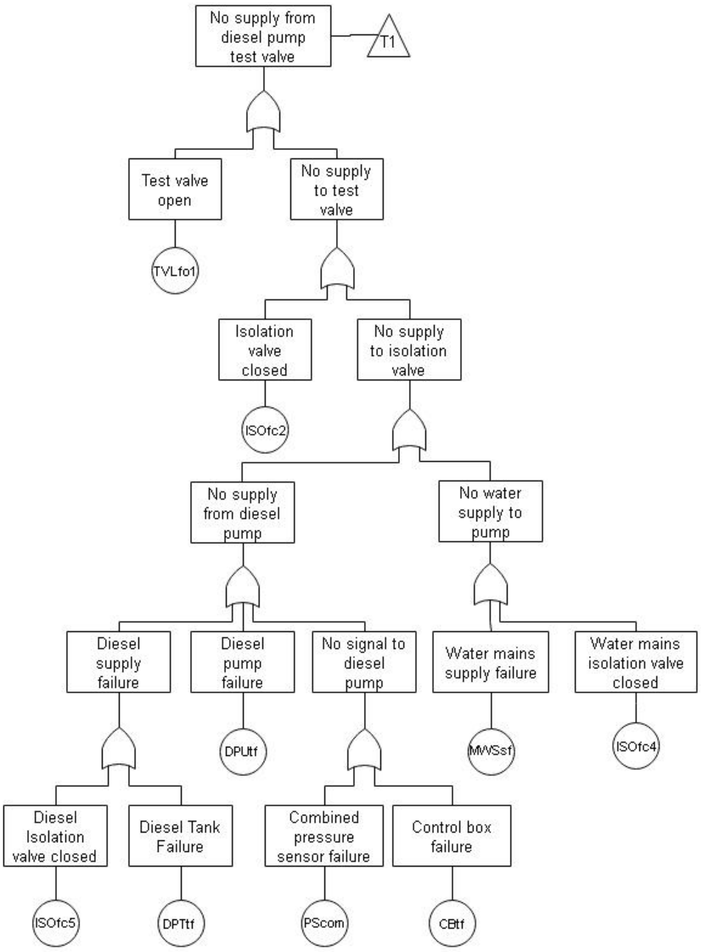


Figure 5.6: A Fault Tree for no water supply from the diesel pump test valve

The Fault Tree for Transfer Event T2, representing no supply from the electric pump test valve is given in Figure 5.7.

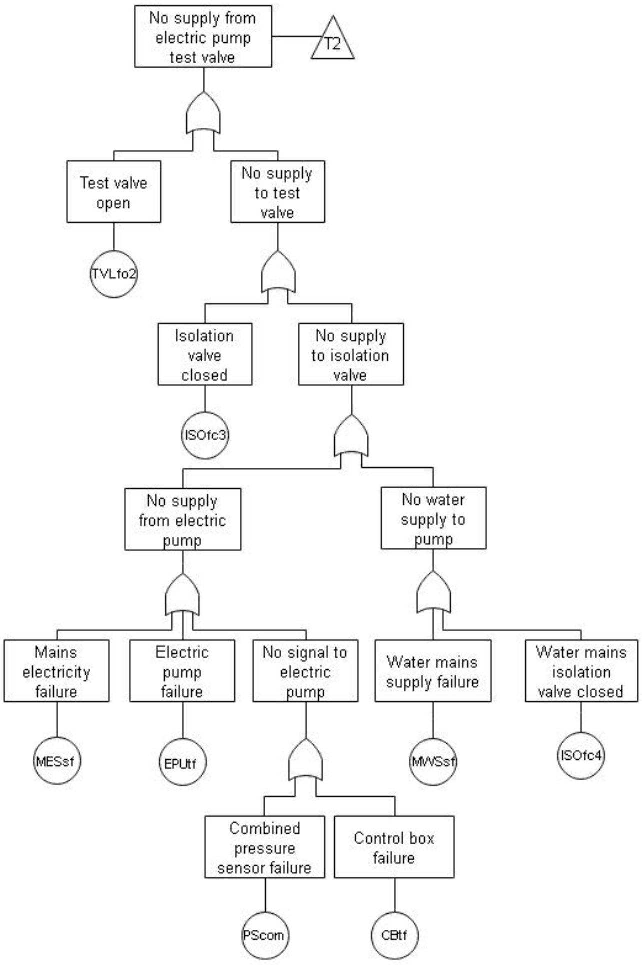

No supply from electric pump test valve
T2
Test valve open
No supply to test valve
TVLfo2
Isolation valve closed
No supply to isolation valve
ISOfc3
No supply from electric pump
No water supply to pump
Mains electricity failure
Electric pump failure
No signal to electric pump
Water mains supply failure
Water mains isolation valve closed
MESsf
EPUtf
MWSsf
ISOfc4
Combined pressure sensor failure
Control box failure
PScom
CBtf

Figure 5.7: A Fault Tree for no water supply from the electric pump test valve

The Fault Tree for Transfer event T3, which represents low initial pressure of the ringmain, is given in Figure 5.8.

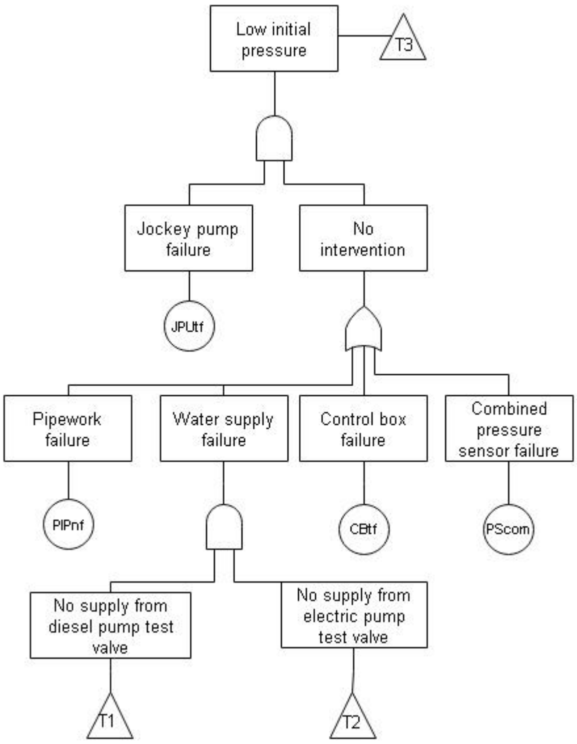
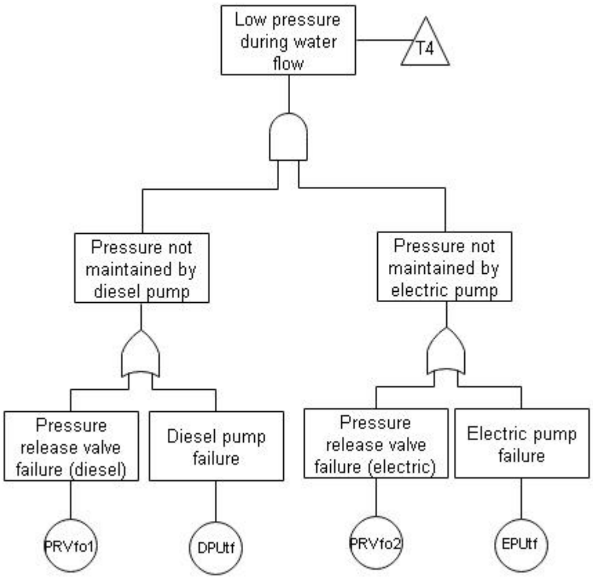


Figure 5.8: A Fault Tree for low initial pressure in the ringmain

The Fault Tree for Transfer event T4, which represents low pressure during water flow, is given in Figure 5.9.



Low pressure during water flow
T4
Pressure not maintained by diesel pump
Pressure not maintained by electric pump
Pressure release valve failure (diesel)
Diesel pump failure
Pressure release valve failure (electric)
Electric pump failure
PRVfo1
DPUtf
PRVfo2
EPUtf

Figure 5.9: A Fault Tree for low pressure during water flow

Following the analysis of the detection system failure mode, the minimal cut sets for the full expanded Fault Tree for the failure of water flow at the nozzle head of the deluge system were found. There are nine first order minimal cut sets for the system, twenty-five second order minimal cut sets for the system, four third order minimal cut sets for the system and four fourth order minimal cut sets for the system. Each minimal cut set, along with a description is given below.

First order minimal cut sets:

1. {SPHfb} - Corresponding to a nozzle failure
2. {PIPnf} - Corresponding to a pipework failure
3. {DELfc} - Corresponding to a deluge valve failure
4. {ISOfc1} - Corresponding to an isolation valve failure in the closed position
5. {RGMnf} - Corresponding to a ringmain failure
6. {PScom} - Corresponding to a combined pressure sensor failure
7. {CBtf} – Corresponding to a control box failure
8. {MWSsf}- Corresponding to a water supply failure from the mains
9. {ISOfc4} – Corresponding to a failure of the water mains isolation valve in the closed position

Second order minimal cut sets

1. {SOLna,MAnna} – Corresponding to a solenoid failure and a manual release mechanism failure
2. {SOLna,HFna1} – Corresponding to a solenoid failure and human error in operating the manual release mechanism.
3. {TVLfo1,TVlfo2}- Corresponding to an open testvalve for the diesel pump and an open test valve for the electric pump.
4. {TVLfo1,ISOfc3}- Corresponding to an open testvalve for the diesel pump and a failure of the electric pump isolation valve in the closed position
5. {TVLfo1,EUPtf}- Corresponding to an open testvalve for the diesel pump and an electric pump failure
6. {TVLfo1,MESsf}- Corresponding to an open testvalve for the diesel pump and a mains electricity failure
7. {TVLfo2,ISOfc2}- Corresponding to an open testvalve for the electric pump and a failure of the diesel pump isolation valve in the closed position
8. {TVLfo2,ISOfc5}- Corresponding to an open testvalve for the electric pump and a failure of the diesel tank isolation valve in the closed position
9. {TVLfo2,DPUtf}- Corresponding to an open testvalve for the electric pump and a diesel pump failure
10. {TVLfo2,DPTtf}- Corresponding to an open testvalve for the electric pump a diesel tank failure
11. {ISOfc2,ISOfc3}- Corresponding to a failure of the diesel pump isolation valve in the closed position and a failure of the electric pump isolation valve in the closed position
12. {ISOfc2,EPUtf}- Corresponding to a failure of the diesel pump isolation valve in the closed position and an electric pump failure
13. {ISOfc2,MESsf}- Corresponding to a failure of the diesel pump isolation valve in the closed position and a mains electricity failure
14. {ISOfc5,ISOfc3}- Corresponding to a failure of the diesel tank isolation valve in the closed position and a failure of the electric pump isolation valve in the closed position
15. {ISOfc5,EPUtf}- Corresponding to a failure of the diesel tank isolation valve in the closed position and an electric pump failure

16. {ISOfc5,MESsf}- Corresponding to a failure of the diesel tank isolation valve in the closed position and a mains electricity failure
17. {DPUtf,ISOfc3}- Corresponding to a diesel pump failure and a failure of the electric pump isolation valve in the closed position
18. {DPUtf, EPUtf}- Corresponding to a diesel pump failure and an electric pump failure
19. {DPUtf,MESsf}- Corresponding to a diesel pump failure and a mains electricity failure
20. {DPTtf,ISOfc3}- Corresponding to a diesel pump tank failure and a failure of the electric pump isolation valve in the closed position
21. {DPTtf,EPUtf}- Corresponding to a diesel pump tank failure and an electric pump failure
22. {DPTtf,MESsf}- Corresponding to a diesel pump tank failure and a mains electricity failure
23. {PRVfo1,PRVfo2}- Corresponding to a diesel pump pressure release valve failure and an electric pump pressure release valve failure.
24. {DPUtf,PRVfo2}- Corresponding to a diesel pump failure and an electric pump pressure release valve failure.
25. {EPUtf,PRVfo1}- Corresponding to an electric pump failure and a diesel pump pressure release valve failure

Third order minimal cut sets

1. {ZNfd1,WIRtf1,MAnna} – Corresponding to a fire detectable in Zone 1, a Zone 1 wiring failure and a manual release mechanism failure.
2. {ZNfd2,WIRtf2,MAnna} – Corresponding to a fire detectable in Zone 2, a Zone 2 wiring failure and a manual release mechanism failure
3. {ZNfd1,WIRtf1,HFna1} – Corresponding to a fire detectable in Zone 1, a Zone 1 wiring failure and a human error in operation of the manual release mechanism
4. {ZNfd2,WIRtf2,HFna1} – Corresponding to a fire detectable in Zone 2, a Zone 2 wiring failure and a human error in operation of the manual release mechanism

Fourth order minimal cut sets

1. {ZNfd1,SKDdf,CPcf1,MAnna}- Corresponding to a fire detectable in Zone 1, an automatic smoke detection failure, a manual call point failure in Zone 1 and a manual release mechanism failure
2. {ZNfd1,SKDdf,CPof1,MAnna) – Corresponding to a fire detectable in Zone 1, an automatic smoke detection failure, a human error by a staff member to operate a manual call point and a manual release mechanism failure.
3. {ZNfd1,SKDdf,CPcf1,HFna1}- Corresponding to a fire detectable in Zone 1, an automatic smoke detection failure, a manual call point failure in Zone 1 and a human error in operation of the manual release mechanism
4. {ZNfd1,SKDdf,CPof1,HFna1) – Corresponding to a fire detectable in Zone 1, an automatic smoke detection failure, a human error by a staff member to operate a manual call point and a human error in operation of the manual release mechanism

Here, control box failure includes failures due to a total power failure or control box hardware failure. This is included in the control box modelling, and considered when combining the models to get the full system failure probabilities.

5.6: Component failure and maintenance

The Fault Tree analysis identified the component failures, or combinations of component failures, that can lead to each of the failure modes considered in this chapter. A Petri net model is presented for each of the component failure modes, so that the probability that each component fails with time can be modelled. These Petri nets are simulated via Monte Carlo Simulation in order to find a quantitative estimate for the probability of each component failure.

In this model, for each component, it is assumed there are two categories of failure, this is represented by two competing transitions in each component Petri net model. The first transition represents random failures that can occur at any point in the component's life, and the second transition corresponds to an increasing number of failures as the component ages. This is completed under the assumption that after installation and the initial burn in period, each component will fail according to the shape of the Reliability Bath Tub Curve, with a constant rate of random failures throughout the components' life followed by an increase in failure rate as the component ages. In this model, it is assumed that the initial high probability of failure commonly shown in the Reliability Bath Tub Curve, often seen due to poor installation or manufacturing defects, is reduced through testing of the components on installation. The probability of failure is modelled as approximately constant while random failures are occurring, such as those due to accidental damage. As the component ages, the probability that it will fail increases due to factors such as wear and corrosion.

There are several methods whereby a failure in a component can be identified. In some cases, a component failure will be immediately revealed, such as in the case where the failure causes a false activation of the system. Alternatively, system testing or component inspection can reveal a failure. The strategies for the identification of a component failure are discussed for each component separately in the following sections. For most components it is assumed that an exact quantification of a partial failure is unreliable and so partially degraded states are not included. However, for the water pumps, it is assumed that a partially degraded state can be quantified through inspection and testing, such as monitoring the flow of water through the pump when activated. This allows condition-based maintenance of the pumps prior to failure. For all components in the model, the inspection interval is periodic. It is also assumed that the system level testing interval is periodic.

There are several maintenance options for each component included in the model. Firstly, if a component failure is identified then a maintenance action is completed on this component after a short scheduling delay. It is assumed that this returns the component to the 'as good as new' state. Secondly, if there is an identified partially degraded state of the component, such as that included in the pump model, then the maintenance of the component is scheduled as a priority. Again, it is assumed that the maintenance returns the component to the 'as good as new' state. There is also age-based maintenance included in the component models. Each component can be assigned two age-based maintenance intervals: one representing early age-based maintenance and one representing routine age-based maintenance. These intervals can be assigned based on historic maintenance records, engineering judgement or estimated from predictions of the behaviour of each component over its lifetime. Alternatively, the method given in Chapter 6 can be extended to optimise the intervals. It is assumed that these actions return the component to the 'as good as new' state. In this case, the 'age' of the component is determined from the time since the most recent maintenance action.

A phased system level maintenance strategy is also included in this model. Here, maintenance phases are defined for the system, based on the age of the whole system since installation. The frequency of the component inspection or system testing can then be varied based on the maintenance phase. The activation of an individual early age-based, or routine age-based, maintenance strategy for each component can also be governed by the system level maintenance phase.

The model developed here can be populated with input data, found in the field to adapt the model to a specific fire protection system. The model provides a framework for analysis of the system given this data.

For application of this model distributions are required for:

1. The random failure rate of each component throughout its life
2. The failure rate of each component as it ages

The strategy for the management of the system also requires the following inputs:

1. The system level maintenance phase entry times
2. The inspection frequency for each component within each maintenance phase
3. The system testing frequency within each maintenance phase
4. The early age-based maintenance interval for each component
5. The routine age-based maintenance interval for each component

In addition to this, in some of the component models, there is a probability associated with different component failure modes, for instance if the failure is immediately revealed or not. These should be evaluated when applying the model to a specific system. These values are also required as input to the models. Input values to the model can be adjusted via an Excel spreadsheet that interfaces with the custom made model simulation software.

For a demonstration of the modelling capability, sample data values are used to give example outputs of the models. Here, a uniform distribution is used for each component to estimate the time to a random component failure and a 2-Parameter Weibull distribution is used to estimate the time to an aged based failure of each component. The sample values used can be found in Appendix 3, a table of the failure modes and asset management strategies of each component modelled here can be found in Appendix 4. Results for each of the component models for these sample values can be found in Appendix 5.

5.7: Using a Petri net to model component failures

This section presents the models for each of the component failures that contribute to the system failure modes identified in the previous section. The aim of these models is to give the probability of each of the component failure modes, for input to the Fault Trees for the system failure modes, and to model how these probabilities change with time.

For demonstration of the model, sample model inputs are used, and results are given for an arbitrary phased system-level maintenance strategy, with three system level maintenance phases. For this sample application, initially the strategy is in the first phase, whereby components are repaired only on the discovery of a failure. Following a 36-month interval, the strategy enters the second phase and routine age-based maintenance for each component is enabled. In this phase, components are also maintained when they are in a revealed failed state, or a failed state has been discovered through inspection or testing. After 156 months from the point of installation, the maintenance strategy enters the third phase. Here, early age-based maintenance of each component is enabled as is maintenance when they are in a revealed failed state, or a failed state has been detected through inspection or testing. Hence, the older the system, the earlier that the preventative maintenance is completed for each component. Different phased strategies can be easily tested by altering the input data to the model.

The component inspection frequency and system level testing frequency can also be varied depending on the maintenance phase. For demonstration of the model in this chapter, a sample inspection and testing strategy is included. Here, in the first system level phase, every component is inspected once every 12 months. In the section system level phase, every component is inspected once every 6 months. In the third system level phase every component is inspected once every 3 months. In this model each component can have an individual inspection frequency for every system phase, this is discussed further and optimised in Chapter 6. The system testing in this model can also vary with each phase. For demonstration in this section, sample intervals are assigned with one system level test every 9 months in the first phase, one system level test every 6 months in the second phase and one system level test every 3 months in the third phase. To simplify the modelling it is assumed that the test valve will only reside in the open state if there is a system test underway, the opening of both test valves is modelled together by the system testing action.

In this model each component type is modelled separately. In some cases, the model for the specific component is unique and in other cases the same Petri net structure can be applied to several different component types. There are three component types with a unique Petri net structure in this model, these are: the control box, the ringmain pressure sensors and the alarm sounder circuit. There are also 6 component model structures that are applied to multiple components cases in this model. These arise due to the similarities between failure modes and the specific inspection and testing strategy applied to each component. An overview of the assumptions made by each component model structure, and a description of each model structure, is given in the following sections. Specific component data must be input to these model structures when they are repeated for each component in the full model.

The results given in Appendix 5 for each component model, are a sample application with the synthetic data values given in Appendix 3. The computational time for Monte Carlo Simulation of the full system model was 77807.315s for 2000 runs of the simulation.

5.7.1: Control box failure

The Petri net shown in Figure 5.10 models the condition and maintenance of the control box. The control box and the control box battery are modelled in this structure, along with a mains power failure. The control box is powered by a mains power source with a battery in back up [161]. The control box can fail with two competing mechanisms: one due to the age of the control box and one due to random failure occurrence. Here, a competing mechanism implies that one mechanism will occur first. There are two failure modes for the control box, one revealed by an internal alarm and one unrevealed. The control box battery can also fail with two competing mechanisms: one due to the age of the control box battery and one due to random failure occurrence. A control box battery failure is assumed to be unrevealed. A mains power failure is assumed to occur with a uniform probability throughout the lifecycle of the system, and last for a short duration. This power failure does not only impact the control box but also other areas of the protection system such as the electric pump, if it is operational at the time. A final failure mode is included in the model to represent the probability that there is a mains power failure and the backup battery fails. In addition, the failure mode for this mains power failure, impacting other areas of the system, is taken from this model.

A failure in the mains power is immediately revealed, in this model it is assumed that this failure lasts for a short delay time. An unrevealed failure of the control box can be identified through inspection of the control box or testing of the system by opening the test valve. A failure in the control box battery can be revealed by inspection of the control box battery.

Maintenance of the control box battery is completed on a revealed battery failure. This returns the battery to the ‘as good as new’ state. Maintenance of the control box is completed on a revealed failure, this returns the control box and the control box battery to the ‘as good as new’ state.

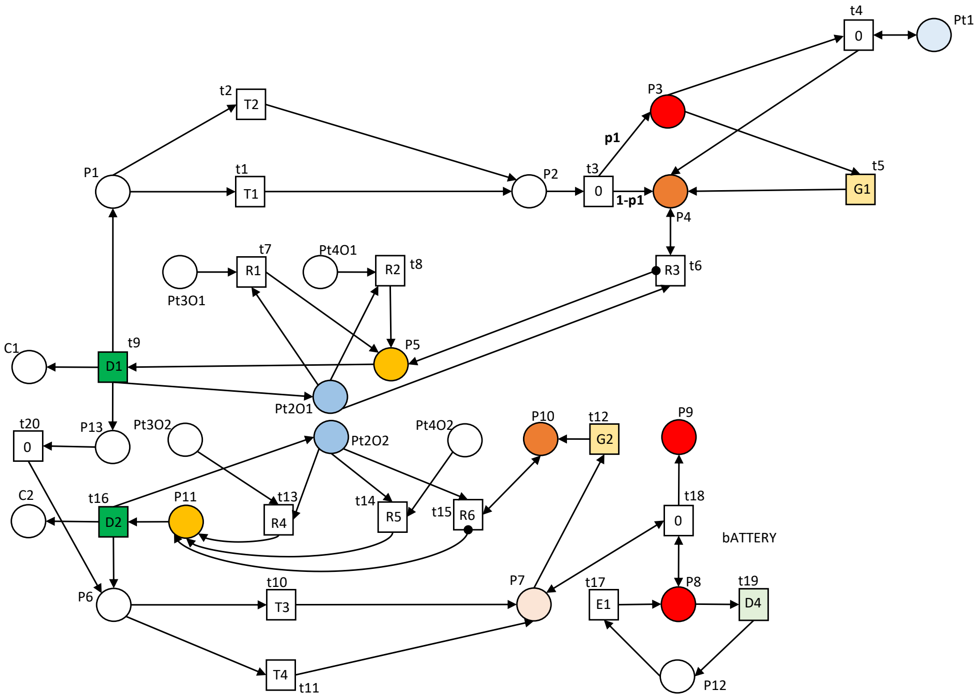


Figure 5.10: A Petri net for the control box and power failure

In this Petri net, place P6 corresponds to a working state of the control box battery. Transition t11 represents random battery failure, for example through damage to the battery, false installation, evaporation of the electrolyte due to high temperatures or thermal runaway due to excess charging current. Transitions t10 represents battery failure due to the age of the battery, such as chemical decomposition of the electrolyte, oxidation of the electrolyte or corrosion of the electrodes. Place P7 corresponds to a failure of the control box battery.

Place P8 corresponds to a mains electricity failure, which occurs at random governed by transition t17. The electricity failure ends after a delay governed by transition t19. P9 corresponds to a combined mains power failure and a battery failure. Transition t12 is a global inspection transition and models the inspection of the control box battery to reveal a failure. The inspection interval for the control box battery can be defined for the component and can vary depending on the system maintenance phase. Place P10 corresponds to a revealed failure of the control box battery.

When place Pt2O2 is marked, maintenance is possible for the control box battery. Transition t15 represents maintenance scheduling when there is a revealed failure of the control box battery. When place Pt4O2 is marked, routine age-based maintenance of the control box battery is possible. Transition t14 corresponds to scheduling of this maintenance. The time until this maintenance is scheduled is governed by the component maintenance strategy and counted from the time since the most recent maintenance intervention. When place Pt3O2 is marked, early age-based maintenance of the control box battery is possible. Transition t13 corresponds to scheduling of this maintenance. Similarly, the time until this maintenance is scheduled is governed by the component maintenance strategy and counted from the time since the most recent maintenance intervention. The marking of places Pt3O2 and Pt4O2, which enable the age-based maintenance of the control box battery, can occur at different time, governed by a system level phased maintenance strategy. Transition t16 is a reset transition that corresponds to maintenance of the control box battery, it is assumed in this model that this returns the control box battery to the ‘as good as new’ state. Place C2 counts the number of maintenance actions on the control box battery.

Place P1 corresponds to a working state of the control box. Transitions t1 represents ageing of the control box such as loose wiring or degradation of the electrical components. Transition t2 represents random failure of the control box, for instance due to water ingress or accidental damage. Place P2 corresponds to a failed state of the control box. On failure of the control box, there is a probability that the failed state will be revealed via an internal control box alarm. This is represented by the probability transition t4, with place P4 corresponding to a revealed control box failure and place P3 corresponding to an unrevealed control box failure.

When place Pt1 is marked the system is under test, by opening the test valve. This can reveal a failure in the control box and transition t4 corresponds to this. The marking of place Pt1, and hence the frequency of system level testing, can vary depending on the system level maintenance phase, as was discussed in Section 5.6. Transition t5 is a global inspection transition that models the periodic inspection of the control box, which can reveal a failure. The inspection interval governing this transition can be assigned based on the individual component asset management strategy and can vary depending on the system level maintenance phase.

Maintenance is possible for the control box when place Pt2O1 is marked. Transition t6 corresponds to maintenance scheduling for the control box on a revealed failure. When place Pt3O4 is marked, routine age-based maintenance of the control box is enabled. Transition t8 corresponds to scheduling of this. The time at which this maintenance is scheduled is dependent on the component maintenance strategy and is counted from the time since the most recent maintenance action on the control box. When place Pt3O3 is marked, early age-based maintenance of the control box is enabled. Transition t7 corresponds to scheduling of this. Again, the time at which this maintenance is scheduled is dependent on the component maintenance strategy and is counted from the time since the most recent maintenance action on the control box. Place P5 corresponds to a scheduled maintenance action of the control box and transition t9 is a reset transition that models the maintenance of the control box. It is assumed that all maintenance actions on the control box return the control box and the control box battery to the ‘as good as new’ state. The marking of places Pt3O3 and Pt3O4 can occur at different times and be governed by the system level maintenance phase.

Distributions are required for the probability that there is an age-based failure for the control box and the control box battery over time, and the rate of randomly occurring failures. These can be gathered from failure data for the components. The probability that the control box failure is revealed by an internal alarm is also required for application of the model. In addition, the rate of mains power failure and rate of repair to the mains power are required as input to the model. For testing of different asset management strategies; the scheduling delays for the age-based maintenance and the maintenance on revealed failure can be assigned, either based on current maintenance strategies or a test case. The component inspection and system level testing frequencies can also be assigned in this way.

Initially places P1, P6 and P2 are marked by tokens. The Petri net can then be simulated. The average marking of places P3 and P4 gives the probability that the control box is in an unrevealed or revealed failed state at each time. The average marking of places P7 and P10 gives the probability that the control box battery is in an unrevealed or revealed failed state at each time. The average marking of place P8 gives the probability that there is a mains power failure at each time and the average marking of P9 gives the probability that there is a combined power failure to the control box. The number of tokens in place C1 at each time represents the total number of combined control box and control box battery maintenance actions that have been completed. The number of tokens in place C2 at each time represents the total number of control box battery maintenance actions that have been completed.

5.7.2: Pressure sensor failure

For the system modelled in this chapter, there are three sensors whose failure logic follows that of the 2/3 voting gate in a Fault Tree. Hence, if there is one reading that is different but two readings that match, the system will follow the reading of the two sensors.

Three pressure sensors are modelled with the assumption that if any two of the sensors fail, then there will be a false reading of the ringmain pressure. Each sensor can fail with two competing mechanisms: one due to the age of the component and one due to random failure occurrence. Each sensor has three failure modes, the first where the sensor gives a reading that is higher than true, the second where the sensor gives a reading that is lower than true and the third where the sensor gives no reading.

Inspection of the sensor readings and system testing by opening the test valve can reveal various combinations of failures of the pressure sensors. Firstly, the failure mode where one or more of the sensors gives no reading, is revealed on inspection of the readings. If two or more of the sensors fail to give a reading, then the deluge system can fail. Secondly, if there is a difference in the readings obtained from the sensors, this failure is revealed on inspection of the readings. This failure mode can arise if one, or two, of the sensors have failed and are providing false readings. Thirdly, if two, or more, of the sensors give a reading that is lower than true, this can activate the pumps to increase the pressure in the ringmain. This can cause a false activation of the system by creating a pressure difference across the deluge valve. The failure mode is revealed in this case. Fourthly, if two, or more, of the sensors give a reading that is higher than true, this can lead to insufficient pressure in the ringmain, with the potential to cause a system failure. In the case that all three sensors give a reading that is higher than true, the failure is unrevealed and not identifiable by inspection of the sensor readings. Testing of the system by opening the test valve can reveal this failure mode.

In this model, the sensors are maintained on a revealed or discovered failure in any of the sensors. The sensors can also be maintained based on the time since the last maintenance intervention. Maintenance returns all sensors to the ‘as good as new’ state.

The Petri net for the combined pressure sensor condition is given in Figure 5.11.

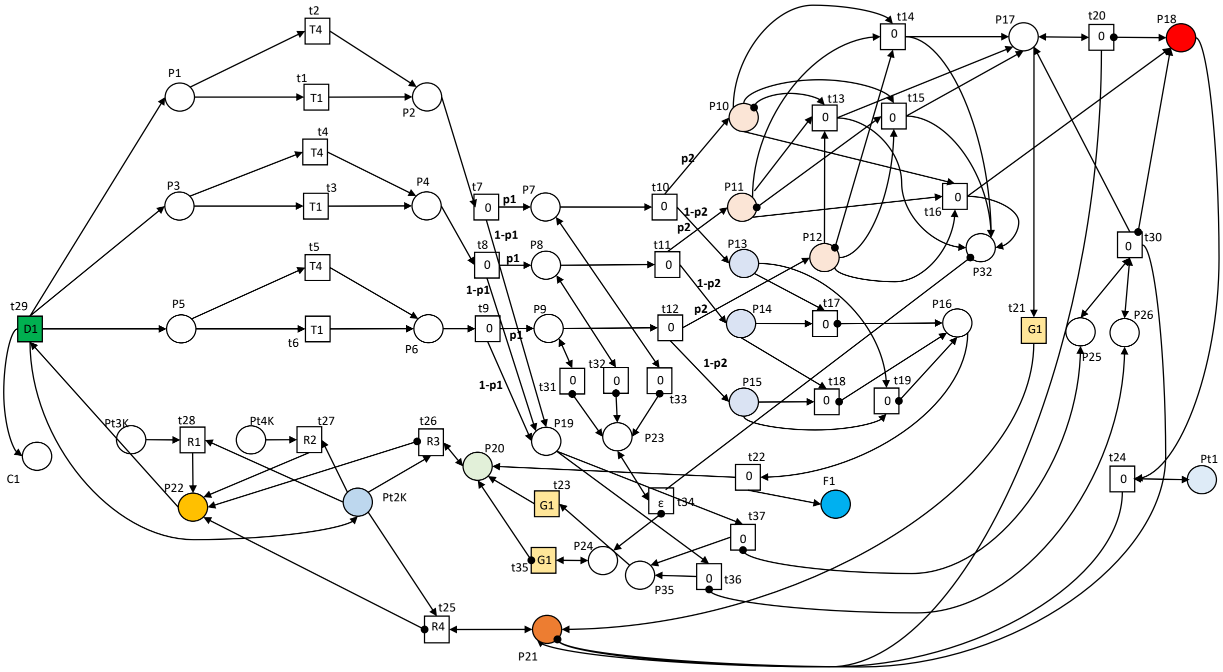


Figure 5.11: A Petri net model for the combined condition of the pressure sensors

In this model places P1 corresponds to the working state of the first sensor, place P3 corresponds to the working state of the second sensor and place P5 corresponds to the working state of the third sensor. Transitions t1, t3 and t5 govern the failure of each of the sensors due to their age and transitions t2, t4 and t6 govern the rate of random failures of each of the sensors. Place P2 corresponds to the failed state of the first sensor, place P4 corresponds to the failed state of the second sensor and place P6 corresponds to the failed state of the third sensor. There is a probability included in the model that the failure will be of the type where no reading is gained from the sensor. Transitions t7, t8 and t9 model this for each sensor in turn, with Place P19 corresponding to a failed state where one or more of the sensors is not giving a reading. Place P7, P8 and P9 represent a false reading from each sensor in turn. For each sensor this false reading can either be higher than true, or lower than true. This is represented in the model by probability transitions t10, t11 and t12 in turn. The places P10, P11 and P12 corresponds to a failure mode of each sensor in turn where the reading is higher than the true value. The places P13, P14 and P15 corresponds to a failure mode of each sensor in turn where the reading is lower than the true value.

Place P16 corresponds to a failure mode where two or more of the sensors give a reading that is lower than the true value, this is assumed to be a failure mode that is revealed by false activation of the deluge system. Place F1 counts the number of false system activations by any component in the system. Place P17 corresponds to a failure mode where two of the sensors are giving a reading that is higher than true, this can cause a failure of the system but is revealed on inspection of the sensor readings. Place P18 corresponds to a failure mode of the system cause by two or more of the sensors giving a reading that is higher than true. This can also be revealed by testing of the system by opening the test valve. Place P24 corresponds to a failure where one sensor gives a false reading, this is revealed by inspection of the sensor readings. Place P19 corresponds to a failure where one or more of the sensors to fail to give a reading, if two of the sensors fail to give a reading then there can be a system failure. This is represented by transition t30. Place P18 corresponds to a failure that has the potential to cause a system failure. Place F1 corresponds to a false activation of the deluge system.

Transitions t21, t32 and t35 model the inspection of the readings from the pressure sensors. These are global transitions where the frequency of inspection can be defined for the pressure sensors and can vary depending on the system level maintenance phase. Transition t24 models whole system testing by opening the test valve, when place Pt1 is marked this testing is underway. The marking of place Pt1, and hence the frequency of system testing, can vary with the system maintenance phase.

In this model maintenance is scheduled for the pressure sensors if there is a discovered failure in any of the sensors. Place P21 corresponds to a revealed or discovered failure that has the potential to cause a system failure. Place P20 corresponds to a revealed or discovered failure that does not have the potential to cause a system failure. The scheduling of maintenance in either case is modelled by transitions t25 and t26, for each of the previous failure modes. The pressure sensors can also be maintained based on their age following an interval since their last maintenance intervention. When place PtK4 is marked then routine age-based replacement is enabled for the pressure sensors, this is scheduled following a delay governed by transition t27. When place PtK3 is marked then early age-based replacement is enabled for the pressure sensors, this is scheduled following a delay governed by transition t28. Transition t29 is a reset transition that models the maintenance of the pressure sensors. It is assumed that all pressure sensors are maintained at the same time, returning all the sensors to the ‘as good as new’ state. Place C1 counts the number of maintenance interventions.

Data is required for the distribution governing the expected failure times of a pressure sensor due to age and due to random failure occurrences. The probability that a pressure sensor will fail in a way such that there is no reading is also required. In addition, the probability that a pressure sensor will give a false reading that is higher than true, or conversely lower than true, is also required as input to the model. To apply a specific asset management strategy an estimate is required for the inspection frequency of the pressure sensor readings, and the system testing frequency. In addition, the time until an early or routine age-based maintenance action, or a distribution representing these times, can be included based on historic data or a specific test scenario. The maintenance scheduling delay for a revealed or discovered failure and the time taken for maintenance to be completed can also be included for a specific case.

The initial marking of the Petri net is set with places P1, P3, P5 and Pt2K marked by tokens. The model can then be simulated, subject to the system level strategy in place. The average marking of the places P18 and P21 can be extracted to give the probability that there is a combined pressure sensor failure at each time from installation of the system. The number of tokens in place C1 can be analysed to give the number of maintenance actions on the pressure sensors at each time. The number of tokens in place F1 can be analysed to give the number of false system activations at each time.

5.7.3: Alarm failure

The Petri net in Figure 5.12 gives the model for the alarm sounder circuit. A circuit with a population of alarm sounders is modelled. The alarm sounders can fail with two competing mechanisms: one due to the age of the sounders and one due to random failure occurrence. A failure in the alarm sounders can have one of two modes. In the first failure mode, the failure is insufficient to cause a system failure due to an inbuilt redundancy in the number of sounders in the circuit. In the second failure mode a failure in the alarm sounders is sufficient to cause a system failure. Both failure modes are unrevealed. The circuit connecting the alarm sounders can also fail with two competing mechanisms: one due to the age of the circuit and one due to random failure occurrence. This failure is unrevealed and assumed to be sufficient to cause a system failure.

Inspection of the alarm circuit can reveal a failure. Included in this inspection is testing of the circuit to ensure that the alarm sounds correctly. The alarm sounder circuit is maintained on a discovered failure. The alarm sounder circuit can also be maintained based on the time since the last maintenance intervention. Maintenance returns all alarm sounders and the connecting circuit to the good as new condition.

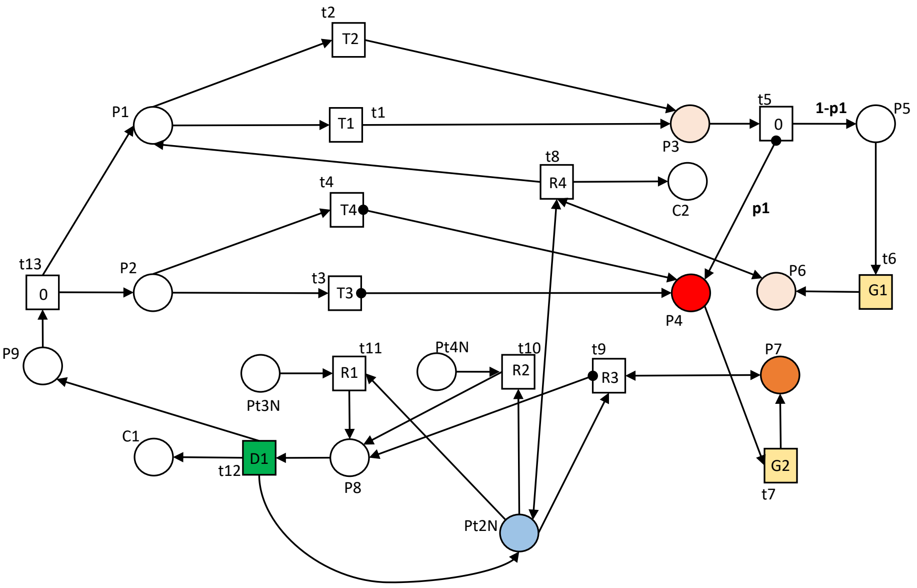


Figure 5.12: A Petri net for the alarm sounder circuit

In this model place P1 corresponds to a good condition of the population of alarm sounders and place P2 corresponds to a good condition of their connecting circuit. Failure of the alarm sounders, due to age, is governed by transition t1, for example degradation of the piezo element, a loose wiring connection or a fault due to an accumulation of dust or water inside the alarm sounder. Random failure of the alarm sounders, such as that due to accidental damage is governed by transition t2.

Failure of the alarm sounder connecting circuit, due to age, is governed by transition t3, this includes degradation of the cable sheath resulting in exposure of the cables and a potential short circuit, moisture ingress into the cable and corrosion of the cable. Random failure of the alarm sounder connecting circuit, such as that due to accidental damage is governed by transition t4, for example failures due to accidental mechanical damage to the wiring or damage caused by rodents. Place P3 corresponds to a failure in the population of alarm sounders. Transition t5 is a probability transition that governs the likelihood of such a failure causing a system level failure, due to several alarm sounders present in the circuit resulting in inbuilt redundancy in the system. Place P5 corresponds to a partial alarm sounder failure that does not have the potential to cause a system level failure. Place P4 corresponds to a failure that does have the potential to cause a system level failure, either due to the condition of the alarm sounders or due to a failure in the connecting circuit.

Inspection of the alarm sounder circuit, including testing of the circuit, can reveal a total failed state in the sounder circuit, or a partial failed state in the sounder circuit. Transition t6 is a global inspection transition that models the periodic inspection of the alarm sounder circuit to reveal a partially failed state. Place P6 corresponds to a discovered partially failed state. Transition t7 is a global inspection transition that models the periodic inspection of the alarm sounder circuit to reveal a total failed state. Place P7 corresponds to a discovered total failed state of the system.

When place Pt2N is marked, maintenance is possible for the alarm sounder circuit. The alarm sounder circuit is maintained on a discovered total, or partial, failure. Transition t8 corresponds to maintenance of the alarm sounders on a discovered partial failure. This returns the alarm sounders to the ‘as good as new’ state but does not return the connecting circuit to the ‘as good as new’ state. Transition t9 models the scheduling of a maintenance action on a revealed total failure in either the population of alarm sounders or the connecting circuit. This maintenance action returns the alarm sounders and the connecting circuit to the ‘as good as new’ state. The alarm sounder circuit can also be maintained based on their age, governed by the time interval since the previous maintenance action. When place Pt4N is marked, routine age-based maintenance of the alarm sounder circuit is enabled. Transition t10 governs the scheduling of this maintenance, from the time since the previous maintenance action. When place Pt3N is marked, routine age-based maintenance of the alarm sounder circuit is enabled. Transition t11 governs the scheduling of this maintenance, from the time since the previous maintenance action. Transition t12 is a reset transition, and combined with transition t13, models the maintenance of the alarm sounder circuit to return the population of alarm sounders and the connecting circuit to the ‘as good as new’ state. Place C1 counts the number of maintenance actions on the alarm sounder circuit.

Data is required to gain a distribution for the time to failure of the population of the alarm sounders, due to the age of the alarm sounders. A distribution is also required for the expected time until a random failure of the alarm sounders. In addition, an estimate of the probability that a failure in the alarm sounders will fail the whole alarm system is required. Data is required for the time to failure of the connecting circuit, also due to age or random failure. To test a system level asset management strategy, distributions are required for the scheduling of early-age-based maintenance, routine age-based maintenance and maintenance on a discovered failure. A distribution governing the time taken to complete the maintenance of the alarm sounder circuit can be included in the model. The frequency of inspection, including testing, of the alarm sounder circuit can also be varied.

Initially places Pt2N and P9 are marked by tokens, following the marking of place P9, places P1 and P2 are marked immediately, corresponding to the good state of the alarm sounders and the connecting circuit. The Petri net can then be simulated to give outputs of the model. The number of maintenance actions at each time can be found by extracting the average number of tokens in place C1. The probability that the alarm sounder circuit is in an unrevealed failed state that can contribute to a system failure can be found by tracking the average marking of place P4. The probability that the alarm sounder circuit is in the corresponding revealed failed state can be found by tracking the marking of Place P7.

5.7.4: Type A component failures

The components that have similar behaviours can be modelled with the same Petri net structure, which is repeated for each component, with different input values. Type A components are those that have the following features:

- The component can fail with two competing mechanisms: one due to the age of the component and one due to random failure occurrence. The failure is unrevealed.
- System testing by opening the test valve can reveal a failure in the component. Inspection of the component can reveal the failure.
- The component is maintained on a revealed or discovered failure. The component can also be maintained based on the time since the last maintenance intervention. Maintenance returns the component to the 'as good as new' state.

The model for the Type A components is given in Figure 5.13 and is repeated to model the pipework, pressurised ringmain and the diesel tank.

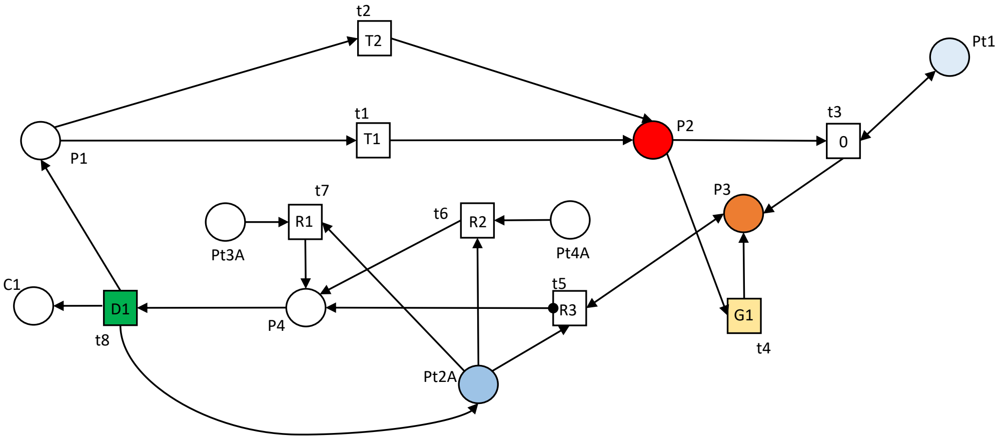


Figure 5.13: A Petri net for the pipework condition in the deluge system

There are two competing types of Type A component failure included in this model. These are random failures such as those that occur due accidental damage, and failures of the component due to ageing. In this model place P1 corresponds to the working state of the pipework and place P2 corresponds to the unrevealed failed state of the pipework. Transition t1 corresponds to ageing of the pipework leading to a failure and transition t2 corresponds to random pipework failures.

In this model a component failure is assumed to be unrevealed until inspection of the component or testing of the system is completed. When place Pt1 is marked a system test is in action, transition t3 corresponds to a system test that reveals a failure in the component. The marking of place Pt1 is discussed in Section 5.9 of this chapter. Transition t4 is a global transition that represents a periodic component inspection that reveals a failure. Place P3 corresponds to a discovered component failure.

It is assumed here that maintenance of the component returns it to the 'as good as new' state. In this model place P4 corresponds to a scheduled maintenance action. Transition t8 corresponds to the maintenance action that returns the condition of the component to the 'as good as new' state. Place C1 counts the number of maintenance actions. A reset transition is used here to model this behaviour.

There are three cases included in this model that can result in a scheduled maintenance action. Firstly, a maintenance action can be scheduled when there is a discovered failure. This is governed by transition t5 which corresponds to a short scheduling delay. Secondly, a maintenance action can be scheduled when it is estimated that the component is close to failure, based on the time since the last maintenance action. This is represented by transition t6 in the model. When place Pt4A is marked this routine age-based maintenance is enabled. Finally, a maintenance action can be scheduled when it is estimated that the component is reaching the end of its useful life, based on the time since the last maintenance action. This is represented by transition t7 in the model. When place Pt3A is marked this early age-based maintenance is enabled. The marking of places Pt3A and Pt4A is discussed in Section 5.9 of this chapter.

A distribution for the probability of an age-based failure at each time is required for transition t1. The distribution for the probability of a random failure at each time is required for transition t2. For application of the model, the inspection interval of the component and the system testing interval are also required. To test different age-based maintenance strategies, estimated distributions to govern the early and routine age-based maintenance actions are required. These can be varied to test the impact on the probability that the pipework is in the failed state. A short delay for scheduling of maintenance if there is a discovered failure and for the time taken for maintenance to be completed can also be varied in the model for a specific system.

Initially places P1 and Pt2A are marked by tokens. A simulation of the Petri net can then be carried out. The probability that the pipework is in a failed state is found by recording the marking of the places corresponding to a discovered or unrevealed failed state and finding the average number of tokens in either place at each time.

5.7.5: Type B components

Type B components are those that have the following features:

- The component can reach the failed state through two competing mechanisms: either through age or random failure occurrence. There is also a quantifiable partially degraded state for the component, due to age. Both partially degraded state and the failed state are unrevealed.
- System testing by opening the test vale can reveal the state of the component. Inspection of the component can reveal the state of the component.
- The component is maintained on a revealed or discovered failure or partially degraded state. The component can also be maintained based on the time since the last maintenance intervention. Maintenance returns the component to the ‘as good as new’ state.

The model for the Type B components is given in Figure 5.14 and is repeated to model the diesel pump, electric pump and jockey pump.

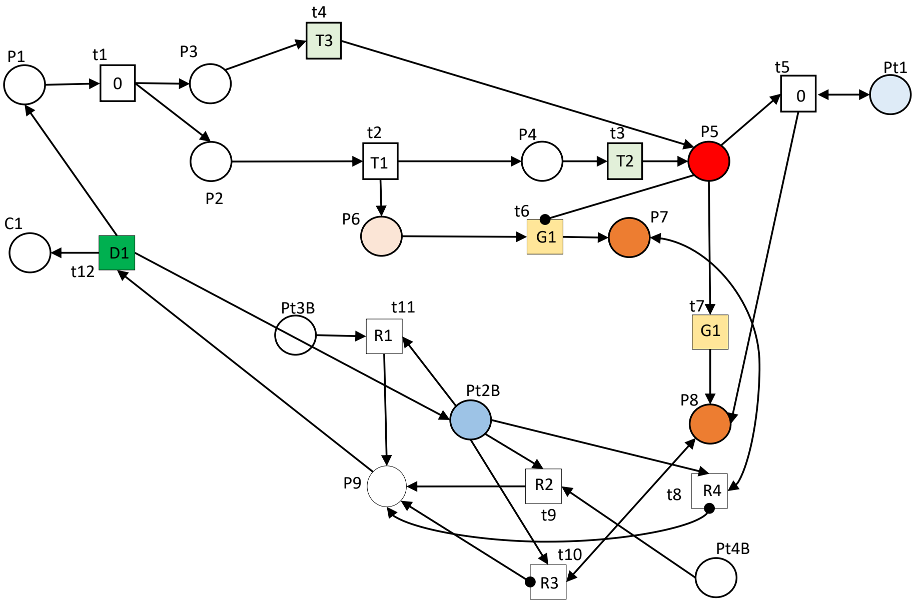


Figure 5.14: A Petri net model for water pump condition

There are two competing failure mechanisms for the Type B components; one is due to the age of the component and one is due to a random failure occurring in the useful life of the component. There is an intermediate degraded state for the Type B components, that forms an intermediate step between the working and failed state. When both of places P2 and P9 are marked this corresponds to the working state of the component. Two places are included here to allow the component to fail due to a random event when it is either in the working or degraded state. Places P4 and P6 are marked simultaneously and correspond to a working but degraded state of the component. Place P5 corresponds to a failed state of the component. Transitions t3 and t4 are reset transitions that prevent the place P5 being marked more than once, by removing the tokens in places that correspond to the alternative mechanism than the one that caused the failure.

For Type B components it is assumed that periodic inspection of the component can reveal the failed or degraded state and the system testing can reveal the failed state. Transition t6 is a global inspection transition that corresponds to an inspection action that reveals a degraded state of the component. Place P7 corresponds to a identified degraded state of the component. Transition t7 is a global inspection transition that reveals a failed state of the component. Place P8 corresponds to a discovered failed state of the component. When place Pt1 is marked a system test is underway, transition t5 corresponds to a discovery of a component failure at this system test.

In this model the component can be maintained in four different scenarios. Firstly, the component can be maintained on a discovered failure, transition t10 corresponds to the scheduling of this. Secondly, the component can be maintained on an identified degraded state, transition t8 corresponds to this. Thirdly, the component can be maintained when it is estimated that the component has reached the end of its useful life, based on the time since the previous maintenance action, transition t9 corresponds to the scheduling of this. This is activated when place Pt4B is marked. Finally, the component can be maintained early, before it is estimated that the component has reached the end of its useful life, transition t11 corresponds to the scheduling of this. This is activated when place Pt3B is marked. When place P9 is marked then maintenance is scheduled, this maintenance is completed after a short delay, modelled by transition t12. Place C1 counts the number of maintenance actions. It is assumed that maintenance returns the component to the ‘as good as new’ state.

Data is required for the rate of random failures of the component and for the time for the component to reach the degraded state from the working state, and from the degraded state to the failed state. The thresholds that define each state can be defined when applying the model, for example the point at which condition monitoring data suggests that the component should be replaced can be used to define the threshold for entry to the degraded state. The inspection intervals of the component are also required, along with the scheduling delays for each of the maintenance actions.

Initially places P1 and Pt2B are marked by tokens. The marking of places Pt3B, Pt4B and Pt1 are governed by a system level strategy, explained in Section 5.9. The number of maintenance actions over time can recorded by tracking the number of tokens in place C1 at each time. The probability that the component is in the unrevealed failed state can be found by tracking the marking pattern of place P9 and the probability that the component is in a discovered failed state can be found by tracking the marking of place P5.

This Petri net is repeated in the model for each of the pump types with the corresponding input data for each pump type. For each pump random pump failures include those resulting from captivation, whereby bubbles in the fluid collapse leading to damage, pressure drops leading to large gas presence, and dry runs of the pump whereby the pump overheats due to a lack of fluid. Ageing failures include those such as erosion, corrosion, damage to bearings, material deposits and damage by oscillations [162]. Pump faults can be diagnosed by several different measurements including: flow rate, inlet and outlet pressure, temperature and vibration [163] [164]. In this example, the failure of the pump is classified as any pump failure scenario such that the pump cannot provide adequate water to the deluge system.

5.7.6: Type C Components

Type C components in this model are those that have the following features:

- The component can fail with two competing mechanisms: one due to the age of the component and one due to random failure occurrence. The failure is unrevealed.
- Inspection of the component can reveal a failure. Testing of the system, by opening the test valve, does not reveal the failure.
- The component is maintained on a discovered failure. The component can also be maintained based on the time since the last maintenance intervention. Maintenance returns the component to the ‘as good as new’ state.

This model is applied to the sprinkler head and detection system wiring.

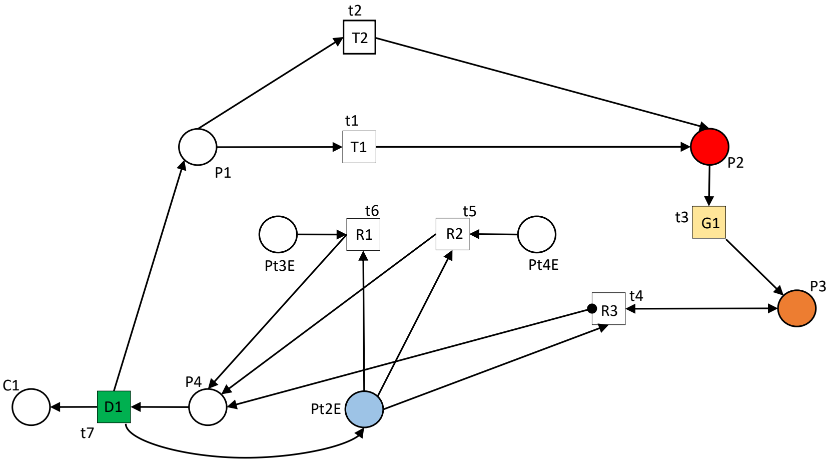


Figure 5.15: A Petri net for the Type C components

There are two competing failure mechanisms for this model. The first corresponds to failures of the component due to ageing, transition t1 models this. The second failure mechanism corresponds to randomly occurring failures. Transition t2 models this. In this model, place P1 corresponds to a working state of the component and Place P2 corresponds to an unrevealed failed state of the component.

A failed state of the component can be discovered by periodic inspection of the component. Transition t3 is a global inspection transition that models this. Place P3 corresponds to a discovered failed state of the component.

Maintenance for the component is modelled as occurring when there is a discovered failure in the component, transition t4 corresponds to the scheduling of this. Age-based maintenance, measured from the time since the most recent maintenance action, is also modelled. Transition t5 corresponds to the scheduling of age-based maintenance when it is estimated that the component has reached the end of its useful life. Transition t6 corresponds to the scheduling of early age-based maintenance, prior to the component reaching the end of its useful life. When Place P4 is marked maintenance is scheduled. Maintenance is modelled after a delay governed by the reset transition t7. Place C1 counts the number of maintenance actions. It is assumed that maintenance returns the component to the 'as good as new' state.

Data is required for the rate of random failure and the time to failure due to the age of the component. Data is also required to govern the inspection interval and the time until each age-based maintenance action. The marking of places Pt3C and Pt4C, which activate the age-based maintenance strategies, are described in Section 5.9.

Initially, places P1 and Pt2C are marked by tokens. The model can be simulated for quantitative analysis. The marking of place C1 can be extracted to give the number of maintenance actions on the component at each time. The average marking of place P2 gives the probability that the component is in the unrevealed failed state at each time. The average marking of place P3 gives the probability that the component is in the discovered failed state at each time.

5.7.7: Type D Components

Type D components within this model have the following characteristics:

- The component can fail with two competing mechanisms: one due to the age of the component and one due to random failure occurrence. The failure is unrevealed but can lie in one of two states. The first failure state does not have the potential to cause a system failure. The second failure state does have the potential to cause a system failure.

- System testing by opening the test valve can reveal a failed state that has the potential to cause a system failure. Inspection of the component can reveal either of the failed states.
- The component is maintained if it is in either discovered failed state. The component can also be maintained based on the time since the most recent maintenance intervention. Maintenance returns the component to the ‘as good as new’ state.

This model is repeated for the isolation valves and pressure release valves.

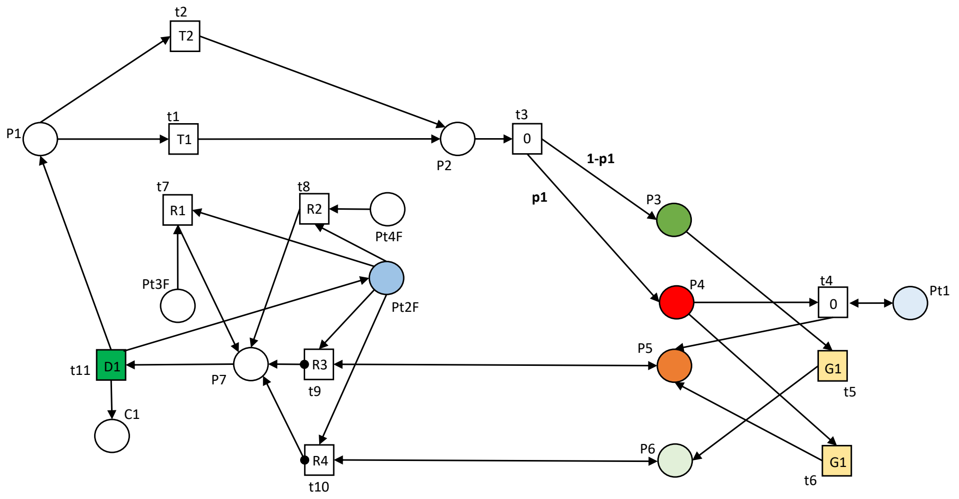


Figure 5.16: A Petri net model for the Type D components

There are two competing failure mechanisms included in this model: one due to the age of the component and one due to random failures. Place P1 corresponds to a working state of the component. Transition t1 corresponds to a failure of the component due to age. Transition t2 corresponds to a randomly occurring failure. Place P2 corresponds to a failed state of the component. On failure there are two failure modes included in the model, the first, modelled by place P3, represents a failure that will not cause a system level failure, and the second, modelled by place P4, represents a failure that will can cause a system level failure. On failure of the component, one of these states is entered in this model, this is represented by the probability transition t3. In this model both failed states are unrevealed.

Both failure modes can be revealed by periodic inspection of the component, in this model. Transitions t5 and t6 are global inspection transitions that model this for each of the failure modes. Place P5 corresponds to a discovered failure that has the potential to cause a system level failure. Place P6 corresponds to a discovered failure that does not have the potential to cause a system level failure. When place Pt1 is marked, a system test is underway. Transition t4 models the revealing of a component failure, which can cause a system level failure, through this system testing.

In this model maintenance is completed when there is a discovered failed state of either type. This maintenance is scheduled after a short delay, modelled by transitions t9 and t10, for each failure type. Also included in this model is the option for early or routine age-based maintenance. When place Pt4D is marked then age-based maintenance is enabled when it is estimated that the component has reached the end of its useful life, based on the time since the previous maintenance action. Transition t8 corresponds to the scheduling of this. When place Pt3D is marked then early age-based maintenance is enabled, prior to the component reaching the end of its useful life. Transition t7 corresponds to this. When place P7 is marked then maintenance is scheduled for the component. Transition t11 is a reset transition that models the maintenance of the component, it is assumed here that this returns the component to the ‘as good as new’ state.

Data is required to govern the time until a failure due to the age of the component and for the rate of random failure occurrences. Data is also required for the maintenance scheduling delay on failure of the component and the time taken for the maintenance to be completed. Different inspection intervals and age-based maintenance intervals can be used as input to the model to test the impact of these on the system failure modes.

Initially place P1 and place Pt2D are marked by tokens. A simulation of the model can then be competed for quantitative analysis. The number of tokens in place C1 can be recorded to give the number of maintenance actions on the component over time. The average marking of places P4 or P5 can be recorded to give the probability that there is a component failure that has the potential to contribute to a system level failure, at each time, for the unrevealed and discovered failed state respectively.

5.7.8: Type E Components:

In this model, Type E components are those with the following shared features:

- The component can fail with two competing mechanisms: one due to the age of the component and one due to random failure occurrence. There are two failure modes of the component. The first failure mode triggers a false activation of the system and is revealed. The second failure mode of the system is unrevealed and has the potential to cause system failure.
- Inspection of the component can reveal a failed state. System testing, by opening the test valve, does not reveal the failed state.
- The component is maintained on a revealed or discovered failure. The component can also be maintained based on the time since the last maintenance intervention. Maintenance returns the component to the ‘as good as new’ state.

In the system model, the model for Type E components is repeated for the deluge valve, solenoid and water closing circuit and the manual start device. Figure 5.17 gives the Petri net that is repeated for each of the Type E components.

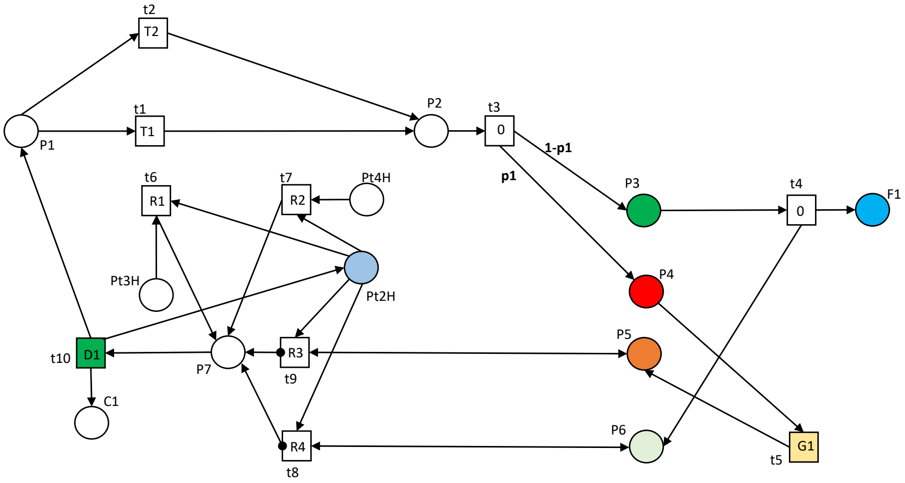


Figure 5.17: A Petri net model for type E components

In this model place P1 corresponds to the working state of the component. The component can fail for one of two reasons in this model. The first reason is that the component fails due to age, this is modelled by transition t1. The second reason is a random failure of the component has occurred, this is modelled by transition t2. There are two failed states of the component, one where the failure causes a false activation of the system and one where the failure is unrevealed and has the potential to prevent the system from responding on demand. Place P3 corresponds to a failed state that causes a false activation of the system and place P4 corresponds the unrevealed failure.

Transition t4 models the false activation of the system due to the component failure, place F1 counts the number of false system activations and place P6 corresponds to a revealed failure due to this false activation. Transition t5 is a global inspection transition that can reveal the failed state of the system, represented by place P4. Place P5 corresponds to a discovered failed state where the failure can lead to a lack of response from the system due to the component failure.

Maintenance of the component is scheduled on a revealed or discovered failure of either state. Transitions t8 and t9 correspond to the scheduling of this. Maintenance can also be completed based on the age of the component, measured by the time since the most recent maintenance action. When place Pt4E is marked then routine age-based maintenance is enabled. The scheduling of this is represented by transition t7. When place Pt3E is marked then early age-based maintenance is enabled. The scheduling of this is represented by transition t6. When place P7 is marked, then maintenance is scheduled for the component. This is completed after a short delay, modelled by the reset transition t10. Place C1 counts the number of maintenance actions and it is assumed that maintenance returns the component to the ‘as good as new’ state.

Data is required for the transitions governing the failure of the component due to age and random failure occurrence. Other inputs to the model are also required, these can be varied to test different strategies, including: the inspection interval, the maintenance scheduling intervals and the time taken for maintenance to be completed.

Initially place P1 and place Pt2E are marked by tokens. A simulation of the model can be completed for quantitative analysis. The marking of place C1 can be tracked over time to give the number of maintenance actions for the component. The average marking of places P4 and P5 can be extracted to give the probability that the component is in a failed state that can prevent the system from responding. The number of tokens in place F1 can be extracted to give the number of false system activations at each time.

5.7.9: Type F components

Type F components in this model are those with the following features:

- This model structure models a population of the same component. The components can fail with two competing mechanisms: one due to the age of the components and one due to random failure occurrence. There are three failure modes included in the model. The first failure mode causes a false activation of the system and is revealed. The second failure mode is an unrevealed failure that does not have the potential to cause a system failure, due to inbuilt redundancy. The third failure mode is an unrevealed failure that has the potential to cause a system failure. If the model is in the second failure mode, a further failure can occur to result in the model residing in the third failure mode. This second failure can arise due to further ageing of the components or a random failure.
- Inspection of the population of components can reveal the second or third failure mode.
- The population of components are maintained on any revealed or discovered failure. The population of components can also be maintained based on the time since the last

maintenance intervention. Maintenance returns the population of components to the 'as good as new' state.

In this model this module is applied to the smoke detectors, heat detectors and manual call points.

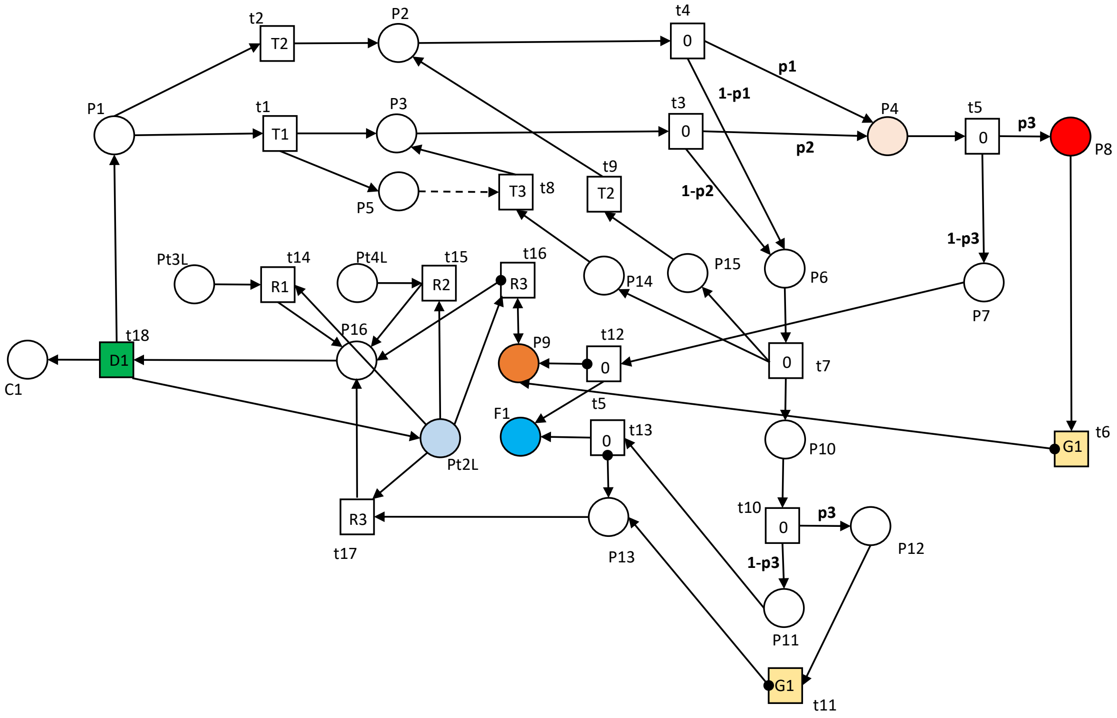


Figure 5.18: A Petri net model for the smoke or heat detectors

In this model a population of components are modelled. Place P1 corresponds to the working state of the population of components. A failure in the population can occur due to either the ageing of the components, represented by transition t1, or a random failure of the components, represented by transition t2. When place P3 is marked a failure has occurred in the population of components due to the age of the components. When place P2 is marked a failure has occurred in the population of components due to a random failure. For each of these failures, in this model there is a probability associated with the failure type to cause a system level failure, this is represented by either of transitions t3 or t4. When place P6, or place P10, is marked the failure is partial and insufficient to cause a system failure and when place P4 is marked the failure is enough to cause a system failure. While the population of components is in a partially failed state, further failures due to age or random occurrences are modelled by transitions t8 and t9 respectively, where transition t8 is a conditional transition whose distribution is dependent on if the previous failure of age related.

For either a partial failure or full failure in the population of components, there is a probability that the failure will be immediately revealed by activating a false alarm of the system. This is represented by the probability transitions t5 and t10 for the full and partial failures respectively. Place F1 counts the number of false system activations. An unrevealed failure in the population of components can be revealed by periodic inspection and testing of the components. The global inspection transitions t6 and t11 model this. Place P9 corresponds to a discovered failed state of the population of components that has the potential to cause a system level failure and place P13 corresponds to a discovered partially failed state of the components.

In this model maintenance is completed in this model when there is either type of revealed or discovered failure. Transition t17 corresponds to the scheduling of maintenance when there is a discovered partial failure. Transition t16 corresponds to the scheduling of maintenance when there is a discovered failure that can cause a system level failure. Maintenance is also completed based on the age of the population of components, measured from the most recent maintenance action. When place Pt4F is marked then routine age-based maintenance is enabled, this occurs after a delay governed by transition t15. When place Pt3F is marked then early age-based maintenance is enabled, this is governed by transition t14. When place P16 is marked then maintenance has been scheduled, maintenance is modelled by the reset transition t18. Place C1 counts the number of maintenance actions on each population of components. It is assumed that maintenance returns the population of components to the ‘as good as new’ state.

Data is required for the transitions governing the random failure rate in the population of components and the failure of the components due to age. Data is also required for the transition governing the rate of a second ageing failure event, given that is a current failure in the population due to age. To test different asset management strategies the scheduling delays for each of the maintenance actions can be varied, along with the marking of the place Pt3F and Pt4F on a system level. Different inspection intervals can also be used as input to the model. These values can be adjusted for each application of the model and to each component population type.

Initially place P1 and place Pt2F are marked by tokens. The model can then be repeated for each component type population and simulated as part of the whole system model, via Monte Carlo simulation. The number of tokens in place C1 over time can be extracted to give the number of maintenance actions for each population of components. The marking of places P8 and P9 can be extracted to give the probability that there is a failure in the population of components that can cause a system level failure. The number of tokens in place F1 can be extracted to give the number of false system activations at each time.

5.8: Discussion of component model results

Full results for the probability of each component failure at each time can be found in Appendix 5, for sample data values given in Appendix 3. There are several key characteristics that are common across the results for each component; the results tend to fit into one of the three categories.

The first characteristic seen in the results occurs in cases where there is a long time to failure, such that it takes some time for any number of failures to be observed within the model. This is seen in the results for the pump models, where a degraded pump condition can be identified and rectified prior to failure. An example of this profile for the output for the model is shown in Figure 5.19. After an initial period where failures are highly unlikely, the probability of an unrevealed failure increases before continuing at a steady rate. At this point in the sample application, the system model is in the third maintenance phase and so age-based preventative maintenance is enabled for components across the system. This balances any expected increase in the probability of component failure as the system ages.

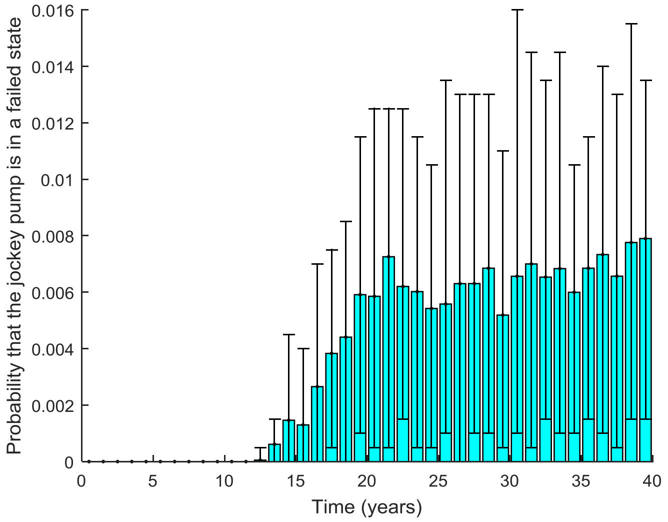


Figure 5.19: The probability that the jockey pump is in a failed state at each time

The second characteristic seen in the results is a case where there are decreases in the probability of failure, after each of the system phase changes. When the system is in the first phase maintenance is only condition-based. When the system is in the second phase, age-based maintenance is enabled. When the system is in the third phase then the interval governing the age-based maintenance is decreased, so that it occurs more frequently. For components with more common failure probabilities, where the component can fail in a shorter time, decreases can be seen in the probability of failure that correspond to the phase change points. An example of this pattern of behaviour can be seen in Figure 5.20, where the phase changes occur after 3 and 13 years in this example.

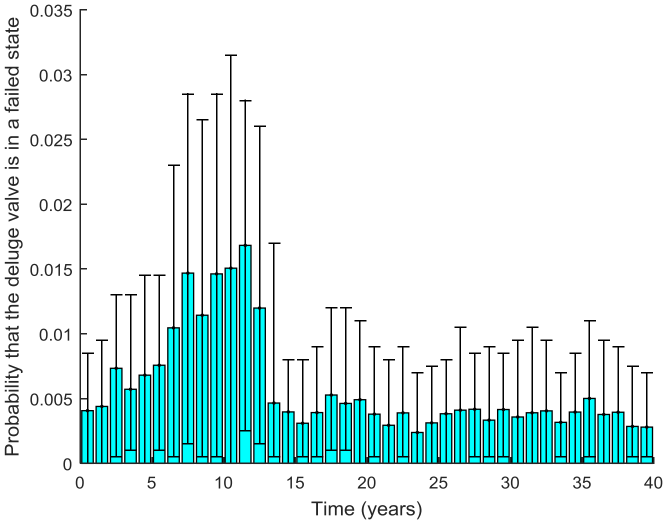


Figure 5.20: The probability that the deluge valve is in a failed state at each time

The third characteristic seen in the results is a steady increase in the probability of failure from the start time, followed by a levelling in the behaviour. This is especially seen for the Type F component models, where a population of components is considered and where the components are maintained as a population if there is a discovered degraded or failed state of one of the components. This levelling behaviour can be attributed to the age-based preventative maintenance introduced in the later phases of the model simulation.

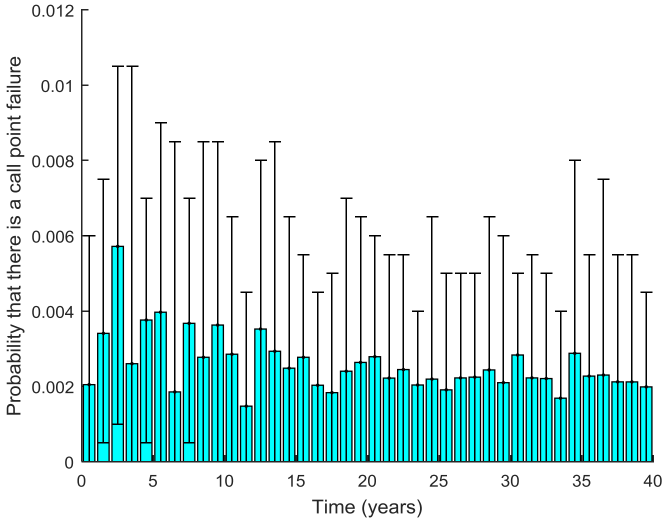


Figure 5.21: The probability that there is a call point failure at each time

Finally, more variable behavior can be observed when there is a lower probability of component failure. Trends can be observed within some of these rare failures that mimic the more common patterns: a decrease at the phase transition points, a steady rate and a gradual increase as the component ages. Examples where there is a lower probability of failure tend to show more variation year on year, relative to their value, due to their rarity of occurrence within the simulation.

5.9: Incorporating system level phased maintenance strategies

The component Petri net models, presented in the previous section, were linked together by overarching maintenance strategies, with different preventive and reactive maintenance strategies activated in each system-level phase. Where multiple types of component were modelled by the same Petri net structure, the structure was repeated with data for each specific type of component. The number of system tests and the inspection frequency of each component also depends on the system phase. The Petri net in Figure 5.22 governs these system level strategies.

In this Petri net, when place P2 is marked, the system is in the first system level maintenance phase. After the firing of transition t3, place P3 is marked. This models the system as residing in the second system level maintenance phase. Place P4 corresponds to the third system level maintenance phase, this is marked when transition t4 fires. Transitions t3 and t4 add a token each to the conditional place Cp1 on firing, which is initially unmarked.

Transition t1 represents the interval between full system tests, where the system is tested by opening the test valve, and transition t2 represents the time taken for a full system test to be completed. Transition t1 is conditional on the marking of place Cp1, which records the phase of the system. The distribution governing the firing of transition t1 is dependent on the number of tokens in place Cp1. Place C1 counts the number of full system tests. The number of full system tests is governed by the delay time in transition t1. In this application of the model, the inspection interval can be different for each component and is conditional on the marking of place Cp1.

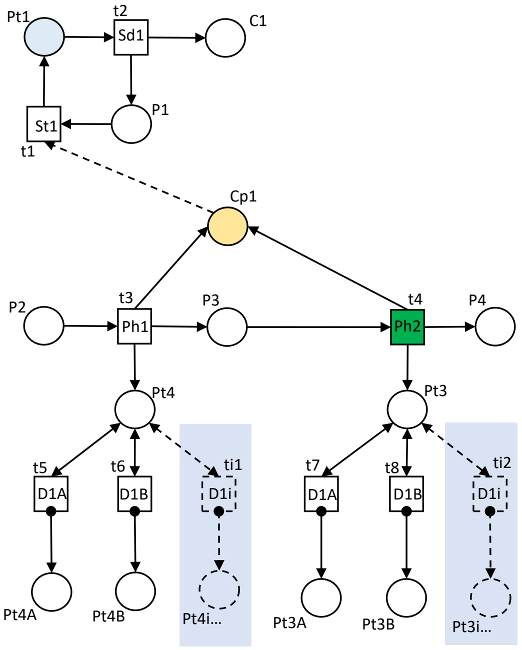


Figure 5.22: Petri nets for testing of the system and full system maintenance strategies

The activation of system level early preventative maintenance is enabled by marking place Pt3 and the activation of system level routine preventative maintenance is enabled by marking place Pt4. This structure could be simplified using a Coloured Petri Net, which would be beneficial for models of a larger size as it would prevent repeating model structures.

In the second and third system maintenance phases, aged-based maintenance is carried out after an interval of time from which the previous component maintenance action was undertaken, or from when the component was installed. The interval is specific to each component and is described in the component model sections presented earlier in this chapter.

In Figure 5.22, as each new phase is entered, tokens are created to indicate the activation of each corresponding component maintenance action. For example, when place Pt4 is marked, places Pt4A, Pt4B, Pt4C etc. are all also marked. Similarly, when place Pt3 is marked, places Pt3A, Pt3B, Pt3C etc. are also marked. In this figure, the marking of these corresponding places is represented by the shaded and dotted structure in the Petri net, representing the repeating of the same module to mark all Pt4i places, for some *i* in the set of all the Petri nets for component models. Transition t4 removes the marking of place Pt4, however, if an age-based maintenance action is scheduled in the previous phase, this scheduling remains by retaining any current marking of the Pt4i places.

In addition, these phases can be used to govern the inspection strategy of each component. For each individual component, an inspection interval can be assigned for each system level maintenance phase. For example, during the first phase, inspection can be carried out with less frequency. Following this, as the system enters the second and third phases, the inspection interval for each component can be decreased so the inspection is carried out on a more frequent basis. The aim of an inspection strategy such as this, is to focus the resources towards the end of the system life where it is more likely that there will be multiple component failures, which can result in a system-level failure. This is included in the Petri net model through conditional transitions. During each phase change, the marking of the conditional place, Cp1, is increased by one token. Within each component level Petri net model, the transitions corresponding to the inspection and testing of the component are conditional on the number of tokens marking this conditional place. The model then selects the correct inspection interval based on the marking of place Cp1.

The distributions Ph1 and Ph2 can be varied to test the impact on the maintenance cost and risk caused by component failures. An optimization based on the varying of these phases is performed in chapter 6 of this thesis. For the transitions corresponding to the preventative maintenance of each component, the lognormal distribution is assigned to govern the transition firing time.

Sample Application

For the sample values, given in Appendix 3 and used to illustrate the models throughout this chapter, the outputs relating to maintenance actions across the system were collated. Figure 5.23 shows the maintenance of the diesel tank, electric pump, diesel pump, jockey pump and the ringmain. The jockey pump has the highest number of maintenance actions per year, corresponding to the faster rate of ageing assigned in the input data. Across the component a higher level of maintenance can be seen, as the components age. This corresponds to the increase in preventative maintenance as the system level maintenance phase increases, and a higher number of maintenance actions due to component failures.

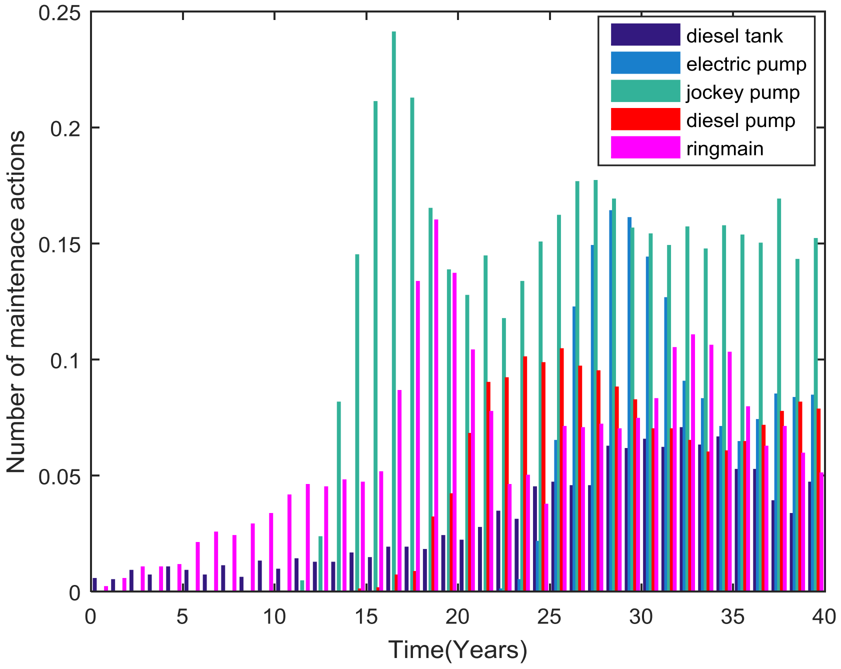


Figure 5.23: The maintenance actions for the diesel tank, pumps and ringmain

Figure 5.24 gives the maintenance actions for the sprinkler heads, isolation valves, pressure release valves and pipework. For the isolation valves and pressure release valves, these results represent the total number of maintenance interventions on a valve of that type within the system. For example, there are five isolation valves in the system and at approximately 20 years, three interventions will be carried out over these five valves. Similarly, there are two pressure release valves modelled in this case. These components can be modelled individually, to consider opportunistic maintenance strategies across similar components, by repeating the model structure used in this chapter and adding dependencies in maintenance actions. This gives a higher number of maintenance actions for the isolation valves and pressure release valves. A low level of pipework maintenance can be seen here corresponding to the low probability of failure and low level of preventative maintenance. A cyclic behaviour can be seen in the other components, which relates to periodic age-based maintenance assigned in this application of the model.

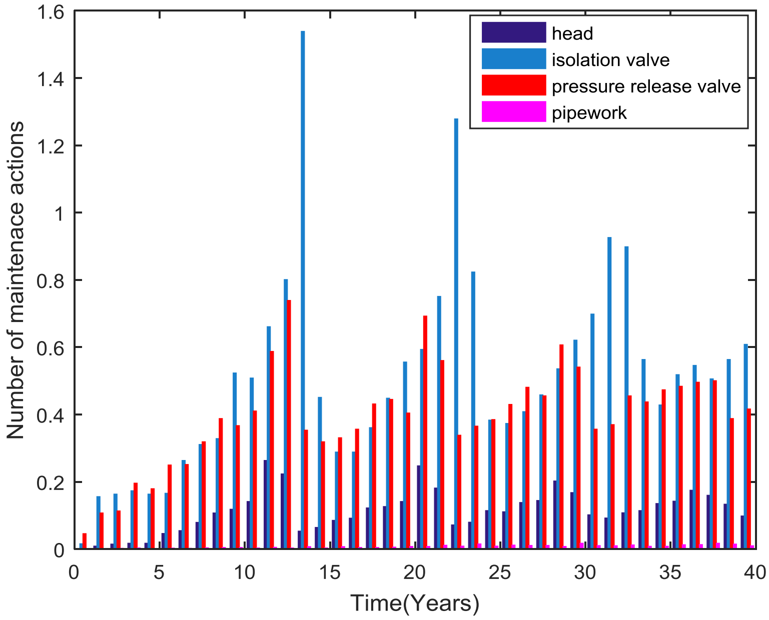


Figure 5.24: The maintenance actions for the sprinkler heads, isolation valves, pressure release valve and pipework

Figure 5.25 gives the maintenance actions for the deluge valve, manual start device, pressure sensor and solenoid. For components with shorter times to failure due to age, or more frequent random failures, it is expected that the maintenance actions will begin sooner and occur more regularly due to the higher probability of a random failure and a shorter age-based maintenance period.

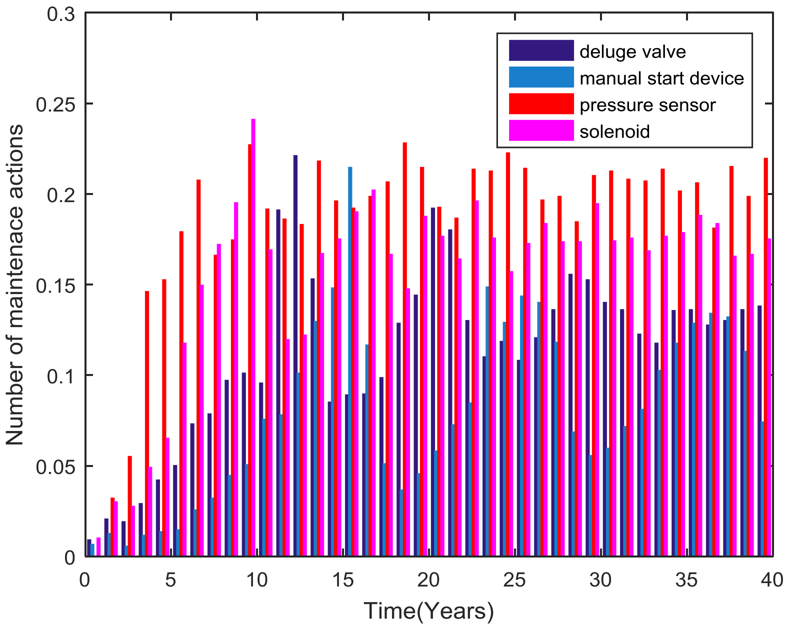


Figure 5.25: The maintenance actions for the deluge valve, call point, pressure release valve, pressure sensor and solenoid

Figure 5.26 gives the maintenance actions for the alarm sounder circuit, the call point, heat detector and smoke detector. Due to the short ageing time of the call point and the higher rate of random failures, the higher levels of call point maintenance can be expected from the input data.

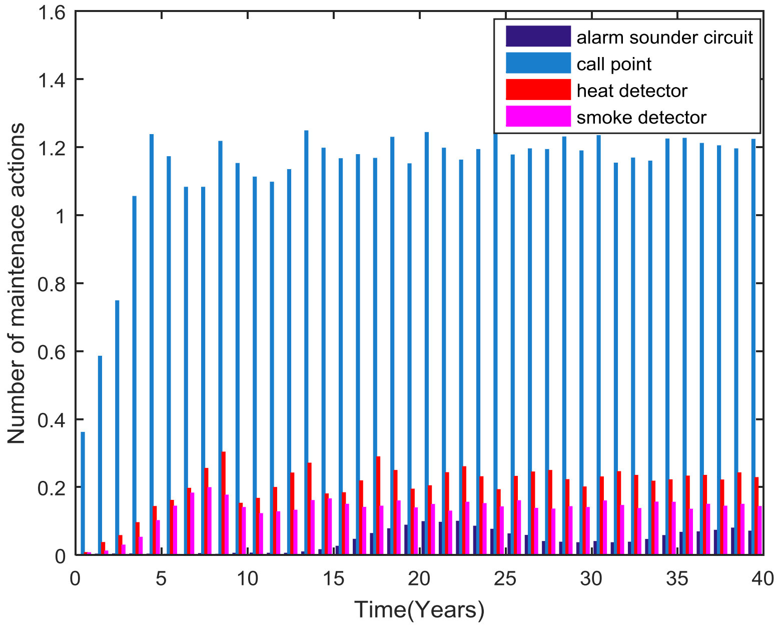


Figure 5.26: The maintenance actions for the alarm sounder circuit, call point, heat detector and smoke detector

Figure 5.27 gives the number of maintenance actions for the control box, the control box battery and the wiring. Here it can be seen that there is a higher level of control box battery maintenance actions, corresponding to the faster aging of the battery in comparison to the other components. Control box maintenance begins soon after installation and increases as the component ages, this maintenance can be attributed to the assigned random failure rate and the revealed failure of the control box. Wiring maintenance begins towards the end of the 40 year period, this can be attributed to the long ageing time, low random failures and scheduled preventative maintenance.

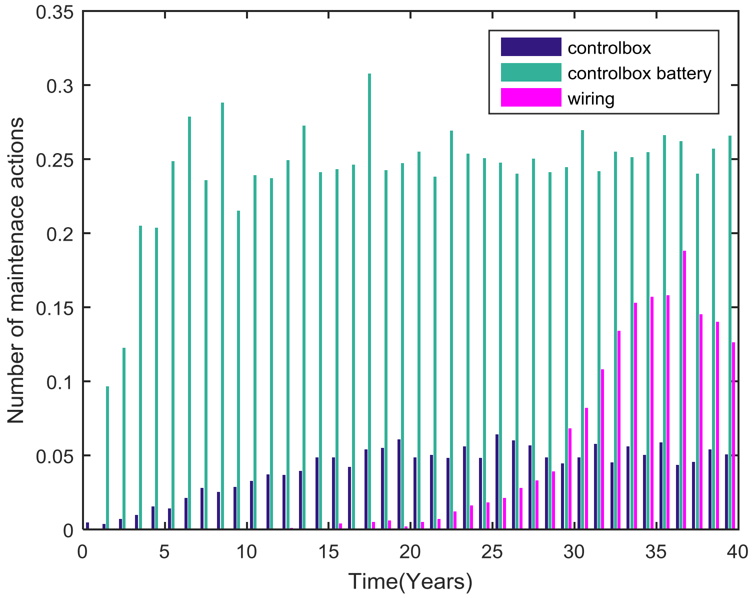


Figure 5.27: The number of maintenance actions for the control box, control box battery and the wiring

Figure 5.28 gives the number of false activations of the fire protection system and the number of system tests, by opening the test valve. An increase in the number of system tests as the system enters the second and third system maintenance phases can be clearly seen at 3 years and 13 years. As the system ages past 8 years there is no notable increase, and some decrease, in the number of false activations of the system each year. This is due to the increased maintenance reducing the number of component failures. This suggests that the maintenance strategy applied in this case can also control the number of false activations of the system, despite the ageing of the system.

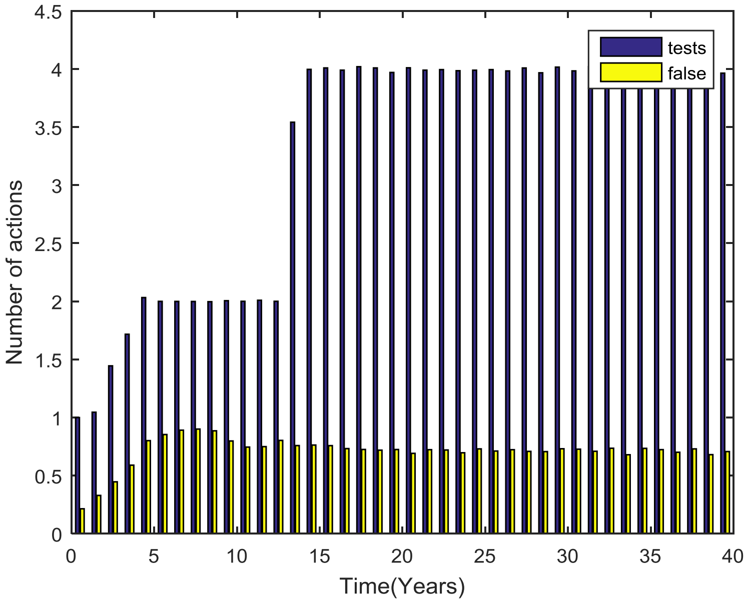


Figure 5.28: The number of false system activations and system tests

5.10: Human Interaction with the system

There are two cases in the system failure modes presented in this chapter, where the action of a member of staff or the public can prevent a system failure mode. The first action is the manual activation of the deluge system by a staff member. The second action is the operation of a call point by a member of the public, or staff member. Petri net models are used to give the probabilities that these actions have been carried out successfully, at each time from fire initiation.

5.10.1: Manual activation of the deluge system

The first Petri net, which models the manual activation of the deluge system, is given in Figure 5.29. This Petri net models the probability that a staff member will operate the system as time progresses from the initiation of the fire.

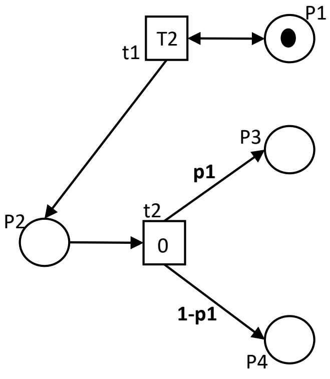


Figure 5.29: A Petri net to model a failure of manual activation of the deluge system

In this Petri net model, place P1 represents the presence of staff in the station. Place P2 corresponds to the arrival of a staff member at the manual activation device. Place P3 corresponds to a successful activation of the system by the staff member, and place P4 to an unsuccessful attempt to activate the system. An unsuccessful attempt could be due to a lack of specific operational training, or a lack of communication and assumption of a false alarm. Transition t1 represents the arrival rate of staff to the activation device. Transition t2 represents the probability of a successful activation.

The results for the simulation of this Petri net for sample staff arrival times and probability of successful activation, are shown in Figure 5.30. Two different staff arrival distributions are used for illustration of the model. The first is a normal distribution with a mean value of 5 minutes and a standard deviation of 2 minutes. The second is a normal distribution with a mean value of 15 minutes and a standard deviation of 5 minutes. A sample probability value of 0.6 for successful activation by each staff member is included in both cases. This model combines the rate of arrival of staff from the point that the fire occurs, with the probability of successful activation of the system. This means that even if the automatic system fails, then the manual activation can still occur. This is included in the modelling in this chapter and Chapter 6.

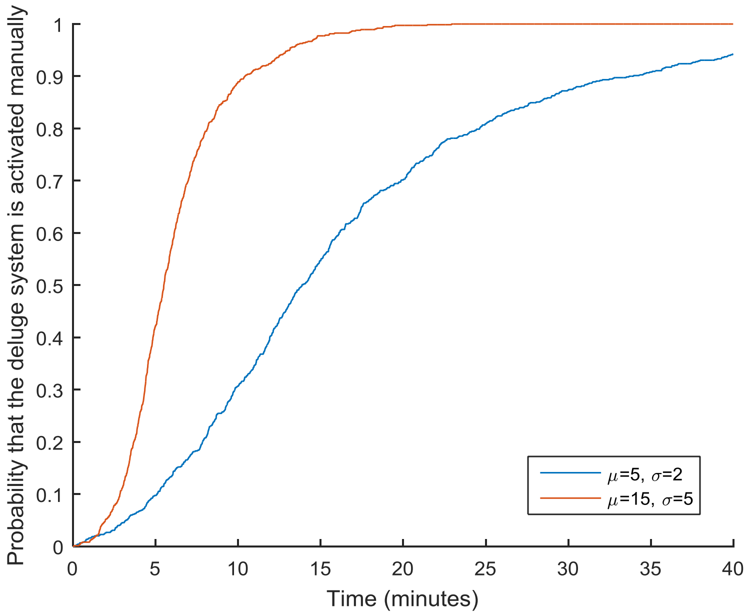


Figure 5.30: A graph showing the probability that the manual detection device is activated over time for sample arrival rates

Here, the time from the initiation of the fire is given along the x-axis and the probability that the manual activation of the deluge system has occurred is given on the y-axis. With an increase in time from the initiation of the fire, the probability that the deluge system has been activated manually increases.

5.10.2: Operation of a call point upon the manual detection of a fire

The second Petri net in this section models the successful operation of a call point by a member of the public or staff upon the manual detection of a fire. This Petri net is given in Figure 5.31. This Petri net can be used to give the probability that the alarm and deluge system will are successfully triggered by the call points as time progresses after the initiation of a fire.

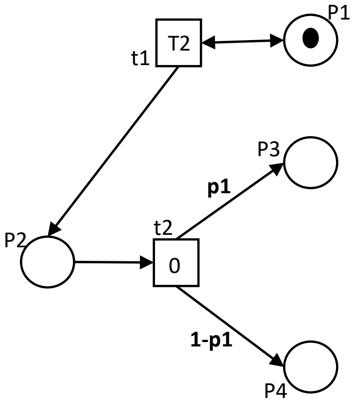


Figure 5.31: A Petri net model for the call point operation failure

In this Petri net, place P1 represents the presence of members of the public or staff in the station. Transition t2 governs the arrival rate of a staff member or member of the public at a call point. Once a member of staff or public arrives at the call point, there is a probability that they will have identified the fire and act appropriately by triggering the call point. Place P3 corresponds to the activation of the call point by the member of the public or staff, and place P4 corresponds to a failure to activate the call point successfully.

As a demonstration of the model sample values were assigned to the arrival rate and probability of activation for two different cases. Figure 5.32 gives the results for two separate sample arrival rate distributions; one normal distribution with a mean of 5 minutes and a standard deviation of 2 minutes and one normal distribution with a mean of 0.7 minutes and a standard deviation of 0.3 minutes. A sample probability value of 0.8 was assigned for successful activation of the call points in both cases.

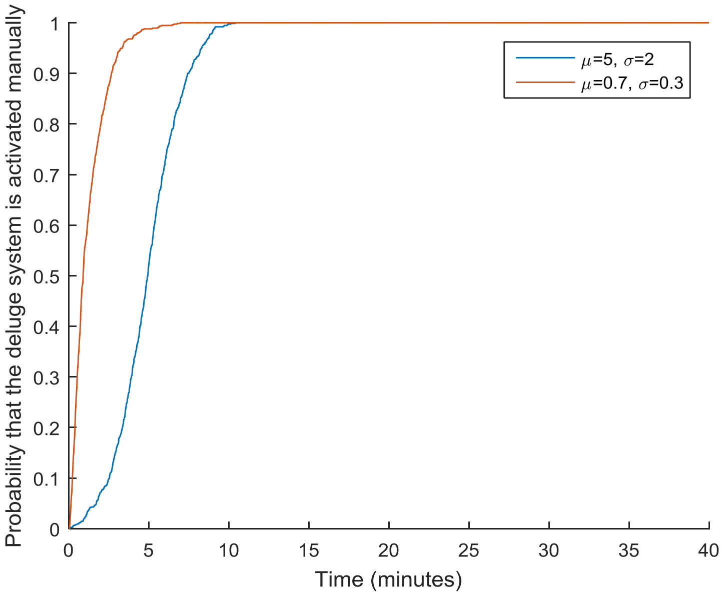


Figure 5.32: A graph showing the probability that a call point is operated at each time for sample arrival rates

In this figure, the time from the initiation of the fire, in minutes, is presented along the x-axis. The probability that the call point has been activated is located on the y-axis. In all cases, as the time increases from the initiation of the fire, the probability that the call point has been activated increases.

A suitable timeframe must be identified to extract the probability that the detection system is activated by a manual call point, and the deluge system is manually activated, before impacting the consequences of a fire. The time frame that is suggested is the time between the initiation of the fire and the time at which evacuation must begin in order to enable all people to leave the station before the fire reaches a critical size.

5.11: Overall failure

The probabilities of each component failure at each time can be used to provide data for the basic events present in the Fault Trees derived earlier in this chapter. The structure of the Fault Trees can be analysed to give the Boolean logic for a total failure of the deluge system, the failure of the detection system and the failure of the alarm system. The probability of each of these system-level failure modes can be found by combining the probability of failure of each of the components at each time, following the structure of the Fault Tree.

Sample Application

The analysis proposed here was completed for the sample values given in Appendix 3, taking the probability of component failure from the models presented in Section 5.7 of this chapter.

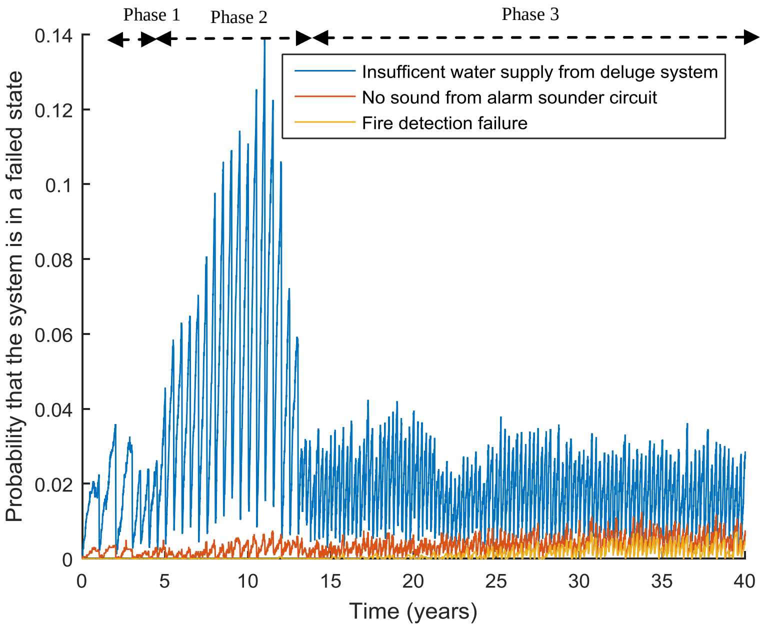


Figure 5.33: The probability that there is a system failure over time

Figure 5.33 gives the probability that each protection system will be in the failed state over time, with the component models combined following the logic of the Fault Tree models presented earlier in this chapter. In these results, a decrease in the probability of system failure can be seen at 13 years and 3 year. This corresponds to the change in system level maintenance phase resulting in an increase in inspection frequency and system testing at these points, meaning that each component spends less time in the failed state, and so reducing the probability that a system-level failure will occur. There is an increase towards 13 years corresponding to the ageing of the components as well as limited preventative maintenance. The reduction following 13 years can also be attributed to the enabling of preventative maintenance strategies. After approximately the 13-year point, preventative maintenance keeps the probability that each system is in a failed state at a relatively constant level, despite the ageing of the system. There is some uncertainty in the solutions, introduced through the Monte Carlo Simulation of the model, and this is shown by the variation in the results with each step.

Figure 5.34, Figure 5.35 and Figure 5.36 show these results for each system separately and are grouped per year, with the maximum and minimum value for that year displayed as an error bar. These error bars can be reduced in size by performing more runs of the simulation. It can be noted that the more common failure events have a smaller error bar in proportion to the average. This corresponds to the proposal that rare events require more runs of a simulation to reach convergence.

These results demonstrate that increased inspection frequency has a positive impact on the probability of a system failure, when concerning the more commonly failed state of components in the deluge system. Also, preventative maintenance can be used to keep the probability that the system is in a failed state at an acceptable level, despite the ageing of components within the system. For the detection and alarm system the probability of failure is lower and increases with time. This suggests that the system fail less frequently and has components with slower ageing rates, such that there are limited failures prior to the entry onto the third system level maintenance phase.

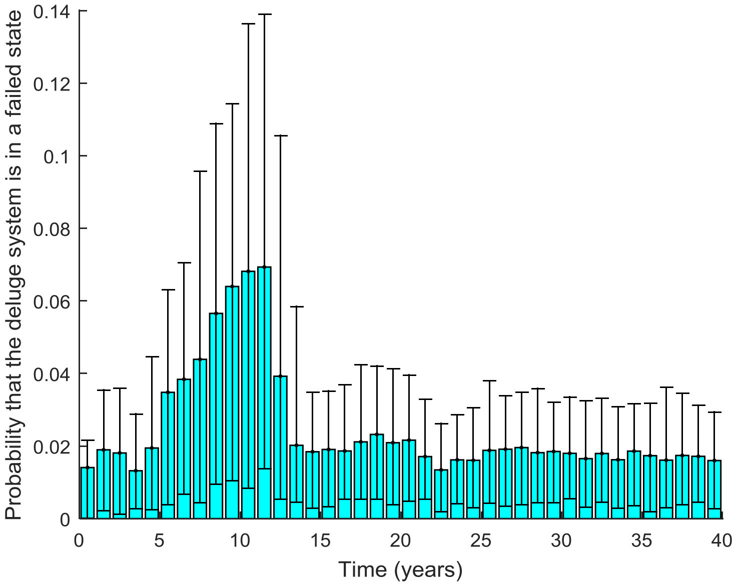


Figure 5.34: The probability that there is a deluge system failure over time

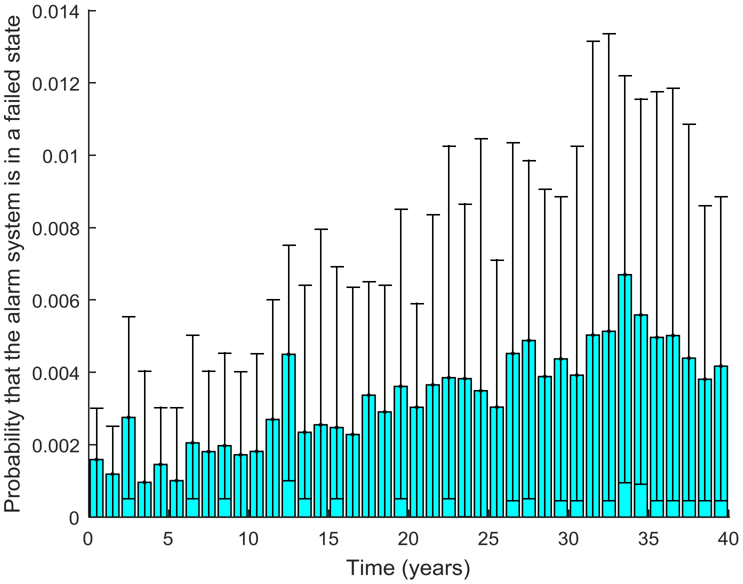


Figure 5.35: The probability that there is an alarm system failure over time

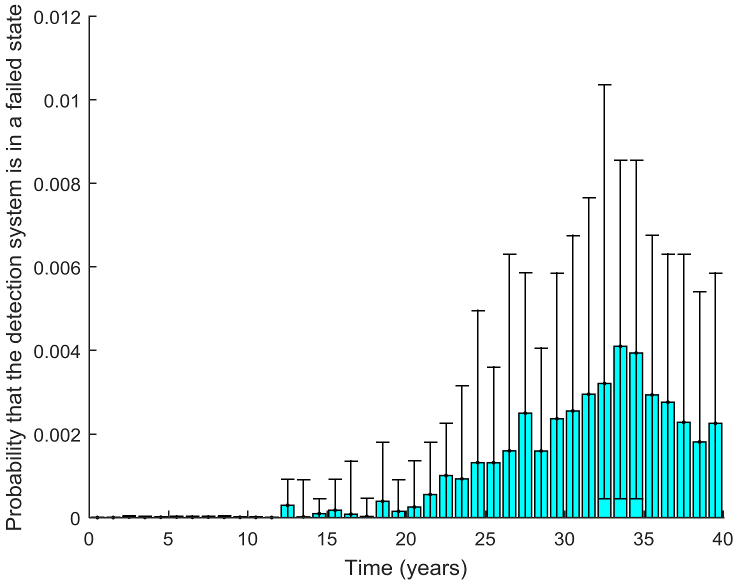


Figure 5.36: The probability that there is a detection system failure over time

5.12: Discussion

The model presented in this chapter demonstrates the capability of a Petri net approach to model component-level ageing and failure, alongside system level phased maintenance, inspection and testing strategies. Component failures are combined through Boolean logic under the assumption that an individual component failure, or group of component failures in some cases, are independent from the remaining component failures in the model. Although every component condition is dependent on underlying system level maintenance strategies, this is deemed a suitable approximation as there is be no interaction modelled here between components ageing across different modules. Where stronger dependencies are introduced through opportunistic maintenance strategies, these components are modelled in one Petri net to incorporate the dependency. The approach improves computational efficiency as it avoids the need for repeat simulation of the same component, where there are multiple in the system, and keeps the Petri net structure to a minimum.

There are some assumptions made in this model. Firstly, it is assumed that the setup and placement of the components of the fire protection systems are sufficient that when in the working state the systems will function correctly. For example, the sprinklers are set up such that they are placed at regular intervals and have the capability to stop a fire. In other words, this model assumes adequate system design.

The application in this chapter has one smoke detector and call point circuit, one heat detector and call point circuit and one sounder circuit. This model can be applied to a more complex fire protection system by repeating the relevant Petri net modules and expanding the Fault Tree structure. The human factors model presented in this chapter is also illustrative. More complex underlying behaviours such as panic or overcrowding and the flow of people in an emergency can be incorporated into this model, if desired. In total, the model developed has 307 transitions and 194 places.

For each of the results in this section, there is a large amount of variation at each time, giving a large range in the probability of component failure each year and some discontinuity in the predicted probability year-on-year. This can be attributed to the low numbers of component failures in each case resulting in a rare event simulation. By increasing the number of runs of the Petri net in the Monte Carlo simulation, this noisy behaviour can be reduced, at the detriment to computational efficiency.

Methods should be given to gain a measure of the risk from modelling approaches such as this one. This is addressed in the next chapter of this thesis. Also, this chapter has highlighted some areas of study where further analysis of a Petri net model of this type can be completed. The first area of interest is a conversion of the Fault Tree structure into a Petri net model for comparison of the results obtained and to provide a framework for incorporation of further dependencies within the system. Secondly, an automatic optimisation method of the phased maintenance strategy presented in this chapter could have useful applications to ageing engineering systems. Finally, a consideration of the convergence and uncertainty of the predictions made by this modelling approach can further validate this methodology. These areas of study are addressed in the following chapter.

5.13: Parameter Assumptions and Use of Data

In this chapter, a sample application of the model has been presented to demonstrate the modelling capability. In this sample application, parameters governing component failure rates have been assumed. Assuming faster failure rates increases the probability of system failure; similarly assuming slower failure rates decreases the probability of system failure. In addition, the parameters governing the maintenance, inspection and testing transitions have been assumed. If the parameters governing the maintenance interval are altered so that the interval between maintenance actions is shorter then there is a reduction in the probability of each system failure. Sample system maintenance phases have also been assigned, but these can be varied to test the system under different phased strategies.

Since the probability of each system failure is small, this outcome of the model is most sensitive to the assumptions of the parameters governing the component conditions. This is especially notable for parameters that govern the state of components contained within the lower order minimal cut sets. In this case, since there are fewer entries in the minimal cut set, if one parameter is falsely assumed it is more likely to impact the probability of system failure.

Despite the assumptions in the model parameters used to give an example application of the model, the approach demonstrates the modelling of the logic of the three sub-systems: the deluge, detection and alarm systems. The parameters within the model can be easily altered within the excel spreadsheet containing the model logic. Furthermore, the sample results presented in this chapter have been extrapolated to non-specific parameter values to show the trends present in some of the results.

In order to improve the model and validate the results further, data should be collected for the time to failure for each component. This data can be used to inform the distribution choice and parameter values for each of the transitions within the model that govern the degradation rate of each component. This can improve the model by bringing each component model more in line with reality. If any adjustments to the modelling of the degradation of each component are required, the model can also be improved, using the collected data. Data should also be collected for testing and inspection strategies, to inform the parameters governing the transitions related to these within the model. Also, this can be used to adjust the model structure, if required, in order to include different strategies present in reality, or remove existing strategies that are not present in reality.

In order to validate the model, data should be collected on the system failure over time, and the circumstances that caused the failure. The model can then be validated and adjusted, such that the model more closely recreates reality. This would improve the model further to increase its accuracy, so it can be used to test scenarios, such as, the use of different system phases.

5.14: Contributions

This chapter presents a novel model for a combined deluge, detection and alarm system. The model improves the state of the art as it applies the Petri net approach to a fire protection system model, modelling the system in a higher level of detail. The modelling of the systems together also allows dependencies between the systems and in the final risk prediction. In addition, a new approach incorporating phased asset management strategies is presented. This is beneficial as it allows different maintenance strategies to be applied at different times, notably as the system ages. This improves current models where the maintenance is based solely on individual component state. Here maintenance can be scheduled based on the time since the last maintenance action. Other areas of novelty include the use of both the probability that the systems are in an unrevealed failed state, and the occurrences of false activation of the system. This improves the functionality of traditional methods for risk modelling of fire protection systems, as it allows consideration of other factors.

5.15: Conclusion

This chapter has considered the application of an integrated Fault Tree and Petri net based modelling approach to the automatic fire protection systems on an underground station. Initially, an introduction to the type of systems present on underground stations is given. The work follows from the literature review in Chapter 2, which highlights a deficiency of modelling approaches that can be used to predict system unavailability where complex maintenance strategies are included. The methodology demonstrated in this chapter is applied to an automatic detection, alarm and ringmain based deluge system. A definition of the system is given in the early stages of the chapter.

A definition of the method applied in this chapter is given in Section 5.3, in addition a sample phased system level maintenance, inspection and testing strategy is described. Following this, the system failure modes are identified and a Fault Tree, with associated minimal cut set analysis, for each safety critical system failure mode is presented. Values for the basic events in these Fault Trees are obtained from a Petri net model, with repeatable modules for each component type. The phased system level maintenance strategy and human interaction with the system are also modelled with a Petri net framework. Sample results from the Petri net modelling framework are presented throughout the chapter.

The modelling framework within this chapter gives the unavailability of each of the deluge, detection and alarm systems at each time over a time period from installation of the system. In the sample application, clear differences in the unavailability can be seen when the system enters each different maintenance phase. This demonstrates how a higher level of preventative maintenance, inspection and testing can reduce the unavailability of the systems, despite the ageing of the systems. The modelling framework also gives the frequency of false system activations, system tests and the number of maintenance actions at each time. There is further analysis possible on this type of model including the risk based optimisation of phased maintenance strategies, this is discussed further in the next chapter.

This chapter has demonstrated the flexibility of a Petri net-based approach to model complex systems and maintenance strategies

Chapter 6 Modelling Risk

6.1: Introduction

In the previous chapter, a Petri net methodology was applied to an automatic fire protection system to give the probability of each component failing, at each time, for a given maintenance strategy. These component failures were combined using Fault Tree logic, under the assumption that the component failure modes, which result in the system failure, are independent of each other. In addition to this, it was assumed that there was no uncertainty in the parameters used as input to the model. Ranges in the values obtained by Monte Carlo simulation of the model were given for each model output at each time. The chapter presented the results for a single phased maintenance strategy.

The system failure modes, given in the previous chapter, can be combined with an estimate of the frequency of fire occurrence and the consequences, for each possible combination of events, to give a value for the fire risk on an underground station.

This chapter explores further possible analysis methods within the Petri net framework demonstrated thus far in this thesis. This includes, the conversion of a Fault Tree structure into a Petri net model, and a comparison of the outputs in each case. Also, a risk based optimization method is presented in this chapter, for a Petri net modelling framework. Within this optimisation methodology a method for estimating the system risk is given. The optimisation method is used to find an optimal solution for the phased maintenance and inspection strategy, presented in the previous chapter, given a constrained budget. Also in this chapter, an analysis of the rate of convergence of the model and the uncertainty introduced through Monte Carlo simulation of the model is presented. Finally, a method for encompassing the uncertainty in the model outputs, given uncertain inputs, is presented.

For exploratory analysis of the methods presented in this chapter, the models presented in Chapter 5 are used as a sample application, with results presented throughout. The methods are not limited to this sample application and can be generalised to different Petri net structures. The failure modes used in the sample applications correspond to those presented in Chapter 5. The models for the component failure used are also the same. The Petri net for the phased maintenance of the system is also the same as that used in Chapter 5.

6.2: Converting the Fault Tree structure to an Petri net

The Fault Tree structure, used in the previous chapter, was converted to a Petri net model. This was completed to allow dependencies between components to be incorporated into the model, to enable real time system level solutions to be obtained from component models and to extend the model to incorporate repeats of the same component so that opportunistic maintenance strategies can be employed.

In the previous chapter, it was assumed that each basic event in the Fault Tree was independent, which enabled the use of Boolean algebra to combine the component failure events following a Fault Tree structure. Variation in the component model results can be present due to the rare nature of some component failures. A high computational effort can be required to reduce this variation in order to achieve convergence for the results of each component failure.

In Chapter 5, the probability of failure for repeated components, such as the isolation valves, was modelled with one Petri net and the result obtained from this was repeated in the Fault Tree structure. To convert a Fault Tree structure, such as that presented in Chapter 5, to a Petri net, can be achieved by duplicating the Petri nets for these repeated components and recreating the logic of the Fault Tree in Petri net format. This is demonstrated in this section.

For the human factors impacting the function of the protection systems, a probability transition can be included in the overall Petri net structure. The probabilities taken from the human factor models, as given in Chapter 5, can then be used as an input to these transitions. For each run of the simulation, the transition is fired immediately and the linked places remain in the resultant marking for the remainder of the run. With sufficient runs of the simulation, this results in a behaviour that approximates the probabilities included in Chapter 5. The probabilities used in these cases, was kept consistent with those applied in Chapter 5.

Figure 6.1 gives a Petri net to combine the component level failures that can result in a lack of detection signal to the control box. The unrevealed and revealed failed states of each component, taken from the component models, are located along the top of the Petri net. These places feed into a place below representing a failure in each component. The meanings of each of these places are annotated on the model. The component failures are combined following the logic from the previous chapter, and using probability transitions to represent the contributions to the system failure by human factors or the fire location. Maintenance actions within the model must remove the marking of the resultant states in this Petri net, and so these places are included in the maintenance reset transition for each of the component Petri net models. Similarly, Figure 6.2 gives a Petri net for an alarm system failure, again following the same structure. Here the probability that there is no detection signal, which was found via the Petri net in Figure 6.1, is included in the model as a contributing factor to the alarm system failure.

Figure 6.3 and Figure 6.4 give the probability that there is no supply from the diesel pump test valve, and the electric pump test valve, respectively. As with the previous Petri nets, the revealed and unrevealed contributing component failures are connected to a place corresponding to either of these failures. A description of the places is given in the figures. The logic from the previous chapter is then used to combine these component failures. In this case, the occurrence of a water mains failure is included. In the previous chapter this was assumed as a constant probability of failure. Here, the occurrence of water mains failure is modelled with a delay time sampled from a uniform distribution, in Transition t55, with an associated delay for the length of time that the failure is likely to occur for, in Transition t76. Transition t76, removes the corresponding system failures and re-marks place P33 to enable further water mains failures.

Figure 6.5 gives a Petri net combining the component failures that result in a low initial pressure in the system. Figure 6.6 gives a Petri net for the combinations of component failures that result in low pressure during water flow of the system.

Figure 6.7 combines component failures and human factors that can result in the deluge valve residing in a failed closed position. There is a probability associated with the lack of manual activation of the deluge system. For each run of the simulation the transition governing this fires once, and the result of either success or failure remains through the run. The probability used in this transition is governed by the human factor Petri net, as in Chapter 5. With multiple simulations this transition approximates the behaviour of the constant probability assigned in the Fault Trees in the previous chapter.

Figure 6.8 gives a Petri net model combining the failures that can result in insufficient flow from the deluge valve. Finally, Figure 6.9 gives a Petri net for insufficient flow from the sprinkler head.

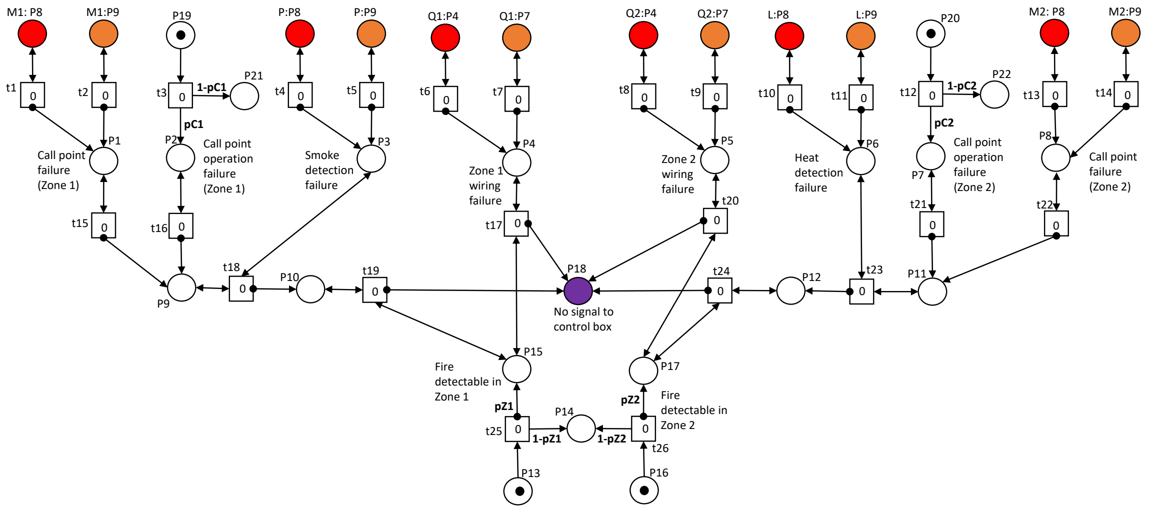


Figure 6.1: A Petri net to combine the failures that result in a lack of signal from the detection systems

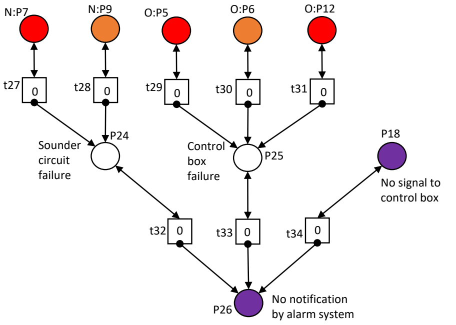


Figure 6.2: A Petri net model for no notification by the alarm system

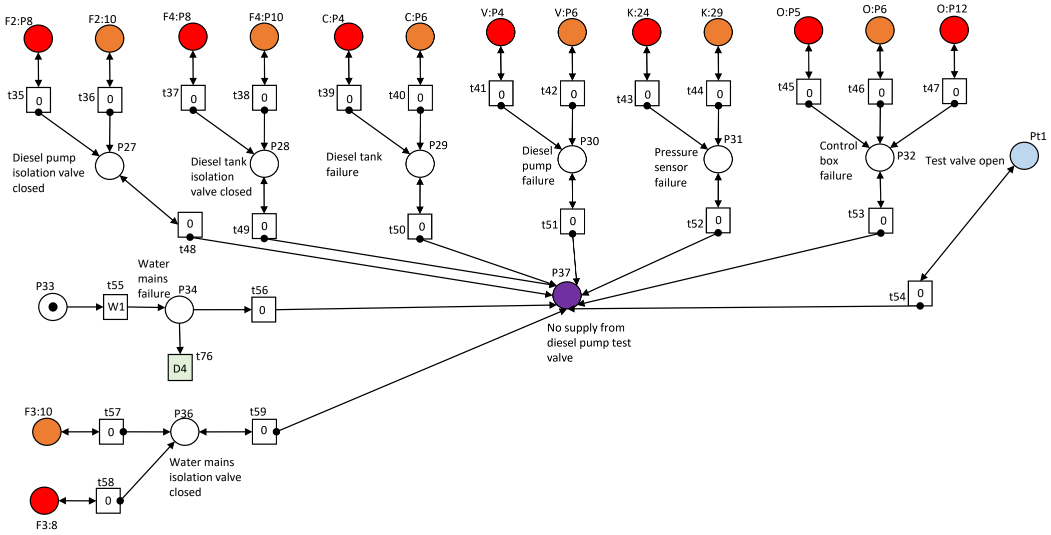


Figure 6.3: A Petri net model for not water supply from the diesel pump test valve

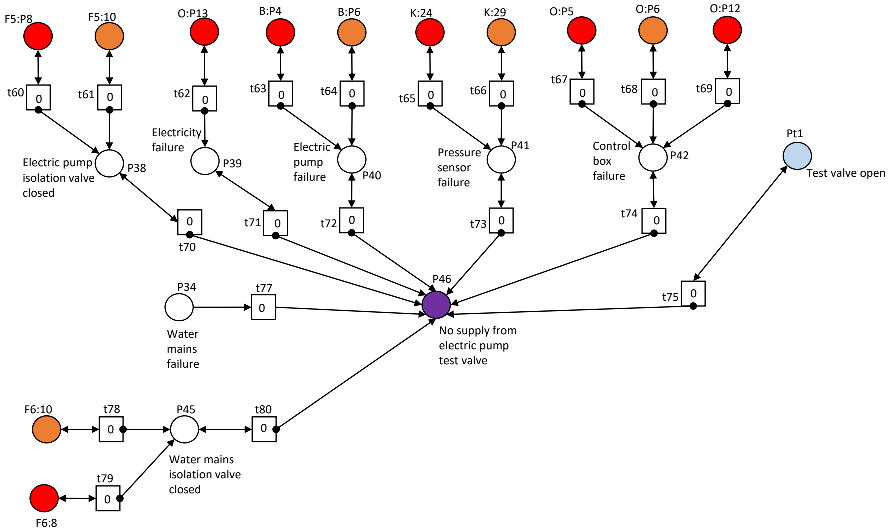


Figure 6.4: A Petri net model for no supply from the electric pump test valve

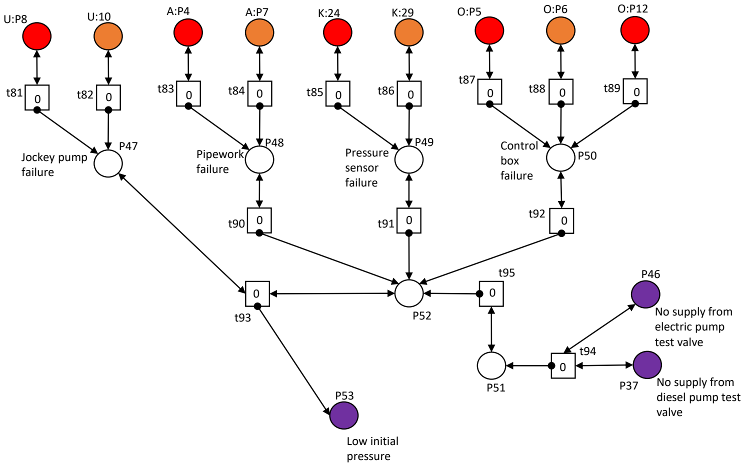


Figure 6.5: A Petri net model for low initial pressure

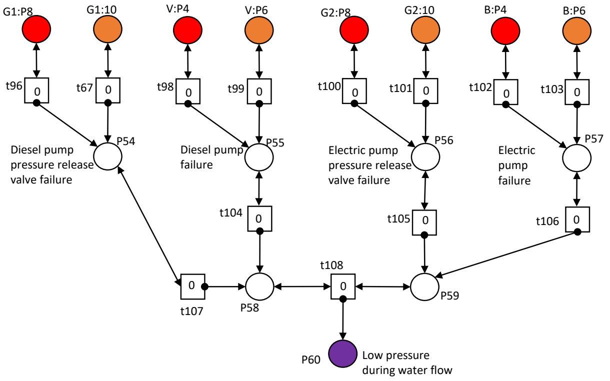


Figure 6.6: A Petri net model for low pressure during water flow

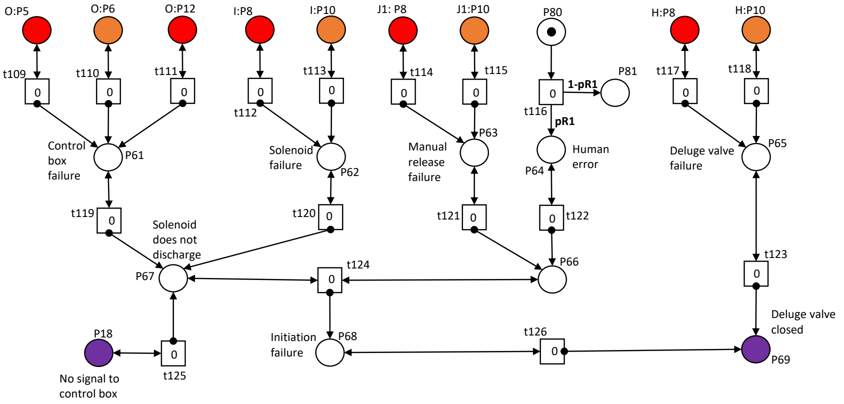


Figure 6.7: A Petri net model for no water flow through the deluge valve

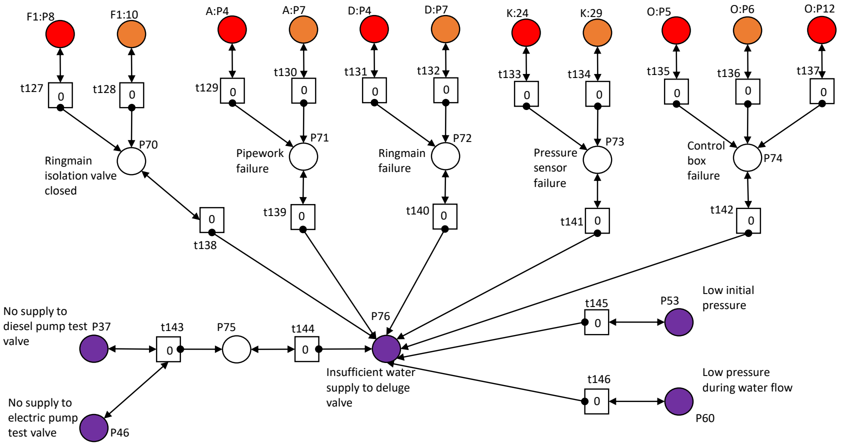


Figure 6.8: A Petri net model for insufficient water flow to the deluge valve

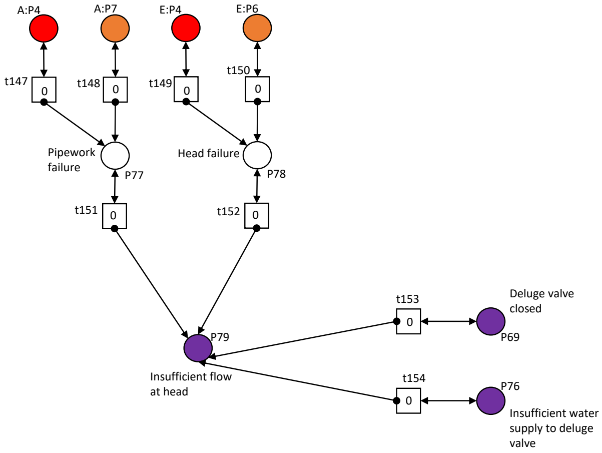


Figure 6.9: A Petri net model for insufficient water flow at the sprinkler head

This overall Petri net structure links the component models together to give the system failure modes. A simulation of this Petri net was carried out. The resulting model was larger in size than the model in Chapter 5, due to the additional transitions and the repeated component models. The structure of this system level Petri net is such that the tokens summarise the behaviour of the tokens in the component models. The resulting model had 560 transitions and 316 places. This simulation took 200599.259 s for 2000 runs.

Figure 6.10 gives the maintenance actions at each time for each component, for the models presented in Chapter 5, combined with the whole system Petri net presented in this section. Figure 6.11 gives the corresponding probabilities that each protection system fails to respond at each time.

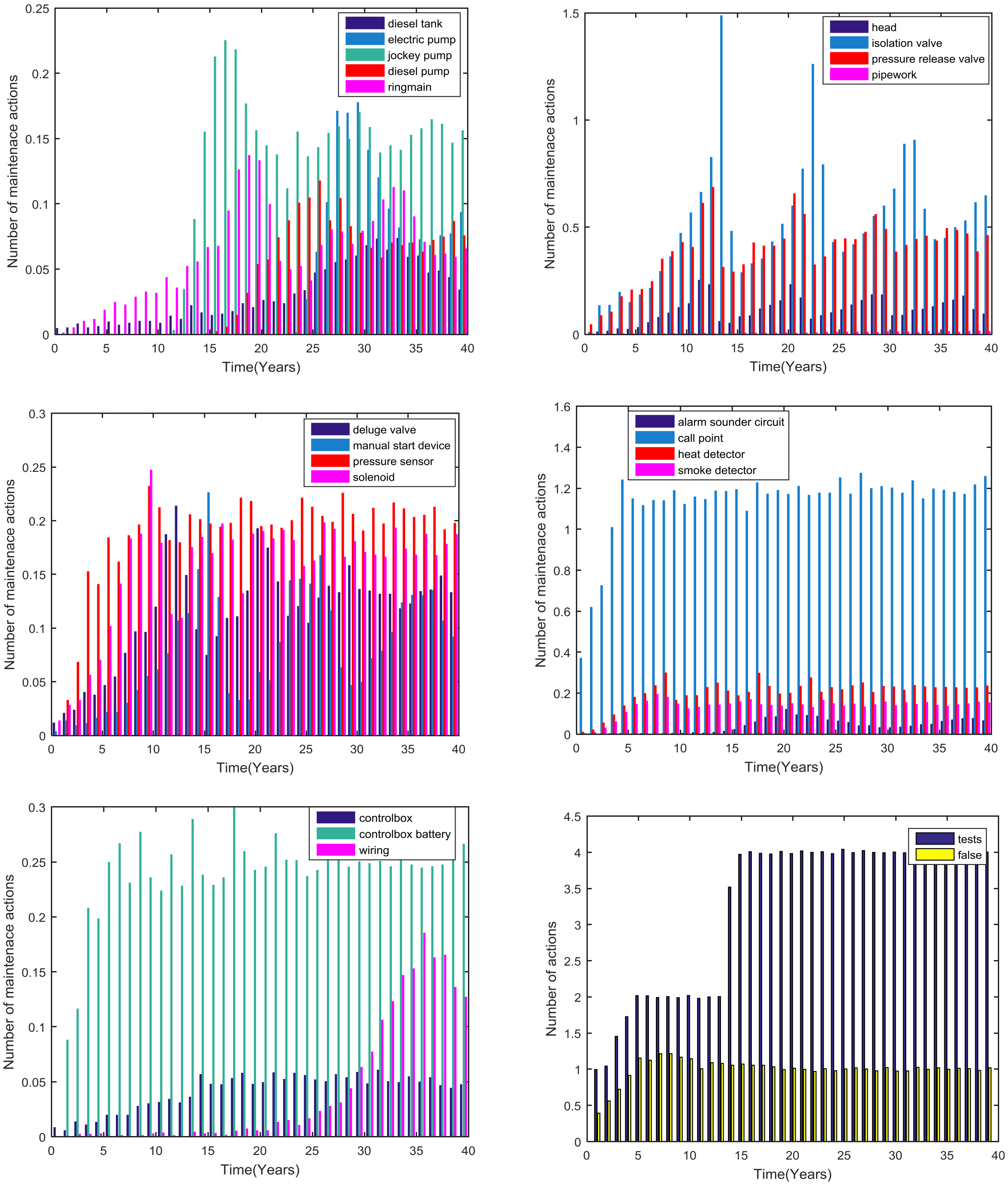


Figure 6.10: The number of maintenance actions for each component for the models presented in Chapter 5, combined with the model representing system logic presented in this chapter

The results following this methodology closely recreate the results seen in the previous chapter. This is as expected due to the same logic present for the combination of events in both of the approaches. There is an increase in the simulation time for this method due to the increased size of the Petri net, however, there are other benefits such as the separate modelling of each of the similar components.

This modelling approach also has the benefit of including all the system logic in the simulated model and hence forms a better facilitating model for the optimization of the asset management strategies. For instance, changes to the system can be analysed immediately during simulation.

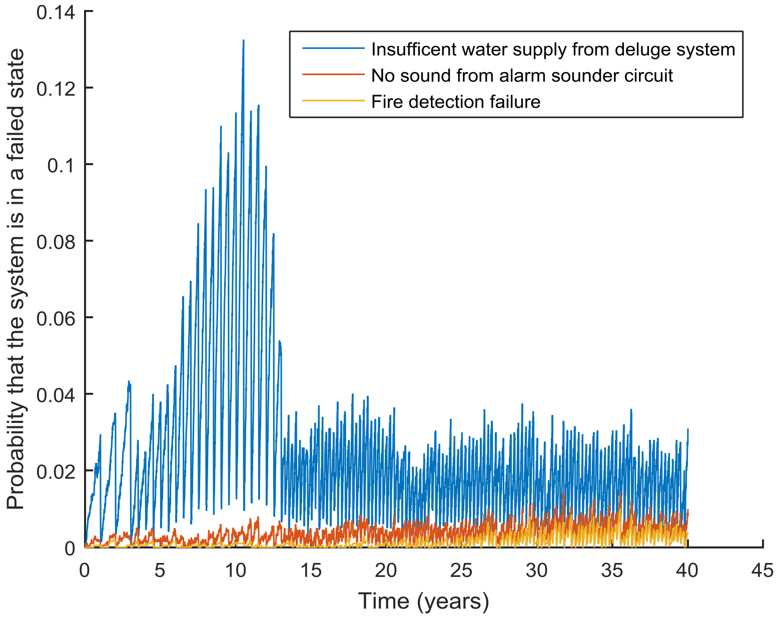


Figure 6.11: The probability that each protection system is in the failed state at each time, from the component models presented in Chapter 5 and the system logic model presented in this chapter

6.3: Optimization of system level asset management strategies

This section presents a methodology for the risk based optimisation of complex ageing systems over their life-cycle. Phased asset management strategies are included to allow the maintenance strategy of the system to evolve as the system ages. This optimization can reduce underground safety risk of the system by optimising the system management over both a system level and a component level, given the system phase.

6.3.2: Method

The optimization methodology proposed in this chapter aims to reduce the life-cycle risk of a system, whilst considering the life-cycle cost. This life-cycle cost can include factors such as: the maintenance cost of the components, the inspection cost of each component, the cost of system testing, a cost assigned to spurious trip occurrence and initial installation cost. The optimisation methodology proposed in this chapter has several stages:

1. Define a phased system level asset management strategy and component level asset management strategy, for each system level phase.
2. Use a Petri net modelling approach to gain system performance metrics, such as life-cycle cost and sub-system failure probability.
3. Model the risk of the system using an Event Tree Framework, ensuring that event branching within the Event Tree Framework is independent.
4. Use a constrained Simulated Annealing algorithm to find the optimal phase entry times for the system level asset management strategy, by considering the risk and life-cycle cost of the system.
5. Use a Genetic Algorithm to optimise the component level asset management strategy, given the system level phases found in the previous step, by considering the risk and life-cycle cost of the system.

6. Interpret the results to give an optimal strategy for the asset management of the system over the life-cycle under consideration.

At each optimisation state, the parameters within the Petri net are updated within the optimisation framework in use.

For the first stage of the optimisation procedure a Simulated Annealing algorithm was selected because the method avoids local minima, generally gives a good solution for arbitrary cost functions and can handle constraints on the optimisation. There is also statistical guarantee of global convergence to the optimal point. Simulated Annealing algorithms are suited to low dimension optimisation problems, however the method becomes increasingly computationally expensive with higher dimensions. For this reason a Simulated Annealing algorithm is suggested for the first stage of the optimisation method presented in this chapter, where there is a two dimensional search space. In addition, the algorithm requires minimal assumptions about the problem under investigation and does not implement gradient decent methods, which makes it suitable for optimisation of complex systems, which may have a complex cost function landscape.

For the second stage of the optimisation a Genetic Algorithm was selected because the methodology can search a number of potential solutions. In the second stage of the optimisation methodology presented in this chapter there are a number of inspection intervals for every component in the model. Hence, there are a large number of potential combinations, with limited information about the relationship between the cost and benefit of changing each parameter. The Genetic Algorithm allows the exploration of numerous points in this high dimensional problem. Typically Genetic Algorithms perform well in noisy environments, such as the one implemented for the second stage of the optimisation methodology presented here.

Both methods were combined in this approach as the Simulated Annealing algorithm reduces the number of parameters required for tuning the Genetic Algorithm, increasing the efficiency. In addition, the most crucial part of the optimisation, which gives the maintenance phase times for the system is performed by the Simulated Annealing algorithm, which has a strong ability to avoid local minima.

6.3.3: Optimisation tools applied in this approach

The Simulated Annealing Algorithm

The Simulated Annealing algorithm works by defining what is known as a cooling schedule where temperatures start high and reduce through the algorithm [165]. Here, solutions are proposed moving away from a starting point. For each temperature, different configurations of the models are explored by taking a step away from the previous configuration. Initially, the higher temperature allows more configurations that do not show an improvement to be accepted. This prevents the algorithm from becoming stuck, in a local optimal solution, and allows it to search more of the solution space with the aim of finding the global optimal solution. At each temperature the best configuration is stored and forms the starting point for the next temperature. As the temperature reduces, it is less likely that a solution will be accepted if it does not give an improvement. This causes the algorithm to condense to an optimal region.

In the proposed methodology the Simulated Annealing algorithm, used in stage 4 of the optimisation procedure, is allowed to explore some solutions that are outside of the maximum constraint applied to the total system life-cycle cost. This is done to allow the algorithm to fully explore the solution space. However, these solutions are only used as in interim step during the algorithm and are not recorded as the best solutions available.

In this optimisation method, the average risk of the system over a defined time period is used to allow quantification of the risk of the system. This time period can be set to the system lifetime allow the allocation of resources to the system in an optimal way over its whole lifetime, as opposed to only considering the immediate future. If an optimisation over a shorter time is required, then the timeframe over which the optimisation is performed can be reduced to the desired timeframe. Risk values at given points, instead of averages, are not used in the optimisation framework, as this can lead to non-representative results. For example, if a risk value used for optimisation happens to lie in a trough in the predicted risk values over time, an optimisation can be performed that ignores the surrounding higher values.

To apply a Simulated Annealing algorithm, to find the optimal solutions, the following must be established:

1) The energy of the system must be defined, in this case it is the average risk of the system over the simulation time period.
2) Either a maximum or minimum valve for the energy can be found, in this case the algorithm seeks to find the minimum energy value corresponding to the lowest risk.
3) A method to alter the configuration of the system must be defined, in this case the configuration changes by altering the maintenance phase entry times of the system.
4) The temperature and initial configuration must be set. In this case, the initial configuration is set to give a maximum risk and the temperature is set to decrease incrementally to give a sufficient cooling schedule for the Simulated Annealing algorithm.

A pseudocode for an unconstrained Simulated Annealing algorithm is given below:

Algorithm 1: Pseudocode implementation for Simulated Annealing

```
Inputs
$T_0$: Initial temperature
$J$: 'Cost' function, in this case the average risk of the system
$s_0$: Initial configuration
$\{T_1, T_2, \ldots, T_i, \ldots, T_N\}$: Cooling schedule for $N$ temperatures
$n(s_k)$: A function to find the neighbour of configuration $s_k$
Algorithm
for $j = 1, \ldots N$ do
    $T = T_j$
    $s_{prev} = s_{best}$ (or $s_0$ in first initialisation)
    for $k = 1, \ldots M$ do
        $s_k = n(s_{prev})$
        $\Delta E = J(s_k) - J(s_{prev})$
        if $\Delta E < 0$ then
            $s_{prev} = s_k$
            if $J(s_k) < J(s_{best})$
                $s_{best} = s_k$
            end if
        else if $(e^{-\Delta E}/_T > rand(0,1))$ then
            $s_{prev} = s_k$
        end if
    end for
end for
return $s_{best}$
```

Constraints can be added to this algorithm. In this case, a constraint on the budget for the system installation, maintenance and inspections can be included. This physical cost is not to be confused with the 'cost' function of the Simulated Annealing algorithm. Where a constraint is added, the proposed configuration at each point is accepted if the value is within a defined threshold for the constraint function. The 'best' configuration is only accepted if it is within the final budget for the constraint function. The constrained Simulated Annealing algorithm applied in this chapter allows the configuration to move outside of the budget during the algorithm, by a small quantity. This allows configurations to be reached that may require the budget to first move outside of the limit in order to find a global optimum solution that is within the budget constraint.

A pseudocode for a constrained Simulated Annealing algorithm is given below [166]:

Algorithm 2: Pseudocode implementation for constrained Simulated Annealing

```
Inputs
$T_0$: Initial temperature
$J$: 'Cost' function, in this case the average risk of the system
$s_0$: Initial configuration
$\{T_1, T_2, \dots, T_i, \dots, T_N\}$: Cooling schedule for $N$ temperatures
$n(s_k)$: A function to find the neighbour of configuration $s_k$
$C$: Constraint function
$B_{max}$: Maximum budget
$\delta_b$: Threshold for the budget
Algorithm
for $j = 1, \dots N$ do
    $T = T_j$
    $s_{prev} = s_{best}$ (or $s_0$ in first initialisation)
    for $k = 1, \dots M$ do
        $s_k = n(s_{prev})$
        $\Delta E = J(s_k) - J(s_{prev})$
        $B = C(s_k)$
        if ($\Delta E < 0\ AND\ B \leq B_{max} + \delta_b$) then
            $s_{prev} = s_k$
            if $J(s_k) < J(s_{best})\ AND\ B \leq B_{max}$ then
                $s_{best} = s_k$
            end if
        else if ($e^{-\Delta E/T} > rand(0,1)$) then
            $s_{prev} = s_k$
        end if
    end for
end for
return $s_{best}$
```

The Genetic Algorithm

Genetic Algorithms were proposed in 1975 in application to an optimization problem with the initial framework of Binary-coded genetic algorithms demonstrated in 1989 [167] [168]. Genetic algorithms can be more efficient than conventional searching algorithms and can be applied to solve multi-objective optimization problems.

A Genetic Algorithm begins with an initial population [169]. The strength of each of these is then assessed with the fittest individuals conserved in each level of the algorithm. These fittest individuals are then combined to give other potential solutions. The Genetic algorithm is applied for the component level asset management strategies, given the fixed system phases found with the Simulated Annealing algorithm.

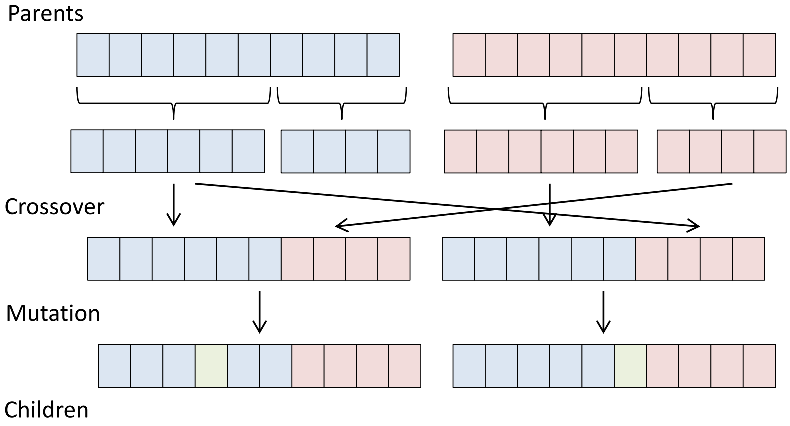
A Pseudocode of a Genetic Algorithm is given below:

```
**Algorithm 3:** Pseudocode implementation of a Genetic Algorithm
Inputs
$P_0 = \{p_1^0,\ p_2^0 \dots, p_N^0\}$: The initial population of $N$ vectors, where $p_n^0$ is a vector of length $L$
$J$: Fitness function, in this case the risk of the system and cost of interventions calculated via Monte Carlo Simulation of the Petri net model
$I$: Number of iterations
$S(P)$: Selection operator, based on the fitness of the solution
$X(p_a, p_b)$: Crossover operator
$M(P)$: Mutation operator
Algorithm
for $j = 0, \dots (I-1)$ do
        $P = P_j$ (set the population at each iteration)
        for $k = 0, \dots N$ do
                $F_k = J(p_k)$
        end for
        $P_{parent}$= $S(P)$, where R is the number of parents following selection
        $P_{j+1} \leftarrow P_{parent}$
        for $q = 0, \dots R/2$ do
        $p_{child\ 1}, p_{child\ 2}$= $X(p_a, p_b)$
        (Where $p_a, p_b$ are vectors from the parent population and each parent are is only used once)
        $P_{j+1} \leftarrow p_{child\ 1}, p_{child\ 2}$
        end for
        $P_{j+1} = M(P_{j+1})$
end for
return $P = P_{j+1}$
```

At each level of the algorithm the weakest 50% are removed and replaced by a new generation of solutions, which are found by combining strong members of the current generation, known as parents. The members of the population selected to form parents in the new generation are chosen based on their fitness. The selection of parents for the population is weighted by the fitness of the parent using a roulette wheel selection. Here, parents are selected based on their fitness with a higher probability of selecting parents that have a lower value of risk and cost.

Following selection of two parents, they are cross bred. A random number generator is used to select a point in each pair of parent parameter vectors for this mixing. The parent parameter vectors are split at this point. The two parents are mixed such that one child is formed with the first part of the first parent and the second part of the second parent and a second child is formed with the second part of the first parent and the first part of the second parent. The Petri net is simulated for each of the child vectors to find the risk and the cost for each. This is repeated for further parent pairs until enough child vectors are produced to replace the discarded population from the previous level. The resulting child population members are combined with the top 50% of parents, which had the lowest risk from the previous level. This process is repeated until a convergent population is found.

Mutations can be added to the Genetic Algorithm. Here, values within the child parameter vectors are switched at random prior to simulation of the Petri net. Figure 6.12 gives a representation of the chromosomes with the Genetic Algorithm and demonstrates the crossover and mutation processes. In this application of a Genetic Algorithm, each chromosome entry contains a value for the input parameters to the model, for instance, the inspection interval of an individual component.



6.12: A diagram of the crossover and mutation of chromosomes within a Genetic Algorithm

Hyper-parameters within the optimisation methodology can be tuned in order to explore the solution space in the most efficient manner. The aim of this tuning methodology is to balance the efficient convergence of the model, with a sufficient exploration of the solution space to prevent the algorithm from becoming stuck in a local minimum. In the application presented in this chapter a 50% selection of parents from the population is implemented, along with a 1% mutation rate, these values demonstrated a good convergence to an optimal solution, for a population of 100 chromosomes. Decreasing the percentage of selected parents, or the mutation rate, increases the likelihood of the algorithm locating a local, instead of global, minimum. Increasing the percentage of parents selected, or the mutation rate, can result in a poor convergence of the model. The tuning of these hyper-parameters should be considered when applying this methodology to new examples.

Within the Genetic Algorithm optimisation, parents are selected at each generation based on the reciprocal of their fitness value. A schematic to demonstrate this method for a population of 4 individuals is given in Figure 6.13; for sample data values selected from the model given to 3.s.f in Table 6.1, where X is a random number in [0,1].

Population member ID number	F	$^1/_F$	Normalised weight	Range for selection
1	2.79	0.358	0.412	$0 \leq X < 0.412$
2	3.37	0.297	0.341	$0.412 \leq X < 0.753$
3	5.59	0.179	0.206	$0.753 \leq X < 0.959$
4	22.7	0.0361	0.0415	$0.959 \leq X$

Table 6.1: Sample data values to demonstrate the sampling method used in the application of a Genetic Algorithm in this chapter

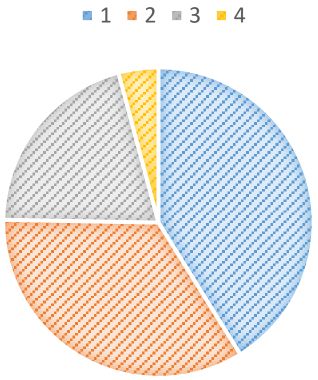


Figure 6.13: Weighted selection sampling of parents within the Genetic Algorithm for sample values given in Table 2 where each entry corresponds to the population member ID number

Here, it can be seen that population member 1 has the highest probability of being selected, corresponding to the largest proportion of the roulette wheel and range for selection. This arises as population member 1 also has the lowest value for the objective function meaning that it is the most desirable parent, since the optimisation aims to minimise the objective function. Likewise, population member 4 has the highest value for the objective function, and hence is the least desirable parent. This selection method assigns a lower probability of selection to this population member, represented by a lower proportion of the roulette wheel and range for selection. This method is generalised to the population of 100 members in the optimisation methodology such that the more desirable parents, with a lower objective function, are more likely to be selected. For every crossover operation, the parents are selected via this method independently of any previous selections. This means that the fittest individuals are likely to form members of the crossover pairs multiple times in the same level of the algorithm.

This sampling method results in a faster convergence to an optimal solution in comparison to uniform parent selection. This is because parents with a lower fitness function are used more frequently in the creation of the next generation. For the model presented in this chapter, twenty generations of the Genetic algorithm were applied.

6.3.4: Sample Application

A sample application of the optimisation methodology was applied to the fire protection system models developed in Chapter 5 of this thesis and Section 6.2 of this chapter.

<u>Stage 1: Definition of asset management strategies for optimisation</u>

The phased system level asset management framework, given in Chapter 5, was used for the sample application of this optimisation methodology. Here, there are three possible system level maintenance phases. Each phase has a different set of maintenance strategies that are applied to components across the system. Each phase also has a different system testing frequency.

The component level asset management strategy used for this sample application assumes that the maintenance intervals for each component are determined by the system level maintenance phase. The component inspection intervals within each phase are allowed to vary in order to optimise the management of the system. It is possible to extend this definition in order to optimise the maintenance intervals within each phase, of every component of the system. However, this increases the computational cost of the optimisation methodology.

Stage 2: Develop a Petri net to gain system metrics:

The system metrics chosen for the optimisation of the asset management strategies, were the average risk of fatalities of the system over the life-cycle and total-lifecycle cost. The total lifecycle cost included the cost of each component maintenance and inspection action, the cost of a system test, the cost of initial installation and the cost of false system activation.

The Petri nets developed in Chapter 5 of this thesis and Section 6.2 of this chapter were used to gain system metrics for this implementation of the optimisation methodology. The model inputs were kept consistent with this earlier work. Some changes were made to the models to give independence in the sub-system failure probabilities, this is described further in Stage 3 of this section.

Sample costs were assigned to the system initiation and the subsequent maintenance and inspection actions, to allow the methodology to be demonstrated. An initial full system cost of 100,000 units is assigned within the model, this represents the initial cost of the system. The average costs assigned to each of the inspection and maintenance actions is given in Table 6.2.

Intervention	**Average maintenance cost (arbitrary units)**
Pipework intervention	3000
Pipework inspection	100
Electric water pump intervention	1000
Electric water pump inspection	50
Diesel water pump intervention	1000
Diesel water pump inspection	50
Jockey pump intervention	1000
Jockey pump inspection	50
Diesel tank intervention	500
Diesel tank inspection	50
Ring main intervention	2000
Ring main inspection	75
Sprinkler head intervention	200
Sprinkler head inspection	25
Isolation valve intervention	400
Isolation valve inspection	75
Pressure release valve intervention	400
Pressure release valve inspection	75
Deluge valve intervention	600
Deluge valve inspection	75
Solenoid intervention	300

Solenoid inspection	75
Manual release mechanism intervention	300
Manual release mechanism inspection	25
Pressure sensor intervention	800
Pressure sensor inspection	75
Smoke detector intervention	1500
Smoke detector inspection	25
Heat detector intervention	1500
Heat detector inspection	25
Wiring intervention	3000
Wiring inspection	100
Call point intervention	100
Call point inspection	25
Alarm intervention	800
Alarm inspection	50
Control box intervention	1500
Control box inspection	50
Control box battery intervention	200
Control box battery inspection	25
System testing by opening the test valve	100
False activation	500

Table 6.2: The average maintenance and inspection costs assigned to components in the model

Stage 3: The Event Tree Framework for risk calculation

The risk calculation, over which the optimizations take place, follows that of an Event Tree structure, as shown in Figure 6.14. The risk of the system is calculated for the combined probabilities of each protection system either failing, or working, the estimated frequency of fire occurring and the estimated consequences. The aim of both levels of the optimization are to reduce this risk. The models presented in the earlier section of this chapter combined with the component models presented in Chapter 5 give rise to the probability that each of the fire protection systems are in the failed or working state.

In this chapter the method is illustrated with sample consequence and fire frequency values. These illustrative values are assigned to the Event Tree in Figure 6.14. The methodology demonstrated in this section can be easily generalised to any fire frequency and consequence values by simply adjusting the parameter inputs to the model. The sample application is incorporated into each stage for demonstration.

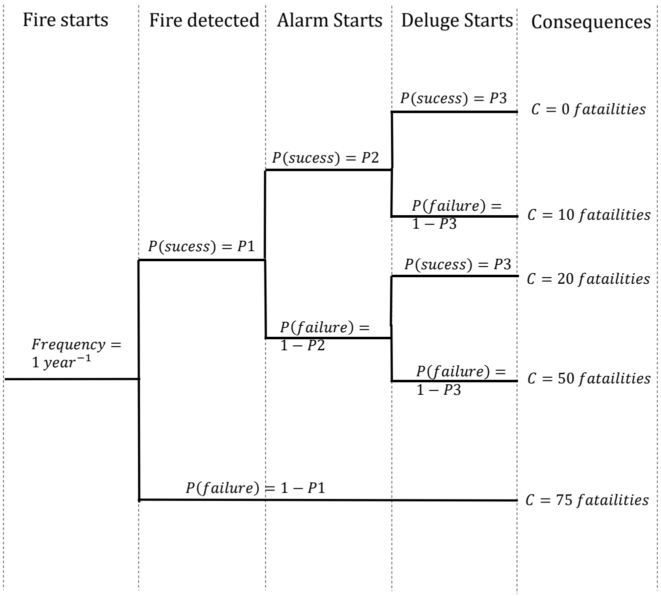


Figure 6.14: An Event Tree for the fire protection systems

In order to apply this method, the branches of the Event Tree must be independent. Up to this point, the probability of a deluge system failure, or an alarm system failure, have been modelled as dependent on the condition of the detection system. The full system Petri net model can be altered to give independent branches of the event tree. This is done by removing the contribution of the detection system to the failure of the deluge and alarm systems, by removing transition t34 and t125 in the full system Petri net model, present in Figure 6.2 and Figure 6.7.

In the Event Tree it is assumed that a lack of fire detection will result in a failure of both the alarm and deluge system and a high level of consequences. The probability for this can be found by tracking the marking of place P18 in Figure 6.1. For the remaining branches it is assumed that the detection system is in the working state and so the fire is detected, and hence cannot contribute to the alarm or deluge system failure. These models can then be used to give the probability of alarm or deluge system failure by recording the marking of place P26 and place P79, in Figure 6.2 and Figure 6.9 respectively, given that the detection system is in the working state.

Incorporated into the state of both the alarm system and deluge system, is the state of the control box. If there is a control box failure, then there will be a failure of both systems and hence a dependency in the branching of the Event Tree. To adjust for this dependency, the control box failure is extracted in the Event Tree structure to form an independent branch. This gives independent branching in the Event Tree for the Petri net models presented in this chapter. This corresponds to removing the transitions t33, t53, t74, t92, t119 and t142, in the full system Petri net model presented in this chapter, in order to give the probability that the alarm and deluge system are in the working, or failed state, given that the control box is in the working state. The marking of places P25, P32, P42, P50, P61, and P74 in the Petri net model can be used to give the probability that the Control box is in the failed state. However, due to these duplicate places that no longer impact the Petri net model, it is advisable to retain one place and corresponding transitions, to contribute to the solution and then remove the duplications to improve efficiency. In both cases, in these Petri net models, if the control box is in the failed state then this leads to a failure of both the alarm and deluge systems and this is incorporated into the Event Tree structure. This is given in Figure 6.15. Following this, there are no further direct dependencies between the branches in the Event Tree.

Hence, the risk of the system at any time for the illustrative values given in this chapter is defined as in Equation 6.1, where the probabilities of system failure at each point can be taken from the Petri net models developed in this chapter and Chapter 5. These values can be adjusted if required prior to applying the optimization methodology.

$$R = 10P1P_{CB}P2(1-P3) + 20P1P_{CB}(1-P2)P3 + 50P1P_{CB}(1-P2)(1-P3) + 50P1(1-P_{CB)} + 75(1-P1) \qquad \textit{(6.1)}$$

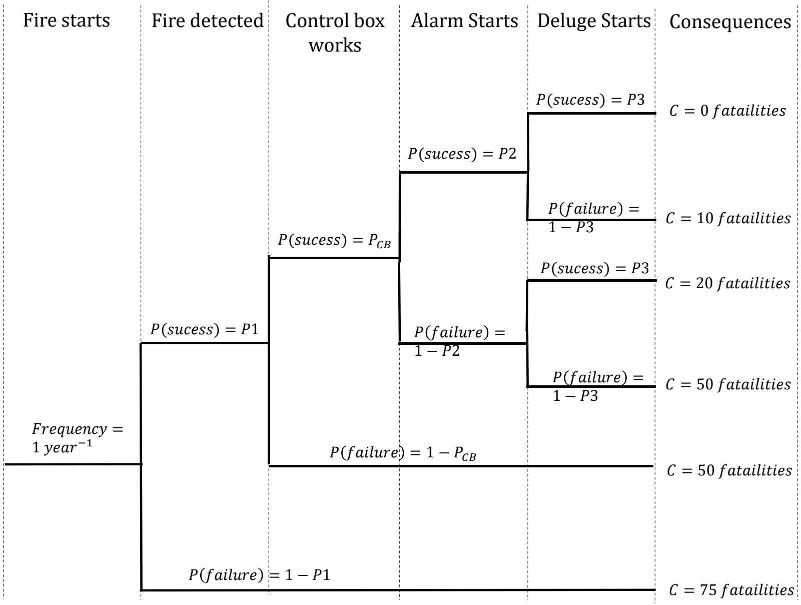


Figure 6.15: An Event Tree for the system with independent branching

The optimization methods used here aim to reduce the risk over the whole lifecycle of the system. To enable this, the average risk for the system is calculated and used as the basis of optimization. This is found by calculating the risk at each time via the method described in Equation 6.1 and taking the average over the 40 year time period. However, it is trivial to switch this value with the risk at a certain point in time by selecting the probability that each component fails at a given time as opposed to finding the average.

Stage 4: Optimising a phased system level asset management strategy

The first level of the optimisation method, applied in this case, aims to find the best time at a system level for the entry to the second or third maintenance phase of the system. These phases dictate when routine and early replacement strategies begin to be employed for each component. To find the optimal solution for these phases a Simulated Annealing algorithm has been implemented. This was chosen due to a relatively smooth relationship between the entry of each phase and the risk of the system. For example, if there is a longer time between the system entering the second phase from the first phase, then there will be fewer preventative maintenance options, a higher probability of component failure and a higher risk to the system. The Simulated Annealing algorithm is also quick to converge and works well with the small number of parameters, such as those sought to be optimized at this stage.

The phase times possible, during the optimization, ranged in one yearly intervals between zero and forty years. The Simulated Annealing algorithm was set up to find the lowest possible risk given a cost constraint. The life time system cost is made up of the cost of each intervention and the initial cost of the system and this is constrained within the algorithm. The algorithm returns a near optimal time for the phase entry times.

In the initial stage of the optimisation, the inspection intervals for each component were set to: 12 months for the first phase, 6 months for the second phase and 3 months for the third phase. Within the first stage of the optimization, with the use of the Simulated Annealing algorithm, a start point is set such that the system remains in the first phase for the full simulation time period. This corresponds to a situation with minimal inspection and no preventative maintenance. Two parameters are optimized in this stage, corresponding to the mean of each of the phase change transitions that govern when the system enters each maintenance phase. At each point the parameters are changed by a fraction of the total lifetime in consideration. One parameter is selected at random and this is changed to give a new potential solution to the algorithm at each trial. A constraint is added on the life time system cost and the risk of the system over the simulation time period is optimized. The mean of the phase transitions is found to the nearest year.

A constraint is applied to the algorithm whereby if the total cost over the systems life goes above a defined threshold value, then the solution is deemed impossible. The algorithm optimizes for the reduction of risk of the system which is calculated via the Event Tree in Figure 6.14.

It is easy to switch the objective function to optimize for the risk at a certain time, or within a certain time period. This can be done by finding the average probability of system failures within a small range or at a point in the system lifetime, as opposed to the average over the simulation period.

In this application, 6 temperatures of the Simulated Annealing algorithm are applied in this case with 20 trial values within each temperature. The algorithm searches for the two best maintenance phase entry times, given within a 40 year period with a value to the nearest year for each. The neighbouring configuration is defined adaptively based on the temperature of the algorithm. Initially, steps of 10 years are taken in both parameter values at the first temperature. At the second temperature, steps of 5 years are taken in both parameter values. At the third temperature, steps of 3 years are taken in the parameter values and at the fourth and fifth temperature steps of 2 years are taken. Finally, at the sixth temperature, steps of 1 year are taken. Each parameter is initiated at the highest value of 40 years and decreased at each point. The new proposed values within each simulation are found by changing one parameter at a time, with a random selection made at each point.

A reduction in the maintenance phase entry times reduces the risk but increases the cost of interventions. There is a 5% variation assigned in the intervention cost constraint, to allow the algorithm to explore slightly outside the maximum cost within each temperature only. The maximum cost constraint, used as a sample value for illustration of the method, was 400000 cost units. The value of each of the parameters, within each temperature, that results in the lowest risk value and has system cost within the constraint, are carried through at each temperature. In the next temperature, these values are again reduced to try and find a lower risk value within the cost constraint.

The Simulated Annealing algorithm applied first resulted in a phase entry time of 12 years for the second phase of the system and an entry time of 30 years for the third phase of the system, following entry into the second phase of the system. This implies an entry time into the third system phase at 42 years after installation. Figure 6.16 shows how the phase entry intervals evolved with each iteration of the Simulated Annealing algorithm. It can be seen that after the 6th iteration that no new solution was proposed inside the algorithm that contributed both a reduction in risk and was within the constrained budget. This is demonstrated through the lack of change in the final four iterations of the algorithm.

This optimisation shows that for the constrained budget applied and within the 40 year period under consideration, that the system should not enter the third system phase. This suggests that the early replacement of components, although shown earlier in this chapter to reduce the probability of system failure, is not an efficient method for the control of risk in this case, under a constrained budget.

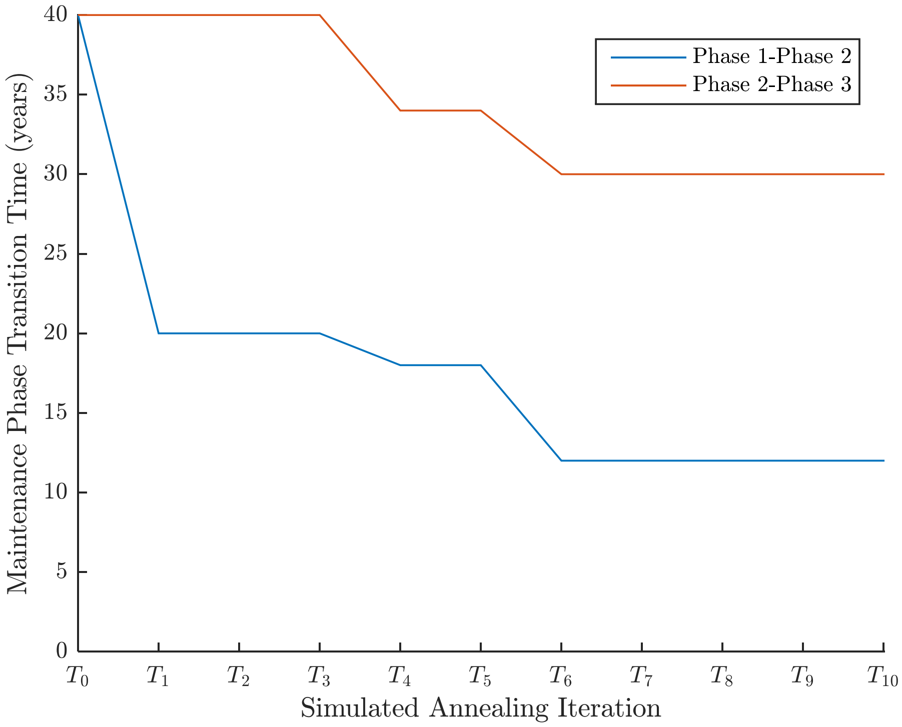


Figure 6.16: The evolution of the maintenance phase entry times for each iteration of the constrained Simulated Annealing algorithm

Stage 5: Optimisation of the component level asset management stategy

The second level of the optimisation method, applied in this case, aims to find inspection intervals for each component within each phase of the system. This was completed to give a further reduction in risk for the resources available.

The inspection intervals were allowed to vary between three months and twelve months, in 3-monthly intervals. An objective function was defined for the optimisation incorporating the system life-cycle cost and the average risk over the system life-cycle. The Genetic Algorithm returns the best inspection intervals within each phase for every component in the model.

Following this, a Genetic Algorithm is applied to optimize the inspection intervals within the optimal solution for the maintenance phase entry times of the system. Throughout this optimization the maintenance delays for each maintenance type are fixed for each component in the system. These are based on the estimated state of each component as in Chapter 5.

In this application of the Genetic Algorithm an initial population is defined for each of the inspection intervals, for every phase of each component. Each member of the initial population has randomly assigned inspection intervals for each component, within each phase, with values of 3 months, 6 months, 9 months or 12 months. The parameters governing these intervals form a vector. For each member of the initial population a simulation of the Petri net is completed and the risk and the cost of the interventions are calculated. In this application the fitness of population members is defined by a function combining the average risk and the cost of the system over the 40 year period. Mutations were included at a rate of 1 in 100.

For this algorithm the initial population of 100 members was defined. The inspection interval for each component was allowed to have the values of 3 months, 6 months, 9 month or 12 months. The algorithm aims to minimise the risk and the cost. The objective function of this is given in Equation 6.2, the algorithm seeks to minimise this function.

$$F = R_{av} + \frac{C_I}{200{,}000} \tag{6.2}$$

Where F is the objection function, R_{av} is the average risk of the system and C_I is the cost of the system over the simulation time period.

The optimisation procedure across both the Simulated Annealing algorithm and Genetic Algorithm took 2,189,016.117 seconds to complete.

Figure 6.17 gives the convergence of the average risk, over the simulation period, to the optimum value, found via the Genetic Algorithm. Each entry denotes the average risk of a member of the population in each generation. Figure 6.18 gives the life-cycle cost at each generation of the algorithm, where each entry denotes the total cost of the system of the population member in each generation. These results demonstrate an initial higher rate of general decrease in the risk predicted by members of the population with each generation. After generation 14 there is limited change in the mean of the average risk predicted by the population. The mean of the population for the life cycle cost shows an initial increase followed by limited change after Generation 7. This suggests that initially the decrease in the risk can be correlated with an increase in lifecycle cost, however after Generation 7 there is still a decrease in average life cycle risk of the population, despite limited increase in cost. This implies that the algorithm is allocating resources in a more efficient way in order to reduce risk at the same life cycle cost.

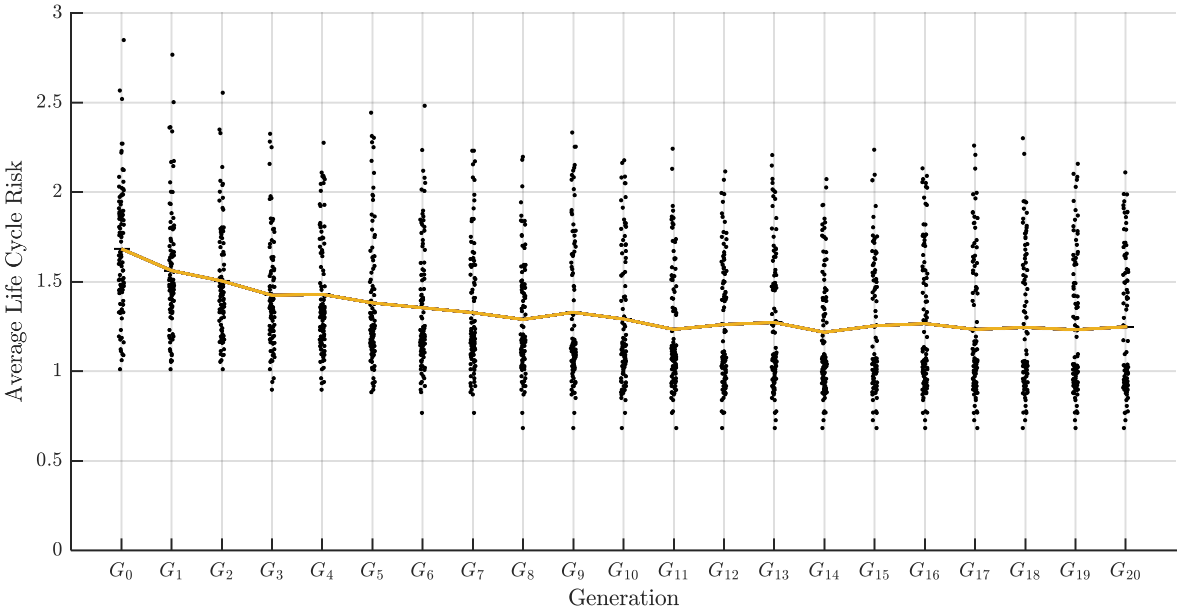


Figure 6.17: The risk of the system with each generation of the Genetic Algorithm

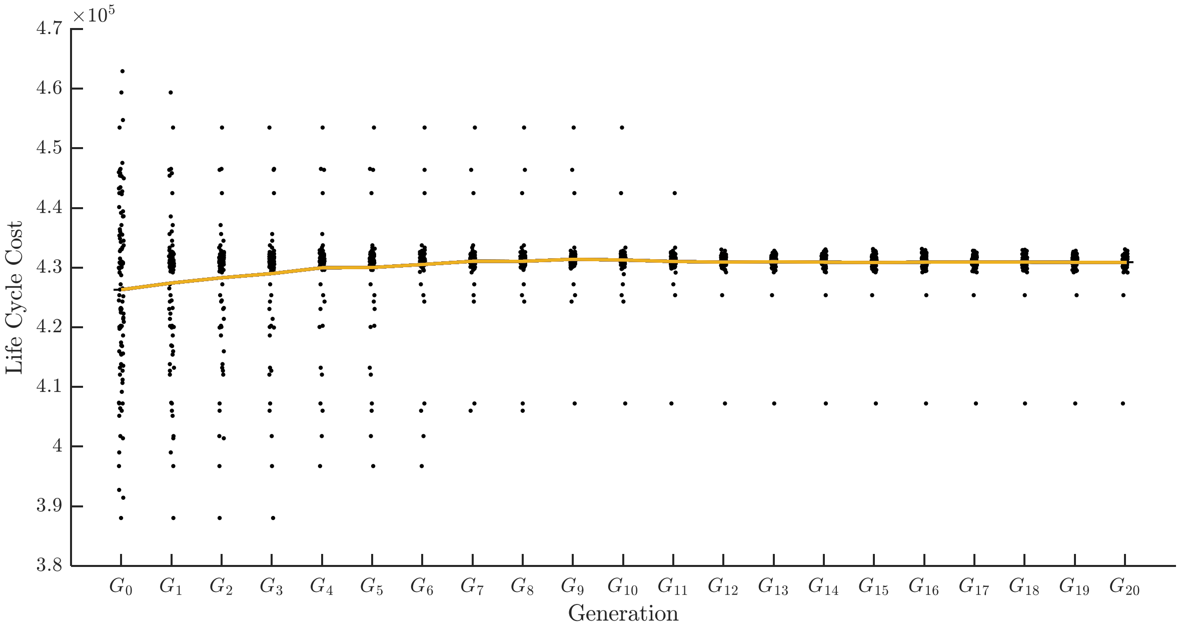


Figure 6.18: The life-cycle cost of the system with each generation of the Genetic Algorithm

Figure 6.19 shows the median, 25th percentile and 75th percentile for average life cycle risk the population over each generation of the Genetic Algorithm. Figure 6.20 shows the median, 25th percentile and 75th percentile for the life cycle cost of the system at each generation of the Genetic Algorithm. The same patterns of results can be seen as in Figure 6.17 and Figure 6.18. There is some instability in the 75th percentile for the average life cycle risk. This can be attributed to the contribution of rare events to the simulation of each member in the population. Namely, the contribution to risk from the rarer failure of the detection system, which also has a higher consequence level associated with it. For example, in some simulations of each population member, this rarer event may not occur, while in others it may occur. This instability could be removed with a higher number of simulations of each population member within the Genetic Algorithm, or with an improved method for rare event simulation of the model. Including a higher number of simulations in each population member of this Genetic Algorithm, with the current model architecture, increases the computational cost. With the large computational cost of this algorithm, with a reduced number of simulations, this suggests that more efficient methods should be developed for more accurate optimisation results.

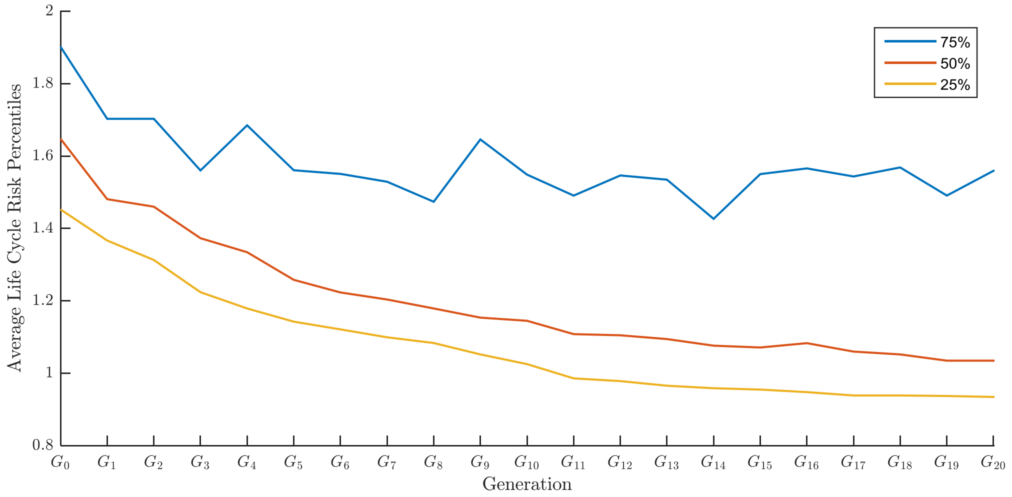


Figure 6.19: A graph showing the median, 25th percentile and 75th percentile for the average life cycle risk for each generation of the Genetic Algorithm

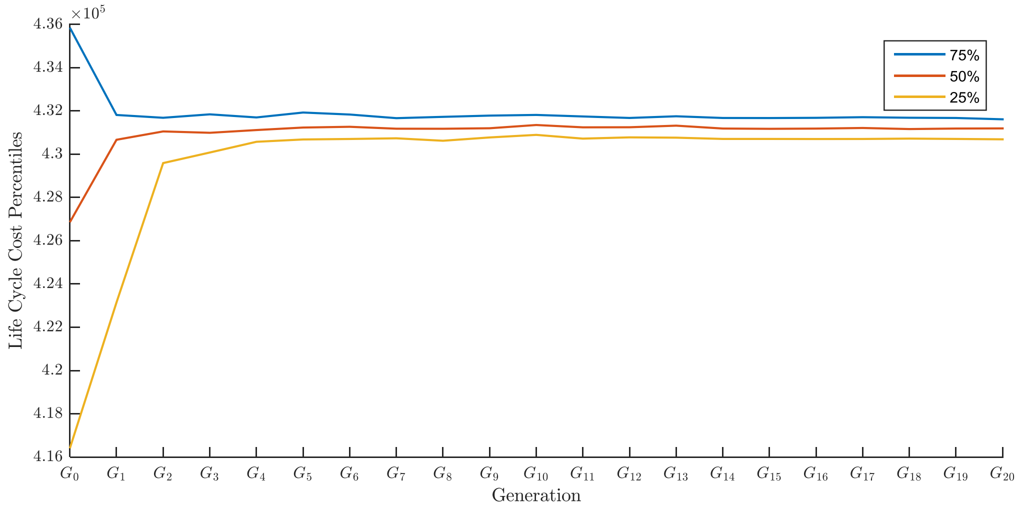


Figure 6.20: A graph showing the median, 25th percentile and 75th percentile for the life cycle cost for each generation of the Genetic Algorithm

The average of the population for each inspection interval can be found for the results gained at the 20th generation of the Genetic Algorithm.

Component	**Phase 1**			**Phase 2**		
	Mean	**Median**	**Mode**	**Mean**	**Median**	**Mode**
Pipework	6.78	6	3	8.04	7.5	6
Electric pump	6.66	6	9	7.935	9	12
Jockey pump	8.55	9	12	8.79	12	12
Diesel pump	7.125	6	9	8.295	9	9
Diesel tank	7.14	6	6	6.84	6	6
Ring main	8.55	7.5	6	5.7	3	3
Sprinkler head and strainer	9.96	12	12	5.31	3	3
Ringmain Isolation valve	7.92	9	9	7.485	7.5	9
Diesel pump pressure release valve	7.485	6	12	7.275	6	12
Deluge valve	5.94	6	6	5.73	3	3
Solenoid	6.6	6	3	8.22	9	6
Manual start device	7.59	7.5	6	9.75	12	12
Pressure sensors	7.28	6	6	8.58	9	12
Heat detectors	6.96	6	6	6.645	6	3
Call points (zone 1)	8.07	9	12	7.785	9	6
Alarm sounders	7.815	9	9	6.495	6	6
Control box	7.38	6	3	5.85	6	3
Control box battery	6.39	6	3	6.33	6	9
Smoke detectors	7.935	9	9	8.07	6	6
Wiring (zone 1)	7.56	6	6	6.93	6	6
Call points (zone 2)	7.53	6	12	6.69	6	3
Wiring (zone 2)	5.91	6	6	9.84	12	12
Diesel pump isolation valve	7.635	6	6	8.10	9	9
Water mains isolation valve	6.375	6	3	7.455	6	6
Diesel tank isolation valve	6.375	6	6	8.61	9	9
Electric pump Isolation valve	8.655	9	12	6.72	6	6
Electric pump pressure release valve	7.68	6	6	7.05	6	3

Table 6.3 gives the optimal inspection interval for each component within each phase. The mean of the population is given for each value in the table. The phase 3 mean inspection intervals, although discovered by the program, are meaningless in this specific case due to the system never reaching the third maintenance phase. They are discarded for this system, with the specific parameter input values used in this example.

Component	**Phase 1**			**Phase 2**		
	Mean	**Median**	**Mode**	**Mean**	**Median**	**Mode**
Pipework	6.78	6	3	8.04	7.5	6
Electric pump	6.66	6	9	7.935	9	12
Jockey pump	8.55	9	12	8.79	12	12
Diesel pump	7.125	6	9	8.295	9	9
Diesel tank	7.14	6	6	6.84	6	6
Ring main	8.55	7.5	6	5.7	3	3
Sprinkler head and strainer	9.96	12	12	5.31	3	3
Ringmain Isolation valve	7.92	9	9	7.485	7.5	9
Diesel pump pressure release valve	7.485	6	12	7.275	6	12
Deluge valve	5.94	6	6	5.73	3	3
Solenoid	6.6	6	3	8.22	9	6
Manual start device	7.59	7.5	6	9.75	12	12
Pressure sensors	7.28	6	6	8.58	9	12
Heat detectors	6.96	6	6	6.645	6	3
Call points (zone 1)	8.07	9	12	7.785	9	6
Alarm sounders	7.815	9	9	6.495	6	6
Control box	7.38	6	3	5.85	6	3
Control box battery	6.39	6	3	6.33	6	9
Smoke detectors	7.935	9	9	8.07	6	6
Wiring (zone 1)	7.56	6	6	6.93	6	6
Call points (zone 2)	7.53	6	12	6.69	6	3
Wiring (zone 2)	5.91	6	6	9.84	12	12
Diesel pump isolation valve	7.635	6	6	8.10	9	9
Water mains isolation valve	6.375	6	3	7.455	6	6
Diesel tank isolation valve	6.375	6	6	8.61	9	9
Electric pump Isolation valve	8.655	9	12	6.72	6	6
Electric pump pressure release valve	7.68	6	6	7.05	6	3

Table 6.3: The optimal inspection intervals for each component at each system phase

This method allows for the complex interactions, especially in system phases where there is age-based maintenance at a variety of intervals, to be modelled and optimised. Some components show an increase in inspection frequency on entry to the second system level maintenance phase, suggesting that failures are more likely as the component ages, despite the addition of age-based maintenance. Alternatively, some components show a decrease in inspection frequency on entry to the second system level maintenance phase, suggesting that the age-based strategy implemented in that individual case is controlling the failure probability such that inspection frequency can be reduced.

This program written to complete this method is generic and the input values to both the model and the optimisation can be easily varied to apply the method with real world data.

Stage 6: Result Interpretation

The results for this optimisation can be used to suggest an optimal asset management strategy with the aim of minimising the risk of fatality over the system life cycle, within a constrained life cycle budget. In this example the modal values found in the inspection optimisation for each component were used to define the inspection intervals. A summary of the following strategy, found for this example is:

1. Between installation and the 12th year following the system installation, components in the system components in the system should not be to subject to age-based maintenance. Components should be maintained on a revealed failure or degraded condition. System testing by opening the test valve should be completed every 9 months.
    - The following component should be inspected every 3 months:

    i. The pipework
    ii. The solenoid
    iii. The control box
    iv. The control box battery
    v. The water mains isolation valve
- The following components should be inspected every 6 months:
    i. The diesel tank
    ii. The ringmain
    iii. The ringmain isolation valve
    iv. The deluge valve
    v. The manual start device
    vi. The pressure sensors
    vii. The heat detectors
    viii. The alarm wiring in zone 1 and zone 2
    ix. The diesel pump isolation valve
    x. The diesel tank isolation valve
    xi. The electric pump pressure release valve
- The following components should be inspected every 9 months:
    i. The electric pump
    ii. The diesel pump
    iii. The alarm sounders
    iv. The smoke detectors
- The following components should be inspected every 12 months:
    i. The jockey pump
    ii. The sprinkler heads and strainers
    iii. The diesel pump pressure release valve
    iv. The call points in zone 1 and zone 2
    v. The electric pump isolation valve

2. From the 12th year following system installation until the 40th year following system installation components should be subject to age-based maintenance when it is assumed that they have reached the end of their useful life. They should also be maintained when it is revealed that they are in a failed or degraded state. System testing by opening the test valve should be completed every 6 months.
    - The following components should be inspected every 3 months:
        i. The ringmain
        ii. The sprinkler head and strainer
        iii. The deluge valve
        iv. The heat detectors
        v. The control box
        vi. The call points in zone 2
        vii. The electric pump pressure release valve
    - The following components should be inspected every 6 months:
        i. The pipework
        ii. The diesel tank
        iii. The solenoid
        iv. The call points in zone 1
        v. The alarm sounders
        vi. The smoke detectors
        vii. The wiring in zone 1
        viii. The water mains isolation valve
        ix. The electric pump isolation valve
    - The following components should be inspected every 9 months:
        i. The diesel pump
        ii. The ringmain isolation valve

iii. The control box battery
iv. The diesel pump isolation valve
v. The diesel tank isolation valve

- The following components should be inspected every 12 months:
  i. The electric pump
  ii. The jockey pump
  iii. The diesel pump pressure release valve
  iv. The manual start device
  v. The pressure sensors
  vi. The zone 2 wiring
  vii.

The optimal maintenance phase entry times and inspection strategy for component, as given above, was simulated to give the risk of the system and to demonstrate how this risk changes over time. A simulation with 1000 runs was carried out. Figure 6.21 gives the risk of the system found in this simulation, following the sample parameters assigned in this chapter. The risk is given as the expected number of fatalities per year. The bars show the average risk predicted by the model within each year. The range bars show the maximum and minimum risk observed within the year as a result of the simulation. As the system ages a general increase in risk can be seen, however a reduction in the risk can be seen after 12 years, corresponding to the entry into the second system maintenance phase. Figure 6.22 gives the probability that each system is in the failed state at each time, also taken from the same simulation.

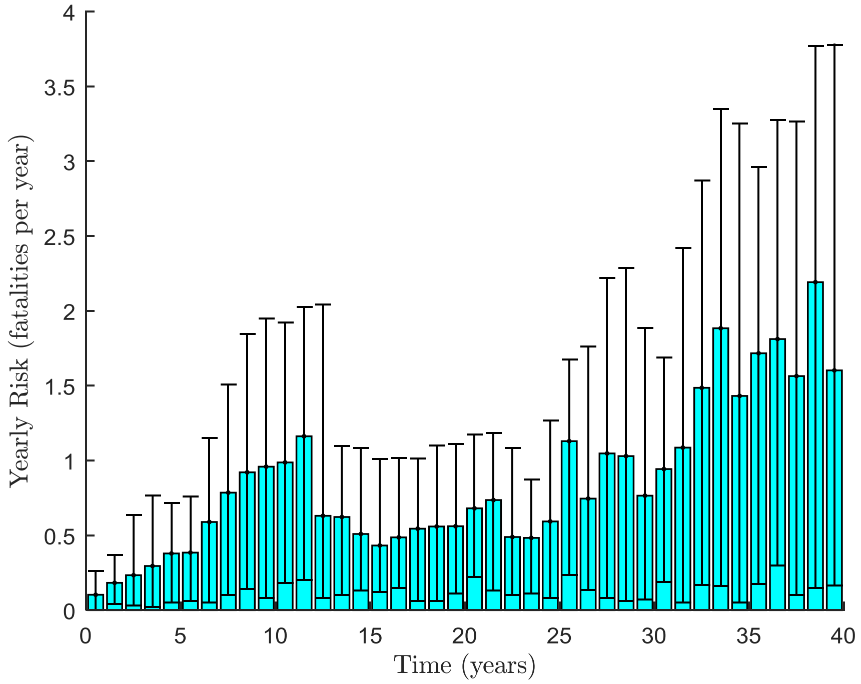


Figure 6.21: The yearly risk, measured in the expected number of fatalities per year, for a simulation of the model with the optimised parameters found in the previous stage.

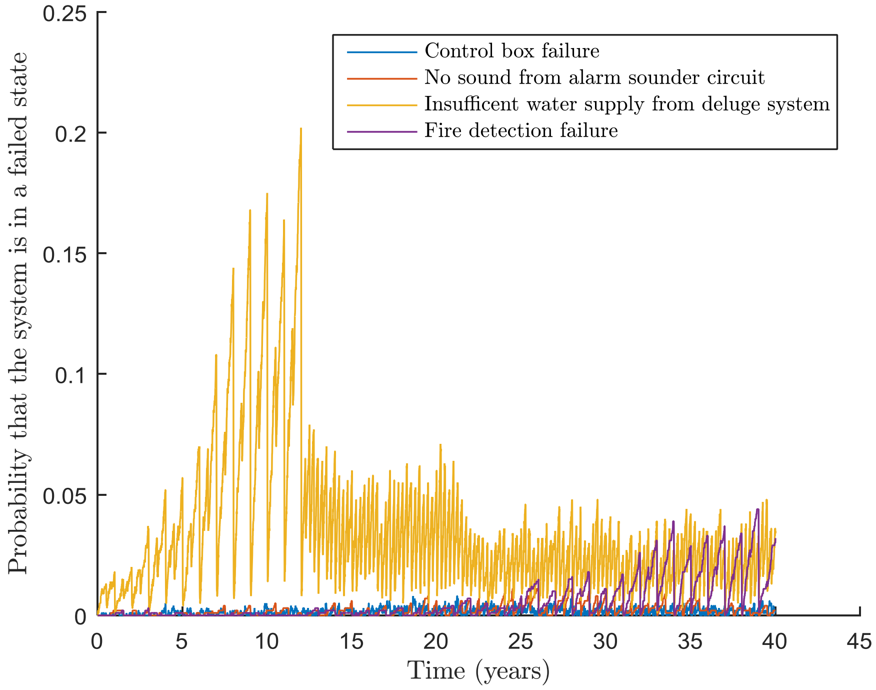


Figure 6.22: The probability of each system failure at evert point in time, gained from a simulation of the model with the optimised parameters found in the previous section

Figure 6.22 shows that the optimisation particularly controls the risk of the control box failure, detection failure and alarm failure. The detection failure has the highest number of fatalities associated with it, in the sample inputs to this optimisation procedure, and causes a total system failure. A failure in the alarm system also has a high level of fatalities associated with in, in the sample inputs to this optimisation procedure. In addition, the control box failure causes a total system failure and has no back-up system in this application. Conversely, the deluge system is allowed to fail more frequently, this is expected as the deluge system failure has less fatalities associated with it in this application of the methodology. There is an increase in the fire detection failure towards the end of the time period under consideration, there is also a periodic pattern displayed. This can be attributed to the Zone 2 wiring failures that where inspection only occurs every 12 months, and hence if there is a failure it can remain undetected for a period of time. Since this failure happens infrequently, a higher number of simulations of the model is expected to reduce the periodic behavior.

6.3.5: Summary

In summary, a methodology has been presented to optimise over a phased system level asset management strategy and a component level asset management strategy, given the optimised system level phases, within a Petri net modelling framework. The optimisation takes place over two levels combining a Simulated Annealing algorithm with a Genetic Algorithm.

The model developed in Chapter 5 and Section 6.2 of this chapter was used for demonstration of the approach. It is assumed here that a failed component is always repaired, on discovery. With the focus on reducing the number of failure occurrences, the first stage of the optimization finds the most effective strategy, within the defined structure of the model, to minimise the risk of the system by considering the phase entry times. With each phase the inspection frequency increases along with the rate of preventative maintenance. A Simulated Annealing algorithm is applied at this stage. Following this, the inspection interval of the components, within each phase is also optimized to find the minimum risk represented by the model given the phase entry times of the system. A Genetic Algorithm is applied in this stage.

The first stage of the optimisation is set to reduce the risk given a constraint on the budget. This was done to ensure the minimization of the risk, for the resources available. The second stage of the optimisation aims to reduce the cost and risk within an objective function.

An alternative option to this approach is to optimize the whole model in one stage, with varying phases and inspection strategies at each proposed solution. This increases the number of options that must be tested at each point in the optimization. In this approach, due to the main requirement to minimise the risk and the expensive cost of maintenance, the combined two-stage Simulated Annealing algorithm and Genetic Algorithm was chosen to save computational cost.

6.4: Convergence and Uncertainty in Petri net models

Modelling uncertainty can arise in a Petri net model due to a lack of knowledge of a system [170]. Uncertainly analysis uses probabilistic methods to find bounds around a predicted value to a certain level of confidence; there is a set probability that another value will occur within that range.

In a Petri net, uncertainty arises from uncertain input data, the assumptions made in the Petri net structure and imperfect convergence due to a finite number of simulations. This is applicable to any modelling approach [171]. The uncertainty in the data input to the model can arise due to the random nature of failures and differences between the ‘same’ components. This results in an imperfect prediction for each individual component behaviour based on the best population data available. Uncertainty is also introduced into the data through lack of knowledge, measurement error, subjective judgement, lack of specific descriptions and ambiguity. These uncertainties are difficult to quantify. The second reason for uncertainty in the Petri net model outputs is due to the imperfect nature of the Petri net model structure used in any scenario. With every application of the Petri net modelling technique different approximations will be made by the modeller, these can result in inaccuracies in the model outputs. Finally, uncertainty is introduced through the simulation of the Petri net model via Monte Carlo Simulation; in this case the higher the number of simulations the lower the uncertainty.

The section considers the uncertainty inherent to the model due to a non-infinite number of runs of a Monte Carlo simulation. With a number of runs of a Monte Carlo simulation of the Petri net model there should be convergence to a mean value for the key outputs of the system. However, there is some distribution in the expected outputs for each run of the model. It is expected that with an increase in the number of simulations the confidence intervals of the mean values obtained will reduce. It is also possible to gain a measure of the order of the convergence for the model outputs, with an increasing number of model simulations. The optimal values found in the previous section are used as input to the full system Petri net model for this analysis.

6.4.1: Example Analysis

The uncertainty in the risk value generated with the optimal solution found in Section 3.6.4 of this chapter is used for a sample analysis in this section. This was selected as a candidate for analysis as the contribution of rare event failures can result in a slow convergence of the risk value from the simulation and the risk encompasses the threat to human life by the system. This increases the importance of gaining the knowledge of the level of convergence of the system and hence the accuracy in the model outputs.

Figure 6.23 gives the life-cycle average risk output from the model, which is obtained from each run of a simulation containing 1000 runs. The average risk is displayed on the x-axis with the proportion of the runs that resulted in each average given on the y-axis. This shows the distribution of the values that can be output from each run of the simulation of the model.

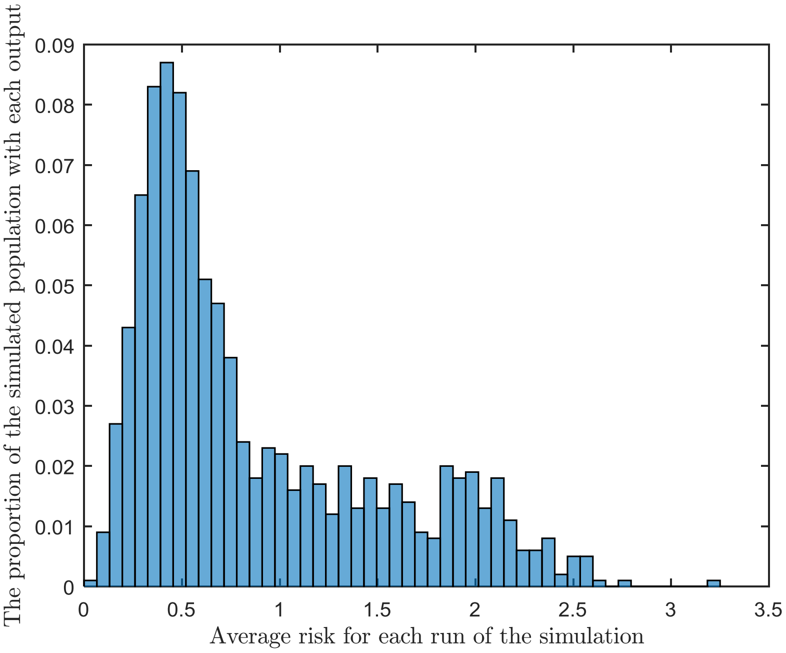


Figure 6.23: A normalised histogram showing the distribution of the average risk values that are output for each run of the simulation

The distribution of the risk values for each run shows a fairly symmetric curve that is similar to a normal distribution for risk values before 1 fatality per year. The modal risk value occurs in the region surrounding 0.5 fatalities per year. The distribution is positively skewed with risk values seen over the level of 2 fatalities per year. The mean value of the risk is 0.88 fatalities per year. This skew can be attributed to the contribution of the different system failures on the total risk, especially the contribution on rare events with a high level of fatalities. There is a spread in the values predicted by each run of the simulation which can result in slow convergence in the model.

The convergence of the average risk value was also analysed. It was assumed that the average risk outputs from the simulation will converge to the true value, with an increasing number of runs. It is also expected that this convergence, in the mean, will follow a normal distribution in accordance to the Central Limit Theorem. Under this assumption, the 95% confidence intervals on the mean value for the average risk, for each number of runs, was calculated. To do this, the distribution of the mean values, for each number of runs, was approximated as a normal distribution and by fitting a normal curve the standard deviation was found. This standard deviation was used to give the 95% confidence interval for each number of runs of the simulation. The mean risk for each number of runs, along with the 95% confidence interval is given in Figure 6.24.

It is important to note the difference between this analysis and the distribution of risk values per run shown in Figure 6.23. In the earlier analysis, the distribution of each value, output by each run, was independently considered. This was done to give an idea of the distribution of model outputs. These model outputs are not assumed to follow a normal distribution. On the contrary, this current analysis section considers the distribution of the mean values, where the mean value for each number of runs is the mean of the outputs for the number of runs at that point. For example, the mean at 50 runs, is the mean of the model outputs for 50 runs of the model. The distribution of the mean values is expected to approximate a normal distribution.

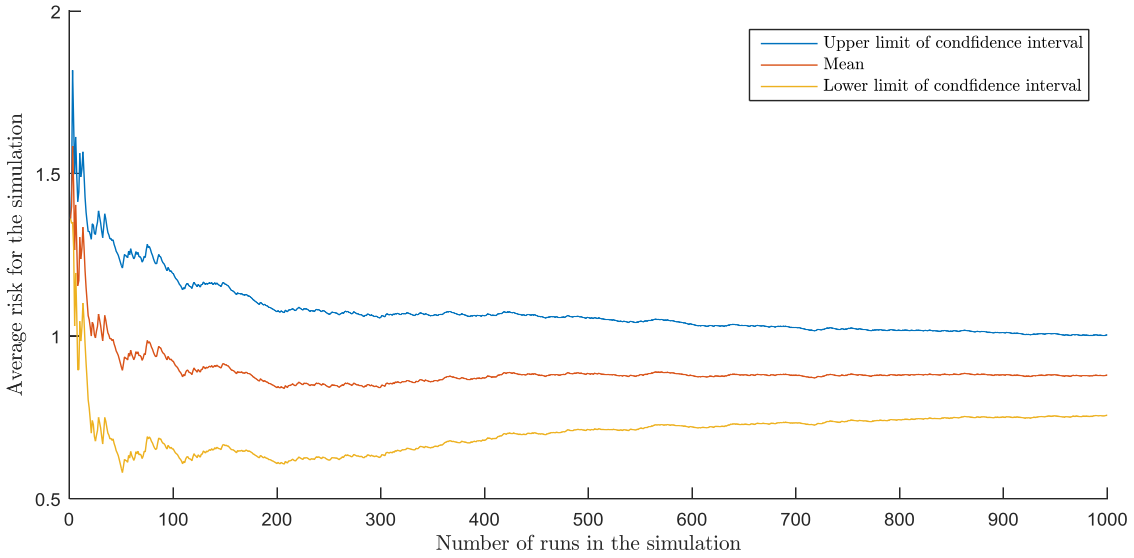


Figure 6.24: A graph showing the convergence of the mean of the average risk value of the system under simulation, with an increase in the number of runs of the simulation. The 95% confidence intervals in the mean value are also displayed.

Figure 6.24 shows the convergence of the mean value of the risk, taken from simulation of the model, with an increasing number of runs of the simulation. There is initially a high level of instability in the mean value. This emphasises the need for a high number of runs, resulting in a level of convergence in the model, for a more stable optimisation process, this has been partially compensated for in this chapter by taking the modal values of the population to improve the stability.

The order of convergence of the average risk value with the number of runs can be analysed to give a better idea of the number of runs required to reach a confidence interval of a defined threshold. The range of the 95% confidence interval for the risk was used as a measure of convergence in this analysis. It is expected that the convergence of the range of the 95% confidence interval, with the number of runs of the simulation, will follow the relationship given in Equation 6.1:

$$\varepsilon = an^k \tag{6.1}$$

Where ε is a measure of the error on the mean values, where the 95% confidence interval is used in this case. n is the number of runs of the simulation and a and k are constants, where k is the order of convergence.

To find the order of convergence the natural logarithm of both sides of the equation can be taken to give:

$$\log(\varepsilon) = k\log(n) + \log(a) \tag{6.2}$$

If the previous assumption is correct, this takes the form of a straight line, where the gradient of the line, k, is the order of convergence.

The average risk values for each number of runs of the simulation were used as input to this analysis. The confidence interval found for a number of runs that were less than 200 were discarded at this stage. This was done as the initial region of high instability is such that there are insufficient values for the Central Limit Theorem to allow the distribution of the mean values to be approximated as a normal distribution, hence resulting in unstable confidence intervals. The log-log graph for this analysis applied to the risk values for each number of runs of a simulation is given in Figure 6.25.

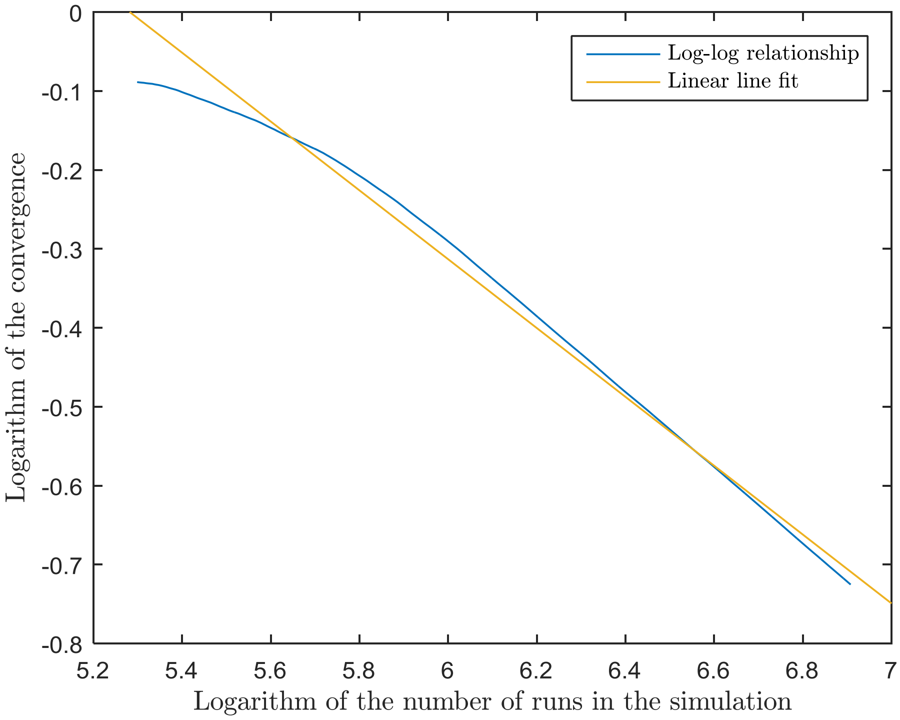


Figure 6.25: A graph showing the logarithm of the 95% confidence interval in comparison to the logarithm of the number of runs, for the risk of the system.

Fitting a straight line to the log-log graph given in Figure 6.25 gives the following equation:

$$\log(\varepsilon) = -0.436 \log(n) + 2.3032 \qquad (6.3)$$

Hence, the order of convergence, $k = -0.436$, where to decrease the confidence interval by a factor of r, the number of simulations must increase with the relationship $r^{-1/k}$.

This approach can be extended to consider the uncertainty on the risk values obtained within each year time period of the simulation. This analysis has assumed that there is no uncertainty in the input values in the model, and has discussed the convergence and a measure of how the uncertainty of the model can be expressed using the Central Limit Theorem. The next section of this model considers a case where there is a Petri net model with uncertain parameter inputs.

6.5: A Petri net with uncertain inputs

There is difficulty in finding a measure of the uncertainty introduced through Petri net modelling, especially in the case where there are uncertain input values for the model. Analytical methods cannot be used as there is no explicit formula for the output of the Petri net models, used in this chapter, as a function of the model inputs. This section presents a method to consider an uncertainty in the values output from a Petri net model, where there are uncertain inputs. In this application it is assumed that the uncertainty introduced by a poor Petri net modelling approach is negligible, and the uncertainty arises from the imperfect and random nature of the data used as input for the model and from the Monte Carlo simulation of the model.

In this chapter, we assume probabilistic uncertainty in model parameters with application to a Petri net model. The method aims to include uncertainty introduced though the simulation of the Petri net and through the uncertainty in the input parameters. The method can be extended further to include a probabilistic fuzzy representation of some input values. The following short exploratory analysis considers the use of an optimisation algorithm within a framework for considering probabilistic uncertainty, in order to improve the computational efficiency required by a double loop methodology.

6.5.1: Method

In this thesis, a methodology is proposed to find the impact on Petri net model outputs given uncertain input values. This methodology uses a Simulated Annealing algorithm, to find a measure of the uncertainty in the model outputs where there is an uncertainty on inputs to the models. The methodology is applied to a simple Petri net and for demonstration, inputs governing the ageing of a component are considered to have the largest uncertainty. In real terms this corresponds to a lack of certain knowledge about the input parameters for the model. For instance, there may be an expected mean time to failure for an aged component of 50 years, however, due to non-ideal data or large data variance, there may be a 95% confidence interval of 20-80 years. In an ideal data set, there may be an expected mean time to failure of 50 years but with a 95% confidence interval of 45-55 years. However, in the outputs of a Petri net model, the average result on convergence gives the same solution for each of these cases; if there is no measure of the confidence in the model outputs even though there are clear differences in the input data.

To combat this, the methodology has been developed to carry through uncertainty in Petri net input parameters and incorporate the uncertainty introduced through Monte Carlo simulation of the model. Uncertain parameters are highlighted in the model and the 95% confidence interval for each is assigned. The uncertain parameters are allowed to vary within this interval. For each variation of the uncertain parameters, a Monte Carlo simulation of the Petri net model is completed. The aim of the method is to find the maximum and minimum average value for the Petri net output at each time, given the 95% confidence interval in the input parameters.

A Simulated Annealing algorithm is repeated twice within the framework: once to find the maximum boundary and once to find the minimum boundary for the averages, given a set number of runs of the simulation, where the input parameters vary between each complete simulation. The objective function of the Simulated Annealing algorithm is defined as, the sum of the output vector in each case with the algorithm seeking to maximise or minimise this sum. It is assumed that the relationship between the uncertain input parameters and the output vector is roughly monotonic. For instance, if the input parameter (time to failure of a component) in the sample simulation is close to the upper 95% confidence interval for that simulation, the failure will occur less frequently. This assumption allows the algorithm to focus on areas where the solution at each time is likely to be near a boundary. However, the end boundary is not found by using the 'best' parameters found by this algorithm. For each parameter that is tested, the average value at each time of the simulation output is compared to the current 'best' upper or lower bound. If the value at any time is found to be more extreme than the current upper or lower bound at this time, it is accepted.

The approach enables the algorithm to incorporate the simulation uncertainty and the uncertainty due to the uncertain input parameters at the same time. For instance, any possible solution variances that are due to a non-infinite number of simulations are also collected into the upper or lower bound and the algorithm makes no distinction between these two types of uncertainty.

To clarify, the lowest value stored in this algorithm is not the lowest value within a run of the simulation, but the lowest average value for a full simulation, taken from a defined number of runs. This average value can change by simulation due to the varying behaviour of the input parameters and the uncertainty introduced through the Monte Carlo Simulation.

6.5.2: Sample Application

This methodology is applied to a component model from Chapter 5, to demonstrate its capability. The process can be applied to the whole model discussed in this chapter and Chapter 5, however is highly computationally expensive and intractable for the software developed in this thesis. Further work can be completed into methodologies to apply this approach to large system models. The Petri net for the pipework used in Chapter 5 and the Petri nets governing the inspection and maintenance at each phase was extracted from the models used previously. The methodology for finding the uncertainty on the outputs, given uncertain model inputs, was applied to this model section.

It is reasonable to assume that in a real world application of the Petri net based model, the data present for the ageing of the components, will not give rise to parameter values governing the failure distribution with 100% accuracy. For instance, if the data follows a Weibull distribution, every data point will not lie directly on the Weibull Distribution. Hence, there is some uncertainty in the parameters governing the distributions used in the model that are based on this data. In the case of the Weibull distribution, there could be uncertainty on the shape parameter β and the scale parameter η.

Assume that a transition is governed by two parameters, θ_1, θ_2, although the transition may be governed by any number of parameters depending on the patterns shown in the data. Instead of representing a transition, $T = T(\theta_1, \theta_2)$, this method aims to include uncertainty in the model parameters such that, $T = T(\theta_1 + \delta\theta_1, \theta_2 + \delta\theta_2)$, where $\delta\theta_i$ is the uncertainty on θ_i.

The results for the small Petri net are given in this section. In this example a Weibull distribution is assigned to the transition governing the ageing of the component and a uniform distribution is assigned to the random failure rate of the component. It is assumed that there is a lack of knowledge about the parameters governing these transitions. For illustration, the parameters and their associated uncertainties are set at:

- Transition t1 (ageing): $\eta = 600 \pm 50,\ \beta = 1.5 \pm 0.25$
- Transition t2 (random failure): $r = 0.000139 \pm 0.00005$

In this example the parameters governing these transitions are allowed to vary between the specified ranges. The Simulated Annealing algorithm looks for the values that correspond to the maximum or minimum value for the probability of failure over a number of simulations. This algorithm stores the maximum or minimum at each point in time at the end of each simulation; this maximum or minimum can arise through error introduced by the simulation of the model or through the uncertain model inputs. The Simulated Annealing algorithm requires repeated simulation of the Petri net.

Figure 6.26 shows the results of this algorithm, applied in this case. Here, the algorithm has been applied four distinct times, with a different number of simulations of the Petri net in each case. It can be seen here that for a small number of simulations, the range of possible average outcomes is large and the average is more unstable in time. The histogram bars show the average unavailability for the component over each year, where the model input values are the mean value of each parameter. The range bars show the maximum and minimum average unavailability for the component over each year for a full simulation, where the input values can range in the confidence intervals. As the number of simulations increases, the average value at each point stabilises and the range of possible values reduces.

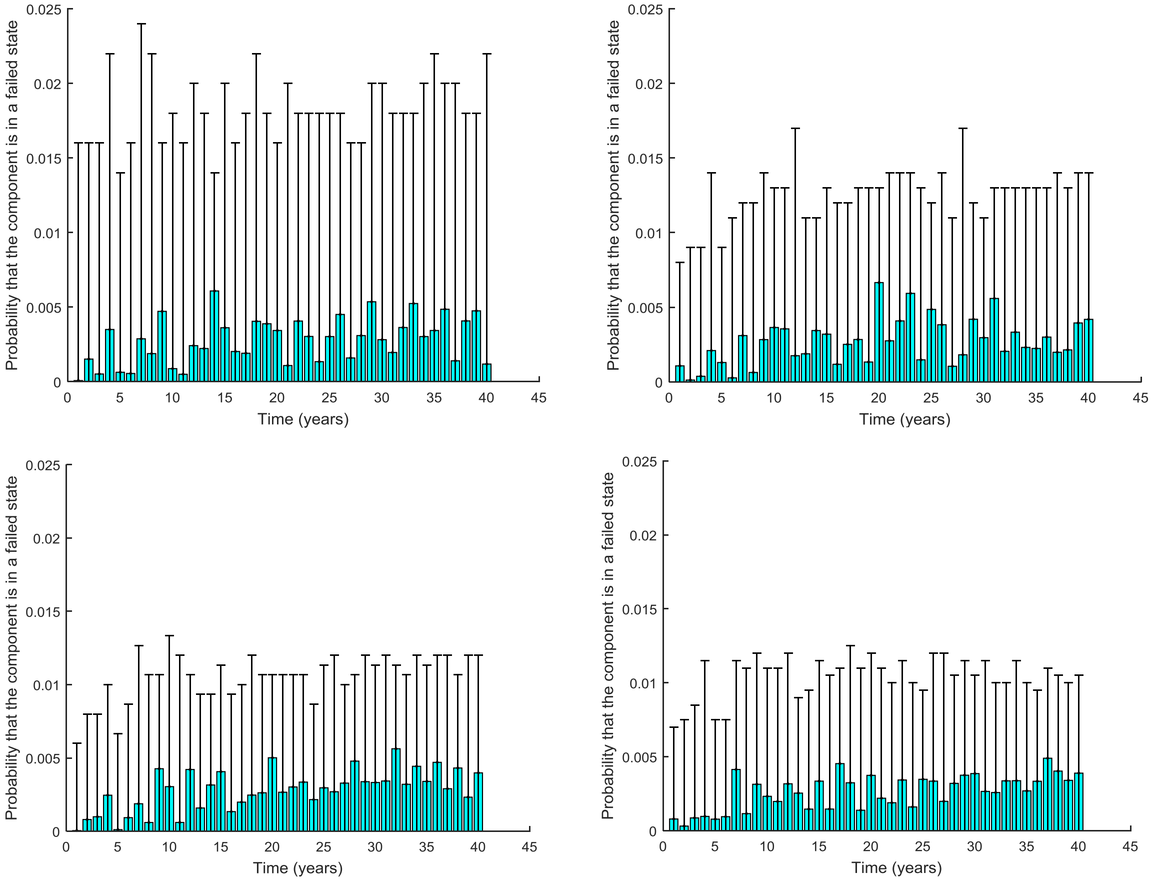


Figure 6.26: A figure showing the probability that the component is in the failed state, with the range bars, for 500 simulations (top left hand side), 1000 simulations (top right hand side), 1500 simulations (bottom left side) and 2000 simulations (bottom right side).

Figure 6.27 gives the range on the average probability of failure at each time for different numbers of simulations. This shows, as with the previous figure, that an increase in the number of simulations reduces the error on the final outcome. However, the reduction is not linear and does not tend towards zero in this case. This is due to the uncertainty introduced in the input parameters leading to an inherent level of uncertainty in the outputs of the model.

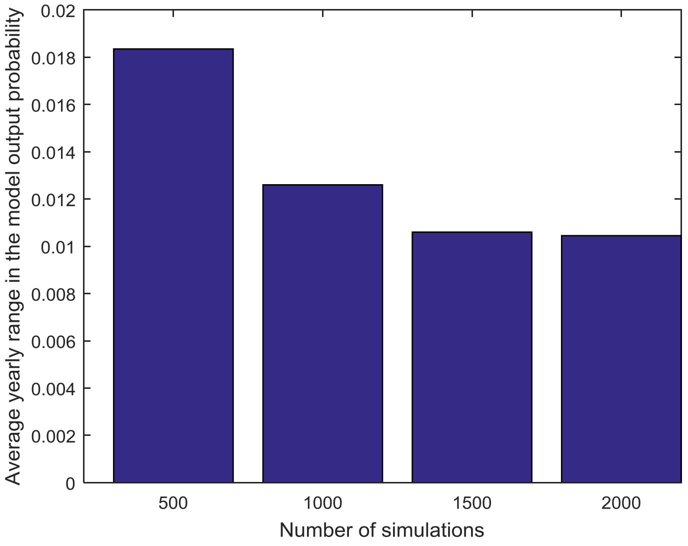


Figure 6.27: A figure showing the average uncertainty on each probability for different numbers of simulations

This demonstration has shown the capability of this approach in dealing with uncertain input parameters and the uncertainty introduced through simulation of the Petri net model. This method enables knowledge about uncertain data to be carried through the analysis in order to give an informed estimate of the outputs of a Petri net model. This also demonstrates the necessity for sufficient convergence of the model outputs due to the Monte Carlo simulation, and if there are insufficient simulations then this is demonstrated by the uncertainty in the results.

This methodology can be applied to a larger Petri net with a number of uncertain input values to give a measure of the uncertainty on the solution. This can be incorporated into the final model outputs, such as an estimate for the risk.

6.6: Discussion

The aim of this chapter has been to further extend the models presented in the previous chapter. The method presented initially in this chapter demonstrated that the logic used to give the system failure probability in Chapter 5 can be represented with Petri net logic. However, this conversion of the Fault Tree structure increases the computational cost of the simulation of the model.

Also, the optimisation method in this chapter has the potential to optimise over a number of system phases. However, due to computational constraints the number of simulations of the model had to be reduced in order to allow the optimisation of the system to be completed in a reasonable timeframe. This resulted in an optimisation based on approximate simulation values that are taken from the model before full convergence is reached. In order to provide optimised results with more confidence, the method can be exactly replicated but with a higher number of model simulations for each trial of the model within the optimisation. This issue has been partially overcome by considering the modal values of the population following the optimisation of the model with a Genetic Algorithm. This was done under the assumption that by considering the average behaviour of the population, as opposed to the behaviour of a single population member, that the approximate optimal solution will more closely approximate the true optimal solution.

A deeper study into model convergence, and the confidence intervals of the risk values predicted by the model, demonstrates that there is a slow convergence of the model. Uncertainty measures can be assigned to the mean values that are output from the model to encompass this. In addition, the method for considering the impact of uncertain parameters on the model outputs has been presented. Again, this method comes at a high computational cost. Further study can be completed into assigning the 95% confidence bands to each time of the simulation, in cases where there can be uncertain inputs.

This chapter has explored potential methods to use traditional Petri net models, commonly seen in literature, in order to expand their usefulness. In each case there has been a limit reached for the applicability of the model due to the computational cost of the Monte Carlo simulation of the Petri net. This is especially visible in methods, such as optimisation or analysis of uncertain inputs, where repeated simulations of the model are required. There are several developments to the modelling approach that can address this issue including: an improvement of hardware for model simulation, an improvement of software for model simulation and an improvement in the simulation approach, such as using a more efficient algorithm to seek convergence than the Monte Carlo Simulation.

6.7: Parameter Assumptions and Use of Data

For the example application of the optimisation approach, given in this chapter, parameters are assumed for the costs of each maintenance, inspection and testing action. Altering these costs will impact the optimised solution of the model. If a component maintenance or inspection cost is increased, the optimisation may find that it is more beneficial to maintain or inspect an alternative component more frequently, if this does not have a large impact on the risk. Attention should be given to the methodology to ensure that the penalty for increased risk is sufficient, so that the optimisation does not recommend a cheaper system strategy that allows the risk to increase. Hence, the optimisation method is sensitive to the parameters that govern the balance between cost and risk.

Despite the assumptions of the parameters in order to demonstrate the method, the optimization approach can be taken from this chapter, and applied with real data. Data should be collected on the cost of different maintenance actions; this can then improve the results given by the optimizations, if combined with real data for the system as discussed in Chapter 5. The results from the optimisation can be used to inform maintenance decisions. Care should be taken with parameters that balance cost with risk, due to ethical implications.

6.8: Contributions

There are a number of novel aspects of this chapter. Firstly, the chapter combines new Petri net models for the deluge, detection and alarm systems to give the risk, using an Event Tree approach. The Fault Tree containing the failure logic of the system is replaced with Petri net logic. This extends the current Fault Tree and Event Tree implementations available in industry, allowing a more detailed analysis of the indirect factors that can impact system failure, such as the applied maintenance actions.

Secondly, a novel optimisation approach for Petri nets is presented. The approach uses a combined Simulated Annealing and Genetic Algorithm optimization, for a phased system asset management strategy. This improves the Genetic Algorithm method as it allows the reduction of the number of parameters required for Genetic Algorithm optimization, hence improving efficiency. The main contribution of this part of the thesis is the use of this approach to optimize a phased system model. The optimization of a phased system shows improvement on current modelling capabilities as it allows optimal choice of when different strategies should be applied over a systems lifecycle. This may be particularly useful for improving maintenance strategies as a system moves past the useful life phase and is still in operation. This optimisation procedure is applied to the combined fire protection system model. It is beneficial to optimise across the three systems, in a combined model, as they share some components.

Secondly, the rate of convergence of the Petri Net model is analysed using a log-log approach. This shows improvements when compared to plotting convergence on a linear scale as it allows the convergence of the Monte Carlo simulation of the model to be quantified. Finally, an area of novelty in this chapter is the use of a Simulated Annealing algorithm to gain an estimate of the uncertainty of the model, given an uncertainty in the input parameters. This improves the state of the art for Petri net modelling and risk models present in industry, by considering the uncertainty introduced through imperfect input parameters. This method gives an uncertainty measure for the final risk value that encompasses both uncertainty in the model input parameters and uncertainty introduced through simulation of the model. This method addresses issues discussed throughout this thesis surrounding the assumptions in model parameters and their impact on the model outputs, by allowing the quantification of uncertainty on input parameters to be specified and carried through to the model outputs. This gives more information to the decision maker and can highlight areas where more data should be collected.

6.9: Conclusion

This chapter has presented a number of aspects covering the analysis and use of models such as those developed in Chapter 5. The aim of this chapter has been to provide methods for a further analysis of Petri net based models.

In the first part of this chapter, the Fault Tree used to combine component models in Chapter 5 was converted to a Petri net structure. This demonstrated results agreeable to those presented in the previous chapter. This methodology can be implemented to convert a model based on a Fault Tree structure to one which can incorporate dependencies between component failures.

Following this, the model developed formed the basis of an optimisation approach for the phase based maintenance of the system over its life-cycle. This approach used a Simulated Annealing algorithm to find the optimal phase entry times, followed by a Genetic Algorithm to assign the inspection intervals for components, given the phased maintenance strategy.

In the penultimate section, a discussion of the convergence of the model with the number of simulations was made. This discussion focused on the rare nature of some failures within the system and how this impacts the number of simulations required to reach a fully convergent answer. A measure of the uncertainty in the model outputs, caused by the Monte Carlo simulation, given a number of simulations was also presented.

Finally, an approach was presented to carry through uncertain model inputs within a Petri net framework. This approach also incorporates the impact of a non-infinite number of simulations of the model on the uncertainty in the outputs. This approach was then applied to a component model used in Chapter 5 of the thesis.

A common issue found through these examples is the computational cost of simulating the Petri net in order to obtain a convergent answer. This is especially highlighted in cases where optimisation of the model is required, or the methodology for incorporating uncertain inputs is implemented, as both methods require many convergent simulations to reach a solution. The next chapter of this thesis presents a methodology for the reduction of Petri net models, to decrease the computational cost of simulation in an attempt to tackle some of these issues.

Chapter 7 Petri net reduction

7.1 Introduction

As demonstrated in the earlier chapters of this thesis, for large and complex stochastic Petri nets with a variety of transition types, finding an exact analytical solution to the Petri net is practically impossible [125]. Simulation tools, such as a Monte Carlo simulation, can be employed to find the average marking sequences based on the probability model associated with the transitions. This type of analysis has been shown to be effective in literature and throughout this thesis [172][173][64][158]. However, this method is computationally expensive due to the requirement of a large number of runs of a simulation to obtain convergence for the marking sequences of the Petri net. Large Petri net models are difficult to simulate via this method, especially in cases where an optimisation of the Petri net model inputs is required. This is because optimization techniques may require repeated convergent simulation of the Petri net, exasperating the already lengthy time of the simulation.

There have been several studies to develop methods to allow very large Petri net based models to be simulated efficiently including the use of parallel computing [174] [175] [176]. These methods, though effective, require specialist hardware or software. In the previous chapter methodologies were introduced to find optimal solutions to a problem, and a measure of the uncertainty, using a Stochastic Petri net model as the system modelling framework. These methods require repeated analysis of the already costly simulation of the Petri net model. For example, take one additional place in a Petri net model that fires twice in the period of interest. If the Petri net requires 2000 Monte Carlo runs to reach a convergent answer, this transition must fire 4000 times in one convergent run of the Petri net. If then during optimization of the Petri net, or a simulation of the uncertainty, 500 variables require testing then this one additional transition must fire 2,000,000 times during the course of the whole analysis. It is clear to see that the reduction of a small number of transitions can impact the computational efficiency greatly.

This chapter presents a technique that can be used when reducing the structural size of Stochastic Petri net models. Simple reduction rules and decomposition techniques can be used to reduce a large Petri net model into a simpler one whilst retaining its properties. These rules include the fusion of places, and transitions, in either parallel or series arrangements as well as removing self-loop transitions, but are limited in application due to the need for existence of specific structures within each Petri net. Hence, these techniques can often only provide a limited reduction to Petri net complexity [177]. The methodology presented in this chapter gives an approach for the reduction of Petri net size that is not limited to specific structures. The methodology aligns with the Monte Carlo simulation method, used throughout this thesis, to allow numerical analysis of varied Petri net structures. In this methodology, a reduced structure is found that can mimic the behaviour of a larger Petri net model, but with a reduced computational cost.

This chapter presents a methodology to produce a reduced Petri net to approximate the outputs of a reference Petri net. This reduced Petri net can then be used to replace the reference Petri net for analysis with a lower computational time.

7.2 Concepts

This section introduces three concepts employed in the methodology presented in this chapter that are incorporated into the parameter updating stage of the methodology. These are: Bayesian Model Updating, Approximate Bayesian Computation (ABC) and Sub-Set Simulation. Finally the ABC-SubSim Algorithm is presented.

7.2.1 Bayesian Model Updating

Bayesian Model Updating provides a methodology to make inferences about parameters of a model based on experimental data [178]. Where there are multiple candidate models, this methodology can also provide a framework to assess the plausibility of each model. Bayesian methods also provide a quantification of the uncertainty in the model parameters introduced by measurement error or the choice of model.

For a stochastic model class, M, Bayes theorem provides a methodology whereby prior knowledge of the parameters of interest, $\theta \in \Theta \subset \boldsymbol{R}^l$, of a system can be updated based on information gained from a set of data, $y \in D \subset \boldsymbol{R}^l$, where D is the observation space and the region in $\boldsymbol{R}^l$ contains all possible observational outcomes according to the model class. A prior distribution, $p(\theta|M)$, represents the initial knowledge about the parameters of interest and a likelihood function, $p(y|\theta, M)$, represents the probability of obtaining data values, θ, for the model class, M. This theory allows the calculation of a posterior distribution for the model class resulting in an updated probability distribution for the parameters of interest given the observed data. For parameter updating, Bayes theorem is given in Equation 7.1 for a posterior PDF, $p(\theta|y, M)$, of the model specified by θ.

$$p(\theta|y, M) = \frac{p(\theta|M)p(y|\theta,M)}{\int_{\boldsymbol{\Theta}} p(\theta|M)p(y|\theta,M)d\theta} \propto p(\theta|M)p(y|\theta, M) \qquad (7.1)$$

In many cases the denominator is intractable but it can be absorbed and replaced by a normalization factor.

In the case of a set of L competing model classes, $\boldsymbol{M} = \{M_{1,}M_2 \dots M_L\}$, the posterior probability of each model class can be found by Bayes theorem at the model-class level:

$$p(M_L|y, \boldsymbol{M}) \propto p(M_L|\boldsymbol{M})p(y|M_L) \qquad (7.1)$$

There are some model classes where the likelihood function is difficult or impossible to calculate. For these models the model updating and model class selection given in Equation 7.1 and Equation 7.2 are not directly applicable however performing parameter updating or model class selection may still be of interest. Approximate Bayesian Computation (ABC) methods can be used to remove the need for computation of the likelihood function while providing a framework for parameter inference and model selection.

7.2.2 Approximate Bayesian Computation (ABC)

Approximate Bayesian Computation (ABC) is a simulation based approach that can evaluate a posterior density whilst avoiding the need for exact knowledge of the likelihood function. ABC also avoids the need to evaluate the intractable integral in the denominator of Equation 7.1. In ABC, a rejection algorithm is used to sample from the posterior in order to find a region, within a set tolerance, that is close to the true posterior values or alternatively, to find the best posterior estimate and give a measure of how close it is to the true posterior [179].

Let $x \in D \subset \boldsymbol{R}^l$, denote a simulated dataset from $p(\cdot\, |\theta, M)$ the forward model of model class, M. An ABC algorithm aims at evaluating the posterior, $p(\theta|y, M) \propto p(\theta|M)p(y|\theta, M)$ by applying Bayes' Theorem to the pair (θ, x).

In Equation 7.3 the model class, M, has been omitted under the assumption that the theory is valid for any specific model class, the equation is arrived at by using the law of total probability.

$$p(\theta, x|y) \propto p(y|x, \theta)p(x|\theta)p(\theta) \qquad (7.3)$$

Higher weights are given to regions where x is close to y by $p(y|x, \theta)$, representing values where simulated parameter values are closer to the true posterior. In a basic ABC algorithm a sample is taken from the posterior in Equation 7.3 and the sample is accepted or rejected based on the equality $x = y$. In practice, since equality is impractical to obtain the ABC algorithm results in a region in $\boldsymbol{R}^l$ where values for x are close to y.

To find a region where $x \approx y$, a tolerance parameter ϵ is introduced that represents closeness of the parameters judged by a metric value ρ gained by a summary statistic $\eta(\cdot)$. Through this approach, the posterior $p(\theta, x|y)$ in Equation 7.3 is approximated by $p_\epsilon(\theta, x|y)$, which assigns higher probability density to those values of (θ, x) that satisfy the condition $\rho(\eta(x), \eta(y)) \leq \epsilon$.

From Bayes' Theorem, the approximate posterior $p_\epsilon(\theta, x|y)$, is given by Equation 7.4.

$$p_\epsilon(\theta, x|y) \propto P(x \in N_\epsilon(y)|x)p(x|\theta)p(\theta) \qquad (7.4)$$

Where $P(x \in N_\epsilon(y)|x) = \boldsymbol{I}_{N_{\epsilon(y)}}(x)$ is an indicator function that assigns a value of 1 if $\rho(\eta(x), \eta(y)) \leq \epsilon$ and 0 otherwise. The output of the ABC algorithm corresponds to samples from the joint probability density function:

$$p_\epsilon(\theta, x|y) \propto p(x|\theta)p(\theta)\boldsymbol{I}_{N_{\epsilon(y)}}(x) \qquad (7.5)$$

The end interest is typically the marginal approximate posterior:

$$p_\epsilon(\theta|y) \propto p(\theta) \int_D p(x|\theta)\, \boldsymbol{I}_{N_{\epsilon(y)}}(x)\, dx \qquad (7.6)$$

An algorithm to generate N posterior sample values by ABC is given below:

Algorithm 1 Standard ABC

```
    for $t = 1$ to $N$ do
        repeat
        1. Simulate $\theta$' from $p(\theta)$
        2. Generate $x' \sim p(x|\theta')$
        until $\rho(\eta(x), \eta(y)) \leq \epsilon$
        Accept $(\theta', x')$
    end for
```

The success of the ABC algorithm is dependent on a good choice of the summary statistic $\eta(\cdot)$, metric choice ρ and tolerance parameter ϵ. For small posterior regions ABC can be computationally heavy as a large quantity of simulations are required to reach a significant number of parameter values within the required tolerance. There have been several algorithms developed to decrease the computational time for the ABC algorithm, several of these can be found in literature [179] [180] [181] [182] [183]. For the reduction methodology presented here the ABC-SubSim algorithm has been chosen as a sufficiently good algorithm to reduce the computational effort of ABC. This algorithm combine's sub-set simulation with ABC, both subset simulation and the ABC-SubSim algorithm are explained in the next sections.

7.2.3 Subset Simulation

Subset simulation is a method presented in [184] to avoid the need for costly or inaccurate rare event simulation that arises due to the existence of a very small failure region. The need for rare event simulation is avoided by the introduction of several smaller intermediary regions between the initial region and the failure region by generating conditional samples.

This corresponds to levels of a performance function $g: \boldsymbol{R}^l \rightarrow \boldsymbol{R}$, which results in a sequence of more frequent events equivalent to the single, low probability, rare event.

Let F be the failure region in the z-space, $z \in Z \subset \boldsymbol{R}^l$**,** corresponding to exceedance of the performance function above some specified threshold level b:

$$F = \{z \in Z | g(z) < b\} \tag{7.7}$$

For simpler notation, we use $P(F) = P(z \in F)$ Let us now assume that F is defined as the intersection m regions $F = \bigcap_{j=1}^{m} F_j$, such that they are arranged as a nested sequence:

$$F_m \subset F_{m-1} \subset \cdots \subset F_2 \subset F_1 = F, \tag{7.8}$$

where $F_j = \{z \in Z | g(z) < b_j\}$, with $b_{j+1} > b_j$.

When the event F_j holds then $\{F_{j-1}, \ldots, F_1\}$ also hold, and hence:

$$P\left(F_j | F_j, \ldots, F_1\right) = P(F_j | F_{j-1)}) \tag{7.9}$$

And it follows that:

$$P(F) = P\left(\bigcap_{j=1}^{m} F_j\right) = P(F_1) \prod_{j=2}^{m} P(F_j | F_{j-1}) \tag{7.10}$$

where $P(F_j | F_{j-1}) \equiv P(z \in F_j | z \in F_{j-1})$, is the conditional failure probability at the $(j-1)^{th}$ level.

The failure region, which corresponds to the occurrence of the rare event, can be expressed by the intersection of several larger intermediate regions arranged in a nested sequence. In Equation 7.10 the intermediate regions can be chosen so that the conditional probabilities are large despite the failure region being small.

To perform a Subset simulation at the first level a Monte Carlo simulation is made to cover the region of interest and estimate $P(F_1)$.

$$P(F_1) \approx \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{I}_{F_1}({z_0}^{(n)}) \tag{7.11}$$

Where ${z_0}^{(n)} \sim p(z_0)$ and $\boldsymbol{I}_{F_1}({z_0}^{(n)})$ is the indicator function for the region F_1 that assigns a value of 1 when $g\left({z_0}^{(n)}\right) > b_0$, and 0 otherwise.

Conditional sampling is used for the intermediate regions. At each level, particle ‘seeds’ from the previous level are selected to generate more samples via Markov Chain Monte Carlo sampling to generate N dependent samples.

These seeds are chosen by taking the best values from the set simulated at the previous level. The Metropolis Hasting algorithm is used to generate successive chain values for each seed, here the value in each chain starts with the seed value and successive chain values are sampled from a symmetric proposal *pdf* centered on the previous value. This demonstrates perfect sampling: if the value in the chain produces a weaker output than the previous value, it is discarded and the previous value is repeated. At each level this process is repeated, again choosing the best values from the previous level to act as seeds. Equation 7.12 gives the method for conditional sampling for the intermediate regions.

$$P\left(F_j | F_{j-1}\right) \approx \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{I}_{F_j}({z_{j-1}}^{(n)}) \tag{7.12}$$

Where ${z_{j-1}}^{(n)} \sim p(z_{j-1} | F_{j-1})$ and $\boldsymbol{I}_{F_j}({z_{j-1}}^{(n)})$ is the indicator function for the region F_j that assigns a value of 1 when $g\left({z_{j-1}}^{(n)}\right) > b_j$, and 0 otherwise.

Figure 7.1 shows the initial stages of SubSet Simulation, where the best seed values from the first level form the seed values for the second level, with each level moving towards the desired failure region.

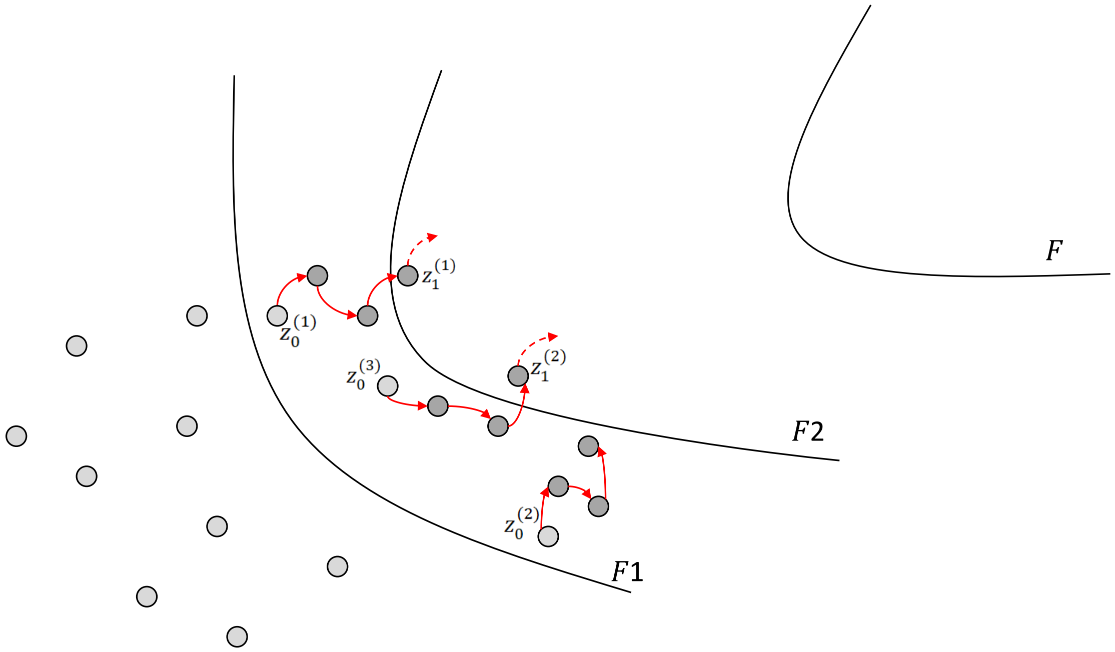


Figure 7.1: A diagram representing the evolution of 'seed' values in Subset Simulation

This method requires a good selection of the proposal *pdf* used to evolve values in each Markov Chain, and the proportion of values to take through from the previous level as seeds. These values must be chosen so that the intermediate regions are large enough to not result in a rare event simulation yet small enough to avoid the need for many simulation levels. Hence, there is a trade-off between the need to explore the entire region and the computational efficiency of the method. It is recommended that the proportion of values chosen at each level should lie between 0.1 and 0.3.

The ABC-SubSim algorithm presented in the next section combines ABC with Subset simulation to decrease the computational effort of ABC.

7.2.4 ABC-SubSim Algorithm

The ABC-SubSim algorithm is presented in [185], and combines Subset Simulation with ABC to improve the efficiency of simple ABC. The combination of these methods results in an algorithm that gives a stepwise improvement to a posterior region over several different levels, until the desired posterior region is found. In effect, the whole region is broadly sampled at a low resolution followed by focused sampling in promising regions at increasingly higher resolutions. This is opposed to a simple ABC methodology where the whole possible region is sampled uniformly resulting in repeated sampling of unnecessary areas.

As with ABC, the ABC-SubSim Algorithm relies the selection of an informative summary statistic. The benefit of this algorithm is an improvement in the computational cost, however, care must be taken to sample sufficiently from the space such that the true posterior region is not passed over and discarded in the early levels. Because of this the algorithm works well where the solution space is ‘smooth’, for example a change in a parameter value does not cause a spike in the metric value. A sufficient summary statistic and metric value must be chosen to facilitate this.

In this algorithm z is defined as $z = (\theta, x) \in Z$, so that $p(z) = p(x|\theta)p(\theta)$. F_j in the subset simulation section is also replaced by a nested sequence of regions $D_{j,}$ for $j = 1, \dots, m$, in Z defined by:

$$D_j = \left\{z \in Z: x \in N_{\epsilon_j}(y)\right\} \equiv \{(\theta, x): \rho(\eta(x), \eta(y)) \le \epsilon_j\} \tag{7.13}$$

Where ρ is a metric based on the summary statistic $\eta(\cdot)$ and the sequence of tolerances $\epsilon_{1,}\epsilon_{2,} \dots \epsilon_m$ is such that $\epsilon_{j+1} < \epsilon_j$ is chosen adaptively and the number of levels, m, is chosen such that ϵ_m is within a specified tolerance.

In this algorithm, the small probability, $P(D_m)$ is expressed as a sequence of larger conditional probabilities, $P(D_j)$ which forms the basis for a Subset Simulation to achieve a faster convergence to a posterior of the desired tolerance. A pseudocode of the algorithm is given below:

Algorithm 2: Pseudocode implementation for ABC-SubSim

```
Inputs
$P_0 \in [0,1]$, {governs the percentage of seeds selected and so that $NP_0$ and $1/P_0$ are integers}
$N$, {the number of samples at each level)
$m$, {the maximum number of simulation levels allowed}
$\epsilon$, {the desired tolerance}
Algorithm
Sample $[(\theta_0^1, x_0^1), (\theta_0^2, x_0^2) \dots (\theta_0^n, x_0^n) \dots (\theta_0^N, x_0^N)]$, where $(\theta, x) \sim p(\theta)p(x|\theta)$
for $j = 1, \dots, m$ do
        for $n = 1, \dots, N$ do
                Evaluate $\rho_j^{(n)} = \rho(\eta(x_{j-1}^n), \eta(y))$
        end for
        Renumber $[(\theta_0^n, x_0^n), n: 1, \dots, N]$ so that $\rho_j^{(1)} \le \rho_j^{(2)} \le \cdots \le \rho_j^{(N)}$
        Fix $\epsilon_j = \frac{1}{2}(\rho_j^{(NP_0)} + \rho_j^{(NP_0+1)})$
        for $k = 1, \dots, NP_0$ do
                Select a seed $\left(\theta_j^{(k),1}, x_j^{(k),1}\right) = \left(\theta_{j-1}^{(k)}, x_{j-1}^{(k)}\right)$
                Run Modified Metropolis Algorithm to generate $1/P_0$ states of a Markov Chain lying
                in $D_j$: $[\left(\theta_j^{(k),1}, x_j^{(k),1}\right), \dots, \left(\theta_j^{(k),1/P_0}, x_j^{(k),1/P_0}\right)]$
        end for
        Renumber $\left[\left(\theta_j^{(k),i}, x_j^{(k),i}\right): k = 1, \dots, NP_0, i = 1, \dots, \frac{1}{P_0}\right]$ as $\left[(\theta_j^1, x_j^1), \dots, (\theta_j^N, x_j^N)\right]$
        if $\epsilon_j \le \epsilon$ then
                End algorithm
        end if
end for
```

A natural outcome of this algorithm is a measure of the acceptability of the model class, this is given in Equation 7.14. This enables the testing of multiple models M_{S_k}in order to test the ABC evidence $P_\epsilon(y|M_{S_k})$ of each model class.

$$P_\epsilon(y|M_{S_k}) = P(D_m|M_{S_k}) = P(D_1) \prod_{j=2}^{m} P(D_j|D_{j-1}) \approx P_0^m \tag{7.14}$$

7.3 Proposed Reduction methodology

This section presents the novel Petri net reduction methodology, proposed in this thesis. The aim of this methodology is to develop a technique for a very large Petri net, referred to in this chapter as the Reference Petri net, M_R, to be represented by a smaller Petri net, M_S referred to as a Reduced Petri net. This methodology also provides a comparative measure of the goodness of representation of a variety of reduced Petri nets to the reference Petri net to allow an informed decision on the model class selection.

This methodology uses Bayesian parameter updating to allow the reduced Petri net to approximate the behaviour of the Reference Petri net. The place, or places, representing the key outputs of the reference Petri net are identified. The corresponding place, or places, in the reduced Petri net are also identified. The marking sequence of these comparison places forms the basis for which the reference Petri net and reduced Petri net can be compared. It is important to choose effective comparison places that hold the same meaning in each Petri net and contain the required information from the reference Petri net.

The similarity of the marking sequence of the comparison places is measured by a summary statistic, $\eta(\cdot)$.The summary statistic should be chosen depending on the application to give as much information as possible for each individual problem.

Initially the reference Petri net is defined, along with the comparison places, $\boldsymbol{p}_{C_R}$. The reference signal is gained by tracking the marking of the comparison place, or places, and this is found via Monte Carlo simulation of the reference Petri net. Following this the reduced model is proposed. In this stage the places, $\boldsymbol{p}_{C_S}$, in the reduced Petri net that correspond to the comparison places in the reference Petri net must be defined.

The firing of the n transitions in the reduced Petri net are governed by distributions $\boldsymbol{d}_s = \{d_1, d_2 \dots d_n\}$ each distribution, d_T is governed by a set of m parameters, $\boldsymbol{\theta}_T = \{\theta_1, \theta_2, \dots, \theta_m\}$. Hence, the reduced Petri net with n transitions is governed by a set of parameters, $\Theta \ = \{\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \dots, \boldsymbol{\theta_n}\}$. The aim of this method it to update the parameters governing the reduced Petri net in order to approximate the output of the reference Petri net. For ease of fitting, a subset of these parameters can be selected for updating, known hereafter as the fitting parameters, $\Theta_f \subset \Theta$. Various fitting parameters can be tested to consider their impact on the final approximation made by the reduced model.

For any proposed set of parameters, Θ_i, a response signal for the reduced Petri net can be obtained by tracking the marking pattern of the comparison places in the reduced Petri net, $\boldsymbol{p}_{C_S}$. Again, this is done via Monte Carlo simulation of the reduced Petri net.

A sufficient metric, ρ, based on the summary statistic, $\eta(\cdot)$, is defined to compare the response signal from the reference Petri net and the reduced Petri net. This metric quantifies the similarity between the reference Petri net and the reduced Petri net for the parameters used and allows for the implementation of the ABC-SubSim algorithm to update the parameters.

Initially, a prior region in the parameter space is defined and N sets of parameter values are sampled uniformly from this space and used as the prior parameter set, $\boldsymbol{\Theta} = \{\Theta_1, \dots, \Theta_N\}$, for the reduced Petri net model. For each parameter set, Θ_i, the reduced Petri net is simulated via Monte Carlo simulation and the corresponding metric value calculated by comparison of the response signal from the reduced Petri net with the response signal of the reference Petri net. The prior parameter sets corresponding to the metric values representing a close approximation to the reference Petri net are selected as the seed values for the first level of the ABC-SubSim algorithm. The number of parameter sets selected depends on the size of P_0, with $P_0 = 0.2$ representing a selection of the top 20% of values that correspond to the closest response signals of the two Petri nets.

Similarly, at each level of the ABC-SubSim algorithm, parameter seed values are chosen from the previous level that resulted in the lowest values for the metric, ρ. Within each level of the ABC-SubSim algorithm, seed values are evolved in a Markov Chain by a proposal *pdf* within a range $(0, L_j]$, where L_j is a defined upper limit for parameter θ_j. Each evolved parameter is accepted or rejected to form the next entry in the Markov Chain. If the evolved parameter results in a metric value that is within the tolerance for that level of the algorithm then the evolved parameter is accepted, otherwise the previous parameter is repeated. Each new evolved parameter is found from the previous entry in the Markov Chain and not from the initial seed value. The sets of proposed parameters at each level of the ABC-SubSim algorithm condenses sequentially to a posterior region.

The required accuracy of the approximation of the reduced Petri net to the reference Petri net must be considered. An increase in the number of parameters selected from the reduced Petri net for parameter fitting improves the ability of the reduced Petri net to approximate the reference Petri net. However, for large Petri nets there is a trade-off between the choice of the number of parameters updated and the computational effort. In the examples in this chapter it has been sufficient to update two parameters in the reduced Petri nets, corresponding to the mean values for two transitions, in order to obtain a reasonable approximation to the reference Petri net.

A natural outcome of this algorithm is a measure of the acceptability of the model class, this is given in Equation 7.14. This enables the testing of multiple models, M_{S_k}, in order to obtain the relative evidence, $P_\epsilon\left(y \middle| M_{S_k}\right)$, of each model. The paper "Approximate Bayesian Computation by Subset Simulation" suggests setting $P_0 = 0.2$, which represents the proportion of the values to be retained as seeds through each level, this was applied in the examples in this chapter[185].

There are several choices when applying this methodology that are dependent on the modelling situation:

- A reduced Petri net must be chosen that retains the capability to sufficiently model the situation. A relative measure of the success of the approximation made by the reduced Petri net is given in Equation 7.14.
- The comparison places must be chosen so that they contain the same meaning across both the reduced Petri net and reference Petri net.
- A sufficient summary statistic must be chosen in order to compare the signals from the reference Petri net and reduced Petri net. With changes in the parameter values, the resulting metric should not contain singularities. If there is a lack of convergence to a good posterior then the summary statistic may not contain sufficient information.
- A decision must be made on which parameters to vary within the reduced Petri net; it is possible to fit multiple parameters but at increased computational effort. The impact of fitting different parameters can be tested.
- A prior parameter region should be chosen to sample a sufficient space, sampling uniformly from a prior region is suggested to evenly cover the potential region of the posterior.
- The proposal *pdf* used to evolve the seed values in the Markov Chains must be chosen to acceptably cover the parameter region. The success of this choice can be tested by considering the acceptance rate of each value generated from the seeds in the Markov Chains. The Proposal *pdf* can change at each level of the ABC-SubSim algorithm to optimize convergence to the posterior region.

A schematic for the Petri net reduction methodology is given in Figure 7.2.

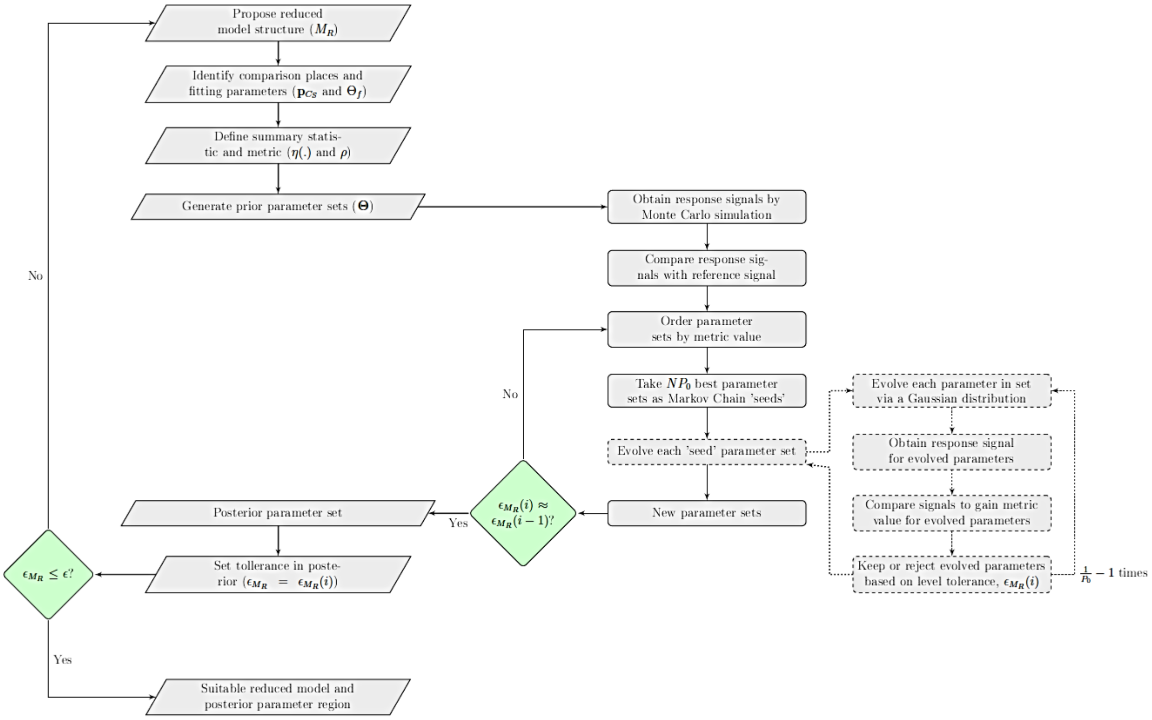

Propose reduced model structure (M_R)
Identify comparison places and fitting parameters ($\mathbf{p}_{Cs}$ and Θ_f)
Define summary statistic and metric ($\eta(.)$ and ρ)
Generate prior parameter sets ($\boldsymbol{\Theta}$)
Obtain response signals by Monte Carlo simulation
Compare response signals with reference signal
Order parameter sets by metric value
Take NP_0 best parameter sets as Markov Chain 'seeds'
Evolve each 'seed' parameter set
Evolve each parameter in set via a Gaussian distribution
Obtain response signal for evolved parameters
Compare signals to gain metric value for evolved parameters
Keep or reject evolved parameters based on level tolerance, $\epsilon_{M_R}(i)$
$\frac{1}{P_0} - 1$ times
New parameter sets
$\epsilon_{M_R}(i) \approx \epsilon_{M_R}(i-1)$?
No
Yes
Posterior parameter set
Set tollerance in posterior ($\epsilon_{M_R} = \epsilon_{M_R}(i)$)
$\epsilon_{M_R} \leq \epsilon$?
No
Yes
Suitable reduced model and posterior parameter region

Figure 7.2: A schematic of the reduction methodology

This method can be used to update several parameters for the reduced Petri net so that the signal outputs over time of the reference Petri net can be closely replicated. The next section gives a discussion of the metric choice followed by two applications of the method.

7.4 Discussion of metric choice

A suitable metric value must be chosen for implementation of the ABC-SubSim algorithm to the comparison of two Petri nets. A review of metric options was completed to identify potential metric choices. Several different metric choices are presented in this section. These were then tested with a sample signal to observe their behaviour. The metrics and distance measures given in this section are given for two signals $\boldsymbol{a}, \boldsymbol{b}$.

The paper “A Framework for Comparing Models of Computation” [186] aims to create a ‘meta model’ that allows some properties of different computational models to be compared. The systems considered include: discrete event systems, dataflow, rendezvous-based systems, Petri nets, and process Petri networks. The concept of assigning a ‘Tag’ to each event in a system is introduced. These tags contain information about the sequence of occurrence of events and can either represent the abstract ordering of events or the time order at which events occur depending on the physicality of the system. The events in the system can be grouped into signals and each of these signals contained in a set of signals specific to that model. In a Petri net a signal can be gained from each place or transition. A distance measure is presented for the comparison of two models based on the signals that are generated by each.

$$d(\boldsymbol{a}, \boldsymbol{b}) = \frac{1}{2^{\tau}} \tag{7.15}$$

Where τ is:

a. The smallest tag at which there is a difference between the signals generated by the models.
b. Infinity if the two signals are identical

c. Minus infinity otherwise

This metric allows the comparison of two models with a focus on the time at which the events begin to occur in a different order between the models. This approach is useful for comparing models of different types to ensure that the logic of the system is preserved when changing the modelling tool. For the application of a metric for this problem it is less applicable as there has to be a consideration of the degree of difference between the models, not solely the time at which a difference occurs. For instance, signals of the two models that greatly diverge at a point and never return to similar results would receive the same metric value as signals of a different model pairing that has a slight deviation at the same tag point.

There was limited literature directly related to comparison of signals generated between two separate Petri net models, however a wider review of metric spaces and signal comparison was carried out.

The book, “Encyclopaedia of Distances” [187], gives an extensive description of different measures of distance and their applications. These includes the Power Distance, given in Equation 7.16.

$$\rho_{p,r}(\boldsymbol{a},\boldsymbol{b}) = \left(\sum_{i=1}^{n}(a_i - b_i)^p\right)^{\frac{1}{r}} \tag{7.16}$$

When $(p,r) = (2,2)$ this corresponds to the Euclidian Distance, when $(p,r) = (1,1)$ this gives the Manhattan distance.

Several measures given include the Natural Metric for real numbers, the M-relative Metric and the Janous-Hametner Metric. The latter two distance measures are weighted versions of the first. These are given below:

$$d_N(a_i,b_i) = |a_i - b_i| \tag{7.17}$$

$$d_{M-rel}(a_i,b_i) = \frac{|a_i-b_i|}{a_i b_i} \tag{7.18}$$

$$d_{J-H}(a_i,b_i) = \frac{|a_i-b_i|}{(a_i+b_i)^t}\ where\ t\ can\ be\ chosen \tag{7.19}$$

A further measure is the Sierpinski Metric, which is similar to a weighted Hamming Distance. Both the Hamming Distance and this distance measure are given below:

$$d_{Ham}(a_i,b_i) = \begin{cases} 1\ if\ a_i \neq b_i \\ 0\ if\ a_i = b_i \end{cases} \tag{7.20}$$

$$d_{Si}(a_i,b_i) = \begin{cases} 1 + \frac{1}{a_i+b_i}\ if\ a_i \neq b_i \\ \quad 0\ if\ a_i = b_i \end{cases} \tag{7.21}$$

The Hamming distance represents the number of changes required to convert one sequence into another. These distances can be combined in the following ways to form a metric for the sequence overall:

$$\rho_{seq}(\boldsymbol{a},\boldsymbol{b}) = \sum_i^n d(a_i,b_i) \tag{7.22}$$

$$\rho_{seq}(\boldsymbol{a},\boldsymbol{b}) = \sqrt{\sum_i^n d(a_i,b_{i_i})^2} \tag{7.23}$$

The intersection distance is given in Equation 7.24 for signals $\boldsymbol{a},\boldsymbol{b}$.

$$\rho_I(\boldsymbol{a},\boldsymbol{b}) = 1 - \frac{\sum_i min\{a_i,b_i\}}{min\{\sum_i a_i,\sum_i a_i\}} \tag{7.24}$$

The Symmetric Chi-Distance is defined by Equation 7.25.

$$\rho_{\chi}(\boldsymbol{a},\boldsymbol{b}) = \sqrt{\sum\nolimits_i \frac{\bar{a}+\bar{b}}{n(a_i+b_i)}(\frac{a_i}{\bar{a}} - \frac{b_i}{\bar{b}})^2} = \sqrt{\sum\nolimits_i \frac{\bar{a}+\bar{b}}{n(\bar{a}\bar{b})^2}\frac{(a_i\bar{b}-b_i\bar{a})^2}{a_i+b_i}} \qquad (7.25)$$

A metric known as Centroid Linkage is also presented. This metric clusters values in sequences around the point of interest and takes their average. These averages are then compared for each point in the sequences. The Euclidean distance of these averages can then be found. This method may be especially useful as it filers small time differences. For two signals $\boldsymbol{a}$ and $\boldsymbol{b}$, where there are sliding regions of length m and midpoint i, for which $\boldsymbol{a}_m^i$ and $\boldsymbol{b}_m^i$, which are subsections of the full signal, are members and where $\widehat{a_m^{i}}$ and $\widehat{b_m^{i}}$ are the mean values of each region, the Centroid Linkage for each midpoint is given by :

$$d_i = \left|\widehat{a_m^{i}} - \widehat{b_m^{i}}\right| \qquad (7.26)$$

Where $\widehat{a_m^{i}} = \frac{1}{m}\sum_{k=i-\frac{m}{2}}^{i+\frac{m}{2}} a_k$ and $\widehat{b_m^{i}} = \frac{1}{m}\sum_{k=i-\frac{m}{2}}^{i+\frac{m}{2}} b_k$ for members a_k and b_k of $\boldsymbol{a}$ and $\boldsymbol{b}$. These midpoints can be then combined as in Equation 7.22 or Equation 7.23 to give a metric for the whole signal.

The paper "Alignment-free sequence comparison-a review" [188] considers different methods for comparing discreet sequences. The methods of measuring similarity are grouped into two types: methods that are based on word frequency and methods that do not require splitting the sequence into fixed length sections. The first type includes metrics such as the Euclidean distance, weighted Euclidean distance, correlation coefficient, covariance and the relative entropy. These are found based on vectors for the frequency of word occurrence. These methods require a finite 'alphabet' from which different 'L-tuples' (similar to words) can be extracted, the frequency of each of these L-tuples is then found in both sequences and stored in vectors. These vectors then form the basis for the comparison of the two sequences. For the application to this problem these 'word' based methods may not be ideal. Firstly, there may not be a finite alphabet as the values in the sequence could range anywhere over the real numbers. Secondly, the distance between individual elements in the sequence is not considered. This is of value when comparing sequences of letters as it is not necessary to have a measure of how close one letter is to another, however when considering a sequence of discreet numbers then the 'closeness' of numbers should be considered.

The second group of methods are based around Kolmogorov complexity theory and scale-independent representation of sequences by iterative maps. The metrics here do not require a specific length of the L-tuples to be defined. Universal Sequence Maps (USM) are founded on Chaos theory and splits the sequence into regions, the difference in the values in these regions is then compared. For instance consider two regions, $a = (a_1, \ldots, a_i),\ b = (b_1, \ldots, b_i)$, each belonging to a sequence then the USM co-ordinates are:

$$d^{USM}(a,b) = -\log_2(\max\nolimits_i|a_i - b_i|) \qquad (7.27)$$

$$\rho^{USM}(\boldsymbol{a},\boldsymbol{b}) = \frac{1}{\sum_j d^{USM}(a,b)_j}\ where\ there\ are\ j\ regions \qquad (7.28)$$

Kolmogorov complexity theory considers the relative decrease in complexity or conditional complexity as a measure of similarity of sequences. These two methods have more potential for adaptation for application to a discreet numerical sequence of positive numbers instead of collections of symbols.

The mutual information between two discreet random variables X, Ym that are jointly distributed according to $p(x,y)$ is given by Equation 7.29 [189]

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \qquad (7.29)$$

The mutual information that is in common to both X and Y. The marginal for the mutual information for a discreet signal can be found via normalising a 2d histogram and summing over the other variable. A distance measure can be formed via:

$$d_{MI}(a,b) = \frac{N}{I(\boldsymbol{a},\boldsymbol{b})} \ for\ some\ scaling\ value\ N \qquad (7.30)$$

A sample signal was generated at random to aid in the metric decision making process for this case. For the Petri nets in the examples presented in the latter parts of this chapter, the signal from each Petri net is expected to follow a behaviour with some variation and a maximum value of one. To mimic this, the sample reference used in this analysis were formed of uniform random numbers between zero and one. This signal is referred to as ref in Figure 7.3.

To test the metric choice in a controlled manner the following sample signals were generated for comparison with the reference signal:

- Signal cp1: An exact copy of the reference signal
- Signal cp2: An exact copy of the reference signal with one value altered
- Signal cp3: An exact copy of the reference signal with three values altered
- Signal cp4: A copy of the reference signal with a 1 unit time lag
- Signal cp5: A copy of the reference signal for 0 to 20 time units followed by a randomly generated signal from 20 to 40 time units.
- Signal cp6: A randomly generated signal

Figure 7.3 gives a plot of these signals, cp1 through to cp6, in comparison to the reference signal.

Following this eleven different metric choices were used to compare the signals. The metric value for each of the signals is given in Figure 7.4 and Figure 7.5, with the Euclidian distance provided as a comparison measure in each case. There are some differences that can be seen in the results. Firstly, there is a difference in the way that the metrics judge the similarity of signal cp4 to the reference signal. In most cases signal cp4 is judged to be further from the reference signal than signal cp5 and for some of these cases the difference is large. For this application, a large bias against signals with a time lag is detrimental to the parameter fitting. This is because in the Petri net application, with a small interval between signal entries, a lag of a low number of steps causes minimal change to the output. The metric choice should reflect this. Secondly, for some of the metrics there is some difficulty distinguishing between the similarity of Signals cp1, cp2 and cp3 to the reference signal. And, for some of these metrics the distance between signal cp1 and the reference signal does not become zero. For the application in this chapter the metric must be able to distinguish between the closer signals to enable accurate parameter fitting in the latter stages. From this analysis a metric based on the Centroid Linkage was selected for the parameter fitting.

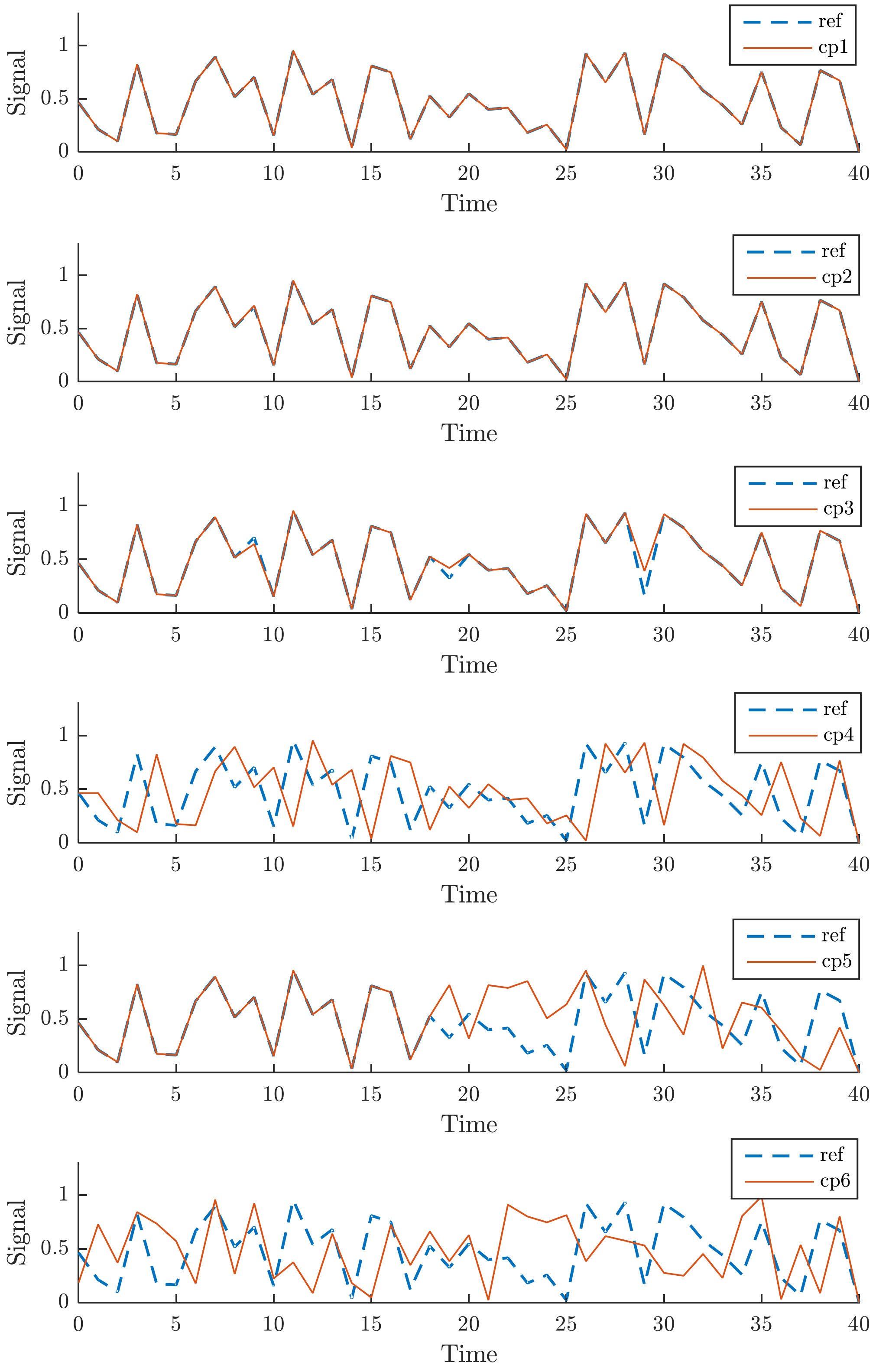


Figure 7.3: Sample signals to form the basis of metric analysis

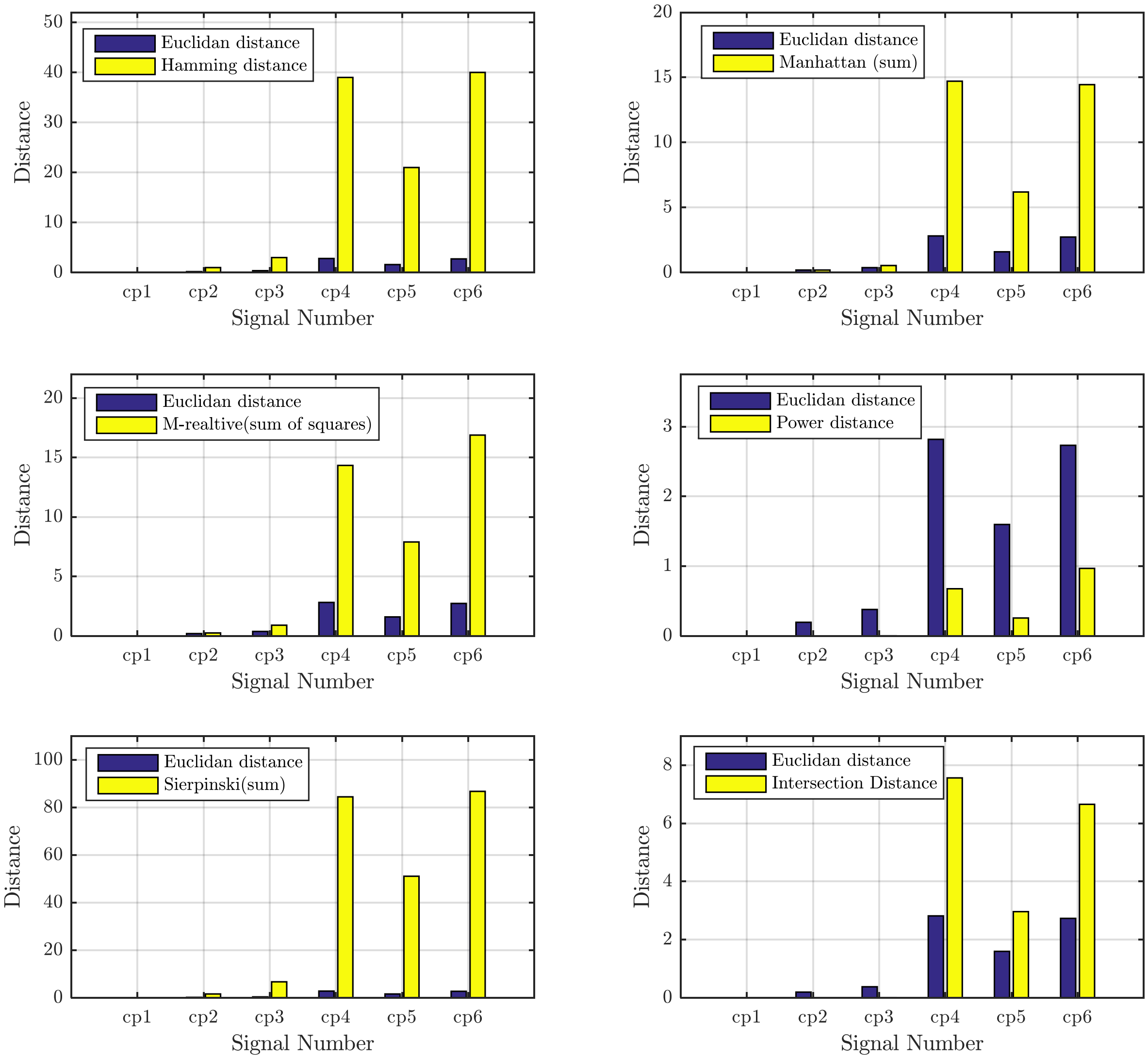


Figure 7.4: Metric test for the sample signals (part I)

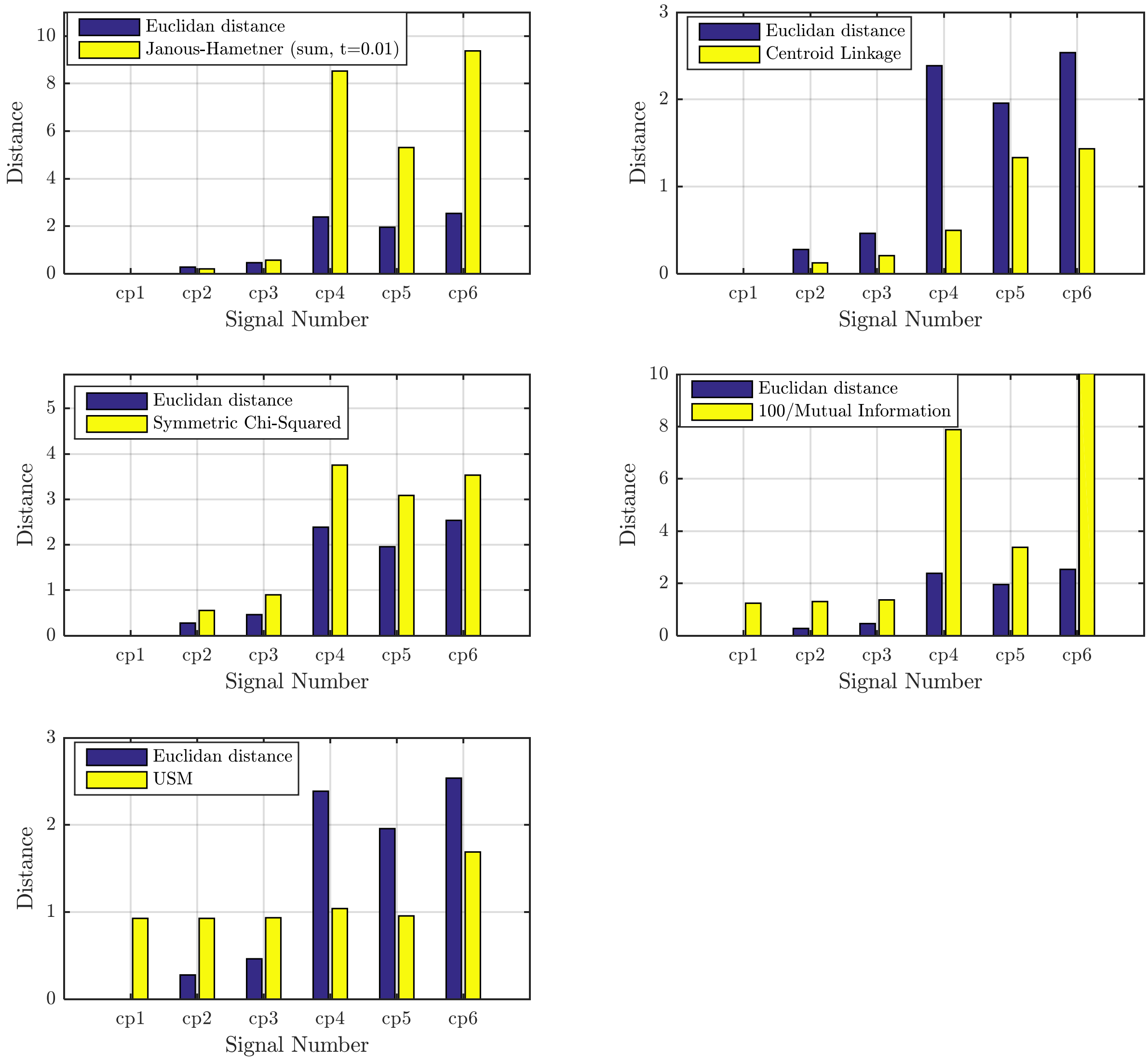


Figure 7.5: Metric test for sample signals (part II)

There is a wide choice of measures that can be used in this methodology in order to update the parameters of a reduced Petri net model. For the specific models a sample signal can be generated in order to form this sort of high level analysis of the possible distance measures in order to narrow down the range of options. Once a metric has been identified as a possibility the algorithm must be completed to test the feasibility of that metric choice. If the metric is unsuitable resulting in a poor convergence of the posterior parameter region then an alternative metric can be tested.

7.5 Examples

7.5.1 Example 1

A simple application of the reduction method is given in this section. For this first example, Figure 7.6 gives the reference Petri net, M_R on the left hand side and the reduced Petri net, M_S on the right hand side.

The reference Petri net represents the failure of a repairable component with three states: the working state represented by $P1$, an intermediate poor state that is repaired if inspection of the component reveals the state, represented by $P2$, and a third revealed failed state, represented by $P3$. Ageing of the component resulting in these states is represented by transitions $t1$ and $t2$, repair of the components is represented by transitions $t3$ and $t4$, and inspection of the component by transitions $t5$ and $t6$. This resembles a simplified model in comparison to models in previous chapters of this thesis. More complex examples follow this one.

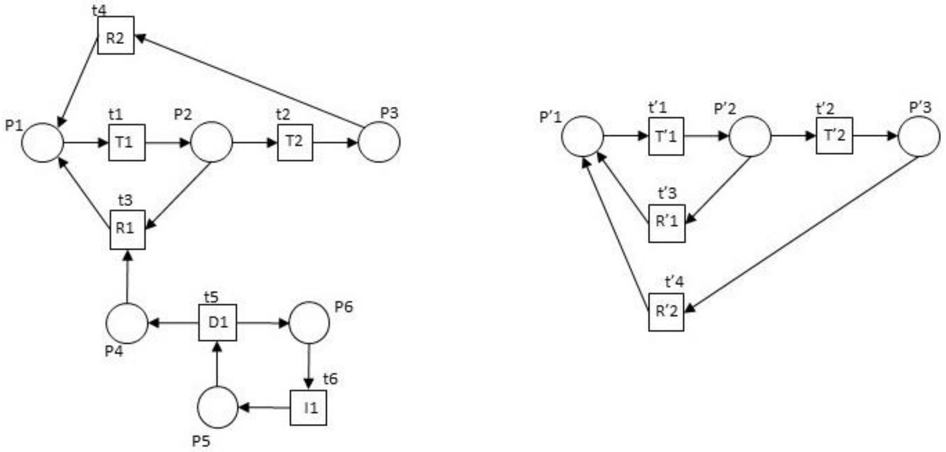


Figure 7.6: The reference Petri net (left hand side) and the reduced Petri net (right had side) for the first example

In the reduced Petri net model, the inspection loop is incorporated into the transition $t'3$, with $P'1$ representing the working state, $P'2$, representing the intermediate poor state and $P'3$ representing the revealed failed state. The reduced model comes with a compromise of loss of information but more efficient simulation. Transitions $t'3$ and $t'4$ represent the repair of the component from the intermediate and failed states respectively. The transitions in both Petri nets are governed by normal distributions with the parameters $\theta_t = (\mu_t, \sigma_t)$. The parameters governing the transition times in the reference Petri net are given in Table 7.1.

Transition	**Distribution**	$\boldsymbol{\mu_t}$	$\boldsymbol{\sigma_t}$
t1	Normal	20	2
t2	Normal	10	2
t3	Normal	2	0.5
t4	Normal	1	0.5
t5	Normal	0.5	0.0001
t6	Normal	2	0.5

Table 7.1: The initial distributions for the reference Petri net in Figure 7.5

Places $P2$ and $P'2$ were chosen as the comparison places in this example. They contain the corresponding information on the intermediate state of the system. The Centroid Linkage was chosen as the basis of a summary statistic, as it provides a good reduction in the prior distribution with each level of the ABC-SubSim algorithm. The Centroid Linkage is given in Equation 7.26.

The metric used in the ABC-SubSim algorithm is chosen to minimise the a function based on the Centroid Linkage, the metric, ρ, is given in Equation 7.31 where d_i is the Centroid Linkage found with Equation 7.26.

$$\rho = \sqrt{\sum_i d_i^2} \qquad (7.31)$$

In this example, the mean of the distributions for $t'1$ and $t'3$, $\mu'_1$and μ'_3, in the reduced Petri net, were chosen for the parameter fitting. Multiple parameter combinations were tested, with these values giving demonstrating the highest impact on the model outputs. During the parameter fitting process, the remaining parameters in the reduced model were kept consistent with the parameters in the reference model: the parameters governing transitions $t'4$ and $t'2$, were the same as those governing transitions $t4$ and $t2$, and the standard deviations of transitions $t'1$ and $t'3$ were kept consistent with those of transitions $t1$ and $t3$. The methodology can be extended to update all parameters in the reduced model, at an increased computational complexity. This was deemed unnecessary for this example, due to the strong approximation obtained when updating two parameters only.

A prior region was defined for each of the parameters that were updated, $\mu'_1$and μ'_3, as a uniform distribution in two dimensions varying from zero, non-inclusive, to twice the mean value of the normal distribution assigned to the corresponding transition in the reference Petri net. For example, for the parameter μ'_1, the prior region existed in the range $(0{,}2\mu_1]$, where μ_1 is the mean of transition t1. For the parameter μ'_3, the prior region existed in the range $(0{,}2\mu_3]$, where μ_3 is the mean of transition t3. Within this prior region 2000 seed values were generated. Within each level of the ABC-SubSim algorithm, each seed values is evolved via a proposal *pdf*. This proposal *pdf* can be chosen adaptively for each level of the algorithm in order to gain a suitable acceptance rate.

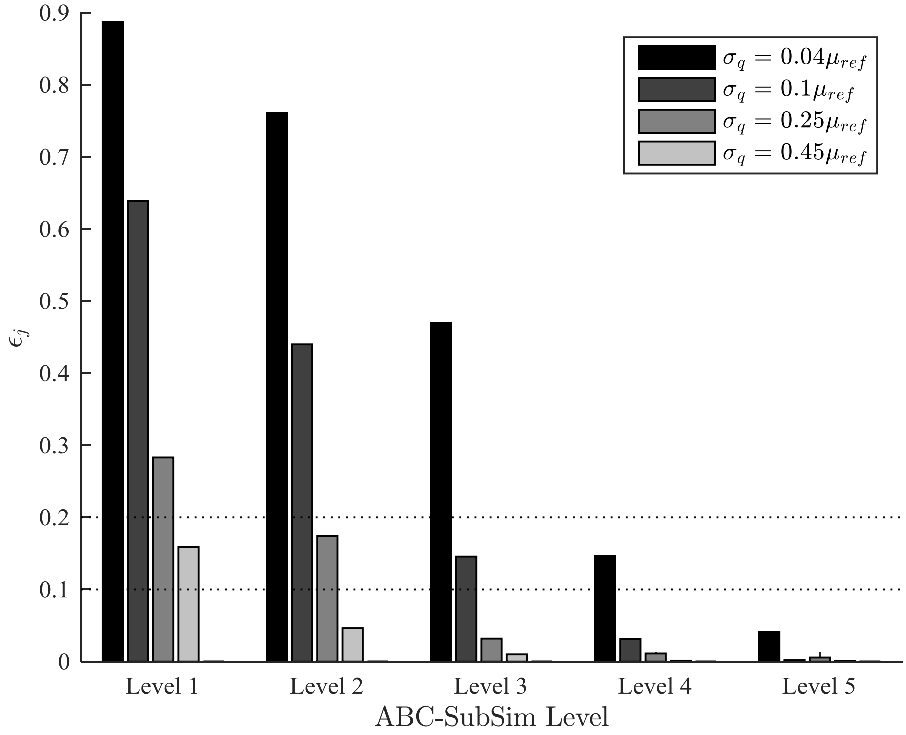


Figure 7.7: Testing the acceptance rate for different Gaussian distributions

In order to evolve parameters within this method, two proposal *pdfs* are required. One of these evolves the solution in the direction of the first parameter and one evolves the solution in the direction of the second parameter. A Gaussian distribution, with a mean of zero, was chosen for each proposal *pdf*. This allows new solutions to be generated at varying distances from the current solution, in two parameter dimensions. In order to achieve an acceptable level of convergence, it is expected that the standard deviation of each of these Gaussians will be somewhat proportional to the magnitude of the parameter direction it is associated with. For example, in the direction of a parameter with a larger value and hence a larger prior region, a larger variation in the step size used to explore the region will be required. Under this assumption, various proposal *pdfs* were tested. Here, each proposed standard deviation for both Gaussians was defined relative to the central point of the prior region for each parameter. The proportion of this central point was varied and the impact of this on the acceptance rate of proposed solutions was tested.

Figure 7.7 gives the graph used to inform the choice of the standard deviation of the Gaussian distribution used for the proposal *pdfs* at each level of the ABC-SubSim algorithm. In this graph, ϵ_j is the acceptance rate of evolved parameters, for Gaussian distributions, centred on zero, with different standard deviations, σ_q. At each level of the algorithm, an acceptance rate of approximately 0.2, and above 0.1, should give optimal convergence to the posterior region.

It can be seen that for the proposal *pdfs* in this example, the acceptance rate reduces as the ABC-SubSim level increases. This corresponds to the condensing of the parameter space to the posterior region. If the standard deviation of the proposal *pdf* is too large, many of the evolved parameters in the Markov Chain are rejected, resulting in a low acceptance rate, a lack of exploration of the parameter space and a poor convergence to the posterior region. Likewise, if the standard deviation is too small, too many of the evolved parameters are accepted resulting in a high acceptance rate, but the evolved parameters are close to the seed values, again resulting in a lack of exploration of the parameter space and a poor convergence to the posterior region.

From the analysis presented in Figure 7.7, with the view of adaptively choosing the posterior *pdf* in order to give a reasonable acceptance rate for each level of the algorithm, the following proposal *pdfs* were chosen to evolve the parameters at each level of the ABC-SubSim algorithm:

- At level 1 of the ABC-SubSim algorithm Gaussian distributions with parameters $\mu = 0, \sigma = 0.45\mu_r$were selected.
- At level 2 of the ABC-SubSim algorithm Gaussian distributions with parameters $\mu = 0, \sigma = 0.25\mu_r$ were selected.
- At level 3 of the ABC-SubSim algorithm Gaussian distributions with parameters $\mu = 0, \sigma = 0.1\mu_r$ were selected.
- At level 4 of the ABC-SubSim algorithm Gaussian distributions with parameters $\mu = 0, \sigma = 0.04\mu_r$were selected.

Where $\mu_r$is the midpoint of the prior region for each parameter, which is the mean of transition t1 in the reference Petri net, μ_1, for the first updated parameter and the mean of transition t3 in the reference Petri net, μ_3, for the second updated parameter. These Gaussian *pdfs* evolve each parameter within each Markov chain, allowing each evolved parameter to step away from the current location. This adaptive choice of the proposal *pdfs* allows larger steps to be taken at the initial stages for more effective exploration of the solution space, with smaller steps taken as the region condenses to the posterior region.

Figure 7.8 shows the step-wise reduction in the two-dimensional parameter space to give the posterior region at each level of the ABC-SubSim algorithm, from the prior region which is enclosed by the dotted lines. Here, θ_1 is the parameter corresponding to the mean of the distribution assigned to transition t'1 in the reduced Petri net and θ_2 is the mean of the distribution assigned to transition t'3 in the reduced Petri net. This results in a posterior region from which the value of both parameters can be selected such that the output of the reduced Petri net most closely matches the output of the reference Petri net.

In this figure, each of the seed values used in the ABC-SubSim algorithm, at any of the levels, is represented by a circular mark. Initially, these seeds are spread evenly through the region enclosed by the dotted lines. This prior distribution is not shown here, to improve clarity in the figure. At each level of the parameter updating ABC-SubSim algorithm, the parameters that allow the reduced Petri net to most closely recreate the behaviour of the reference Petri net are discovered, and form seed values for the next level of the algorithm. The figure shows each updated seed value at each level of the algorithm. After Level 4 there was limited reduction in the region that held the seed values, suggesting that the seed values at the 4th level of the algorithm lie within the most accurate posterior region discoverable by this approach. The figure shows the reduction of the region that contains the most suitable parameter values, to a condensed region containing the updated parameter values. These parameter values can be used in the reduced Petri net to recreate the behaviour of the reference Petri net.

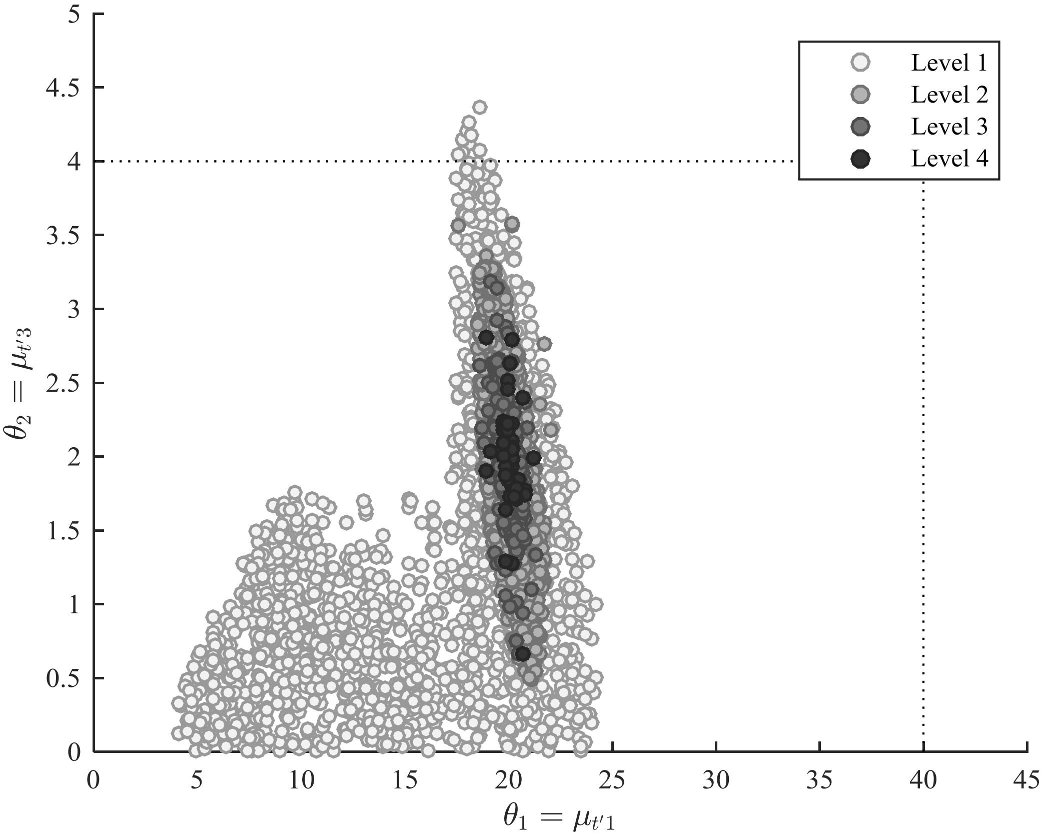


Figure 7.8: Stepwise posterior reduction

Figure 7.9 shows a plot of the signal output from the reference Petri net along with the average signal output generated from the reduced Petri net, with pairwise parameter values found in the posterior region of the 4th level of the ABC-SubSim algorithm. These pairs of parameter values correspond to the darkest circular marks taken from Figure 7.8, and are used to complete the set of input parameters for simulation of the reduced Petri net. This figure shows how the marking of Place P2 for the reference model changes with time, where the time period of interest for simulation of the model is on the x-axis and the probability that Place P2 is marked is on the y-axis. The figure also shows how the marking of Place P’2 of the reduced Petri net model changes, with the time period of interest for simulation of the model, where the input parameters used for the reduced model are those in the discovered posterior region and the fixed values described at the beginning of this example. All parameter pairs from the discovered posterior region are used as input to the reduced model, a simulation in each case is completed, and the average output was taken. This was done to give a robust representation of the discovered posterior region. The figure demonstrates a close approximation of the outputs gained by the reduced model, with updated parameters, to the outputs given by the reference model.

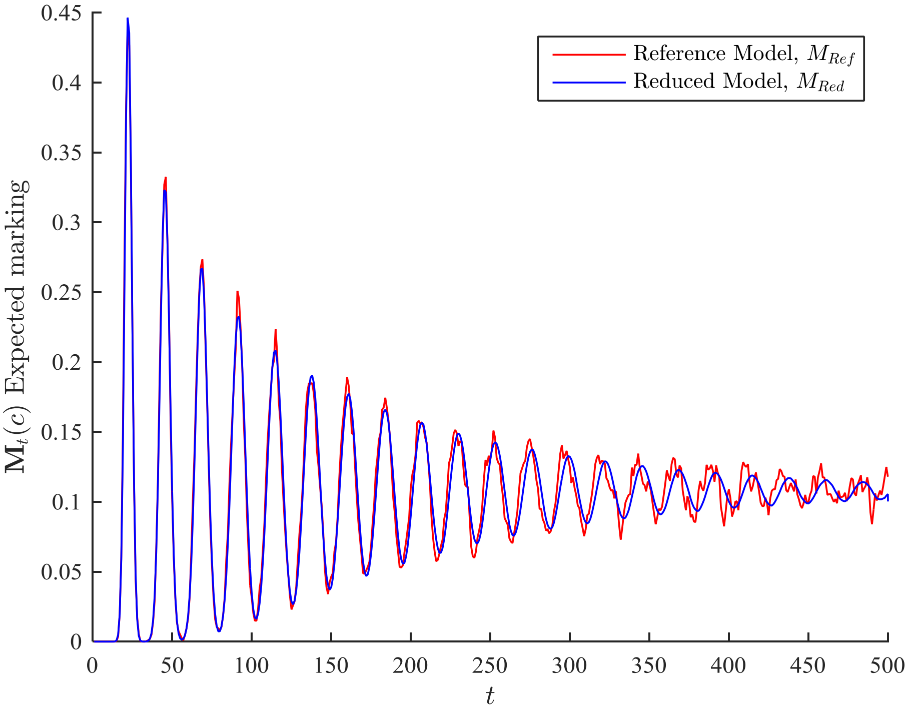


Figure 7.9: A signal plot for the reference Model and the reduced model

The above figure suggests that, in this simple case, the reduced model could be used to replace the reference model, with limited impact on the model outputs. The reference Petri net completed 2000 simulations in 64.432s and the reduced Petri net completed 2000 simulations in 6.603s. If the reduced model were to be used instead of the reference model, this gives a reduction in computational time of just under 90%. The signal outputs from the reduced Petri net, with input parameters in the discovered posterior region, are all within a tolerance of 0.168 in comparison the output from the reference Petri net, when measured via the metric, based on the Centroid Linkage, given in Equation 7.31. This demonstrates the ability of this methodology to reduce the computational time of Petri net models whilst reproducing the desired signal outputs.

7.5.2: Example 2

For the second example application, a larger Petri net model was chosen as the reference model. This is given in Figure 7.10. This reference Petri net model represents a component with states moving from the working state, through two intermediate states, and finally resulting in a failure. The failure can be revealed or unrevealed and there is a probability associated with each. There is an inspection loop to identify any unrevealed failures of the component and once a failure has occurred then repair of the component is scheduled immediately. When the component is in an intermediate state, repair is scheduled following a delay which incorporates the time taken for the state to be identified and the usual wait time for intervention for the component in that condition. For repairs of the component in the first intermediate state, the maintenance action can be enabled if early replacement of the component is activated. In this Petri net, maintenance returns the component to the working state. This reference Petri net resembles the simpler component models from Chapter 5.

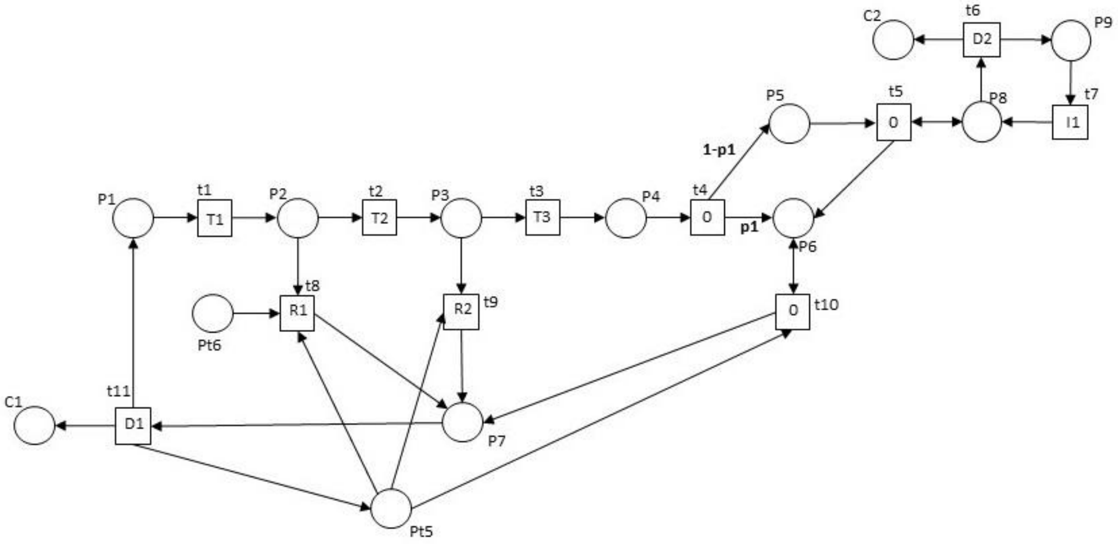


Figure 7.10: The reference Petri net for the second application

In the reference Petri net, Place $P1$ represents the working state of the component, place $P2$ represents the first intermediate state of the component, place $P3$ represents the second, and more severe, intermediate state of the component and place $P4$ represents a failed state of the component. Place $P5$ represents an unrevealed failed state of the component and place $P6$ represents a revealed state of the component. Places $P8$, $P9$ and $C2$ represent an active inspection action, the delay between inspections and the number of completed inspection actions respectively. Place $P7$ represents a scheduled maintenance action and place $C1$ counts the number of competed maintenance actions. Place $Pt5$ represents the availability of resources for the maintenance action and place $Pt6$ allows early replacement to occur if it is marked by a token. The distributions governing the transition firing times and any associated parameter values used in this illustrative model are given in Table 7.2.

Transition	**Distribution**	μ_t	σ_t	**p1**	**1-p1**
t1	Normal	40	10		
t2	Normal	10	2		
t3	Normal	2	0.5		
t4	Probability			0.5	0.5
t5	Immediate				
t6	Normal	1	0.25		
t7	Normal	5	0.5		
t8	Normal	5	0.5		
t9	Normal	1	0.001		
t10	Immediate				
t11	Normal	1	0.5		

Table 7.2: A table giving the distribution of firing times for each transition in the reference Petri net in Figure 7.9, and any associated parameter values

The reduced model is given in Figure 7.12. Different model structures of the reduced model were tested prior to the selection of this reduced structure, reduced Petri nets of a smaller size resulted in weaker approximations of the reference Petri net. The methodology presented in this chapter is a trade-off between higher levels of reductions in model structure, which results in a weaker approximation of the reference Petri net, and the improved computational time that comes with the reduction of the model size. In order to test different model structures the initial requirements of the reduced Petri net were first defined as follows:

- The reduced model must contain a working state
- The reduced model must contain an unrevealed failed and an associated repair action
- The reduced model must have some maintenance preventing an unrevealed failure

In order to arrive at a suitable structure, a highly reduced structure was tested first. The structure of this model is given in Figure 7.11 and is the same reduced structure as that in Example 1 of this chapter. The Petri net given here has three places, corresponding to a working state, P'1, a state encompassing any revealed failures or revealed degraded states, P'2, and an unrevealed failed state, P'3. There are two transitions governing the degradation and failure, t'1 and t'2, and two transitions representing repair, t'3 and t'4. Even for updated parameters, this model structure showed a poor approximation to the reference model structure.

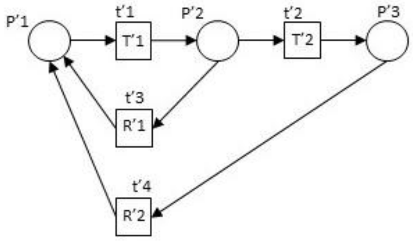


Figure 7.11: The most reduced Petri net structure, to fit defined initial requirements

As an improvement to the smallest model structure, given in Figure 7.11, a failed state encompassing both the revealed failed and unrevealed failed states was added to the Petri net structure, along with a transition to split this state into revealed and unrevealed failed sates, each with an associated maintenance action. This improved structure is given in Figure 7.12. The parameter updating methodology was applied to this structure, and with updated parameters, showed a comparatively better approximation to the reference Petri net model than the smallest model structure. This structure was chosen in this case as it gives a better approximation to the reference model, and the results are presented here. A further option is to add additional places and transitions to the reduced model, to give a more accurate approximation, however this increases the computational cost. Further work is presented in later examples on the comparison of different model structures.

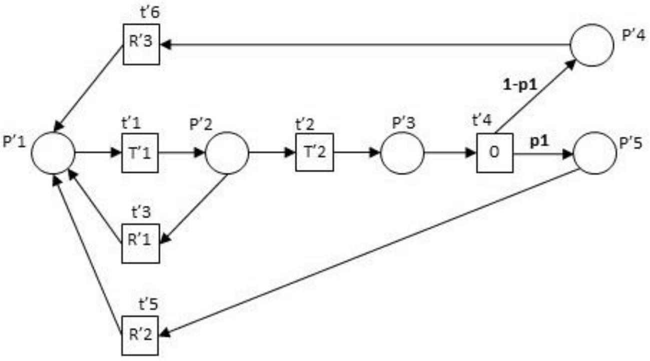


Figure 7.12: The reduced Petri net for the second application

In the reduced Petri net, in Figure 7.12, one of the intermediate states has been absorbed. Here, $P'1$ represents a working state of the component, $P'2$ represents and intermediate state of the component and $P'3$ represents a failed state of the component. $P'4$ represents an unrevealed failed state of the component and in the reduced Petri net the inspection of this state is incorporated into transition $t'6$, which represents maintenance of a component following an unrevealed failure. Similarly, $P'5$ represents a revealed state of the component with the transition $t'5$ relating to repair of the component following a revealed failure. All maintenance of the component prior to failure is incorporated into transition $t'3$. As with the reference Petri net, the maintenance of the component returns it to the working state.

The response signal chosen as the basis for the reduction methodology was the marking sequence of place $P5$ in the reference Petri net and the corresponding place, $P'4$, in the reduced Petri net. This state, in both Petri nets, corresponds to an unrevealed failure of the component. In order to gain the response signal of the reduced Petri net for each proposed parameter set, a Monte Carlo simulation of the reduced Petri net was completed, 1000 time-steps were taken and 1000 runs of the Monte Carlo simulation were completed. The average marking of the place $P'4$, was then found for each time-step.

Various parameters were tested for updating in the reduced model, to consider their impact on the accuracy of the reduced model to approximate the reference model. The standard deviations of the distributions governing each transition had limited impact on the model outputs. In addition, the transitions with shorter mean values, t'5 and t'6, had limited impact on the model outputs. Since transition t'1 and transition t'3 encompass much of the reduced structure from the reference Petri net, the mean values of the parameters governing these transitions, $\mu_{t'1}$and $\mu_{t'3}$, were selected for parameter updating. The standard deviations of the distributions governing these transitions was set to $\sigma_{t'1} = \sigma_{t1} = 10$ and $\sigma_{t'3} = \sigma_{t8} = 0.5$, such that the parameters that are not updated in the reduced Petri net reflect those of places with a similar logic in the reference Petri net. The parameters in transitions t'5 and t'6 were fixed to the same value as those governing transition t11 in the reference model. The parameters in transition t'2 were set to the same value as those governing transition t3 in the reference model. The parameters governing transition t'4 were set to the same value as those governing transition t4 in the reference model.

It is possible to update more parameters for transitions in the reduced Petri net, however this increases the computational cost required to find the posterior region in the parameter space. There is a trade-off between the increased computational cost to find the posterior region and the improvement in the approximation made by the reduced Petri net with a higher number of updated parameters. This is explored further in later examples in this Chapter.

As in the first example, the metric based on the Centroid Linkage, given in Equation 7.31, was used as the summary statistic in the ABC-SubSim algorithm, as it demonstrated that it enabled a good convergence of the proposed parameters to the posterior region. The prior region was defined in the parameter space as zero, non-inclusive, to twice the largest mean of a similar transition in the reference Petri net, μ_r. In this example the mean of the transition $t1$ was used for μ_{ref}^{1} and the mean of transition $t8$ was used for μ_{ref}^{2}. Hence the prior region was defined as: $(0,0) < (\mu_1, \mu_2) \leq \left(2\mu_{ref}^{1}, 2\mu_{ref}^{2}\right)$. Similarly to the first example, 2000 seed values were used in the methodology and initially these values were sampled uniformly from the prior parameter space. Each seed value consists of a pair of parameters, which are then evolved through the ABC-SubSim algorithm, to explore the parameter space, in order to discover the parameters that allow the reduced Petri net to most closely approximate the reference Petri net.

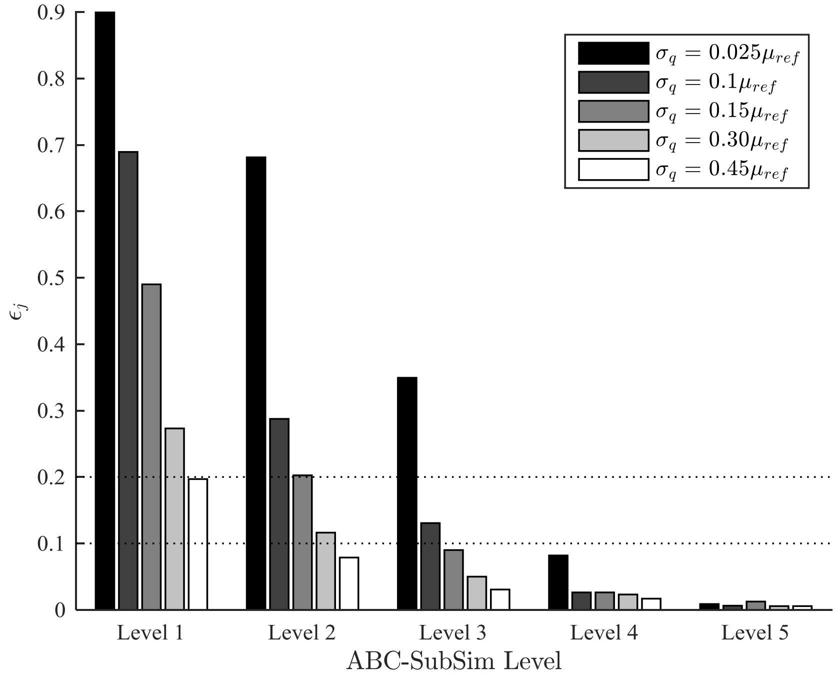


Figure 7.13: Testing different proposal pdfs for each of the ABC-SubSim levels

As with the first example, Gaussian distributions, with a mean of zero, were selected as the proposal *pdfs* for the evolution of the seed values in the Markov Chains in the ABC-SubSim algorithm. A Gaussian with a mean of zero is a valid choice for parameter proposal within the Markov chains, as it is a symmetric function that allows solutions to be proposed with varying closeness to the current accepted solution. The updating of two parameters was completed, and hence two proposal *pdf* were required, one to evolve each entry in the Markov chain in the direction of the first parameter and one to evolve each entry in the direction of the second parameter. The standard deviations of each Gaussian can be optimised to improve convergence to the posterior parameter region.

In this example, various standard deviations of these Gaussian distributions were tested. This allowed the adaptive selection of the proposal *pdf* with each level of the ABC-SubSim algorithm. The standard deviations were varied for each proposal *pdf* such that each standard deviation tested was proportional to the magnitude of the parameter in the direction that it evolves the seed values. In the same way as the analysis was completed in the first example, Figure 7.13 shows the changes in the acceptance rates, ϵ_j, of proposed parameters with the ABC-SubSim level, for Gaussian distributions with different standard deviations, σ_q.

Looking for proposal *pdfs* that result in an acceptance rate of approximately 0.2, resulted in the following selection of the proposal *pdfs* for each level of the ABC-SubSim algorithm:

- At level 1 of the ABC-SubSim algorithm Gaussian distribution were selected to evolve each of the two parameters, with $\mu = 0, \sigma = 0.5\mu_{ref}$.
- At level 2 of the ABC-SubSim algorithm Gaussian distribution were selected to evolve each of the two parameters, with $\mu = 0, \sigma = 0.3\mu_{ref}$.
- At level 3 of the ABC-SubSim algorithm Gaussian distribution were selected to evolve each of the two parameters with $\mu = 0, \sigma = 0.1\mu_{ref}$.
- At level 4 of the ABC-SubSim algorithm Gaussian distribution were selected to evolve each of the two parameters with $\mu = 0, \sigma = 0.01\mu_{ref}$.

Where μ_{ref} is the midpoint of the prior region for each parameter: the mean of the distribution governing transition $t1$ was used for μ_{ref}^1 and the mean of the distribution governing transition $t8$ was used for μ_{ref}^2.

Figure 7.14 shows the step-wise reduction of the seed values at each level of the ABC-SubSim algorithm, until they condense to the posterior region. The axis show the parameter space in two dimensions, with the x-axis showing the parameter space for the mean of transition t'1 and the y-axis showing the parameter space for the mean of transition t'3. The prior region in the parameter space is outlined with a dotted line. With each level of the algorithm it can be seen that the seed values evolve towards a region in the bottom left of the figure. After Level 4 of the algorithm there was limited change to the seed values, with almost all of the proposed parameters rejected. This suggests that the region has condensed to the true posterior region, with the applied framework, and no further improvements can be found via the parameter updating process.

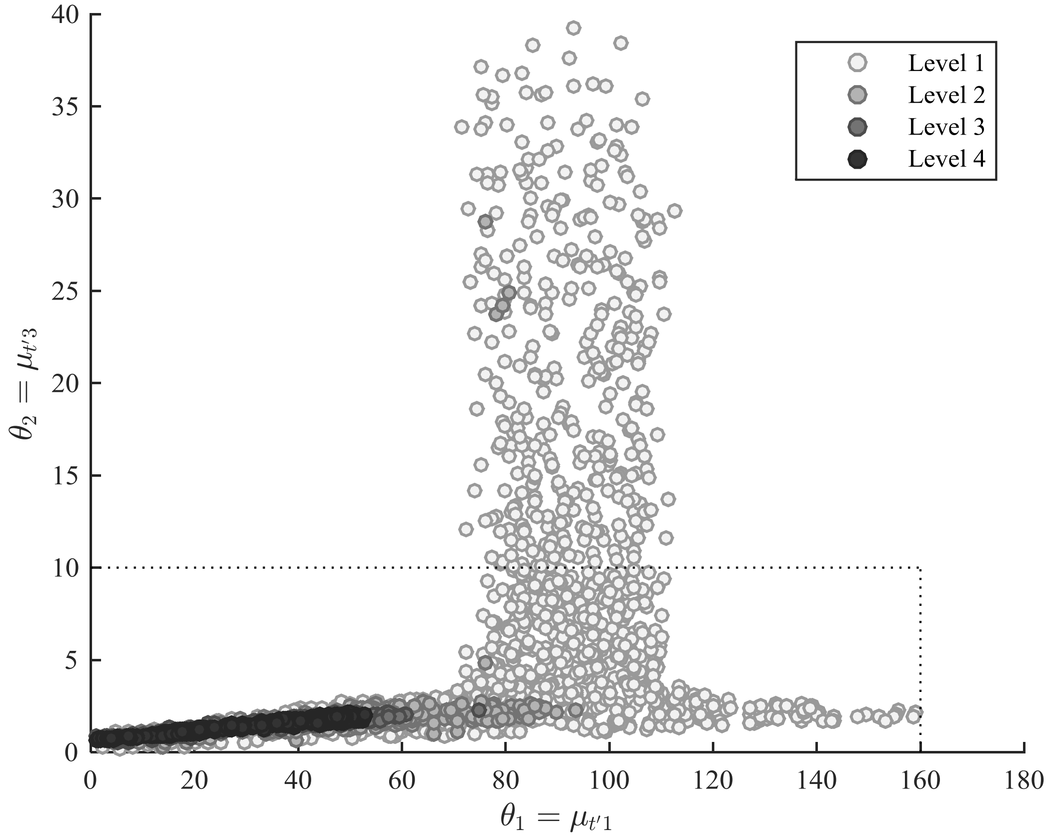


Figure 7.14: The stepwise reduction of the posterior parameter region with each level of the ABC-SubSim algorithm

Figure 7.15 shows the average signal response of the parameters in the posterior region for the reduced Petri net in comparison to the signal response from the reference Petri net. This is the evolution of the marking of the place in both Petri nets that represents an unrevealed failure of the component. For the summary statistic used, the tolerance of the posterior region at the 4th level of the ABC-SubSim algorithm was 0.083. This means that every pairwise parameter solution within the posterior region results in a reduced model output that is within a metric value of 0.083 of the reference model output, when measured by the metric value given in Equation 7.31. Initially there is some deviation of the reduced Petri net response signal from the reference Petri net response signal. However, following this initial period of instability the approximation made by the reduced Petri net levels to a position that is a small margin above the signal from the reference Petri net. A notable difference between this example and the first is that the marking of the place used to generate the response signals in this case is far less frequent. This results in a larger number of Monte Carlo runs required to gain convergence for the response signal. Consequently, there is a poorer matching of response signals across the reference Petri net and reduced Petri net as a limited number of runs were carried out. In addition to this, there is an increased level of model simplification in this example resulting in a higher level of approximation made with the reduced Petri net.

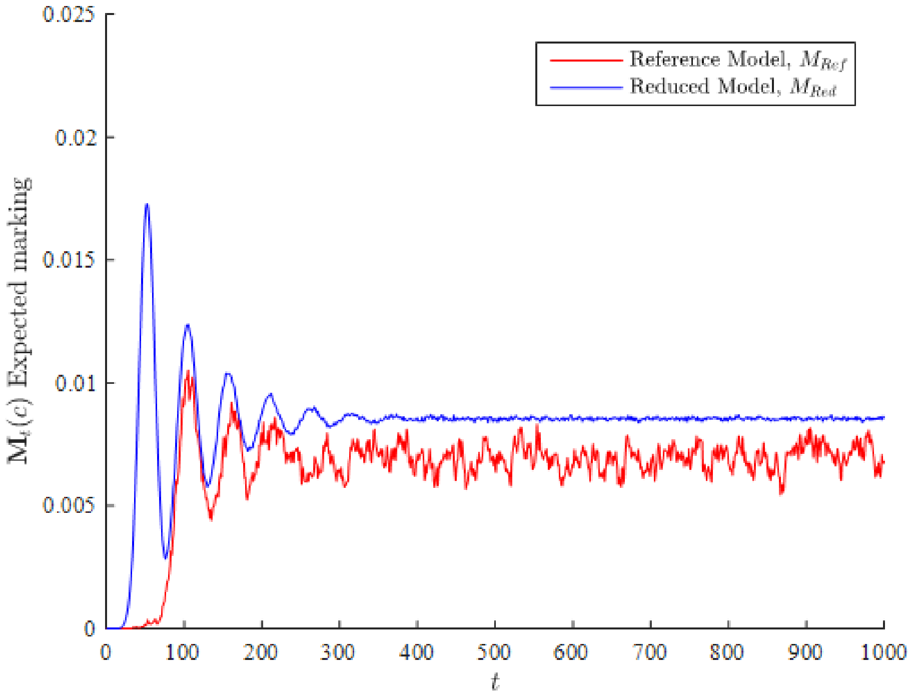


Figure 7.15: The signal response of the reduced Petri net model in comparison to the reference Petri net model

In this example, the reference Petri net has a simulation time of 66.482s for a Monte Carlo simulation with 2000 runs, the reduced Petri net has a simulation time of 20.984s for 2000 runs. This represents a 68% reduction in the computational time.

7.5.3: Example 3

This example implements the reduction methodology to improve the efficiency of optimisation of the reference Petri net model. This is done to demonstrate a use-case of the methodology, whereby a reduced model structure can replace the full model structure, in order to approximate the solution. In this approach, the reduced model structure is used to approximate the solution space in the early stages of the optimisation, followed by an optimisation of the reference model in the final stages. In the latter, the approximate solution space is used as a basis of the optimisation; this reduces the computational effort of the optimisation of the larger reference model since a smaller solution space is explored. This example also applies the updating procedure to parameters governing a Weibull distribution in the reduced Petri net structure, to demonstrate the flexibility of the approach whereby different distributions can be used within the reduced and reference model structures.

The optimisation methodology implemented here uses the following steps, as set out in the conference paper related to this work [190]:

1. Define the key outputs of the reference Petri net for comparison with a proposed reduced structure,
2. Define the reduced model structure,
3. Identify parameters in the reduced model structure for updating,
4. Update parameters,
5. Validate the reduced structure by comparing the reduced model outputs to the outputs for the reference Petri net,
6. Find the approximate optimal solution space using the reduced model structure,

7. Find the optimal solution space for the reference Petri net by searching a reduced solution space based on the approximate optimal solution found in the previous step.

Central to this reduction method is the definition of key model outputs that are present in both the reference and reduced model structures. For this methodology to be applied, the reduced Petri net must have at least:

- The capacity to reproduce the key output, or outputs, of the reference Petri net;
- The capacity to incorporate the behaviours requiring optimisation.

A Genetic Algorithm was implemented to find the optimal solutions for the maintenance and inspection intervals of a component in order to reduce the probability that the component is in the unrevealed failed state. This follows the Genetic Algorithm methodology detailed in Chapter 6. A Genetic Algorithm was selected to demonstrate this approach as it allows an initial definition of the search space, through the assignment of the first parents in the algorithm.

In this optimisation approach, the reduced model forms part of an intermediate step when finding the optimal solution. Initially, an optimal solution space is found for the reduced Petri net, by completing a number of generations of a Genetic Algorithm. The second step of the optimisation approach is to define a region encompassing these solutions to give an approximate solution space for the reference model. The third step is to use the approximate solution space as the initial population of a Genetic Algorithm, and to apply a low number of generations of the algorithm to the reference Petri net to gain the optimal solution space. In summary, the optimisation process is mostly performed on a smaller, more efficient model, with the larger model reintroduced in the latter stages to fine-tune the solutions.

There are two further decisions to be made when implementing this approach, in addition to those required when applying a Genetic Algorithm:

- Firstly, the number of generations that are applied to the reduced Petri net before re-introducing the reference Petri net must be decided;
- Secondly, the definition of the approximate solution space, given the population obtained from the reduced Petri net optimisation, must be defined.

These decisions are problem-dependent and more research should be completed to produce an automated approach for this. It is recommended that the number of generations applied to the reduced Petri net is sufficient to see a good level of convergence in the solutions, so that the search space can be adequately reduced. It is also recommended that the optimal solution space uniformly covers all values gained from the reduced Petri net optimisation, with some values outside of this range. This is recommended to allow the algorithm to explore the approximate space and to check for values outside, but close to, the values found from the reduced Petri net optimisation.

Within this methodology, the assumption is made that the optimal solutions from the reduced Petri net will approximate the optimal solutions for the reference Petri net.

Figure 7.16 gives the reference Petri net used in this example. In this Petri net, place P1 corresponds to the working state of the component and place P4 corresponds to the failed state of the component. There are two pathways that can result in a failure, firstly through the age of the component modelled by transitions t1, t2 and t3, and secondly through a randomly occurring failure modelled by transitions t1 and t5. Place P6 corresponds to a revealed failure and place P7 corresponds to a scheduled maintenance action, either due to a revealed failure, or the age of the component, represented by transitions t6 and t8 respectively. Place P8 counts the number of maintenance actions.

The shaded regions cover the areas of the model that were reduced, and place P4 is emphasised as its marking pattern over time is used as the key output for the Petri net.

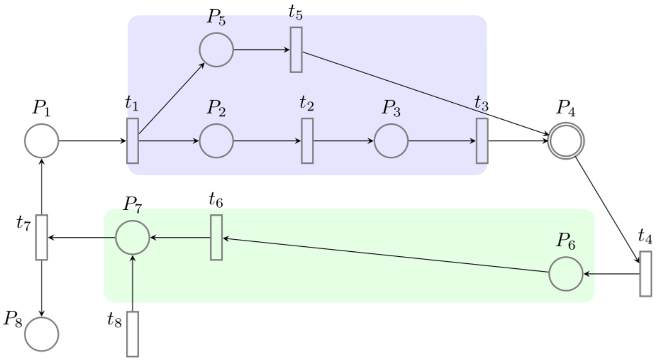


Figure 7.16: The reference Petri net to illustrate a combined reduction and optimisation approach

The optimisation problem for this Petri net aims to reduce the time that the component is in the unrevealed failed state by finding the optimal inspection and age-based maintenance intervals for the component, within a given cost constraint. This corresponds to reducing the time that place P4 is marked, representing the unrevealed failed state, by altering the parameter values of the distributions governing transitions t4 for the inspection interval and t8 for the age-based maintenance.

Figure 7.17 gives the reduced Petri net structure implemented in this example. Here, place P'1 corresponds to the working state of the component and place P'2 corresponds to the unrevealed failed state of the component. Place P'3 corresponds to the revealed failed state of the component and place P'4 counts the number of maintenance actions, either due to a revealed failure or the age of the component. Transition t'4 represents periodic age-based maintenance. Place P'2 corresponds to the unrevealed failed state of the component represented by place P4 in the reference Petri net model.

The shaded areas in the reduced Petri net correspond to the shaded areas in the reference Petri net and highlight the following reductions:

- The intermediate states of the component that lie between the working and failed states are absorbed into the single transition t'1;
- The maintenance scheduling delay on failure is assumed to be much less than the inspection interval and so this delay, and the state corresponding to scheduled maintenance, are absorbed into the place P'3 to represent a state where failure is revealed and maintenance is scheduled.

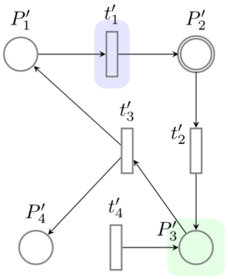


Figure 7.17: The reduced Petri net used to show the combined reduction and optimisation methodology

The parameters governing the transition t'1 were updated. A 2-parameter Weibull distribution was assigned to this transition. As with the other examples in this chapter, the ABC-SubSim algorithm was implemented to find the region in the parameter space where the parameters governing this Weibull distribution resulted in the most similar reduced model output to that of the reference Petri net. The following parameter values were used as input to the reference model in this example:

Transition	**Distribution**	**Parameters**
t1	Normal	$\mu = 20, \sigma = 7$
t2	Normal	$\mu = 13, \sigma = 4$
t3	Normal	$\mu = 8, \sigma = 2$
t4	Interval (Global)	$I = 6$
t5	Uniform	$C = 0.0005$
t6	Normal	$\mu = 1, \sigma = 0.5$
t7	Normal	$\mu = 0.1, \sigma = 0.01$
t8	Normal	$\mu = 30, \sigma = 10$

In addition, the following parameters were assumed in the reduced model, where the updating of the two parameters that govern the firing rate of transition t'1 was completed under these assumed values, for the rest of the parameters in the model:

Transition	**Distribution**	**Parameters**
t’1	2-Parameter Weibull	$\eta = \theta_1, \beta = \theta_2$
t’2	Interval (Global)	$I = 6$
t’3	Normal	$\mu = 1, \sigma = 0.5$
t’4	Normal	$\mu = 30, \sigma = 10$

The output used here for governing the similarity of the models was the marking of the place corresponding to an unrevealed failure in both Petri nets. In this case, the centroid linkage was used as the summary statistic, as with the previous examples in this chapter. The metric value used for updating the parameters was the squared sum of the centroid linkage, for the marking of this place at each time. Four levels of the ABC-SubSim algorithm were applied to update the parameters from a uniform prior region.

Figure 7.18 shows the reduction of the posterior parameter region with each level of the parameter updating process, where the prior region is enclosed in the dotted lines and each of the circles represents the pair of parameter values within the estimated posterior region. The evolution of the parameters within each level was chosen adaptively to maximise convergence to the posterior region, as with the previous examples in this chapter. With repeated further levels of the algorithm, there was limited reduction in the posterior region, showing that after a point the approximation made by the reduced model could not become any more exact.

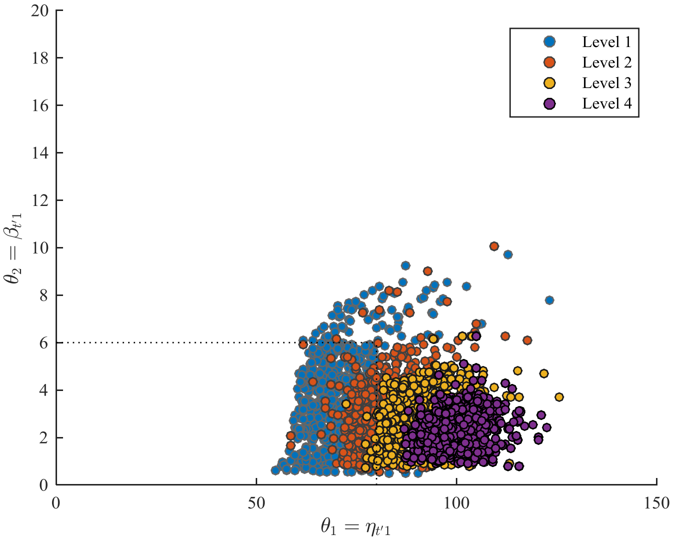


Figure 7.18: The reduction of the parameters in the population to the posterior region

The updated parameter values, which resulted in the lowest metric value and hence the closest approximation, were $\theta_1 = 100.29$ and $\theta_2 = 1.64$, where transition t'1 follows a 2-Parameter Weibull distribution with $\eta = \theta_1, \beta = \theta_2$. Figure 7.19 gives the average marking over time of place P4 for the reference Petri net and place P'2 for the reduced Petri net, with the updated parameter values supplied to the reduced Petri net. Across the two nets, these places have the same representation: when marked, the component is in the unrevealed failed state. This signal can be interpreted as the probability that the component is in the unrevealed failed state as time progresses. The models show a good level of agreement.

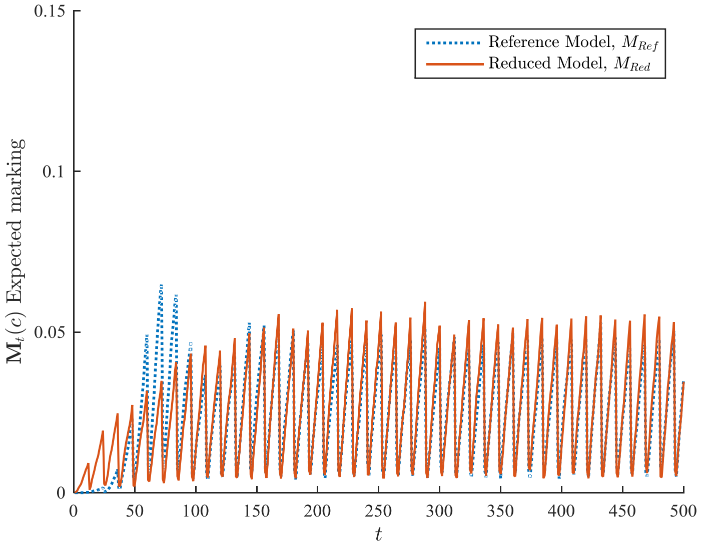


Figure 7.19: The model output, representing the probability that the component is in the unrevealed failed state, over time, for the reference and reduced Petri net models.

A Genetic Algorithm was applied to the Petri nets presented in this example. For comparison, the algorithm was applied to both the reference Petri net and the reduced Petri net in isolation. Finally, the Genetic Algorithm was applied across both Petri nets using the two-stage approach.

In this example, the Genetic Algorithm had a population of 100 vectors and a mutation rate of 1 in 100. The initial population of 100 vectors was defined for parameter values where entries ranged from 1 to 100. For example, the first member of the population was a vector where all entries had the value

1 and for the last population member, all entries had the value of 100. For the reference Petri net in isolation and the reduced Petri net in isolation, eight generations of the Genetic Algorithm were completed. For the two-stage approach, five levels of the Genetic Algorithm were completed for the reduced Petri net to find the approximate solution space. Following this, three levels of the Genetic Algorithm were applied to the reference Petri net using the approximate solution space as the initial population. The selection operator was weighted such that the fittest individuals in each population had a higher probability of selection.

An arbitrary cost was assigned to each of the inspection and maintenance actions and this was constrained to within 1000 units over the time period in question. The time that the component was in an unrevealed failed state was minimised subject to this constraint.

The approximate solution space was found by taking the maximum and minimum value of each variable in the population that resulted at the 5th generation of the Genetic Algorithm for the reduced Petri net. The maximum and minimum values for each variable were found along with the difference between these. The approximate solution space, for this case, is defined by the region given by Equation 7.32 and Equation 7.33 for the two variables θ_1 and θ_2.

$$d_{\theta_n} = \sigma_n^{max} - \sigma_n^{min} \tag{7.32}$$

$$\sigma_n^{min} - d_{\theta_n} \leq \theta_n \leq \sigma_n^{max} + d_{\theta_n} \tag{7.33}$$

where σ_n^{max} is the maximum value for parameter n, and σ_n^{min} is the minimum value for parameter n, as found in the 5th generation of the Genetic Algorithm optimisation of the reduced Petri net, for values of $n\ in\ \{1,2\}$.

The results for this two-level optimisation are given in Figure 7.20 for the optimal inspection interval and age-based maintenance interval. Each mark within the generation represents the parameter value of the individual population member, with the horizontal bar representing the mean value of the population for the variable in question. The mean of the population for the optimisation applied to the reference Petri net in isolation was 9.21 time units and 9.03 time units for the inspection and maintenance intervals, respectively. The mean of the population for the optimisation applied to the reduced Petri net in isolation was 11.00 time units for the inspection interval and 11.00 time units for the maintenance interval. The mean of the population for the two-level approach was 8.87 time units and 9.84 time units for the inspection and maintenance intervals, respectively.

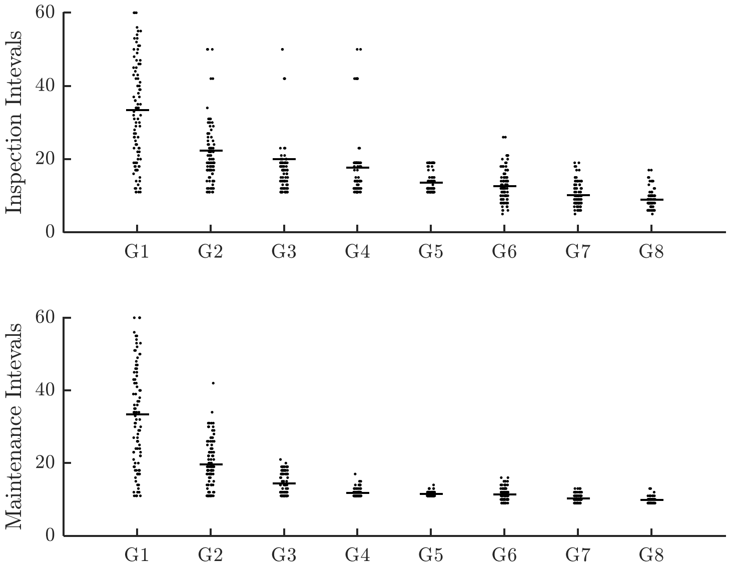


Figure 7.20: Results from the two-level optimisation where the reduced Petri net is used for the first 5 generations, and then substituted to the reference Petri net in the final three generations.

It is notable that the reduced Petri net gives solutions that are close to the reference Petri net, and that for some modelling scenarios it may be suitable to solely take the optimal values from the reduced Petri net. However, the reduced Petri net may not closely reproduce the time that the component is in the failed state given the parameter changes to the system. This is due to dependencies within the model that are absorbed during the parameter fitting process. To clarify, the reduced Petri net may be sufficient to mimic trends in the behaviour of the reference Petri net for different parameter values, to enable an approximation to be made in an optimisation process, but may not be able to reproduce exactly the key model outputs when parameters are changed within the model.

Figure 7.21 gives the time for the optimisation procedure for both the reference model in isolation, the reduced model in isolation and the two-level optimisation combining both the reduced and reference models. In this example, a reduction in the computational cost of the optimisation procedure can be seen, however this is offset in practical application by the time required to update the model parameters, prior to the optimisation process.

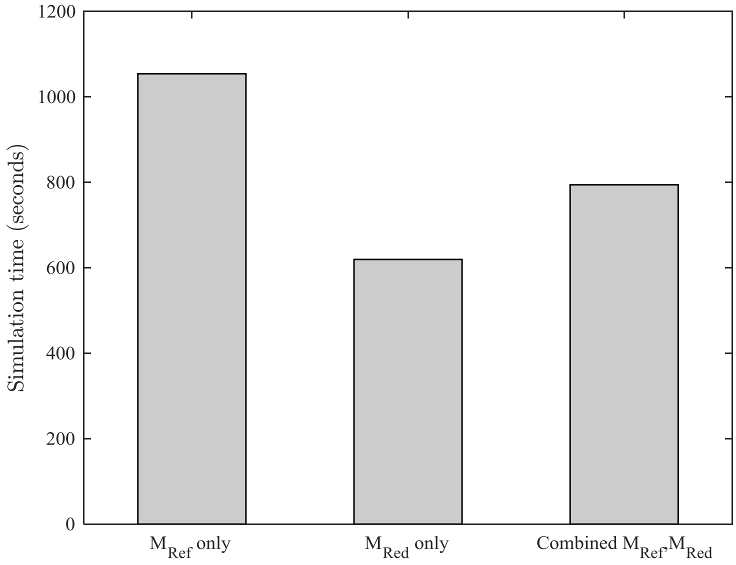


Figure 7.21: The time for the optimisation of the reduced model in isolation, the reference model in isolation and for the two-level optimisation combining both the reference and reduced models.

This example has demonstrated a potential application of the reduction methodology, in order to extend the Genetic Algorithm approach by implementing a reduced model structure as an intermediate step. This demonstrates a benefit of the methodology, where a reduced structure can be used as an approximation of the full structure, in order to reduce the computational cost of optimisation of the model. There are three drawbacks of this method, though. Firstly, the computational cost of fitting the parameters in the reduced model is high. Secondly, the success of the method is highly dependent on the assumption that the reduced structure closely approximates the reference model structure. Thirdly, a choice must be made on the number and location of parameters to be updated in the reduced model structure. The computational cost of fitting and simulation, with a variety of model structures and updated parameter choices, is explored more fully in the following example.

7.5.4: Example 4

In this example, a larger reference model is considered. The reference model has two components, with one component operating as a backup to the other. The model considers the condition, inspection and maintenance of each component, and the overall system state. This example further explores the benefits and limitations of the reduction methodology. To accomplish this, four different reduced model structures are implemented to explore their suitability and demonstrate the capability of the modelling approach to rank reduced model structures. In addition, the example explores the impact of different choices of which parameters to update. In this example, the reference model is presented first. Following this, a description of each of the reduced model structures, and their corresponding results, are presented. Next, a comparison of the reduced model structures is presented. Finally, an example is presented where different model parameters have been updated. The method proposed in this chapter is discussed throughout.

Figure 7.22 gives the reference model for this example. The reference Petri net represents a system with two repairable components, one of which is in back-up. The system fails if both components are in the failed state. The area at the top of the figure, shaded in light grey, governs the overall system state. Here, place P1 corresponds to the working state of the system. Place P2 corresponds to the failed state of the system.

In this reference model, the first component is modelled by the dark-grey shaded region on the left-hand side. This is a repairable component that has an unrevealed failure. Inspection of the component is modelled, and it is assumed that this can discover a failure, or a degraded component state. Maintenance is modelled for each of these states. In this part of the model, place P3 represents a good working state of the component, place P4 represents a degraded state and place P5 represents a failed state. When place P9 is marked then inspection is enabled, and when place P8 is marked, inspection of the component is underway. Place P7 is marked when a degraded component state is discovered, and maintenance is requested. When Place P6 is marked then there is a discovered failure of the component and maintenance is requested.

The second component is modelled by the region shaded in mid-grey on the right-hand side. This is a repairable component that has a revealed failure. Inspection is also modelled for this component and it can reveal a degraded state of the component. Maintenance of the component is scheduled if there is a revealed failure, or inspection identifies a degraded state of the component. In this part of the model, place P10 corresponds to a working state of the component. Place P11 corresponds to a degraded state of the component. Place P12 corresponds to a failed state of the component. When place P15 is marked then inspection is enabled for this component, and when place P14 is marked then inspection is underway. Place P13 is marked when there is a discovered degraded state, and here maintenance is requested for the component.

The remaining parts of the model govern the maintenance scheduling for the components. For both components, maintenance is requested if there is a known failure, or if there is an identified degraded state. It is assumed for both components that the maintenance for a degraded state will occur before there is a failure of the component. Place P18 and transition t25 model the availability of maintenance resources and when place P19 is marked then maintenance is possible. When place P17 is marked then maintenance is requested for a component in the system, due to an identified failure in one of the components. When P23 is marked then maintenance is available to either, or both, of the components to repair a failure. If neither component is in the failed state, then the available maintenance resources can be assigned to repair a degraded state; this is modelled by the marking of place P21. When place P22 is marked then maintenance to improve a degraded state of either component is enabled. Initially, places P1, P3, P10, P9, P15 and P18 are marked with tokens.

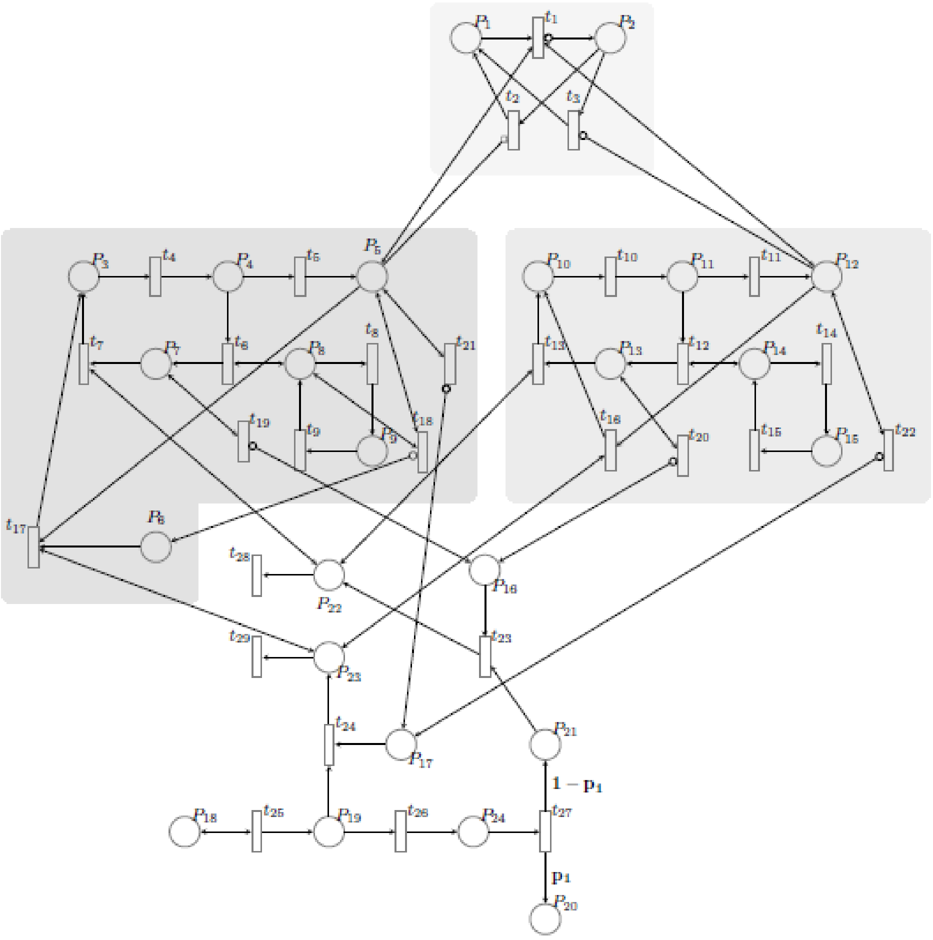


Figure 7.22: The reference model for the fourth example in Chapter 7

In the remaining work given in this example, different reduced model structures are implemented to demonstrate their capability to approximate this reference model. To compare the model outputs, the probability that the system is in the failed state was selected as the key model output for model comparison. This can be found from this reference model by considering the average marking of place P2, for a simulation of the model. The parameters used for each of the models in this example can be found in Appendix 6.

For each proposed reduced model structure, the parameters to update are discussed. This includes a definition of the prior region and details which parameters were updated in each case. In each case, a uniform prior region is used and 2000 seed values are used in the parameter updating process. Varying proposal *pdfs* were tested for their impact on the acceptance rate and convergence of the parameters. A Gaussian distribution with a varying standard deviation at each ABC-SubSim level was chosen as it provided a good reduction from the prior region.

Reduced Model 1

The first proposed reduced model structure simply considers the overall system behaviour. The Petri net in Figure 7.23 gives this reduced model. This is the smallest reduced model structure tested in this example. In this model there are two places, one representing the working state of the system, P1, and one representing the failed state of the system, P2. Transition t1 governs the time to total system failure and transition t2 governs the time to repair of the system.

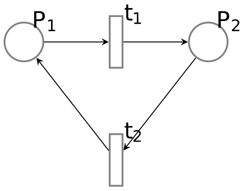


Figure 7.23: The first reduced model structure, in example 4 of Chapter 7

In this reduced model, the output marking of place P2 was used to compare the reduced model output to the output of the reference model. This place corresponds to the failed state of the system, which holds the same representation as place P2 in the reference model.

This reduced model structure combines the total behaviour of both components in the system, and their maintenance and inspection strategies, to solely provide an estimate of the system state. For this model, a normal distribution was assigned to each of the transitions, and the mean of each of the distributions was discovered through the parameter inference process. This was completed to gain the closest approximation of the output of this reduced model structure to the output of the reference model structure. Full parameters used in the models can be found in Appendix 6.

The values for the mean of each of transition t1 and t2 were updated, from a prior region. The prior region varied between [0, 14] for the mean of transition t2, and [0, 200] for the mean of transition t1, where the unit of the values is given in months. These prior regions were defined by considering the physical interpretation in the reference model, to ensure that the prior region was suitably large to contain the parameters governing the system model. For instance, inspection is expected every 6 months in the reference model, with an approximate delay of 1 month on maintenance, and hence the prior was defined as twice this interval. Within each level of the ABC-SubSim algorithm, seed values were varied by a Gaussian distribution, such that the acceptance rate for each level remained within the optimal range of [0.1, 0.2]. Figure 7.24 shows the seed values at each level of the ABC-SubSim algorithm, demonstrating the condensing to a posterior region. Here, the prior region is shown with dashed lines.

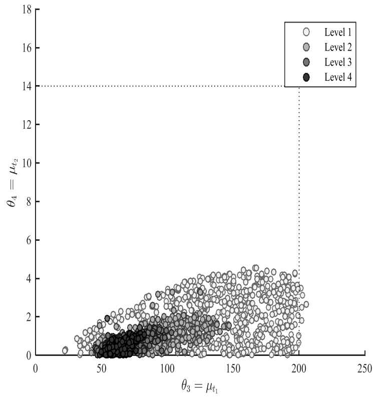


Figure 7.24: Parameter updating process for the first reduced model in example 4 of Chapter 7

The values within the posterior region which resulted in the lowest errors were selected. These values were 59.2 months for transition t1 and 0.213 months for transition t2. This demonstrates a slow time to failure, and a short time between failure and repair, for the system. Since this model summarises the behaviour of both components into a single cycle of transitions, the fitting processes assign this fast repair time, encompassing the behaviour of the components wherein it is very unlikely for both to be in the failed state at the same time.

The reduced model was simulated with the updated parameters, and the marking of place P2 was recorded. In addition, the reference model was simulated, with the marking of corresponding place P2 recorded. Figure 7.25 gives the marking of these places, for both the reference and reduced models. This corresponds to the probability that the system is in the failed state at each time. For this reduced model, the data follows a similar trend. However, on average, the output data from the reduced model is lower than the output data from the reference model. There is some fluctuation in the results, which is on a similar level for both models.

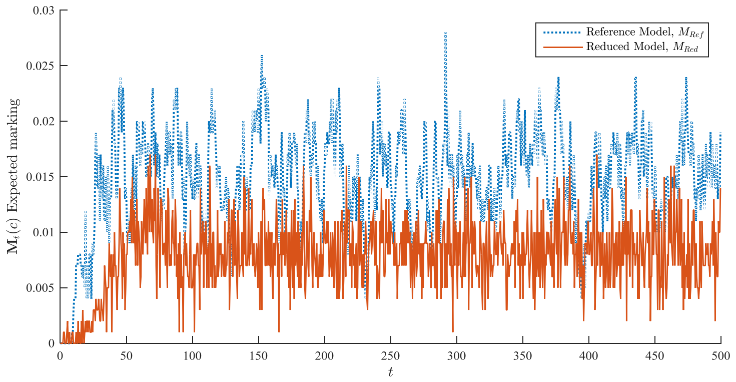


Figure 7.25: Model outputs for the first reduced model and the reference model, in example 4 of Chapter 7

At the end of this example, numeric measures of the model fit and simulation time are presented, along with the hyperparameters for the updating process. This is provided for each reduced model structure. Also, comparisons between different reduced model structures are made.

Reduced Model 2

The second reduced model structure tested in this example is given in Figure 7.26. This model consists of a separate simple model for each component in the system, combined with three transitions that govern the system state. Here, the system state is dependent on the component state. The components in the system are in parallel, such that both must be in the failed state for a failure of the system to occur.

In this reduced model structure, place P1 corresponds to the working state of the system. Place P2 corresponds to the failed state of the system. Transition t1 immediately marks place P2 if both the components are in the failed state. Then, if one of the components returns to the working state, the marking of place P2 is immediately removed and place P1 is marked. This represents the return of the system to the working state and occurs by firing either transition t2 or t3.

In this reduced structure, each of the component models contains only the working and failed state. For each component, there is a transition that governs the time to failure and a transition that governs the time to repair.

The first component is modelled by the places P3 and P4, and transitions t4 and t5. Here, place P3 corresponds to the working state of the first component and place P4 corresponds to the failed state of the first component. The second component is modelled by places P5 and P6, and transitions t6 and t7. Here, place P5 corresponds to the working state of the second component and place P6 corresponds to the failed state of the second component.

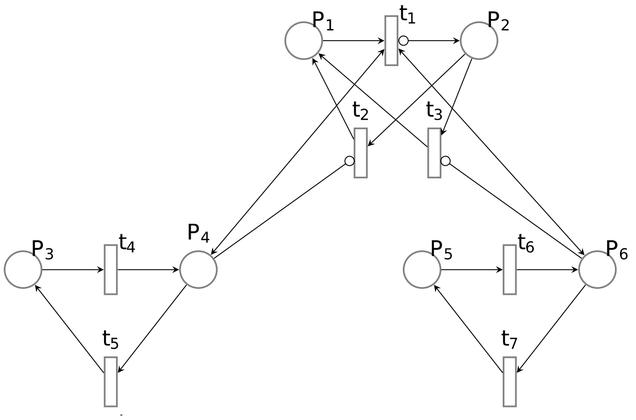


Figure 7.26: The second reduced model structure in example 4 of Chapter 7

In this model, the marking sequence of place P2 over time was selected for comparison between the reduced model and the reference model. This place holds the same interpretation as place P2 in the reference model, which is the probability that the system is in the failed state.

In this reduced model structure, it is assumed that repair times for each component are largely governed by inspection and testing times. The parameters governing the time to each component failure are updated according to this assumption. This is discussed further, with different parameters updated, in a further example at the end of this chapter. Normal distributions are assigned to transitions t4, t5, t6, and t7, with the mean of transitions t4 and t6 decided within the parameter updating process. For transition t5, it is assumed that the repair time of the first component is largely dominated by the 6-month inspection interval, combined with the 1-month maintenance scheduling interval. For transition t7, it is assumed that the repair time for the second component, with a revealed failure, is largely governed by the 1-month maintenance scheduling delay. Transitions t1, t2 and t3 fire instantaneously if enabled. Full parameter values can be found in Appendix 6.

A prior region was defined for the mean values of transition t4 and transition t6. The prior region was defined to include approximate time to failure for each component. The prior region for transition t4 was defined uniformly in the range [0, 100]. The prior region for transition t6 was defined uniformly in the range [0, 80]. This reflects the failure rates assigned to the reference model. The ABC-SubSim algorithm was used to update these parameters, with condensation to a posterior region, following four levels of the algorithm. Figure 7.27 demonstrates this. Hyperparameters of the algorithm were tuned to improve convergence of the algorithm. Details of this can be found in Table 7.4. The posterior region for this reduced structure has some discontinuities, suggesting that certain small changes in parameter values give rise to cases that greatly change the output signal. This implies less stability in the posterior region.

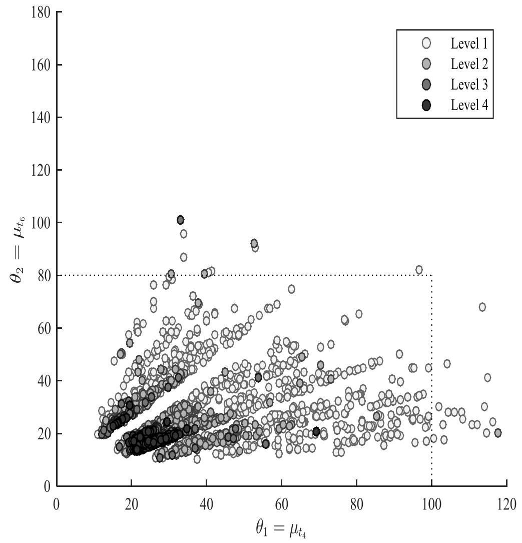


Figure 7.27: The parameter updating process for the second reduced model in example 4 of Chapter 7

The optimal fitted parameters were extracted from the updated posterior region. For the mean of transition t4, this value was 26 months, and for the mean of transition t6, this value was 16 months. These values reflect the faster failure rates assigned to the second component in the reference model.

The reduced model was simulated with the updated parameter values. The probability of system failure, for the reference and reduced model, is given in Figure 7.28. Here it can be seen that, on average magnitude, the reference model approximates the reference model. However, there are periodic peaks in the results of the reduced model, which cannot be seen in the results of the reference model. This demonstrates that this reduced model has limited capacity to recreate the behaviour of the reference model, under the assumptions applied.

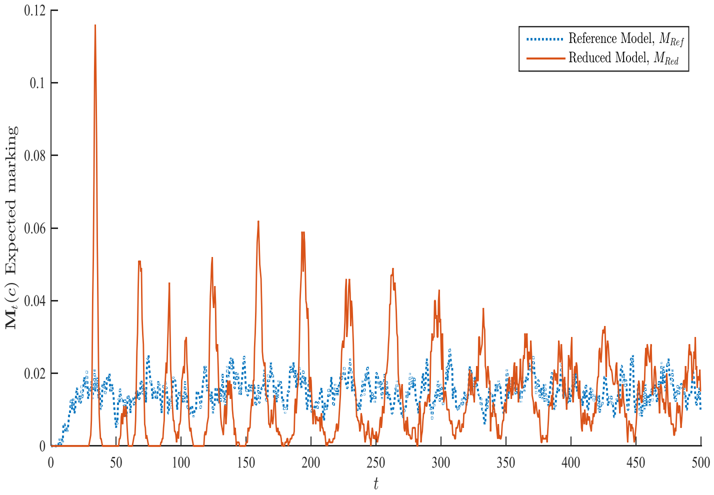


Figure 7.28: The output signals for the second reduced model and the reference model in the fourth example in Chapter 7

A comparison of the different reduced model structures is provided at the end of this example. Also, a further parameter updating process for this model structure is presented, where different assumptions are made.

Reduced Model 3

The third reduced model structure is given in Figure 7.29. In this example, the inspection and maintenance loops are removed, so that there is intermediate repair of each component and repair upon failure. This reduced model has a similar structure to the reduced model structure in Figure 7.26, with the transitions and places that govern the overall system state at the top, the transitions and places that govern the first component on the lower left hand side and the transitions and places that govern the second component on the lower right hand side. In this model, places P1 and P2, and transitions t1, t2 and t3, represent the system condition. These places have the same meaning as described for the model in Figure 7.26, with place P1 representing the working state of the system and place P2 representing the failed state of the system.

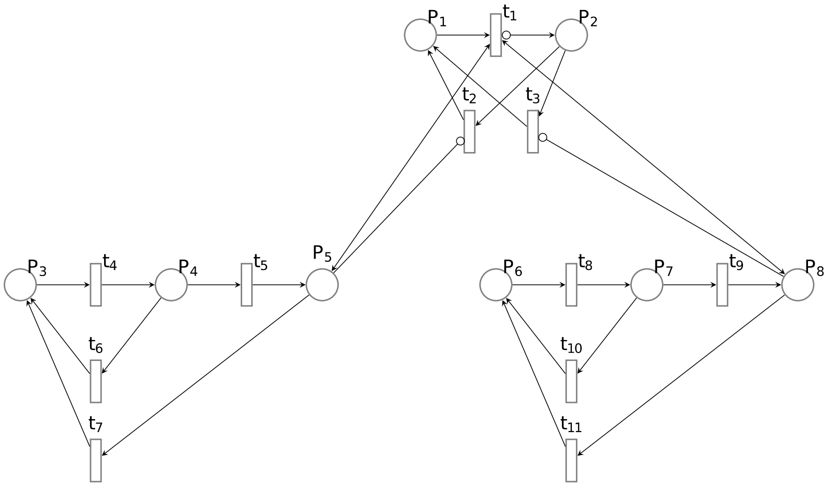


Figure 7.29: The third reduced model structure in example 4 of Chapter 7

In this reduced model, each of the components is modelled with three states: the working state, an intermediate degraded state and the failed state. There is a repair action assigned to the intermediate degraded state and to the failed state. The working state is modelled by place P3 for the first component, and place P6 for the second component. The intermediate degraded state is modelled by place P4 for the first component, and place P7 for the second component. The failed state is modelled by place P5 for the first component, and place P8 for the second component. Transitions t4 and t5 model the degradation and failure of the first component. Transitions t6 and t7 model the maintenance of the first component, for the degraded and failed state, respectively. Transitions t8 and t9 model the degradation and failure of the second component. Transitions t10 and t11 model the maintenance of the second component, for the degraded and failed state, respectively.

During the parameter updating process, the marking pattern of place P2 in this reduced model was compared to the marking pattern of place P2 in the reference model. In both the reference model and this reduced model, this marking pattern corresponds to the probability of system failure, at each time.

In this reduced model, the reduction removes the maintenance and inspection modelling. For this reason, the parameters governing the maintenance transitions were selected for updating. Normal distributions were assigned to each of t6, t7, t10 and t11, and the mean of each of these distributions was discovered through the parameter updating procedure. Transitions t1, t2 and t3 have no associated delay time. The distributions that govern firing times for transitions t4, t5, t8 and t9 were set to the parameters taken from the reference model, which had the corresponding component state changes. Full details of the parameters used can be found in Appendix 6.

A prior region was defined that contained a dimension for each of the four parameters that were selected for the updating process. This prior region was defined uniformly in [0, 20] for the mean of transition t6, uniformly in [0, 40] for the mean of transition t7, [0, 40] for transition t10 and [0, 4] for transition t11. Here it is expected that the distribution governing transition t11 will lead to shorter repair times, since this part of the reduced model considers the behaviour of the component with the revealed failure.

Figure 7.30 shows the seed values within each level of the ABC-SubSim algorithm, which was used to update the parameters. Since all four parameters governing the maintenance transitions were updated, an interesting result is seen for the distributions governing the first component maintenance transitions, t6 and t7. Here the parameter values condense to a curved posterior region for pairwise values in each set of the updated parameters. This shows that where the intermediate repair action occurs less frequently (every few months), then the repair on failure occurs quickly. Conversely if the intermediate repair occurs within a short time-frame, meaning that the component rarely fails, this allows the repair time on a revealed failure to increase. These times may not reflect reality. The parameters governing the maintenance transitions for the second component, t10 and t11, largely show the reduction of the parameter value governing transition t11. This corresponds to the time to repair on failure of the second component. This is in line with expectations: since the failure is revealed, maintenance is expected shortly after the failure.

In this example, we have an extra degree of freedom for each of the component models, where the updated parameters can interact to show some dependence in the posterior region. This can be a limitation of the methodology, since unrealistic values may be arrived at. To address this, the use of another metric for the parameter updating process can be explored further. For example, the failure probability of the system at each time, and the number of maintenance actions at each time can be combined to give a summary statistic for each proposed set of parameter values. As we have some understanding of the system in the reference model, in this case we can select the parameter values that are in line with reality.

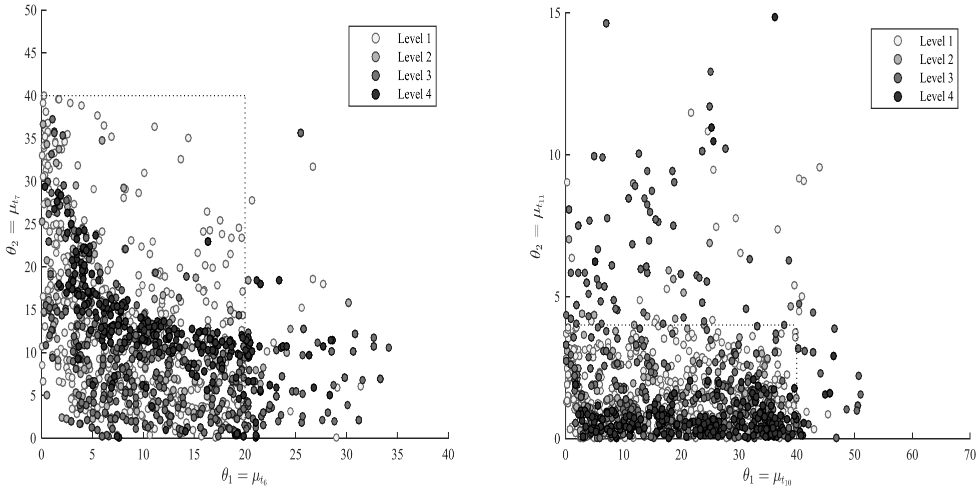


Figure 7.30: Parameter updating procedure for the third reduced model of the fourth example of Chapter 7

The parameters that resulted in the lowest errors were selected from the posterior region. For each of the transitions, the values were 11.39 for the mean of transition t6, 13.57 for the mean of transition t7, 36.90 for the mean of transition t10 and 0.136 for the mean of transition t11. The reduced model was simulated with these parameters and the marking of place P2 was recorded, corresponding to the probability of system failure at each time. Figure 7.31 shows this and the probability of system failure, taken from the reference model. The output from the reduced model follows similar average values, and has a similar level of fluctuation to the output from the reference model. This model therefore more closely matches the reference model than the outputs of the two previous model structures. The model also does not show a large increase in simulation time in comparison to the previous reduced structures, and hence is likely a better choice of reduced model structure. This is discussed further at the end of this example.

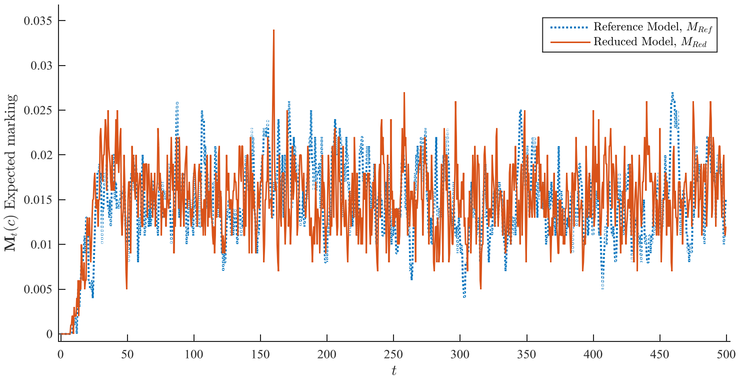


Figure 7.31: Model outputs for the third reduced model and the reference model in example 4 of Chapter 7

There are some interesting results in this reduced model. For the first component there is a relationship between the intermediate maintenance and the maintenance on failure. This shows how the different maintenance actions can compensate for each other. This could be an interesting method for optimising a system, so as to keep the failure to a similar level but discover the optimal maintenance strategy. A faster maintenance before failure can allow a slower maintenance on failure, and vica verca, to still give the same model output. For the parameters on the second component, the behaviour is largely dominated by reducing the time to repair on failure. This short time allows the time governing the maintenance prior to failure to move more freely.

Reduced Model 4

The fourth reduced model structure in this example is given in Figure 7.32. This model differs from the reference model as it does not include the maintenance scheduling logic. In this reduced model, place P1 corresponds to the working state of the system and place P2 corresponds to the failed state of the system. The system state is determined by the state of each of the two components.

The first component is modelled by the places and transitions on the lower left-hand side. Place P3 corresponds to the working state of the first component, place P4 corresponds to a degraded state of the first component and place P5 corresponds to the failed state of the first component. This failure is unrevealed. Inspection of the component is modelled by places P8 and P9. The maintenance of the component, on a discovered failure, is modelled by transition t17. The maintenance of the first component, when a degraded state is discovered, is modelled by transition t7.

The second component is modelled by the places and transitions on the lower right-hand side. Place P10 corresponds to the working state of the second component, place P11 corresponds to a degraded state of the second component and place P12 corresponds to the failed state of the second component. This failure is revealed. Inspection of the component is modelled by places P14 and P15. The maintenance of the component, on failure, is modelled by transition t16. The maintenance of the second component, when a degraded state is discovered, is modelled by transition t13.

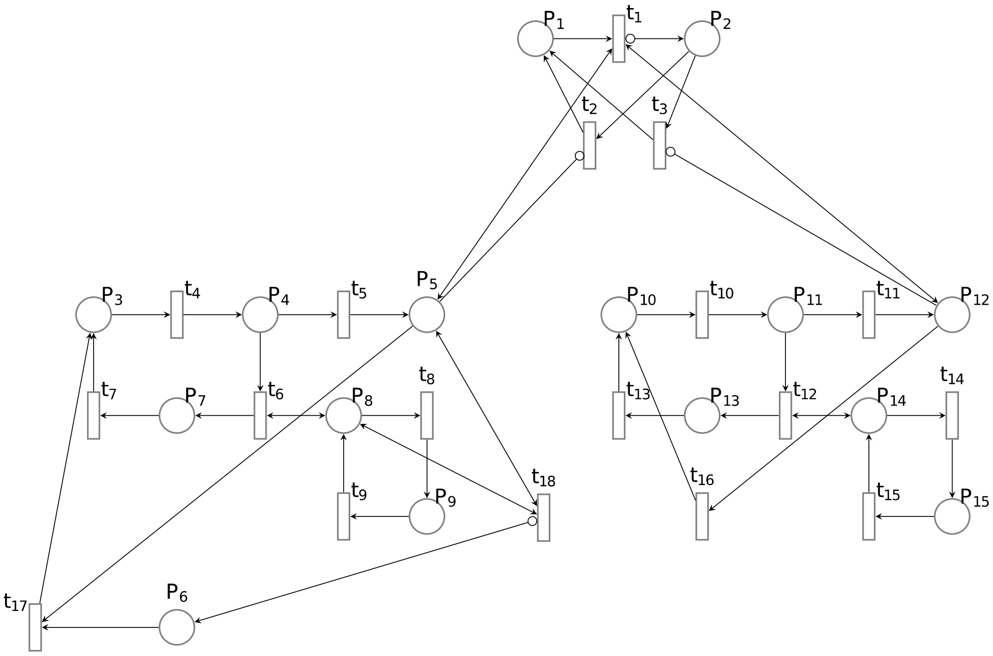


Figure 7.32: The fourth reduced model structure in the fourth example in Chapter 7

The marking of place P2 in this reduced model was used as the output signal for comparison to the output signal of the reference model. The marking of this place represents the probability of system failure at each time. Parameters in this reduced model were updated based on the similarity of the output signals from both the reduced and reference model.

The transitions t1, t2 and t3 have no associated delay times. Here the mean values of transitions t7, t13, t16 and t17 were updated in this example. These transitions relate to the maintenance scheduling times, which are adjusted with this reduction. The remaining transitions in this reduced model were assigned distributions so that they mirror the corresponding transitions in the reference model. For instance, transitions t8 and t9, which govern the inspection of the first component, were given the same distributions as the first component inspection loop transitions in the reference model.

A prior region was defined for each of the parameters for updating. For the mean of each transition, the prior region was defined uniformly in the range [0, 10]. The procedure for updating the parameters can be seen in Figure 7.33. Here, we can see that the updating is largely dominated by reducing the value for the parameter corresponding to the mean of transition t16 to a low level. This represents a short time to repair for the component with revealed failures. The parameter for the mean of transition t13 can vary without much impact on the final system failure, in this case. This parameter corresponds to preventative maintenance of a component with a revealed failure. Likewise, the parameters governing the component with unrevealed failures do not show a strong convergence to a fixed set of values. This suggests that, across the system, failures can be dominated by the repair time of the component with the revealed failure. This makes some sense, since both components need to be in the failed state for a system failure, and the repair of the second component is not dependent on an inspection interval, since the failure is revealed.

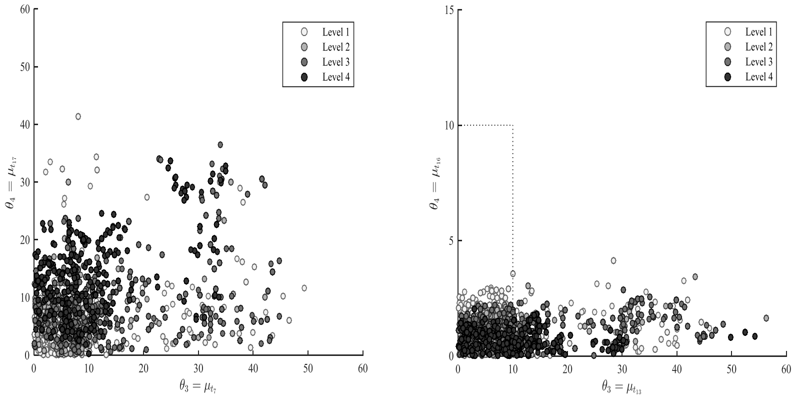


Figure 7.33: The parameter updating process for the fourth reduced model of example 4 in Chapter 7

The parameter values that resulted in the closest recreation of the output signal of the reference model were selected from the posterior region. For the mean of transition t7 this value was 11.18 months, while for the mean of transition t17 this value was 10.36 months. These values both suggest a slow maintenance rate of the component with the non-revealed failure. In contrast, the updated parameter value for the mean of transition t13 was 0.45 months and the updated parameter for the mean of transition t16 was 0.64 months. Both imply that the updated model has almost immediate maintenance for the component with the revealed failure.

This reduced model was simulated for 1000 runs with the updated parameter values. The output showing the probability of failure of the system, taken from this reduced model and the reference model, is shown in Figure 7.34. It is notable that both the third and fourth reduced models have a similar approximation to the reference model, however this model comes at a higher computational cost for fitting and running in general, due to its increased size and the looping behaviour.

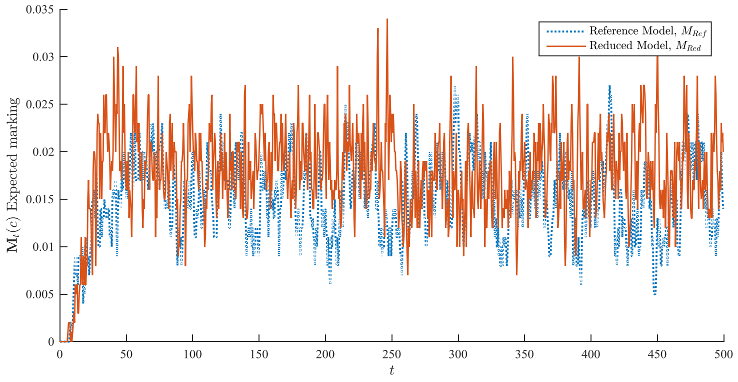


Figure 7.34: The signal comparison for the fourth reduced model in example 4 of Chapter 7

A limitation of this method seems to be that the larger and more complex models cannot have their behaviour succinctly described by a single output parameter of the model. This is demonstrated in this example, where the updated parameters contribute to a large reduction in maintenance times for the second component, allowing larger maintenance times for the first component. The parameter updating process applied here can be extended to incorporate several model outputs, to gain more information on the suitability of different parameter values.

Reduced model summary and comparison metrics

This section gives summary information for the different reduced model structures and compares their suitability. As a baseline, for 1000 runs, simulation of the reference model took 544.223 seconds to complete. In this example, the reference model had an average probability of system failure of 0.0147, when averaged over 500 months for the simulation time period. Table 7.3 gives a summary of the data for each of the results, for each reference model.

This table includes some information on the model and updating procedure: the number of transitions in the reduced model, the number of updated parameters, the percentages that govern the variation of the seed values in each level of the ABC-SubSim algorithm and the acceptance rates at each level of the ABC-SubSim algorithm. These data give a summary of the model size, the complexity of the updating procedure and detail on the hyperparameter tuning of the algorithm.

In addition, the table includes several measures of the suitability of the model. Firstly, the error tolerance is given for each model. This quantifies the associated error in the approximation made by the posterior region, for the final level of the ABC-SubSim algorithm. A lower tolerance indicates a better approximation to the reference model. Here, reduced model 3 has the lowest error tolerance. The time to fit the parameters is also given here. The time to fit parameters increases in a non-linear manner with increasing model size. This lengthy time is a major limitation of the current implementation of the approach, especially when considering larger and more complex models. The simulation time for 1000 runs of each model is also presented. This increases in a non-linear manner with the number of transitions. There is a large difference between the simulation time for reduced model 3 and reduced model 4. Finally, the average probability of system failure, for each model is given. This can be compared to the same average for the reference model: reduced model 3 shows the best approximation in this case.

By considering the data within this table, and the plots given already in this example, a user can make an informed decision on which model to choose. This depends on the modelling requirements, as there is a trade-off between the level of reduction and the time to simulate the model. For these data, reduced model 3 seems a reasonable choice, since it has the lowest tolerance, the closest approximation to the average probability of system failure and reduces the simulation time of the model to less than 10% of the reference model simulation time.

	Reduced Model 1	**Reduced Model 2**	**Reduced Model 3**	**Reduced Model 4**
Number of transitions	2	7	11	18
Error Tolerance	0.233	0.492	0.191	0.197
Time to Fit parameters (seconds)	32241.156	292780.81	460990.87	2171825.7
Number of parameters updated	2 (mean of both transitions)	2 (mean of degradation transitions)	4 (mean of each maintenance transition)	4 (mean of maintenance transition)
Simulation time-1000 runs (seconds)	2.434	16.622	28.048	127.408
Average probability of system failure	0.00791	0.0135	0.0149	0.0177
Gaussian variance percentages	80, 50, 30, 20	200,100,50,20	50,25,15,10	60, 30, 20, 10
Acceptance rates	0.20, 0.11,0.13, 0.13	0.30,0.22,0.13,0.19	0.11, 0.12, 0.17, 0.08	0.13,0.11,0.14,0.10

Table 7.3: Summary information for the reduced models in example 4 of Chapter 7

This section has provided a summary of the analysis completed on the different reduced model structures presented in this example. It demonstrates how this procedure can be used to rank reduced model structures, to aid in decision making. A couple of limitations were discussed. Firstly, updating the parameters is a lengthy process, and alternative approaches for this should be considered. Secondly, unfeasible solutions can arise when looking at more complex models with multiple interacting behaviours. Use of a more comprehensive summary statistic to address this has been discussed, as a potential further step of this work. This summary statistic could include both the system state and the maintenance history, for example. The next and final section of this example explores this limitation further.

A further discussion on parameter fitting selection

In addition to the analysis presented so far in this example, different model parameter fitting choices were explored. This section presents an aside to the main part of Example 4 in this chapter, where results are given for a different parameter updating choice. This is applied to the second reduced model structure, given earlier in this example in Figure 7.26. In this model, there are two components and each component is modelled by one failure transition and one repair transition. The system state is determined by the state of each of the components. This example highlights limitations already discussed for more complex models, whereby a summary statistic, based on a single model output, may be insufficient for parameter choice.

In the earlier parameter updating attempt for the second reduced model, the means of the transitions relating to the repair of the components were updated. The transitions for the failure of each component were assigned fixed values, which somewhat constrained the possible parameter values governing the repair of the component. Here, an attempt was made to fit distributions to parameters governing the failure rates and repair rates of both of the components. Hence, probability of failure of each component can be decreased by either increasing the time to failure or reducing the time to repair. The system failure can be determined by a balance of parameters in all four of the distributions. This allows more freedom for the parameter updating algorithm to find a closer match to the desired output signal but has a greater possibility of returning values that are not in-line with reality.

The ABC-SubSim algorithm was run with a uniform prior region for the mean of each of the transitions governing each component state. This was done to find the optimal parameter solution in order to minimise the difference in the output signals of this reduced model and the reference model. However, in this case it resulted in a scenario where the optimal matching is found outside of the physical thresholds of the system. Here, the algorithm returned parameters that resulted in fast degradation rates for each component, which were counteracted by a short repair time.

In this case, the optimal parameter value for the mean of the transition governing component failure was 11 months for the first component and 4 months for the second component. The mean of the parameter governing the repair rate was 0.1 months for the first component and 1 month for the second component. For both components in the reference model, the inspection interval was set to 6 months, hence, the parameter fitting has given solutions that are not physically reasonable, in an attempt to most closely recreate the output signal of the reference model.

The reduced model was simulated with these updated parameters. Figure 7.35 gives the result of this simulation, and the result of the simulation of the reference model. These parameters, although unfeasible, result in a close approximation to the reference model. However, this is only considering a single output of the model. Table 7.4 gives a summary of the metrics for this example. This parameter selection gives a closer approximation to the output signal than the parameter updating process demonstrated earlier for the same reduced structure. However, the resulting parameter values are unreasonable. Hence, care should be taken when applying this method, since a closer approximation of a single output signal may not necessarily mean a better representation by the reduced model structure.

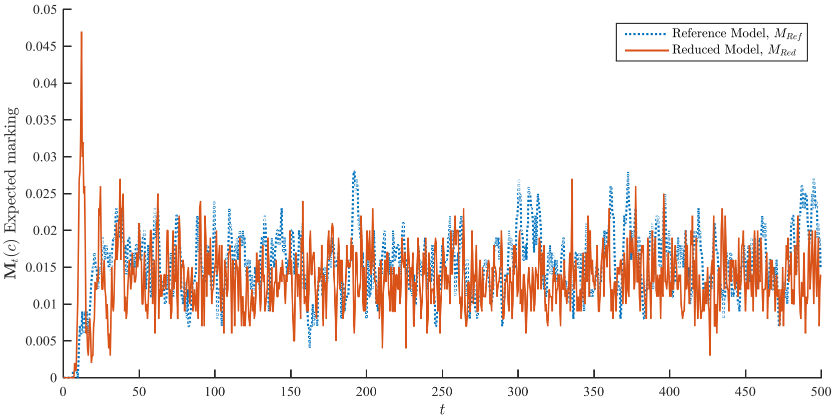


Figure 7.35:The signal of the reference and reduced model fitting for case 2b

	Reduced Model 2
Number of transitions	7
Error Tolerance	0.290
Time to Fit parameters (seconds)	193260.177
Number of parameters updated	4 (mean of failure and repair transitions)
Simulation time-1000 runs (seconds)	18.048
Average probability of system failure	0.01329
Gaussian variance percentages	60, 30, 20, 10
Acceptance rates	0.17, 0.14,0.16, 0.11

Table 7.4: Summary results for updating all timed transitions in example 4 of Chapter 7

To address limitations seen in this example, a summary statistic can be based on multiple outputs of the model, instead of a single output. This should be explored further with the aim of avoiding cases where there is a high number of degrees of freedom and a single model output used for quantifying model similarity. For example, the fitting of the model parameters can be adjusted to include the expected number of intervention actions, in addition to the predicted probability of failure.

7.6: Discussion

There are several benefits to applying the method proposed in this chapter. Firstly, there can be a reduction in the computational time taken for Monte Carlo simulation of the reduced Petri net, in comparison to the reference Petri net. Secondly, the methodology provides a framework to reduce complexity in Petri net models. Thirdly, the methodology can be used to justify model selections, especially when justifying assumptions made by the modeller to keep the model at a reasonable size. The method provides a framework to test different model structures, and to give a comparative measure of their suitability. The method has also been implemented within a two-stage optimisation procedure, to decrease the computational cost of the optimisation.

There are several challenges faced by this methodology. The reduced Petri net approximates the reference Petri net and so does not exactly recreate the results. The higher the level of simplification in the reduced model the greater the level of approximation. It is possible to fit every parameter in the reduced Petri net, to gain as close as possible approximation to the output, however, the computational effort to fit many parameters counteracts the gain made by reducing the model size in the first place. Also, updating multiple parameters can result in a scenario whereby the output is closely approximated but the parameters are not in line with reality. Hence, careful consideration of the choice of summary statistic for the model comparison must be undertaken. Also, with the reduced model, there may be dependencies that are contained in the reference model that are not carried through during the reduction process. This could be problematic if the reduced Petri net alone is used to find the optimal solutions of the system. In addition, it is time-consuming to manually create and test different model structures. Finally, it is computationally expensive to perform the parameter updating process.

Central to the method is a suitable choice of a summary statistic and subsequent metric for the comparison of signals from the reduced model to the reference model. A high-level framework for the assessment of a suitable metric has been presented in this chapter. However, further work can be completed to develop a methodology within the ABC field where the suitable metric choice by the user is less crucial to the success of the approach. In addition, summary statistics based on multiple model outputs, or behaviours can be explored. This should further constrain allowed parameter values to those that have a reasonable physical interpretation.

The method presented in this chapter could be particularly useful in situations where there are large Petri nets simulations with multiple repeated similar units. This is because one reduced Petri net structure could be found for each unit and then this reduced structure could be repeated to improve computational efficiency. As demonstrated, this method can also be incorporated into an optimisation strategy. The optimisation carried out earlier in this thesis was computationally expensive due to the need for repeated convergent simulations of the Petri net, for different input values. As demonstrated in this chapter, a reduced Petri net model could be used to find the approximate region that the optimal solutions lie in. This approximate region could then be used as input to the reference Petri net model for a second level of optimisation to find the exact optimal solution. This would reduce computation time as most of the simulations would be carried out on the reduced Petri net with only the latter stages carried out on the larger reference Petri net.

7.7: Contributions

A novel reduction method for Petri nets, within a parameter updating framework, was developed, and is presented in this chapter. It is demonstrated in some detail in the first two examples in this chapter. This improves the state of the art for Petri net reduction methods, which contain specific rule-based reductions, as it is more flexible. The method also provides a way of quantifying the reduced model suitability. Included in the development of this approach is new research into comparison metrics for Petri Net model outputs. This contributes to the wider body of literature by applying signal processing techniques to quantify the similarity of model outputs in a time-dependent manner.

In addition, the reduction approach is implemented in a new way, demonstrating its use within an optimization procedure that uses both the reduced and full model structures. This improves on the state of the art by providing an approach for reducing the computational cost of optimisation of system models. Finally, this chapter presents an application of the reduction methodology exploring the implications for a larger model, and the use of the method for informing the choice of reduced model structure and parameter fitting. This adds to the state of the art by giving a novel methodology for quantitatively informing the choice of one model structure over another.

7.8: Conclusion

The review of reduction methodologies in Chapter 2 highlights that there are commonly restrictions on Petri net reduction methodologies to specific structures within the model. With a view to developing a flexible methodology that is not limited to specific structures or transition types, the methodology proposed in this chapter was presented, along with a description of the concepts used.

This chapter has presented a new methodology to reduce the complexity of Petri net models to retain key outputs, while reducing computational time for simulation. This methodology can be applied to a complex Petri net with large size or various transition types. This methodology could be especially useful in situations where a process contains many multi-state components resulting in a Petri net that is timely to simulate, especially if there are no structures within the Petri net that can be reduced with rules currently available in literature. This method provides a framework for the reduction of models and an approach for comparing the suitability of the resulting reduced models.

Central to the methodology applied in this chapter is the choice of the metric value used to compare across a reference and reduced Petri net model. An example comparison of metric is presented and this gives a framework for the selection of metric that are most likely to result in a reasonable application of the methodology. A metric based on the Centroid Linkage is proposed for application of the methodology, based on the analysis.

For illustration, the methodology proposed in this chapter was applied in four separate examples. The first two examples demonstrate the methodology for small models. These applications demonstrate the potential of the methodology to allow a reference Petri net model to be replaced with a smaller model, such that the smaller model approximates the outputs of the larger model. In the third example, the use of the method in an optimisation approach is explored. Here, the methodology presented in this chapter allows an approximate optimal solution to be found using a reduced model, followed by a final optimal solution found using the reference model and a reduced search space, based on the approximate optimal region. The fourth example explores the use of the methodology in determining model structure and parameter fitting choice. In this example, multiple reduced model structures are tested for their suitability to approximate a larger reference model structure.

There are three main limitations of the approach proposed in this chapter. Firstly, the time taken to update the parameters is large, this can reduce any benefit of improved speed of computation gained by implementing a reduced model structure. Secondly, the choice of reduced model structure is an iterative process, where several structures can be required for testing. This is time consuming for a user. Finally, if the summary statistic is chosen such that it does not contain sufficient information for the updating, then a number of solutions can arise, that well recreate the desired output across the models, but that have nonsensical meanings. This is especially the case for larger reference, or reduced, models with several degrees of freedom. This has been explored in the fourth example in the chapter. A suggestion is made to further research different summary statistics, which may represent the flow of tokens across different parts of the model or contain multiple model outputs.

In conclusion, the proposed method has been implemented with some success but to apply the methodology presented in this chapter, an in-depth knowledge of the technique and the problem is required for successful application of this method. This is inherent to the nature of the decisions needed to perform any sort of ABC analysis. This is a trade-off of this sort of approximate analysis which avoids the need for a likelihood function. There is further work to be done on how to best choose summary statistics, model class and how to select output signals from the models. Further work is suggested to tie in existing methods for optimally suggesting the structure of a reduced Petri net model prior to matching the results to that of a reference Petri net.

Chapter 8 Conclusion and Further Work

This thesis has intended to address the aim set out at the start of the project. The aim of this thesis was:

To develop a method that can be used to accurately model risk on an underground railway especially as the network ages and utilisation increases.

To complete this, the objectives have been addressed with a proposed modelling approach, applied to two real world systems to highlight different capabilities. Increasing failure rates as the system ages has been incorporated into the models developed in Chapter 4, where imperfect maintenance is employed along with dependencies introduced through system level maintenance strategies. In addition, the work presented in Chapter 5 has developed methods for complex asset management strategies whereby the component is replaced on its individual age, condition and the system phase, and the component is inspected and tested at a frequency dependent on the system phase.

A methodology for the optimisation of asset management strategies is given in Chapter 6, including an optimisation of the system phases presented in Chapter 5. Chapter 6 also presents a measure of uncertainty for a predicted value of risk, and gives a new methodology for incorporating uncertain input parameters into the modelling framework. Finally, the computational efficiency of Petri net models has been discussed in Chapter 7 and a new methodology for the reduction of such models in order to improve this efficiency has been presented.

This chapter provides a summary of each chapter presented within this thesis, following this a discussion of the work is presented. The next section presents recommendations for future work. Then, the key findings of the thesis are presented. Finally, an overall conclusion of the thesis is presented.

8.1: Chapter Summary

Chapter 1 provides a background on world-wide underground railways and presents research on historical accidents that have occurred on these systems. This highlights the importance of managing risk for these safety critical networks. The chapter also provides a brief introduction to concepts such as risks, hazards and ageing systems and summarises the key hazards on the London Underground as identified by current risk assessment methods within London Underground. The aims and objectives of the project are outlined.

Chapter 2 gives a literature review of risk modelling methodologies, along with a review of asset management methodologies. A review of the UK industry underground and over ground railway risk modelling methods is presented. Following this, a review of risk modelling methods in other industries is given. This provides the justification for the selection of the Petri net, Monte Carlo simulation, Fault Tree and Event Tree methods that are implemented in the remainder of this thesis. In addition, the chapter gives a review of the application specific models available in literature, for S&C and fire protection systems. In Chapter 4 and Chapter 5, models are presented for these systems. The review provides justification for modelling choices made when creating these models. In addition, Chapter 2 gives a review of optimisation methods, which forms the bases of the work in Chapter 6. Also included in Chapter 2, is a review of methods to reduce the complexity of Petri net models, to address identified issues with computational time and the incorporation of uncertainty in model outputs. This part of the review informs the development of methodologies proposed at the end of Chapter 6 and in Chapter 7.

Chapter 3 introduces the methodologies implemented in this thesis and is provided as an aid to the reader. The Fault Tree method, Event Tree method, Petri net method and Monte Carlo simulation method are introduced. A discussion of the strengths and weaknesses of each method is given along with simple illustrative examples. This chapter also presents the proposed modelling approach applied to system models in this thesis, which implements the methodologies described in the earlier parts of the chapter. In addition, a description of the custom software developed during this project, for the analysis of the models developed, is given.

Chapter 4 presents a model for a railway S&C using the methodology presented in Chapter 3. There is a focus in this chapter on imperfect maintenance actions and system level opportunistic maintenance strategies. Sample input values are used to demonstrate the potential of the modelling approach for quantitative analysis. Different maintenance strategies are applied in the chapter to validate the capability of the methodology, developed in this thesis, to test the impact of different management strategies on the system state and derailment occurrence.

Chapter 5 presents a model for underground station fire protection systems. The model includes a fire detection, alarm and deluge system. The condition and management of each component within the systems is modelled, in order to assess the unavailability of each system over time. In this chapter, there is also consideration of how human actions can interact with a system with the potential to cause a system failure. In addition, there is a focus on a phased asset management strategy where components are inspected, tested or maintained at a time dependent on their age and the system age, or their condition.

Chapter 6 presents several further analysis methods for the methodologies proposed in this thesis, applied to the model developed in Chapter 5. Initially a modelling approach for changing a Fault Tree structure into a Petri net is presented, showing a good agreement of results. Secondly, a methodology is proposed for the risk based asset management optimisation of ageing systems, with a phased asset management strategy. An application of this is given, using the structure developed in the first part of the chapter. Following this, a discussion on the convergence and uncertainty of the model is given. Also, a novel methodology is presented and applied for incorporating uncertain model inputs in a Petri net framework. Finally, a discussion of the challenges faced when implementing these approaches is given, namely the computational cost of repeated convergent simulation of a large Petri net model, and the potential instability of the modelling approach when considering uncertain model inputs.

Chapter 7 presents a novel Petri net reduction methodology to combat the potentially large computational cost of Monte Carlo simulation of Petri net models. Here, a parameter inference method is employed to allow updating of parameters within a proposed reduced model, such that the output of the reduced model approximates that of the original larger model. The methodology is applied in four examples. The first two examples demonstrate the methodology, the third example combines the reduction with a two-stage optimisation procedure and the final example explores the use of the method for reduced model selection. Any limitations and benefits of the methodology are discussed throughout the chapter.

8.2: Parameter Assumptions and Use of Data

The parameters within the model applications in Chapter 4, Chapter 5 and Chapter 6 have been assumed, in order to demonstrate the capability of the models. These assumptions affect the sample results for the model outputs. The outcomes that are most sensitive to these assumptions are those that relate to the rare event occurrences, such as a system failure, as a change in one of the parameters, can lead to a large increase in these system level outputs. For instance, if there is not a component in backup and the component fails regularly, this can largely impact the model outputs on a system level. In comparison, more common events, such as the total number of maintenance actions, are less sensitive to such assumptions. Despite the assumed parameters in the model demonstrations, the modelling approach can be easily adapted when data becomes available. It is simply a case of varying the parameters in the excel spreadsheet that contains each model logic in order to use real data values. In addition, typical trends were extracted from the sample model runs, to give a generalize commentary on the trends of each model example. To address the issue of inherent assumptions or inaccuracies associated with model input parameters, an uncertainty propagation approach is given in Chapter 6. Hence, where there is higher uncertainty on a parameter value this can be propagated through the model simulation to the model outputs, giving decision makers more information.

To apply the models developed in this thesis, data should be collected for the degradation rate of each component within each model, this can be from extended life testing or data gathered in the field. The data would be used to find the parameters and distributions for the transitions governing the degradation transitions within each component level model. Collection of this data can improve the model allowing assumptions made within each component model to be either justified or adjusted, based on the available evidence. In addition, data can be collected for the maintenance and inspection strategies currently applied in each modeled system, such as the time interval between identifying a failure in each component and the components repair. This data can be used to determine the parameters for the transitions governing repair and inspection within the model. Hence, improving the model by allowing the removal of assumptions about the maintenance actions applied to each of the components. However, these parameters are less crucial to the model success as they can be varied in order to test different strategies. Data should also be collected on the overall system state for each modeled system, which can be used to validate the current model predictions and make any required amendments to the model structure to bring the model more in line with reality, hence improving any future predictions. Finally, data should be collected on the cost of different maintenance actions, this can then improve the results given by the optimizations of the model, these results can then be used to inform maintenance decisions.

In addition, the parameters governing the examples in Chapter 7 are assumed. These examples are present to demonstrate and explore the methodology, rather than represent a real-world system.

8.3: Key Contributions

The key contributions of this work are:

1. A risk modelling approach that can be used to evaluate the impact of complex asset management strategies. A novel aspect of this approach is that it extends existing methodologies applied in industry; this improves on the state of the art as it allows more in depth modelling of components with complex degradation, maintenance and inspection strategies.
2. An asset management model for a railway S&C that considers derailment frequency and system state. This model goes into further depth than S&C models available in literature, including the modelling of imperfect maintenance and inspection with the application of a Petri net approach to the problem. A range of maintenance actions, including opportunistic maintenance, are modelled. The model improves on the state of the art as it removes assumptions of perfect maintenance and inspection and allows dependencies to be introduced through maintenance actions. In addition, system level restrictions or closures due to the combined condition of components across the system are modelled, hence, the model can be used to predict derailment occurrence and the total cost of maintaining the system.
3. An asset management model for an automatic fire protection system that predicts the probability that each of the deluge, detection and alarm system is in a failed state. This model includes areas of novelty in comparison to models available in literature by implementing a Petri net approach. This allows the model to feature: a phased asset management strategy, a probability model for system failure and modelling of false activations of each system. The introduction of the phased asset management improves the state of the art as it allows exploration of strategies that can change throughout a system lifecycle. Also, since the model considers the system in a higher level of detail, it can be used to estimate the cost of the system due to maintenance, inspection, testing and any penalties for false activation of the

system. This contributes to the state of the art for risk modelling of fire protection systems, as other factors contributing to failure and cost are considered. The modelling also allows dependencies between the systems to be modelled, such as combined testing.

4. A novel approach for optimisation of a phased asset management strategy, such that different maintenance strategies are applied at different times. This improves the state of the art for modelling the management of the system over time, where strategies are applied based on component condition, as it allows different strategies to be applied at different phases as a system ages. The method used combines a Simulated Annealing and Genetic Algorithm optimization approach, to improve on the Genetic Algorithm approach by reducing the number of search parameters, hence improving efficiency.
5. A new approach for incorporating uncertain inputs in a Petri net model, and evaluating the impact on the model outputs. This improves cases of Petri net modelling, where the uncertainty in the output of the Petri net model is unstated, and model parameters are assumed true. Inclusion of uncertainty can give decision makers a higher level of knowledge when using the predictions of such models to make decisions. A method for studying the convergence of the model is also applied; this improves on current convergence checks used for Petri net models where the convergence is viewed on a linear scale, which does not clearly demonstrate the rate of convergence.
6. A novel reduction methodology for Petri net models. As part of this work, research was conducted into comparison metrics between Petri net model outputs, to quantify the difference in outputs of such models over time, as opposed to comparing point estimates of the model. This can give a wider picture of the difference in the predictions of two models. The developed reduction methodology is a flexible tool which improves on current Petri net reduction methods that are highly rule based. Research exploring this method is also presented, including the use of the method to improve current optimization methods using a approximate solutions space approach. Also, the use of the approach to choose model structure is explored. This improves on the state of the art for model selection, as Petri Net models are usually user defined, and this approach provides a quantitative measure to support model structure choice.

8.4: Further Work

There are several areas where future work can be completed in this area of study. Firstly, the methodology demonstrated in this thesis can be applied to further real-world applications. Also, data can be collected from these systems to validate the outputs of the specific models developed in this thesis.

Secondly, a lack of convergence of the simulation of Petri net models can result in unreliable predictions, especially when considering rare events. The rare event may not occur in the defined number of simulations giving the user the idea that the model has fully converged. With more simulations, the rare event may occur and impact the risk predicted by the model, however this contribution can be overlooked, by analysing the convergence of the model with insufficient simulations. Further work can be completed on the efficient simulation of rare events using a Petri net methodology. This could include simulation tools that use more intelligent methods for sampling from input distributions, to improve the rate of convergence to a solution for a Petri net model. This could also be used to reduce the computational cost of simulation of Petri net models.

Thirdly, future work can be completed on the new methodology presented in this thesis for Petri net model reduction. Further work can be completed on an optimal method for the simplification of the Petri net model, along with the optimal number of parameters for reduction. This reduction methodology can also be applied to a large real-world system model and research can be completed into reducing the computational time of the parameter updating methodology.

Finally, the combined optimisation technique incorporating a reduced model structure and the full model structure can be explored further and attempted with different model structures or optimisation approaches.

8.5: Conclusion

This thesis has presented work to improve the existing methodologies to model risk for ageing systems on an underground railway. An approach centred on Petri net modelling and Monte Carlo simulation has been proposed.

The first contribution of this thesis is a risk modelling approach that incorporates the impact of complex asset management strategies. The approach allows detailed modelling of component condition, maintenance, and inspection. The approach was applied to two systems, with a model created for each. Firstly, a new model was presented for a railway S&C. This model allows the prediction of derailment occurrence, along with the different interventions required across the system life cycle. The model is beneficial as it allows component condition dependencies that are introduced through maintenance actions to be modelled, introduces opportunistic maintenance, and allows imperfect maintenance and inspection to be modelled. Secondly, a new model is presented for an automatic fire protection system. The system modelled includes a detection, deluge, and alarm sub-systems. Main benefits of this model are the incorporation of a phased asset management strategy, modelling false system activation and modelling component condition dependencies introduced through system-level maintenance strategies. In addition, the model also considers interactions between the sub-systems, such as system level testing and their combined function on activation. In both cases data collection is required, for a real-world system, to validate the results of the model. A disadvantage of the models is the computational cost for simulation, to gain numerical results.

Several further novel approaches were also proposed in the thesis. Firstly, an optimisation approach for a phased asset management strategy was presented. This allows different strategies to be applied at different times, where the optimisation procedure considers where the strategy should change phase, and what the optimal values within each phase should be. The approach was applied to the fire protection system model. Secondly, an approach was proposed for quantifying the uncertainty of Petri net model outputs, given uncertain model inputs. The approach encapsulates uncertainty introduced through the simulation of the model and the uncertainty in the input parameters. The approach was applied to a sub-section of the fire protection system model. Finally, a novel Petri net reduction methodology was proposed. This gives a flexible method for reducing the complexity of Petri net models and gives a numerical quantification for the level of approximation of the reduction. This method was applied in several scenarios, including as part of a two-step optimisation procedure and to demonstrate its suitability for model structure selection.

In conclusion, the work presented in this thesis expands current quantitative methods and models that consider risk for ageing systems, and how management of the system can be optimised to reduce the risk. The methodologies can enable a more rigorous analysis process, given the current constrains on the data available for modelling railway processes. There are also several approaches that develop current methodology within the Petri net modelling field.

Focus should be put on collecting a larger base of reliable data, as this is one of the major limiting factors discovered throughout this thesis. This is especially the case for development of intelligent risk modelling for railway systems.

References

[1] UITP, “World Metro Figures: Statistics Brief,” UITP, Brussels, 2018.

[2] J. Glover, London's Underground (12th Edition), Ian Allan Publishing, 2015.

[3] N. Darroch, “A brief introduction to London's underground railways and land use,” *Journal of Transport and Land Use,* vol. 7, no. 1, pp. 105-116, 2014.

[4] TfL, “Underground services performance, Performace data almanac,” 2018. [Online]. Available: https://tfl.gov.uk/corporate/publications-and-reports/underground-services-performance. [Accessed 15 May 2018].

[5] Transport for London, “Annual Report and Statement of Accounts 2014/15,” Transport for London, London, 2015.

[6] Interborough Rapid Transit Co., “The New York Subway, it's Construction and Equipment,” McGraw Publishing Co., 1904.

[7] Metropolitan Transportation Authority, “2018 Annual Report,” 2018.

[8] P. Hall, “Underground as City Maker: London Versus Paris, 1863–2013,” *The London Journal,* vol. 38, no. 3, pp. 177-183, 2013.

[9] RAPT Group, “Activity and sustainable development report 2018,” 2018.

[10] Moskovsky Metropoliten, “Метрополитен в цифрах [Metropolitan in figures],” Moskovsky Metropoliten, 2019. [Online]. Available: https://www.mosmetro.ru/press/digits/. [Accessed 30 July 2019].

[11] Tokyo Metro Co., “Tokyo Metro Corporate Profile,” Tokyo, 2017.

[12] Rail and Underground Panel, “Rail and Underground International Benchmarking Report,” TfL, London, 2015.

[13] D. Fennel, “Inverstigation into the King's Cross Underground Fire,” The Department of Transport , London, 1988.

[14] T. Taig and M. Hunt, “Review of LU and RSSB Saftey Risk Models,” (TTAC Limted) Office of Rail Regualtion, 2012.

[15] Health and Saftey Executive, “Chancery Lane Derailment - HSE interim report on the findings from the technical investigation,” Health and Saftey Executive, London, 2003.

[16] Health and Safety Executive, “Derailments on London Underground at Camden,” HSE, London, 2005.

[17] Health and Saftey Executive, “White City train derailment,” HSE, London, 2005.

[18] Rail Accident Investigation Branch, “Rail Accident Report: Derailment at Archway 2 June 2006,” RAIB, Derby, 2006.

[19] Rail Accident Investigation Branch, “Rail Accident Report: Derailment of a London Underground Central Line train near Mile End station 5 July 2007,” RAIB, Derby, 2008.

[20] Rail Accident Investigation Branch, “Rail Accident Report: Passenger trapped in a closed train door, Tooting Broadway, Northern Line, London Underground 1 November 2007,” RAIB, Derby, 2008.

[21] London Underground, “Investigation into 3 Customers Struck by a Loose Inter-Car Barrier at Mile End Station on the 17 November 2009,” London Underground, London, 2010.

[22] Rail Accident Investigation Branch , “Derailment of an engineering train between Gloucester Road and Earl’s Court stations on London Underground 12 May 2010,” RAIB, Derby, 2011.

[23] Rail Accident Inverstigation Branch, “Passenger dragged a short distance by a train at Holborn station 3 February 2014,” RAIB, Derby, 2014.

[24] Rail Accident Investigation Branch, “Passenger trapped in train doors and dragged at Clapham South station 12 March 2015,” RAIB, Derby, 2016.

[25] Rail Accident Investigation Branch, “Derailment of a passenger train at Ealing Broadway 2 March 2016,” RAIB, Derby, 2016.

[26] ORR, “Rail Safety Statistics, 2016-2017 Annual Statistical Release,” OGL, London, 2017.

[27] Y. Chen, *Identifying organizational and contractual drivers behind metro accidents in Shanghai,* Faculty of Covil Engineering: Technology University of Delft, 2013.

[28] The commitee of Inquiry, “The incident at the mrt circle line work site that led to the collapse of the Nicoll Highway on 20 April 2004,” Singapore, 2004.

[29] National Transportation Safety Board, “Collision Between Two Washington Metropolitan Area Transit Authority Trains at the Woodley Park Zoo/Adams Morgan Station in Washington, D.C. November 3, 2004,” National Transportation Safety Board, Washington, 2006.

[30] National Transportation Sefety Board, “Collision of Two Washington Metropolitan Area Transit Authority Metrorail Trains Near Fort Totten Station,” National Transportation Sefety Board, Washington, 2010.

[31] M. Quatre, B. Koubi Karsenti, B. Desbazeille and J. Ville, “Rapport d'enquête sur l'accident survenu sur la ligne 12 du métro parisien, le 30 août 2000, Intersection Saint-Georges-Notre Dame de Lorette, sens nord-sud,” Conseil Général de Ponts et Chaussées, 2000.

[32] U.S Fire Administration/Techincal Report Series, “Case Study Number Ten: Union Square Srarion, New York City- August 28, 1991,” in *Special Report: Rail Emergencies,* Homeland Security, 2003, p. 27.

[33] Toronto Transit Commission, “1995 Annual Report,” 1995.

[34] L. Poon and R. Lau, “Fire Risk in Metro Tunnels and Stations,” *International Journal of Performability Engineering,* vol. 3, no. 3, pp. 355-368, 2007.

[35] Railway Accidents Investigation Commission, “Derailment at the Teito Rapid Transit Authority's Naka-Meguro Station on the Hibiya Line, Japan, October 26, 2000,” The Ministry

of Transport of Japan, 2000.

[36] A. Newman, "Train Derails In Brooklyn, Injuring Scores," *The New York Times* , 21 June 2000.

[37] Health and Saftey Authority, "Hazard and Risk," 2015. [Online]. Available: http://www.hsa.ie/eng/Topics/Hazards/. [Accessed 12 11 2015].

[38] J. D. Andrews and T. R. Moss, Reliability and Risk Assessment, London and Bury St Edmunds: Professional Engineering Publishing Ltd, 2002.

[39] T. Packman, "London Underground Qualitative Risk Assesment Update 2014.02," TfL HSE Specialist Advisers Team , 2014.

[40] T. Taig and M. Hunt, "Review of LU and RSSB Safety Risk Models (for ORR)," TTAC Limited, 2012.

[41] S. Turner, D. Keeley, M. Glossop and G. Brownless, "Review of Railway Safety's Safety Risk Model," HSL, 2002.

[42] C. Yao, Improving Railway Saftey Risk Assessment Study, PhD thesis, University of Birmingham, 2012.

[43] RSSB, "Safety Risk Model," Rail Safety and Standards Board , [Online]. Available: http://www.rssb.co.uk/rail-risk-portal/safety. [Accessed 27 01 2016].

[44] RSSB, "Saftey Risk Model: Risk Profile Bulletin, version 8.1," RSSB, London, 2014.

[45] U.S Nucelar Regulatory Commission, Fault Tree Handbook, Washington D.C.: U.S. Government Printing Office, 1981.

[46] J. Dugan, S. Bavuso and M. Boyd, "Dynamic Fault-Tree Models for Fault-Tolerant Computer Systems," *IEE Transactions on Reliability,* vol. 41, no. 3, pp. 363-377, 1992.

[47] R. Gulati and J. B. Dugan, "A Modular Approach for Analyzing Static and Dynamic Fault Trees," in *Reliability and Maintainability Symposium. 1997 Proceedings, Annual,* Philadelphia, 1997.

[48] J. Magott and P. Skrobanek, "Timing analysis of saftey properties using fault trees with time dependence and times state-charts," *Reliability Engineering and System Saftey,* vol. 97, pp. 14-26, 2012.

[49] T. Khanh Nguyen, J. Beugin and J. Marais, "Method foe evaluating an extended Fault Tree to analyse the dependability of complex systems: Application to a satellite-based railway system," *Reliability Engineering and System Saftey,* vol. 133, pp. 300-313, 2015.

[50] US Nuclear Regulatory Commission, "Reactor Safety Study," WASH-1400 (NUREG-75/014), 1975.

[51] S. Kaplan, "Matrix Theory Formalism for Event Tree Analysis: Application to Nuclear-Risk Analysis," *Risk Analysis,* vol. 2, no. 1, pp. 9-18, 1982.

[52] C. G. Acosta and N. O. Siu, "Dynamic Event tree Analysis Method (DETAM) for Accident Sequence Analysis," Office of Nuclear Regulatory Research, United States Nuclear Regulatory

Commission, Washington, D.C., 1991.

[53] B. Rutt, U. Catalyurek, A. Hakobyan, K. Metzroth, T. Aldemir, R. Denning, S. Dunagan and D. Kunsman, “Distributed Dynamic Event Tree Generation for Reliability and Risk Assessment,” in *CLADE 2006 Workshop*, Paris, 2006.

[54] D. Marsh and G. Bearfield, “Generalizing event trees using Bayesian networks,” *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability,* vol. 222, no. 2, pp. 105-114, 2008.

[55] G. Bearfield and W. Marsh, “Generalising Event Trees Using Bayesian Networks with a Case Study of Train Derailment,” in *Winther R., Gran B.A., Dahll G. (eds) Computer Safety, Reliability, and Security. SAFECOMP 2005. Lecture Notes in Computer Science, vol 3688.,* Springer, Berlin, Heidelberg, 2005.

[56] J. Andrews and C. Fecarotti, “Modelling Life Extension of Safety Critical Systems,” in *Safety and Reliability of Complex Engineered Systems*, Zurich, 2015.

[57] L. Oukhellou, E. Côme, L. Bouillauta and P. Aknina, “Combined use of sensor data and structural knowledge processed by Bayesian network: Application to a railway diagnosis aid scheme,” *Transportation Research Part C: Emerging Technologies,* vol. 16, no. 6, pp. 755-767, 2008.

[58] G. Wang, T. Xu, T. Tang, T. Yuan and H. Wang, “A Bayesian network model for prediction of weather-related failures in railway turnout systems,” *Expert Systems with Applications,* vol. 69, pp. 247-256, 2017.

[59] E. Castillo and Z. Grande, “Bayesian Networks-Based Probabilistic Safety Analysis for Railway Lines,” *Computer-Aided Civil and Infrastructure Engineering,* vol. 31, pp. 681-700, 2016.

[60] E. Castillo, A. Calviño, S. Sánchez-Cambronero, I. Gallego, A. Rivas and J. M. Menéndez, “A Markovian–Bayesian Network for Risk Analysisof High Speed and Conventional Railway Lines Integrating Human Errors,” *Computer-Aided Civil and Infrastructure Engineering,* vol. 31, pp. 193-218, 2016.

[61] T. S. Liu and S. B. Chiou, “The application of Petri nets to failure analysis,” *Reliability Engineering and System Safety,* vol. 57, pp. 129-142, 1997.

[62] M. Ghazel, “Using Stochastic Petri Nets for Level-Crossing Collision Risk Assessment,” *IEE Transactions on Intelligent Transportation Systems,* vol. 10, no. 4, pp. 668-677, 2009.

[63] X. She, J. Zhao and J. Yang, “Functional Verification on Railway Signaling System with Colored Petri Nets,” in *IEEE 17th International Conference on Intelliget Transportation Systems* , Qingdao, 2014.

[64] J. Andrews, “A modelling approach to railway track asset management,” *IMechE Part F: Journal of Rail and Rapid Transit,* vol. 227, no. 1, pp. 56-73, 2012.

[65] L. Podofillini, E. Zio and J. Vatn, “Risk-informed optimisation of railway tracks inspection and maintenance procedures,” *Relaiiblity Engineering and System Saftey,* vol. 91, pp. 20-35, 2006.

[66] L. Bai, R. Liu, Q. Sun, F. Wang and P. Xu, “Markov-based model for the prediction of railway

track irregularities," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit,* vol. 229, no. 2, pp. 150-159, 2015.

[67] F. J. Restel and M. Zajac, "Reliability model of the railway transportation system with respect to hazard states," in *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM),* Singapore, 2015.

[68] D. Prescott and J. Andrews, "Investigating railway track asset management using a Markov analysis," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit,* vol. 229, no. 4, pp. 402-416, 2015.

[69] D. Y. Yang and D. M. Frangopol, "Risk-informed bridge ranking at project and network levels," *Journal of Infrastructure Systems,* vol. 24, no. 3, 2018.

[70] R. Dekker, "Applications of maintenance optimization models: A review and analysis," *Reliability Engineering and System Safety,* vol. 51, no. 3, pp. 229-240, 1996.

[71] M. Rausand, "Reliability centered maintenance," *Reliability Engineering and System Safety,* vol. 60, no. 2, pp. 121-132, 1998.

[72] A. Grall, C. Bérenguer and L. Dieulle, "A condition-based maintenance policy for stochastically deteriorating systems," *Reliability Engineering and System Safety,* vol. 76, no. 2, pp. 167-180, 2002.

[73] L. Di, L. Si and T. Ze, "Based on Petri Nets and Hybrid Genetic-Tabu Search Approach to Scheduling Optimization for Dual-Resource Constrained Job Shop," in *2nd International Conference on Electronic & Mechanical Engineering and Information Technology,* Paris, 2012.

[74] A. Zimmerman, D. Rodriguez and M. Silva, "A two phase optimization method for Petri net models of manufacturing systems," *Journal of Intelligent Manufacturing,* vol. 12, no. 5-6, pp. 409-420, 2001.

[75] V. Jain, R. Swarnkar and M. Tiwari, "Modelling and analysis of wafer fabrication scheduling via generalized stochastic Petri net and simulated annealing," *International Journal of Production Research,* vol. 41, no. 15, pp. 3501-3527, 2003.

[76] S. Apeland and T. Aven, "Risk based maintenance optimization: Foundational issues," *Reliability Engineering and System Safety,* vol. 67, no. 3, pp. 285-292, 2000.

[77] N. S. Arunraj and J. Maiti, "Risk-based maintenance-Techniques and applications," *Journal of Hazardous Materials,* vol. 142, no. 3, pp. 653-661, 2007.

[78] R. Faddoul, W. Raphael and A. Chateauneuf, "Maintenance optimization of series systems subject to reliability constraints," *Reliability Engineering and System Safety,* vol. 180, pp. 179-188, 2018.

[79] C. Yang, R. Remenyte-Prescott and J. Andrews, "Pavement Maintenance Scheduling using Genetic Algorithms," *International Journal of Performability Engineering,* vol. 11, no. 2, pp. 135-152, 2015.

[80] B. L. H. Le, Modelling railway bridge asset management, Univeristy of Nottingham : PhD Thesis, 2014.

[81] P. Yianni, A Modelling Approach to Railway Bridge Asset Management, University of Nottingham : PhD Thesis, 2017.

[82] Z. Su and B. De Schutter, “Optimal scheduling of track maintenance activities for railway networks,” in *15th IFAC Symposium on Control in Transportation Systems CTS 2018,* 2018.

[83] J. D. Andrews and L. M. Bartlett, “Genetic Algorithm Optimization of a Firewater Deluge System,” *Quality of Reliability Engineering International,* vol. 19, pp. 39-52, 2003.

[84] J. Andrews and R. Pattison, “Optimal safety system performance,” in *Annual Reliability and Maintainability Symposium, pp. 76-83,* Philadelphia, PA, USA, 1997.

[85] L. Podofillini, E. Zio and J. Vatn, “Risk-informed optimisation of railway tracks inspection and maintenance procedures,” *Reliability Engineering & System Safety,* vol. 91, no. 1, pp. 20-35, 2006.

[86] N. Arunraj, S. Mandal and J. Maiti, “Modeling uncertainty in risk assessment: An integrated approach with fuzzy set theory and Monte Carlo simulation,” *Accident Analysis & Prevention,* vol. 55, pp. 242-255, 2013.

[87] D. Guyonnet, B. Bourgine, D. Dubois, H. Fargier, B. Coˆme and J.-P. Chilès, “Hybrid Approach for Addressing Uncertainty in Risk Assessments,” *Journal of Environmental Engineering,* vol. 129, no. 1, pp. 68-78, 2003.

[88] J. Cardoso, R. Valette and D. Dubois, “Fuzzy Petri Nets: An Overview,” in *13th Triennial World Congress,* San Francisco, 1996.

[89] N. Sadeghi, A. Fayek and W. Pedrycz, “Fuzzy Monte Carlo Simulation and Risk Assessment in Construction,” *Computer-Aided Civil and Infrastructure Engineering,* vol. 25, pp. 238-252, 2010.

[90] R. Kenarangui, “Event-Tree Analysis by Fuzzy Probability,” *IEEE Transactions on reliability,* vol. 40, no. 1, pp. 120-124, 1991.

[91] P. Baraldi and E. Zio, “A Combined Monte Carlo and Possibilistic Approach to Uncertainty Propagation in Event Tree Analysis,” *Risk Analysis,* vol. 28, no. 5, pp. 1309-1325, 2008.

[92] A. Senderovich, A. Shleyfman, M. Weidlich, A. Gal and A. Mandelbaum, “To aggregate or to eliminate? Optimal model simplification for improved process performance prediction,” *Information Systems,* vol. 78, pp. 96-111, 2018.

[93] A. Andreas Rogge-Solti, W. M.P. van der Aalst and M. Weske, “Discovering Stochastic Petri Nets with Arbitrary Delay Distributions from Event Logs,” *Lecture Notes in Business Information Processing,* pp. 15-27, 2014.

[94] E. Leclercq, D. Lefebvre and S. El Medhi, “Identification of timed stochastic Petri net models with normal distributions of firing periods,” in *Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing,* Moscow, 2009.

[95] R. Buchholz, C. Krull and G. Horton, “Reconstructing model parameters in partially-observable discrete stochastic systems,” *ASMTA,* vol. 6751, p. 159–174, 2011.

[96] S. Senderovich, A. Rogge-Solti, A. Gal, J. Mendling, A. Mandelbaum, S. Kadish and C.

Bunnell, “Data-Driven Performance Analysis of Scheduled Processes,” in *Business Process Management. BPM 2016. Lecture Notes in Computer Science, vol 9253*, vol. 9253, Springer, Cham, 2015.

[97] G. Balbo, S. Bruell and S. Ghanta, “Combining queueing networks and generalized stochastic petri nets for the solution of complex models of system behavior,” *IEEE Transactions on Computers,* vol. 37, no. 10, pp. 1251-1268, 1988.

[98] K. M. van Hee and L. Zerguini, “A new reduction method for the analysis of large workflow models,” in *Lecture Notes in Informatics; Vol. P-21*, Potsdam, Germany, 2002.

[99] G. Ciardo and T. K.S., “A decomposition approach for stochastic Petri net models,” in *Proceedings of the Fourth International Workshop on Petri Nets and Performance Models ,* Melbourne, 1991.

[100] C. Woodside and Y. Li, “Performance Petri net analysis of communications protocol software by delay-equivalent aggregation,” in *Proceedings of the Fourth International Workshop on Petri Nets and Performance Models* , Melbourne, 1991.

[101] X. Liu, M. Saat and C. Barkan, “Analysis of causes of major train derailment and their effect on accident rates,” *Transportation Research Record: Journal of the Transportation Research Board,* no. 2289, pp. 154-163, 2012.

[102] S. Dindar and S. Kaewunruen, “Assessment of Turnout-Related Derailments by Various Causes,” in *Pombo J., Jing G. (eds) Recent Developments in Railway Track and Transportation Engineering. GeoMEast 2017. Sustainable Civil Infrastructures.*, Springer, Cham, 2018.

[103] M. Ishak and S. K. S. DIndar, “Safety-based maintenance for geometry restoration of railway turnout systems in various operational environments,” in *Proceedings on The 21st National Convention on Civil Engineering,* Songkhala, Thailand, 2016.

[104] W. Zwanenburg, “The Swiss experience on the wear of railway switches & crossings,” in *7th Swiss Transport Research Conference*, Asconta, 2007.

[105] S. Hassankiadeh, Failure analysis of railway switches and crossings for the purpose of preventive maintenance, Stockholm: School of Architecture and the Built Environment, KTH Royal Institute of Technology: PhD Thesis, 2011.

[106] S. Dindar, S. Kaewunruen, M. An and A. Gigante-Barrera, “Derailment-based Fault Tree Analysis on Risk Management of Railway Turnout Systems,” in *IOP Conference Series: Materials Science and Engineering*, 2017.

[107] A. Beard, “Tunnel safety, risk assessment and decision-making,” *Tunnelling and Underground Space Technology,* no. 25, pp. 91-94, 2010.

[108] J. Roh, H. Ryou, W. Park and Y. Jang, “CFD simulation and assessment of life safety in a subway train fire,” *Tunnelling and Underground Space Technology* , no. 24, pp. 447-453, 2009.

[109] F. Taranda and M. King, “Structural Fire Protection of Railway Tunnels,” in *Railway Engineering Conference*, University of Westminster, 2009.

[110] L. Poon and R. Lau, “Fire Risk in Metro Tunnels and Stations,” *International Journal of Performability Engineering,,* vol. 3, no. 3, pp. 355-368, 2007.

[111] D. Howarth and C. Kara-Zaitri, “Fire saftey management at passenger terminals,” *Disaster Prevention and Management: An International Journal,* vol. 8, no. 5, pp. 362-369, 1999.

[112] l.-w. Pan, S. Lo, G.-x. Liao and B.-h. Cong, “Experimental Study of Smoke Control in Subway Station for Tunnel Area Fire by Water Mist System,” in *The 5th Conference on Performance-based Fire and Fire Protection Engineering,* 2011.

[113] K. Frank, N. Gravestock, M. Spearpoint and C. Fleischmann, “A review of sprinkler system effectiveness studies,” *Fire Science Reviews,* vol. 2, no. 6, 2013.

[114] H. D. Boyd and C. A. Locurto, “Reliability and Maintainability of Fire Protection Systems,” *Fire Safety Science* , vol. 1, pp. 963-970, 1986.

[115] W. Schneeweiss, The Fault Tree Method (in the Fields of Reliability and Safety Technology), Hagen: LiLoLe-Verlag GmbH, 1999.

[116] R. Taylor, Risk Analysis for Process Plant, Pipelines and Transport, London: E & FN Spon, 1994.

[117] M. Walter and W. Schneeweiss, The Modeling World of Reliability/Saftey Engineering, Hagen: LiLoLe-Verlag, 2005.

[118] R. Goodstein, Boolean Algebra, Oxford: Pergamon Press, 1963.

[119] J. D. Andrews and S. Beeson, “Birnbaum's Meeasure of Component Importance for Noncoherent Systems,” *IEE Transactions on Reliability,* vol. 52, no. 2, pp. 213-219, 2003.

[120] C. Wilhelmsen and L. Ostrom, Risk Assesment: Tools, Techniques, and Their Applications, John Wiley & Sons, 2012.

[121] The Institution of Engineering and Technology (IET), *Quantified Risk Assessment Techniques-Part 2: Event Tree Analysis,* Stevenage: IET, 2015.

[122] N. Thomopoulos, Essentials of Monte Carlo Simulation : Statistical Methods for Building Simulation Models, New York: Springer, 2012.

[123] J. Gentle, Random Number Generation and Monte Carlo Methods, New York: Springer, 2003.

[124] W. Dunn and J. Kenneth Shultis, Expolring Monte Carlo Methods, Amsterdam: Elsevier, 2012.

[125] T. Murata, “Petri Nets: Properties, Analysis and Applications,” *Proceesings of the IEEE,* vol. 77, no. 4, pp. 541-580, 1989.

[126] W. Schneeweiss, Petri Net Picture Book (An Elementary Introduction to the Best Pictoral Description of Temporal Changes), Hagen: LiLoLe-Verlag GmbH, 2004.

[127] J. Peterson, Petri net theory and the modeling of systems, New Jersey: Prentice-Hall, 1981.

[128] M. Zhou and K. Venhatesh, Modeling, simulation, and control of flexible manufacturing systems: a Petri net approach, Singapore: World Scientific publishing Co. Pte. Ltd., 1998.

[129] M. Audley, Rail Track Geometry Degredation and Maintenance Decision Making, University of Nottingham : PhD Thesis, 2014.

[130] D. Rama and J. Andrews, “A reliability analysis of railway switches,” *IMechE Part F: Journal of Rail and Rapid Transit,* vol. 227, no. 4, pp. 344-363, 2013.

[131] H. Wang, “A survey of maintenance policies of deteriorating systems,” *European Journal of Operational Research,* vol. 139, no. 3, pp. 469-489, 2002.

[132] B. Ghodrati, A. Ahmadi and D. Galar, “Reliability Analysis of Switches and Crossings – A Case Study in Swedish Railway,” *International Journal of Railway Research,* vol. 4, no. 1, pp. 1-12, 2017.

[133] INNOTRACK, “D3.3.1-List of key parameters for switch and crossing monitoring,” 2008.

[134] S. Kaewunruen, “Monitoring structural deterioration of railway turnout systems via dynamic wheel/rail interaction,” *Case Studies in Nondestructive Testing and Evaluation,* vol. 1, pp. 19-21, 2014.

[135] S. Kaewunruen, “Monitoring structural deterioration of railway turnout systems via dynamic wheel/rail interaction,” *Case Studies in Nondestructive Testing and Evaluation,* vol. 1, no. 1, pp. 19-24, 2014.

[136] A. Johansson, B. Pålsson, M. Ekh, J. Nielsen, M. Ander, J. Brouzoulis and E. Kassa, “Simulation of wheel–rail contact and damage in switches & crossings,” *Wear,* vol. 271, no. 1-2, pp. 472-481, 2011.

[137] Federal Railroad Administration, “Track Safety Data Base,” 2014. [Online]. Available: www.fra.dot.gov. [Accessed 21 11 2018].

[138] London Undeground, “Fault trees for derailment risk on the Jubilee line,” 2014.

[139] D. Prescott and J. Andrews, “A track ballast maintenance and inspection model for a rail network,” *Proc IMechE, Part O: Journal of Risk and Reliability,* no. 227, p. 251:266, 2013.

[140] M. Audley and J. Andrews, “The effects of tamping on railway track geometry degradation,” *IMechE Part F: Journal of Rail and Rapid Transit,* vol. 227, no. 4, pp. 376-391, 2013.

[141] E. Magel, M. Roney, J. Kalousek and P. Sroba, “The blending of theory and practice in modern rail grinding,” *Fatigue and Fracture of Engineering Materials and Structures,* vol. 26, no. 10, pp. 921-929, 2003.

[142] A. T. Cornish, “Life-time monitoing of in service switches and crossings through field experimentation,” Imperial College London , 2014.

[143] J. Andrews, D. Prescott and F. De Rozières, “A stochastic model for railway track asset management,” *Reliability Engineering Saftey Systems,* vol. 130, pp. 76-84, 2014.

[144] RailKonsult, “Review of European Renewal and Maintenance Methodologies. Techincal Appendix Number 3,” Office of Rail Regualtion, Surrey, 2012.

[145] MAINLINE, “Deliverablle 3.3: Rail Switches and Crossings. Development of new technologies for replacement,” 2014.

[146] C4R (Capacity for Rail), “Operational failue modes of Switches and Crossings, Public deliverable D 1.3.1,” 2015.

[147] D. Cannon, “Rail defects: an overview,” *Fatigue and Fracture of Engineering Materials and Structures,* vol. 26, no. 10, pp. 865-886, 2003.

[148] H. Wu, X. Shu and V. Wilson, “TCRP Report 71: Flange Climb Derailment Criteria and Wheel/Rail Profile Management and Maintenance Guidelines for Transit Operations,” Transportation Research Board, Washington D.C., 2005.

[149] Australian Rail Track Corportation Ltd. , “Maintenace of Crossings (ETM-03-03),” 2011.

[150] M. F. Rusu, Automation of Railway Switch and Crossing Inspection (PhD Thesis), University of Birmingham, 2015.

[151] W.-J. Zwanenburg, “A model for the life expectancy of railway switches and crossings for maintenace renewal planning in asset management systems,” *WIT Transactions on The Built Environmnet,* vol. 103, pp. 765-773, 2008.

[152] W.-J. Zwanenburg, “Degradation Processes of Switches and Crossings,” in *The Institution of Engineering and Technology International Conference on Railway Condition Monitoring.,* Birmingham, 2006.

[153] I. Gailienë, I. Podagëlis and S. O, “Research on the lifetime of the switch and assumptions of increasing it,” *Transport,* vol. 32, no. 2, pp. 150-155, 2008.

[154] A. Nissen, “LCC Analysis for Switches and Crossings: A Case Study from the Swedish Railway Network,” *International Journal of COMADEM,* vol. 12, no. 2, pp. 10-19, 2009.

[155] R. Carvel and G. Marlair, “A history of fire incidents in tunnels,” in *Handbook of Tunnel Fire Saftey,* London, ICE Publishing, 2015, pp. 1-41.

[156] Legislation - UK, “Fire Precautions (Sub-surface Railway Stations) Regulations,” 1989.

[157] Legislation - UK, “SI 2009/782 Fire Precautions (Sub-surface Railway Stations) (England),” 2009.

[158] J. Andrews and C. Fecarotti, “System design and maintenance modelling for saftey in extended life operation,” *Reliability Engineering & System Safety,* vol. 163, pp. 95-108, 2017.

[159] Y.-F. Tu, “Assessment of the Current False Alarm Situation from Fire Detection Systems in New Zealand and the Development of an Expert System for Their Identifications,” University of Canterbury, Civil Engineering, Christchurch, New Zealand, 2002.

[160] J. Taylor, Risk Analysis for Process Plant, Pipelines and Transport, Oxon: Taylor & Francis, 1994.

[161] T. Nyyssonen, J. Rajakko and O. Keski-Rahkonen, “On the reliability of fire detection and alarm systems: Exploration and analysis of data from nuclear and non-nuclear installations,” VTT, 2005.

[162] R. Isermann, Fault-Diagnosis Applications: Model-Based Condition Monitoring: Actuators, Drives, Machinery, Plants, Sensors and Fault-tolerant Systems, Berlin: Springer-Verlag, 2011.

[163] R. Beebe, Predictive Maintenace of Pumps Using Condition Monitoring, Elsevier Science and Technology Books, 2004.

[164] D. J. Moore, Thesis: Condition Monitoring of Diesel Engines, Manchester: The University of Manchester, 2013.

[165] D. Pham and D. Karaboga, Intelligent Optimisation Techniques, London: Springer-Verlag London Limited, 2000.

[166] H. Romeijn and R. Smith, “Simulated annealing for constrained global optimization,” *Journal of Global Optimization,* vol. 5, no. 2, pp. 101-126, 1994.

[167] J. Holand, Adaptations in Natural and Atrificial Systems, Ann Arbor, Michigan: University of Michigan Press, 1975.

[168] D. Goldberg, Genetic Algorithms in Search, Optimization and Machiene Learning, Reading: Addison-Wesley Publishing Company, 1989.

[169] D. Coley, An Introduction to Genetic Algorithms for Scientists and Engineers, Singapore: World Scientific Publishing Co. Pte. Ltd., 1999.

[170] P. Dunn and M. Davis, Measurement and Data Analysis for Engineering and Science, Fourth Edition, CRC Press, 2017.

[171] K. Lowell and K. Benke, “Uncertainty and risk analysis in hydrological models for land-use management,” in *Proceedings of the 7th International Symposium on Spacial Accuracy Assesment in Natural Resources and Environmnetal Sciences*, 2006.

[172] S. Chew, S. Dunnett and J. Andrews, “Phased mission modelling of systems with maintenance-free operating periods using simulated Petri nets,” *Reliability Engineering anf System Safety,* no. 93, pp. 980-994, 2008.

[173] J. D. Andrews and C. Fecarotti, “Modelling Life Extension on Saftey Critical Systems,” in *Safety and Reliability of Complex Engineered Systems*, Zurich, 2015.

[174] P. C. Yianni, L. C. Neves, D. Rama and J. D. Andrews, “Accelerating Petri-Net simulations using NVIDIA Graphics Processing Units,” *European Journal of Operational Research,* vol. 1, no. 265, pp. 361-371, 2018.

[175] G. Chalkidis, M. Nagasaki and S. and Miyano, “High performance hybrid functional petri net simulations of biological pathway models on cuda,” *IEEE/ACM transactions on computational biology and bioinformatics,* vol. 8, no. 6, pp. 1545-1556, 2011.

[176] R. Geist, J. Hicks, M. Smotherman and J. Westall, “Parallel Simulation of Petri Nets on Desktop PC Hardware,” in *Proceedings of the 2005 Winter Simulation Conference,* Orlando, 2005.

[177] R. Sloan and U. Buy, “Reduction rules for time Petri nets,” *Acta Informatica,* vol. 33, no. 7, pp. 687-706, 1996.

[178] M. Vakilzadeh, J. Beck and T. Abrahamsson, “Using Approximate Bayesian Computation by Subset Simulation for Efficient Posterior Assessment of Dynamic State-Space Model Classes,” *SIAM Journal of Scientific Computing,* vol. 40, no. 1, pp. B168-B195, 2018.

[179] J.-M. Michel, P. Pudlo, C. Robert and R. Ryder, “Approximate Bayesian Computation methods,” *Statistics and Computing,* vol. 22, no. 6, pp. 1167-1180, 2012.

[180] B. Turner and T. Van Zandt, “A tutorial on approximate Bayesian computation,” *Journal of Mathematical Psychology,* vol. 56, pp. 69-85, 2012.

[181] P. Del Moral, A. Doucet and A. Jasra, “An adaptive sequential Monte Carlo method for approximate Bayesian computation,” *Stastics and Computing,* vol. 22, pp. 1009-1020, 2012.

[182] M. Beaumont, J. Cournuet, J. Martin and C. Robert, “Adaptive approximate Bayesian computation,” *Biometrika,* vol. 94, no. 4, pp. 983-990, 2009.

[183] S. Sisson, Y. Fan and M. Takana, “Sequential Monte Carlo without liklihoods,” *Proceedings of the National Academy of Sciences,* vol. 104, no. 6, pp. 1760-1765, 2007.

[184] S.-K. Au and J. Beck, “Estimation of small failure probabilities in high dimensions by subset simulation,” *Probabilistic Engineering Mechanics,* vol. 16, no. 4, pp. 263-277, 2001.

[185] M. Chiachio, J. Beck, J. Chiachio and G. Rus, “Approximate Bayesian Computation by Subset Simulation,” *SIAM Journal of Scientific Computing,* vol. 36, no. 3, pp. A1339-A1358, 2014.

[186] E. A. Lee and A. Sangiovanni-Vincentelli, “A Framework for Comparing Models of Computation,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* , vol. 17, no. 12, pp. 1217 - 1229, 1998.

[187] M. M. Deza and E. Deza, Encyclopedia of Distances, Springer, 2009.

[188] S. Vinga and J. Almeida, “Alignment-free sequence comparison—a review,” *Bioinformatics,* vol. 19, no. 4, p. 513–523, 2003.

[189] A. Kraskov, H. Stögbauer and P. Grassberger, “Estimating mutual information,” *Physical Review E,* vol. 96, no. 6, p. 066138, 2011.

[190] S. Naybour, J. Andrews and M. Chiachio-Ruano, “Efficient Risk Based Optimization of Large System Models using a Reduced Petri Net,” in *Proceedings of the 29th European Safety and Reliability Conference (ESREL),* Hannover, Germany, 2019.

[191] J. Pfaendtner, “Material Failures in Fire Protection Systems,” in *Suppression, Detection and Signalling Reasearch and Applications Conference,* Orlando, 2014.

[192] S. Ross, S. Nowlen and T. Tanaka, “Ageing Assessment for Active Fire Protection Systems,” Sandia National Laboratories, 1995.

[193] Exidia Consulting, “Failure Modes, Effects and Diagnostic Analysis: Valve Control and Monitoring System,” Report number Abc 11/12-345 R001, 2010.

[194] H. Liu, T. Walski, G. Fu and C. Zhang, “Failure Impact Analysis of Isolation Valves in a Water Distribution Network,” *Journal of Water Resources Planning and Management,* vol. 143, no. 7, p. 04017019, 2017.

[195] P. Smith and R. Zappe, Valve Selection Handbook (5th Edition), Oxford: Elsevier, 2004.

[196] J. Riauke and L. Bartlett, “An offshore saftey system optimization using an SPEA2-based approach,” *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* , vol. 222, no. 3, pp. 271-282, 2008.

[197] K. Wong, “Study on the Reliability of Manual Call Points in Residential Buildings,” *International Journal on Engineering Performance-Based Fire Codes,* vol. 6, no. 4, pp. 344-352, 2004.

Appendix 1

This appendix details the categories used for different rail defects in the model prediction derailment due to a rail break or excessive wear, used in Chapter 4.

Category 1: Subsurface Cracking

Defect	Deep Seated Shelling	Bolt Hole Crack	Split Head defect	Base Defect	Web Defect
Description	Loss of rail material due to collapse of the rail gauge corner. Due to excessive loading and sheer failure.	A crack across the rail web that originates from a bolt hole. Caused by bending, residual and thermal stresses.	A split on the inside of the rail head initiated by shells and weaknesses in the metal or by RCF. Propagates due to bending, residual and thermal stress.	Any defect or break in the base of the rail.	Any internal or external fracture or defect in the rail web. For instance, piped rail or head and web separation.
Measurement	Ultrasonic and visual (late stage only) testing.	Ultrasonic and visual testing.	Ultrasonic testing or visual inspection to note resulting surface defects.	Visual and ultrasonic testing.	Visual and ultrasonic testing.
Quantification	Area of rail affected.	Length of crack.	Length of crack.	The length of the crack or break.	The size of the defect.
Associated Maintenance Activities	Replacement	Replacement	Replacement	Replacement	Replacement

Table A1.1: Rail subsurface cracking defects

Category 2: Loss of Rail head Material

Defect	Vertical Wear	Plastic Flow	Corrosion
Description	A misshape of the rail head due to wheel rail interaction and grinding processes.	Deformation of the rail due to wheel rail contact stresses.	The decaying of metal within the rail due to wet or damp areas. Results in cavities in the rail.
Measurement	Manual testing via a measurement gauge.	Manual testing via measurement gauge.	Visual inspection.
Quantification	Extent of profile change.	Extent of profile change.	Area of corrosion.
Associated Maintenance Activities	Replacement	Replacement	Replacement

Table A1.2: Rail defects where there is a loss of rail head material

Category 3: Surface Cracking, Rolling Contact Fatigue and Wear

Defect	Rail Gauge Corner Cracking	Surface Shelling	Squats	Lateral Wear
Description	Thin cracks on the gauge corner of the rails, result of high contact and sheer stresses.	Loss of lumps of rail due to combinations of surface and subsurface cracks.	A depression in the rail caused by the combination of sub-surface cracking and deposits of debris in the depression.	Occurs on the gauge face due to high wheel flange force.
Measurement	Visual or ultrasonic testing	Visual or ultrasonic testing	Visual or ultrasonic testing	Visual or ultrasonic testing.
Quantification	The quantity and depth of the cracks present.	Length and depth of the shell.	Depth of Squat depression.	Extend of misshape of rail.
Associated Maintenance Activities	Replacement or rail grinding.	Replacement or rail grinding.	Replacement or rail grinding.	Replacement or rail grinding.

Table A1.3:Rail surface defects

Appendix 2

Data used in the sample application of the models in Chapter 4, where normal distribution parameters are given in order of $N(\mu,\sigma)$, Weibull distribution parameters are given in order of $Wei(\eta,\beta)$ and Lognormal distribution parameters are given in order of $Lognormal(\mu,\sigma^2)$.

Part A: Component degradation, inspection and maintenance models

Label	Description	Distribution	Parameters (months)
	Ballast		
T3	Ballast condition moves from State I to II. This conditional on number of previous tamps. (5 distributions, distribution changes when number of tamps reaches 1,2,4 and 6)	Weibull (5 distributions)	250,15 200,15 180,10 160,5 150,5
T6	Ballast condition moves from State II to III. This is conditional on the number of previous tamps. (3 distributions, distributions changes when number of tamps reaches 2 and 4)	Weibull (3 distributions)	100,10 50,10 20,5
T9	Ballast condition moves from State III to IV. This is conditional on the number of tamps. (2 distributions, distribution changes when then number of tamps reaches 2)	Weibull (2 distributions)	10,6 8,10
T12	Ballast condition moves from State IV to State V.	Weibull	8,3
Ta1	Early tamping (Resets the condition of the ballast)	Normal	0.0001,0.00001
Ta2	Routine tamping (Resets the condition of the ballast)	Normal	0.0175,0.001
Ta3	Priority tamping (Resets the condition of the ballast)	Normal	0.035,0.001
Ta4	Emergency tamping (Resets the condition of the ballast)	Normal	0.035,0.001

U1	Early undercutting (Resets the condition of the ballast)	Normal	0.0001,0.00001
U2	Routine undercutting (Resets the condition of the ballast)	Normal	0.0175,0.001
U3	Priority undercutting (Resets the condition of the ballast)	Normal	0.035,0.001
U4	Emergency undercutting (Resets the condition of the ballast)	Normal	0.035,0.001
q1	Inspection is successful	Probability	0.9,0.1
q2	Inspection is successful	Probability	0.95,0.05
	Sleepers and Clips		
T1	Sleeper condition moves from working to failed state.	Weibull	450,10
T2	Clip condition moves from working to failed state	Weibull	360,4
I1	Sleeper and clip inspection interval	Global transition	3
D1	Sleeper and replacement delay	Normal	0.5,0.05
D2	Clip replacement delay	Normal	0.5,0.05
R1	Sleeper unit replacement	Normal	0.0001,0.00001
R2	Clip unit replacement	Normal	0.0001,0.00001
D3	Sleeper and clip population scheduling interval	Normal	0.01,0.001
RA	Sleeper and clip population replacement time	Normal	0.015,0.01
	Rail components		
q1	Driver detects a rail break	Probability	0.7,0.3
q2,q3	Inspection is successful	Probability	0.95,0.5
T1	Sub-surface cracking of the fixed rails (State I to II)	Weibull	150,5
T2	Wear of the fixed rails (State I to II)	Weibull	80,4
T3	RCF and surface cracking of the fixed rails (State I to II)	Weibull (3 distributions)	80,3,18,4,12,6
T4	Sub-surface cracking of the fixed rails (State II to III)	Weibull	30,8
T5	Wear of the fixed rails (State II to III)	Weibull	24,7
T6	RCF and surface cracking of fixed rails (State II to III)	Weibull (3 distributions)	24,6,18,9,12,10
T7	Sub-surface cracking of the fixed rails (State III to IV)	Weibull	6,3
T8	Wear of the fixed rails (State III to IV)	Weibull	12,4
T9	RCF and surface cracking of the fixed rails (State III to IV)	Weibull (2 distributions)	12,4,9,3
T10	Sub-surface cracking of the fixed rails (State IV to V)	Weibull	5,3
T11	Wear of the fixed rails (State IV to I)	Weibull	6,3
T12	RCF and surface cracking of the fixed rails (State IV to V)	Weibull	5,3
R1	Early replacement of the fixed rails	Normal	0.0001,0.00001

R2	Routine replacement of the fixed rails	Normal	0.0175,0.001
R3	Priority replacement of the fixed rails	Normal	0.035,0.001
R4	Emergency replacement of the fixed rails	Normal	0.035,0.001
G1	Early grinding of the fixed rails.	Normal	0.0001,0.00001
G2	Routine grinding of the fixed rails	Normal	0.0175,0.001
G3	Priority grinding of the fixed rails	Normal	0.035,0.001
q7	Driver detects a rail break	Probability	0.8,0.2
q8,q9	Inspection is successful	Probability	0.95,0.5
T13	Internal cracking of switch rails (State I to II)	Weibull	60,8
T14	Wear of switch rails (State I to II)	Weibull	48,6
T15	RCF and surface cracking of switch rails (State I to II)	Weibull (3 distributions)	24,6,18,8,12,6
T16	Misalignment of switch rail (State I to II)	Weibull	12,4
T17	Internal cracking of switch rails (State II to III)	Weibull	30,5
T18	Wear of switch rails (State II to III)	Weibull	24,6
T19	RCF and surface cracking of switch rails (State II to III)	Weibull (3 distributions)	24,6,18,7,12,8
T20	Misalignment of switch rail (State II to III)	Weibull	6,3
T21	Internal cracking of switch rails (State III to IV)	Weibull	6,3
T22	Wear of switch rails (State III to IV)	Weibull	12,4
T23	RCF and surface cracking of switch rails (State III to IV)	Weibull (3 distributions)	12,4,9,3
T24	Misalignment of switch rail (State III to IV)	Weibull	4,6
T25	Internal cracking of switch rails (State IV to V)	Weibull	3,7
T26	Wear of switch rails (State IV to V)	Weibull	6,6
T27	RCF and surface cracking of switch rails (State IV to V)	Weibull	4,8
T28	Misalignment of switch rail (State IV to V)	Weibull	4,9
R1	Early replacement of the switch rails	Normal	0.0001,0.00001
R2	Routine replacement of the switch rails	Normal	0.0175,0.001
R3	Priority replacement of the switch rails	Normal	0.035,0.001
R4	Emergency replacement of the switch rails	Normal	0.035,0.001
G1	Early grinding of the switch rails.	Normal	0.0001,0.00001
G2	Routine grinding of the switch rails	Normal	0.0175,0.001
G3	Priority grinding of the switch rails	Normal	0.035,0.001
A1	Early replacement of the switch rails.	Normal	0.0001,0.00001

A2	Routine replacement of the switch rails	Normal	0.0175,0.001
A3	Priority replacement of the switch rails	Normal	0.035,0.001
A4	Emergency replacement of the switch rails.	Normal	0.035,0.001
q15,q16	Inspection is successful	Probability	0.95,0.5
T29	Surface cracking of nose (State I to II)	Weibull (3 distributions)	24,6,18,7,12,8
T30	Sub-surface cracking of nose (State I to II)	Weibull	42,7
T31	Plastic deformation of nose (State I to II)	Weibull (3 distributions)	48,4,12,5,6,7
T32	Surface cracking of nose (State II to III)	Weibull (2 distributions)	12,4,6,7
T33	Sub-surface cracking of nose (State II to III)	Weibull	12,3
T34	Plastic deformation of nose (State II to III)	Weibull (2 distributions)	24,4,12,5
T35	Surface cracking of nose (State III to IV)	Weibull	6,4
T36	Sub-surface cracking of nose (State III to IV)	Weibull	6,3
T37	Plastic deformation of nose (State III to IV)	Weibull	6,5
T38	Surface cracking of nose (State IV to V)	Weibull	5,7
T39	Sub-surface cracking of nose (State IV to V)	Weibull	5,6
T40	Plastic deformation of nose (State IV to V)	Weibull	4,3
R1	Early replacement of the crossing nose	Normal	0.0001,0.00001
R2	Routine replacement of the crossing nose	Normal	0.0175,0.001
R3	Priority replacement of the crossing nose	Normal	0.035,0.001
R4	Emergency replacement of the crossing nose	Normal	0.035,0.001
G1	Early grinding of the crossing nose.	Normal	0.0001,0.00001
G2	Routine grinding of the crossing nose	Normal	0.0175,0.001
G3	Priority grinding of the crossing nose	Normal	0.035,0.001
W1	Early welding of the crossing nose.	Normal	0.0001,0.00001
W2	Routine welding of the crossing nose	Normal	0.0175,0.001
W3	Priority welding of the crossing nose	Normal	0.035,0.001
q22,q23	Inspection is successful	Probability	0.95,0.5
T41	Plastic deformation of check rails (State I to II)	Weibull (2 distributions)	36,8,28,4
T42	Longitudinal cracking of check rails (State I to II)	Weibull	24,6
T43	Plastic deformation of check rails (State II to III)	Weibull	12,3,6,4

		(2 distributions)	
T44	Longitudinal cracking of check rails (State II to III)	Weibull	18,6
T45	Plastic deformation of check rails (State III to IV)	Weibull	6,3
T46	Longitudinal cracking of check rails (State III to IV)	Weibull	6,3
T47	Plastic deformation of check rails (State IV to V)	Weibull	5,3
T48	Longitudinal cracking of check rails (State IV to V)	Weibull	5,4
R1	Early replacement of the check rails	Normal	0.0001,0.00001
R2	Routine replacement of the check rails	Normal	0.0175,0.001
R3	Priority replacement of the check rails	Normal	0.035,0.001
R4	Emergency replacement of the check rails	Normal	0.035,0.001
G1	Early grinding of the check rails	Normal	0.0001,0.00001
G2	Routine grinding of the check rails	Normal	0.0175,0.001
G3	Priority grinding of the check rails	Normal	0.035,0.001
I1-I6	Inspection interval	Normal	2,0.05
	Stretcher bars		
T1	Stretcher bar condition moves from State I to II	Weibull	36,9
T2	Stretcher bar condition moves from State II to III	Weibull	9,3
T3	Stretcher bar condition moves from State III to IV	Weibull	6,4
T4	Stretcher bar condition moves from State IV to V	Weibull	3,4
R1	Early replacement of the stretcher bars	Normal	0.0001,0.00001
R2	Routine replacement of the stretcher bars	Normal	0.0175,0.001
R3	Priority replacement of the stretcher bars	Normal	0.035,0.001
R4	Emergency replacement of the stretcher bars	Normal	0.035,0.001
q1	Probability that inspection is successful		0.85,0.15
	Slide Chairs		
T1	Slide chair condition moves from State I to II (ageing)	Weibull	60,4
T4	Slide chair condition moves from State II to III (ageing)	Weibull	20,3
T6	Slide chair condition moves from State III to IV (ageing)	Weibull	4,3
T8	Slide chair condition moves from State IV to V (ageing)	Weibull	3,3
T2	Slide chair condition moves from State I to II (lubrication and debris build up)	Weibull	9,5
T5	Slide chair condition moves from State II to III (lubrication and debris build up)	Weibull	3,3
T7	Slide chair condition moves from State III to IV (lubrication and debris build up)	Weibull	2,4
T9	Slide chair condition moves from State IV to V (lubrication and debris build up)	Weibull	3,4

T3	External debris falls into slide chairs	Uniform	0.0016
R1	Early replacement of the slide chairs	Normal	0.0001,0.00001
R2	Routine replacement of the slide chairs	Normal	0.0175,0.001
R3	Priority replacement of the slide chairs	Normal	0.035,0.001
R4	Emergency replacement of the slide chairs	Normal	0.035,0.001
L1	Early clearing and lubrication of the slide chairs	Normal	0.0175,0.001
L2	Routine clearing and lubrication of the slide chairs	Normal	0.0175,0.001
L3	Priority clearing and lubrication of the slide chairs	Normal	0.0175,0.001
L4	Emergency clearing and lubrication of the slide chairs	Normal	0.0175,0.001
q1	Probability that inspection is successful		0.05,0.95
D1	The time taken for an inspection to be completed	Normal	0.01,0.001
	POE and Locking device		
T1	POE condition moves from State I to II	Weibull	80,6
T2	Locking device condition moves from State I to V	Weibull	95,6
T3	POE condition moves from State II to III	Weibull	60,7
T4	POE condition moves from State III to IV	Weibull	40,6
T5	POE condition moves from State IV to V	Weibull	20,6
T8	Aged based early maintenance interval for the locking device	Lognormal	3.5,0.1
Rp1	Aged based routine maintenance interval for the locking device	Lognormal	3.9,0.05
Rp2	Aged based priority maintenance interval for the locking device	Lognormal	4.1,0.05
Rl1	Time taken for an inspection to be completed	Normal	0.01,0.001
Rl2	Probability that an inspection is unsuccessful		0.1,0.9
Rl3	Early replacement of the POE	Normal	0.0001,0.00001
I1	Routine replacement of the POE	Normal	0.0001,0.00001
D1	Priority replacement of the POE	Normal	0.0001,0.00001
q1	Emergency replacement of the POE	Normal	0.0001,0.00001
R1	Early replacement of the locking device	Normal	0.0001,0.00001
R2	Routine replacement of the locking device	Normal	0.0001,0.00001
R3	Priority replacement of the locking device	Normal	0.0001,0.00001
R4	Emergency replacement of the locking device	Normal	0.0001,0.00001
	Switch position detector		
T1	Switch position detector condition moves from State I to V	Weibull	48,10
Rs1	Aged based early maintenance interval for the switch position detector	Lognormal	2.5,0.3
Rs2	Aged based routine maintenance interval for the switch position detector	Lognormal	3,0.25

Rs3	Aged based priority maintenance interval for the switch position detector	Lognormal	3.5,0.3
q1	Probability that the inspection is unsuccessful		0.1,0.9
D1	Time taken for an inspection to be completed	Normal	0.01,0.001
R1	Early replacement of the switch position detector	Normal	0.0001,0.00001
R2	Routine replacement of the switch position detector	Normal	0.0001,0.00001
R3	Priority replacement of the switch position detector	Normal	0.0001,0.00001
R4	Emergency replacement of the switch position detector	Normal	0.0001,0.00001
	External signal failure		
T1	Time taken for an external signal failure to occur	Weibull	60,10
D1	Time taken for the external signal failure to be corrected	Normal	0.0001,0.00001
q1	Probability that the external signal failure results in the switch rails lying in a hazardous position		0.1,0.9

Table A2.1: Sample data values for the component degradation, maintenance and inspection transitions used in the application in Chapter 4

Part B: Maintenance scheduling models

Label	Description	Distribution	Parameters
	Maintenance scheduling models		
T1	The time interval between full replacement of the S&C	Normal	96,1
D1	The time for a full replacement to be completed	Normal	0.5,0.05
D2	The delay between a derailment occurring and full S&C replacement scheduling.	Normal	0.0001,0.00001
W1	The time between an identified need for routine ballast maintenance and the maintenance being carried out	Normal	3,1
D3	The time between the identified need for priority ballast maintenance and the maintenance being carried out	Normal	0.5,0.25
D4	The time between the identified need for emergency ballast maintenance and the maintenance being carried out	Normal	0.125,0.001
D5	The time taken for the maintenance to be de-activated	Normal	0.01,0.001
q1	The probability of undercutting the ballast as opposed to tamping		0.25,0.75
W2	The time between an identified need for routine component replacement or manual intervention and the maintenance being carried out	Normal	2,1
D6	The time between the identified need for priority component replacement or manual intervention and the maintenance being carried out	Normal	0.5,0.25
D7	The time between the identified need for emergency component replacement or manual intervention and the maintenance being carried out	Normal	0.01,0.001
D8	The time taken for the maintenance to be deactivated	Normal	0.125,0.001

W3	The time between the identified need for routine track grinding and the maintenance being carried out	Normal	3,1
D9	The time between the identified need for priority track grinding and the maintenance being carried out	Normal	1,0.5
D10	The time taken for the maintenance to be deactivated	Normal	0.01,0.001
	Over speeding		
T1	The arrival rate of an over speeding train (conditional on any applied speed restrictions)	Normal	240,60 (X1=0) 120,40 (X1=1)
q1	The probability that an over speeding train causes a derailment		0.1,0.9
I1	Inspection interval for visual inspection	Normal	1,0.5
I2	Inspection interval for specialist inspection	Normal	3,1
I3	Inspection interval for the POE testing	Normal	3,1

Table A2.2: Sample data values used in the maintenance scheduling models in the application in Chapter 4

Part C: Derailment models

Label	Description	Distribution	Parameters
	Failure models		
T1	Time taken for a derailment to occur, if no restrictions are applied, due to a failed switch rail position	Normal	0.01,0.001
T2	Time taken for a derailment to occur, if only speed restrictions are applied, due to a failed switch rail position	Normal	0.02,0.001
T3	Time taken for a derailment to occur, if no restrictions are applied, due to a geometry failure	Normal	0.3,0.01
T4	Time taken for a derailment to occur, if only speed restrictions are applied, due to a geometry failure	Normal	0.4,0.01
D1	Delay assigned for less likely derailment, due to geometry error caused by the first failed state of the sleeper and clip population.	Normal	3,2
T5	Time taken for a derailment to occur, if no restrictions are applied, due to wear on the rail causing wheel climb	Normal	0.5,0.01
T6	Time taken for a derailment to occur, if only speed restrictions are applied, due to wear on the rail causing wheel climb	Normal	1,0.01
T7	Time taken for a derailment to occur, if no restrictions are applied, due to a rail break	Normal	0.01,0.001
T8	Time taken for a derailment to occur, if only speed restrictions are applied, due to a rail break	Normal	0.02,0.001
D1	The time for the failed state to be identified if it is revealed that the switch rail is in the incorrect position	Normal	0.01,0.001

Table A2.3: Sample data values used for the derailment models in the application in Chapter 4

Appendix 3:

Sample data used in the Petri net models in Chapter 5, where normal distribution parameters are given in order of $N(\mu,\sigma)$, Weibull distribution parameters are given in order of $Wei(\eta,\beta)$ and Lognormal distribution parameters are given in order of $Lognormal(\mu,\sigma^2)$.

Net	Transition	Description	Distribution	Parameters
A	T1	Pipework fails due to age	Weibull	600,1.5
A	T2	Random useful life pipework failure	Uniform	0.000139
A	R1	Early maintenance scheduling of pipework.	Lognormal	5.5,0.04
A	R2	Routine maintenance scheduling of pipework.	Lognormal	6,0.02
A	R3	Maintenance scheduling of pipework on failure.	Lognormal	-5,0.5
A	I1	Inspection interval of pipework.	Global (3 intervals)	12,6,3
A	D1	Maintenance of pipework.	Lognormal	-2,0.5
B	T1	Electric pump moves from good state to degraded state due to age	Weibull	80,2
B	T2	Electric pump moves from degraded state to failed state due to age	Weibull	40,3.5
B	T3	Random useful life electric pump failure	Uniform	0.00104
B	R1	Early maintenance scheduling of pump.	Lognormal	3.5,0.03
B	R2	Routine maintenance scheduling of pump.	Lognormal	4,0.02
B	R3	Maintenance scheduling of pump on failure.	Lognormal	-5,0.5
B	R4	Maintenance scheduling of pump on partial failure.	Lognormal	-3,0.6
B	I1	Inspection interval of pump.	Global (3 intervals)	12,6,3
B	D1	Maintenance of pump.	Lognormal	-2,0.5
U	T1	Jockey pump moves from good state to degraded state due to age	Weibull	40,3
U	T2	Jockey pump moves from degraded state to failed state due to age	Weibull	20,6
U	T3	Random useful life pump failure	Uniform	0.0042
U	R1	Early maintenance scheduling of pump.	Lognormal	2.75,0.01
U	R2	Routine maintenance scheduling of pump.	Lognormal	3.5,0.02
U	R3	Maintenance scheduling of pump on failure.	Lognormal	-5,0.05
U	R4	Maintenance scheduling of pump on partial failure.	Lognormal	-3,0.6
U	I1	Inspection interval of pump.	Global (3 intervals)	12,6,3
U	D1	Maintenance of pump.	Lognormal	-2,0.5
V	T1	Diesel pump moves from good state to degraded state due to age	Weibull	100,1.5

V	T2	Diesel pump moves from degraded state to failed state due to age	Weibull	56,3
V	T4	Random useful life pump failure	Uniform	0.000834
V	R1	Early maintenance scheduling of pump.	Lognormal	4,0.02
V	R2	Routine maintenance scheduling of pump.	Lognormal	4.5,0.01
V	R3	Maintenance scheduling of pump on failure.	Lognormal	-5,0.5
V	R4	Maintenance scheduling of pump on partial failure.	Lognormal	-3,0.6
V	I1	Inspection interval of pump.	Global (3 intervals)	12,6,3
V	D1	Maintenance of pump.	Lognormal	-2,0.5
C	T1	Diesel tank failure due to age	Weibull	240,1.25
C	T2	Random useful-life diesel tank failures	Uniform	0.000556
C	R1	Early maintenance scheduling of the diesel tank.	Lognormal	4.25,0.01
C	R2	Routine maintenance scheduling of the diesel tank.	Lognormal	4.8,0.05
C	R3	Maintenance scheduling of the diesel tank on failure.	Lognormal	-5,0.5
C	I1	Inspection of the diesel tank.	Global (3 intervals)	12,6,3
C	D1	Maintenance of the diesel tank.	Lognormal	-2,0.5
D	T1	Ring main failure due to age	Weibull	500,1.75
D	T2	Useful life ringmain failure	Uniform	0.00021
D	R1	Early maintenance scheduling of ringmain.	Lognormal	5.1,0.05
D	R2	Routine maintenance scheduling of ringmain.	Lognormal	5.7,0.05
D	R3	Maintenance scheduling of ringmain on failure.	Lognormal	-5,0.5
D	I1	Inspection of ringmain.	Global (3 intervals)	12,6,3
D	D1	Maintenance of ringmain.	Lognormal	-2,0.5
E	T1	Head and strainer failure due to age	Weibull	72,2
E	T2	Useful life head and strainer failure	Uniform	0.00081
E	R1	Early maintenance scheduling of the head and strainers.	Lognormal	3.4,0.05
E	R2	Routine maintenance scheduling of the head and strainers.	Lognormal	3.8,0.025
E	R3	Replacement of the head and strainers on failure.	Lognormal	-5,0.5
E	I1	Inspection interval of the head and strainers.	Global (3 intervals)	12,6,3
E	D1	Replacement of the head and strainers.	Lognormal	-2,0.5
F	T1	Isolation valve failure due to age	Weibull	84,3
F	T2	Useful life isolation valve failures	Uniform	0.0021
F	R1	Early maintenance scheduling of the isolation valve.	Lognormal	2.6,0.15

F	R2	Routine maintenance scheduling of the isolation valve.	Lognormal	3.1,0.1
F	R3	Maintenance of the isolation valve on failure.	Lognormal	-5,0.5
F	I1	Inspection interval of the isolation valve.	Global (3 intervals)	12,6,3
F	D1	Repair time of the isolation valve.	Lognormal	-2,0.5
F	p1	Probability that the isolation valve fails in a closed position.		0.1,0.9
G	T1	Pressure release valve failure due to age	Weibull	48,4
G	T2	Useful life pressure valve failure	Uniform	0.00279
G	R1	Early maintenance scheduling of the pressure release valve.	Lognormal	2.9,0.01
G	R2	Routine maintenance scheduling of the pressure release valve.	Lognormal	3.25,0.01
G	R3	Maintenance of the pressure release valve on failure.	Lognormal	-5,0.5
G	I1	Inspection interval of the pressure release valve.	Global (3 intervals)	12,6,3
G	D1	Repair time of the pressure release valve.	Lognormal	-2,0.5
G	p1	Probability that the pressure release valve fails in a closed position.		0.5,0.5
H	T1	Deluge valve failure due to age	Weibull	110,3.5
H	T2	Useful life deluge valve failures	Uniform	0.00139
H	R1	Early maintenance scheduling of the deluge valve.	Lognormal	3.5,0.05
H	R2	Routine maintenance scheduling of the deluge valve.	Lognormal	4,0.02
H	R3	Maintenance of the deluge valve on failure.	Lognormal	-5,0.5
H	I1	Inspection interval of the deluge valve.	Global (3 intervals)	12,6,3
H	D1	Repair time of the deluge valve.	Lognormal	-2,0.5
H	p1	Probability that the deluge valve fails in a closed position.		0.5,0.5
I	T1	Solenoid failure due to age	Weibull	56,4
I	T4	Useful life solenoid failure	Uniform	0.00218
I	R1	Early maintenance scheduling of the solenoid.	Lognormal	2.6,0.1
I	R2	Routine maintenance scheduling of the solenoid.	Lognormal	3.1,0.05
I	R3	Maintenance of the solenoid on failure.	Lognormal	-5,0.5
I	I1	Inspection interval of the solenoid.	Global (3 intervals)	12,6,3
I	D1	Repair time of the solenoid.	Lognormal	-2,0.5
I	p1	Probability that the solenoid fails triggering a failure.		0.5,0.5
J	T1	Manual initiation device fails due to age	Weibull	72,3.75
J	T4	Random manual initiation device failure.	Uniform	0.00083
J	R1	Early maintenance of the manual initiation valve.	Lognormal	3.2,0.05

J	R2	Routine maintenance of manual initiation valve.	Lognormal	3.7,0.02
J	R3	Maintenance of manual initiation valve on failure.	Lognormal	-5,0.5
J	I1	Inspection of manual initiation valve.	Global (3 intervals)	12,6,3
J	D1	Maintenance of manual initiation valve.	Lognormal	-2,0.5
K	T1	Pressure sensor failure due to age	Weibull	120,5
K	T4	Useful life pressure sensor failure	Uniform	0.000834
K	R1	Early maintenance scheduling of pressure sensor.	Lognormal	4.24,0.02
K	R2	Routine maintenance scheduling of pressure sensor.	Lognormal	4.61,0.02
K	R3	Pressure sensor maintenance scheduling on failure.	Lognormal	-5,0.5
K	I1	Inspection interval of the pressure sensors.	Global (3 intervals)	12,6,3
K	D1	Maintenance of pressure sensors	Lognormal	-2,0.5
K	p1	Probability that pressure sensor failure gives no reading		0.5,0.5
K	p2	Probability that pressure sensor failure gives a reading higher than true		0.2,0.8
L	T1	A smoke detector in the population fails due to age	Weibull	144,3
L	T2	Useful life smoke detector failure	Uniform	0.000583
L	T3	A second smoke detector in the population fails due to age, given there has been a failure already (depends on type of failure)	Weibull (2distribuitons)	120,3,100,3
L	R1	Early maintenance scheduling of the smoke detector.	Lognormal	4.1,0.01
L	R2	Routine maintenance scheduling of the smoke detector.	Lognormal	4.6,0.02
L	R3	Maintenance scheduling of the smoke detector on failure.	Lognormal	-5,0.5
L	I1	Inspection interval of the smoke detectors.	Global (3 intervals)	12,6,3
L	D1	Maintenance of the smoke detectors.	Lognormal	-2,0.5
L	p1	Probability that the smoke detector failure is sufficient to cause a system failure (random)		0.1,0.9
L	p2	Probability that the smoke detector failure is sufficient to cause a system failure (age)		0.3,0.7
L	p3	Probability that the smoke detector failure is unrevealed		0.2,0.8
P	T1	A heat detector in the population fails due to age	Weibull	180,2.5
P	T2	Useful life heat detector failure	Uniform	0.000347
P	T3	A second heat detector in the population fails due to age, given there has been a failure already (depends on type of failure)	Weibull (2distribuitons)	150,2.5,80,2.5

P	R1	Early maintenance scheduling of the heat detector.	Lognormal	4.2,0.02
P	R2	Routine maintenance scheduling of the heat detector.	Lognormal	4.67,0.01
P	R3	Maintenance scheduling of the heat detector on failure.	Lognormal	-5,0.5
P	I1	Inspection interval of the heat detectors.	Global (3 intervals)	12,6,3
P	D1	Maintenance of the heat detectors.	Lognormal	-2,0.5
P	p1	Probability that the heat detector failure is sufficient to cause a system failure (random)		0.1,0.9
P	p2	Probability that the heat detector failure is sufficient to cause a system failure (age)		0.1,0.7
P	p3	Probability that the heat detector failure is unrevealed		0.2,0.8
M	T1	A call point in the population fails due to age	Weibull	96,3.5
M	T2	Useful life call point failure	Uniform	0.0159
M	T3	A second call point in the population fails due to age, given there has been a failure already (depends on type of failure)	Weibull (2distribuitons)	80,3.5,60,3.5
M	R1	Early maintenance scheduling of the call point.	Lognormal	3.2,0.03
M	R2	Routine maintenance scheduling of the call point.	Lognormal	3.7,0.02
M	R3	Maintenance scheduling of the call point on failure.	Lognormal	-5,0.5
M	I1	Inspection interval of the call point.	Global (3 intervals)	12,6,3
M	D1	Maintenance of the call point.	Lognormal	-2,0.5
M	p1	Probability that the call point failure is sufficient to cause a system failure (random)		0.1,0.9
M	p2	Probability that the call point failure is sufficient to cause a system failure (age)		0.1,0.7
M	p3	Probability that the call point failure is unrevealed		0.2,0.8
N	T1	The alarm sounders fail due to age	Weibull	300,2
N	T2	Useful life sounder failures	Uniform	0.000463
N	T3	The alarm wiring fails due to age	Weibull	480,2
N	T4	Useful life wiring failures	Uniform	0.000119
N	R1	Early maintenance scheduling of the alarm system.	Lognormal	4.38,0.02
N	R2	Routine maintenance scheduling of the alarm system.	Lognormal	4.82,0.01
N	R3	Maintenance scheduling of the alarm system on failure.	Lognormal	-5,0.5
N	I1	Inspection interval of the alarm system	Global (3 intervals)	12,6,3
N	D1	Maintenance of the alarm system	Lognormal	-2,0.5
N	p1	Probability that the failure of the alarm sounder leads to	Normal	0.1,0.9

		complete alarm system failure.		
O	T1	The control box fails due to age	Weibull	144,2.5
O	T3	The control box battery fails due to age	Weibull	60,7
O	T7	Useful life control box failure	Uniform	0.00054
O	T8	Useful life battery failure	Uniform	0.00836
O	R1	Early maintenance scheduling of the control box.	Lognormal	3.8,0.02
O	R2	Routine maintenance scheduling of the control box.	Lognormal	4.25,0.01
O	R3	Maintenance of the control box on failure.	Lognormal	-5,0.5
O	R4	Early maintenance scheduling of the control box battery.	Lognormal	3,0.01
O	R5	Routine maintenance scheduling of the control box battery.	Lognormal	3.55,0.02
O	R6	Maintenance of the control box battery on failure.	Lognormal	-5,0.5
O	D1	Control box maintenance.	Lognormal	-2,0.5
O	D2	Control box battery maintenance.	Lognormal	-2,0.5
O	I1	Inspection interval of the control box.	Global (3 intervals)	12,6,3
O	p1	Probability that the control box failure is unrevealed.		0.5,0.5
O	E1	Random power failure.	Uniform	0.0041
O	D4	End of power failure.	Normal	0.001,0.0001
Q	T1	Wiring fails due to age	Weibull	480,2
Q	T4	Random useful life wiring failure	Uniform	0.000119
Q	R1	Early maintenance scheduling of wiring.	Lognormal	5.52,0.005
Q	R2	Routine maintenance scheduling of wiring.	Lognormal	6.05,0.005
Q	R3	Maintenance scheduling of wiring on failure.	Lognormal	-5,0.5
Q	I1	Inspection interval of wiring.	Global (3 intervals)	12,6,3
Q	D1	Maintenance of wiring.	Lognormal	-2,0.5
X	St1	Full system testing interval	Normal (3 distributions)	9,1,6,1,3,1
X	Sd1	Full system testing time	Lognormal	-10,0.8
X	Ph1	System enters Phase 1 from Phase 0	Normal	36,1
X	Ph2	System enters Phase 2 from Phase 1	Normal	120,1

Table A3.1: Sample data values for the Petri net model application given in Chapter 5

Appendix 4

A summary of the components modelled in Chapter 5.

Components	Failure modes	Inspection and testing modelled	Maintenance modelled
Control box	Unrevealed and revealed	Periodic inspection and testing, and testing when system level tests are underway.	Early age-based maintenance, routine age-based maintenance, maintenance on discovered failure.
Control box battery	Unrevealed failure	Periodic inspection and testing.	Early age-based maintenance, routine age-based maintenance, maintenance on discovered failure.
Pressure Sensors	Each sensor can either fail to give a reading, give a reading that is higher than true or give a reading that is lower than true. Failures are unrevealed, unless they cause a false activation of the deluge system. The system is assumed to be in a dangerous state if 2 or more sensors give a reading that is higher than true.	Periodic inspection of sensor readings, system level testing.	Early age-based maintenance of all sensors, routine age-based maintenance of all sensors, maintenance of all sensors on discovered failure.
Alarm	Single and multiple unrevealed alarm sounder failure, unrevealed wiring failure.	Periodic inspection and testing.	Early age-based maintenance, routine age-based maintenance, maintenance on discovered failure.
Pipework, Ringmain, Diesel Tank (Type A models)	Unrevealed failure	Periodic inspection, system level testing.	Early age-based maintenance, routine age-based maintenance, maintenance on discovered failure.
Diesel pump, Electric pump, Jockey pump (Type B models)	Unrevealed failed state (in addition an unrevealed degraded state)	Periodic inspection and testing, system level testing.	Early age-based maintenance, routine age-based maintenance, maintenance on discovered degraded state maintenance on discovered failure.
Sprinkler head, wiring (Type C models)	Unrevealed failure	Periodic inspection and testing.	Early age-based maintenance, routine age-based maintenance, maintenance on discovered failure.
Isolation valve, pressure release valve (Type D models)	Unrevealed open failure and unrevealed closed failure	Periodic inspection and testing, system level testing.	Early age-based maintenance, routine age-based maintenance, maintenance on discovered failure.
Deluge valve, solenoid, manual start device (Type E models)	Revealed and unrevealed failure	Periodic inspection and testing	Early age-based maintenance, routine age-based maintenance, maintenance on discovered failure.
Heat detectors, smoke detectors, manual call points (Type F components)	Unrevealed non-hazardous failure of member in population, unrevealed hazardous failure, revealed failure.	Periodic inspection and testing.	Early age-based maintenance, routine age-based maintenance, maintenance on discovered failure.

Table A4.1: A table summarising the components modelled in Chapter 5

In addition, mains power failure and mains water failure are considered.

Appendix 5

For components modelled in Chapter 5, sample results for each component model are presented here. In these sample results a Weibull distribution governs the time that it takes for each component to fail. The mean of the Weibull distributions used in each of the models, has been used to discuss the results in relation to the input data. This is calculated as in Equation A1 for a 2-Parameter Weibull distribution, with shape parameter β and scale parameter η.

$$MTTF = \eta\Gamma(1 + 1/\beta) \qquad (A1)$$

Where $\Gamma(x)$ is the Gamma Function.

Where a log-normal distribution has been used to specify intervals between maintenance actions the arithmetic mean of the distribution can be calculated as in Equation A2, where μ and σ are the location and shape parameters respectivley.

$$E[X] = e^{\mu + \frac{1}{2}\sigma^2} \qquad (A2)$$

Control Box Model

To demonstrate the control box model in Chapter 5, sample model inputs were assigned. These are given in Appendix 3. The system level maintenance strategy defined at the beginning of the Chapter 5, was also applied. Figure A5.1 gives the probability that the control box is in a failed state with each year. Figure A5.2 gives the probability that the control box battery is in a failed state each year. Figure A5.4 gives the probability that there is a complete control box power failure and Figure A5.3 gives the probability that there is a mains power failure over time. The number of maintenance actions at each time is given in Section 5.9.

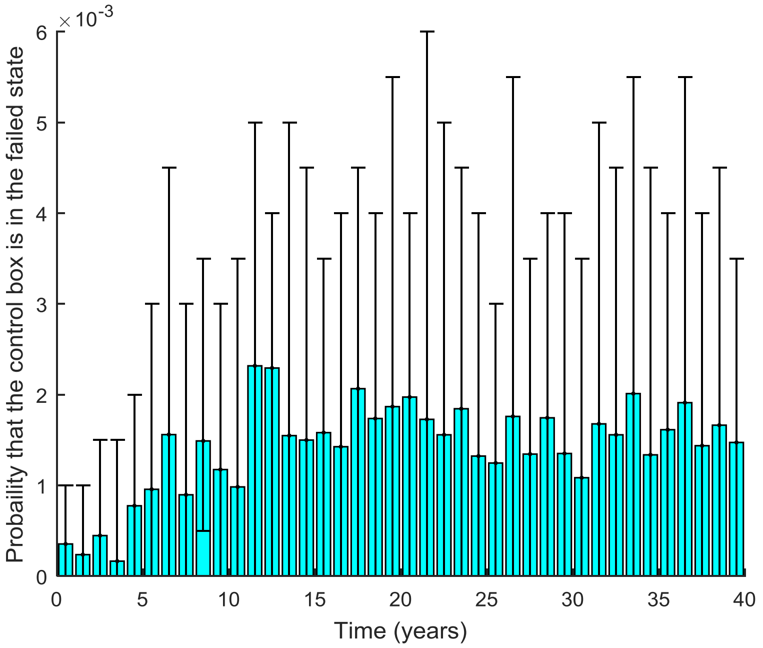


Figure A5.1: The probability that the control box is in the failed state over time

Figure A5.1 gives the probability of control box failure for each year, these results include both revealed and unrevealed failures. The bars show the average probability of failure over the year with the range bars showing the maximum and minimum average simulated value within each year. From the Weibull input data it is expected that a failure due to age will occur with a mean time of approximately 11 years. The routine age-based maintenance scheduling interval is with a mean value of approximately 6 years and early age-based maintenance scheduling interval is set with a mean value of approximately 4 years. With the maintenance strategy applied in this simulation, these results show a low level of control box failure suggesting that the age-based maintenance included in the model prevents an increase in failures as the component ages.

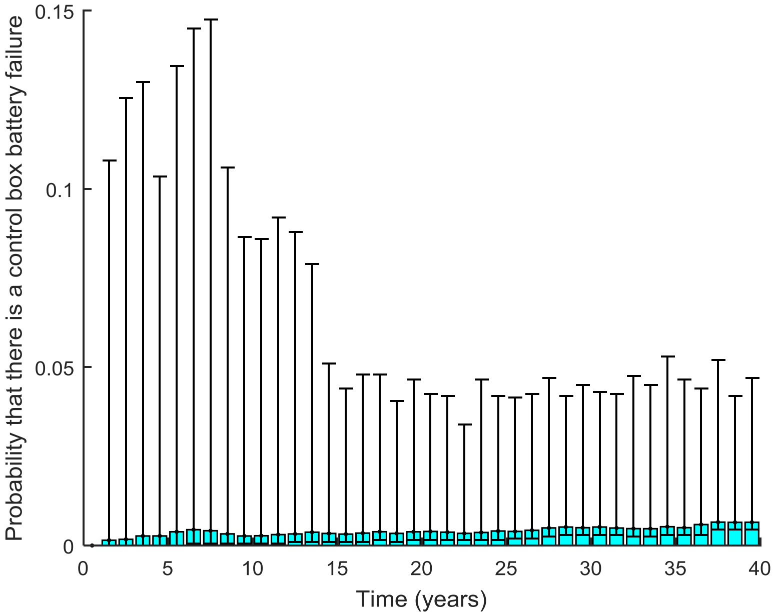


Figure A5.2: The probability that the control box battery is in the failed state over time

Figure A5.2 gives the probability that the control box battery is in the failed state at each year, with the bars showing the average value across the year and the range bars showing the maximum and minimum value within the year. From the input data it is expected that failures will occur with a mean time of approximately 5 years. A mean value of approximately 3 years was assigned to the routine maintenance scheduling interval and a mean value of approximately 2 years was assigned to the early maintenance scheduling interval. These results show a higher level of control box battery failure in comparison to that of the control box. This can be attributed to the shorter time to failure assigned to the battery ageing transition in this demonstration of the model and a higher assigned rate of random failures.

Figure A5.3 gives the probability that there is a mains power failure over time, this is approximately constant which is expected from the input data for this case. Figure A5.4 shows that the combination of a battery failure and a mains power failure occurs very rarely in this application of the model.

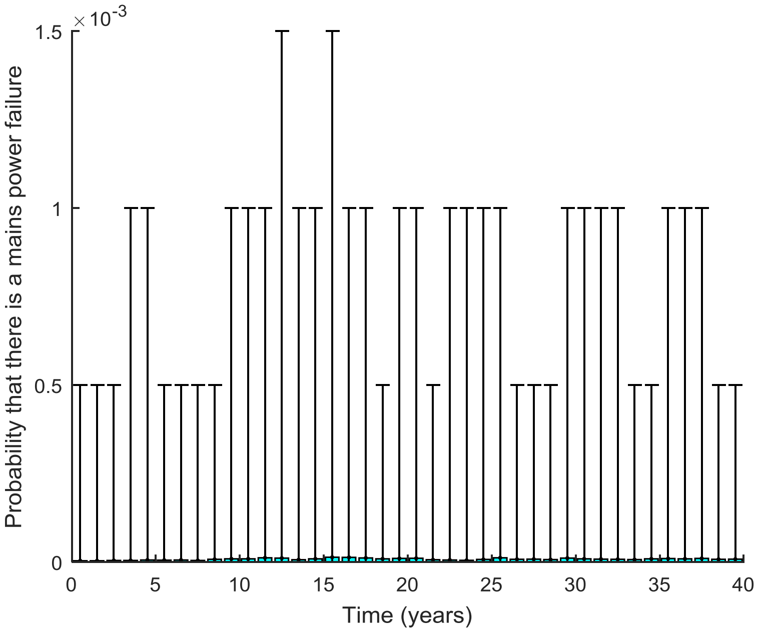


Figure A5.3: The probability that there is a mains power failure over time

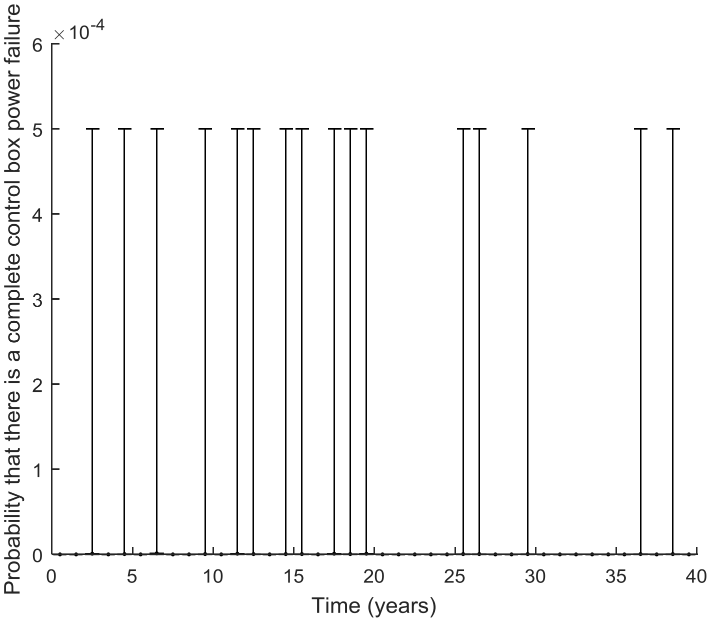


Figure A5.4: The probability that there is a complete control box power failure over time

Pressure Sensors Model

To demonstrate the model for the pressure sensors, given in Chapter 5, sample values were used as input to the model. The values used in this demonstration can be found in Appendix 3. Figure A5.5 gives the probability that there is a combined pressure sensor failure that can result in the system not functioning when it is required, each bar gives the average probability of failure for the year with the upper and lower values within the year represented by the range bars. From the input data it is expected that the pressure sensors will fail due to age with a mean time of approximately 9 years. The routine maintenance scheduling interval was set with a mean value of approximately 8 years and the early replacement scheduling interval was set with a mean value of approximately 6 years. Despite the faster ageing assigned to this component in comparison to other components in the model, there is still a low probability of failure. This can be attributed to the redundancy in the pressure sensors and the commonly revealed nature of the failures.

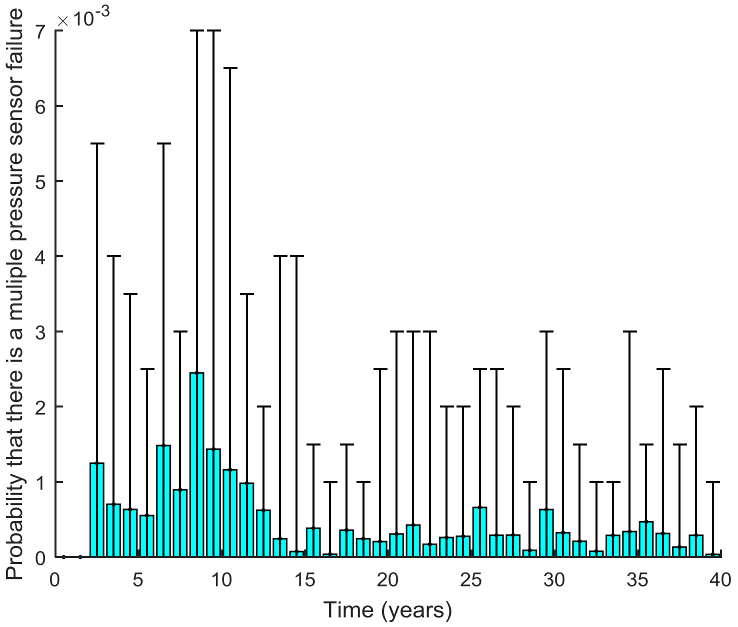


Figure A5.5: The results for the probability of a combined pressure sensor failure

Alarm Sounder Model

To demonstrate the model given in Chapter 5 for the alarm sounder failure, sample input values were used. These can be found in Appendix 3. The results of this Petri net model for this input data can be found in Figure A5.6, which gives the probability that there is a total failure in the alarm sounder circuit. The bars give the average probability of failure over each year, with range bars showing the maximum and minimum simulated value within each year. From the input results it is expected that there will be failures due to the age of the wiring with a mean value of approximately 35 years and failures due to the age of the sounders with a mean value of approximately 22 years. The routine maintenance scheduling interval is set with a mean value of approximately 10 years for the sounders and the early maintenance scheduling interval is set with a mean value of approximately 7 years. The routine maintenance scheduling interval is set with a mean value of approximately 35 years for the wiring and the early maintenance scheduling interval is set with a mean value of approximately 20 years. From the long ageing times assigned in this application of the model, a low probability of failure is expected.

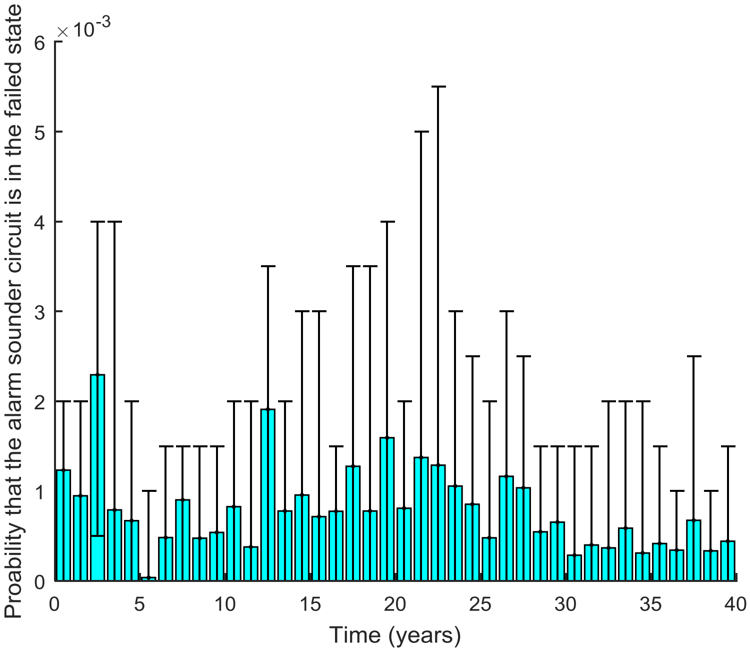


Figure A5.6: The probability that the alarm sounder circuit is in the failed state at each time

Type A Component Model

To demonstrate the model given in Chapter 5 for Type A components, the sample data, given in Appendix 3, was used as input to the pipework model and results were obtained via simulation of the model. The results for the probability of pipework failure in this case are given in Figure A5.7. The time since system installation in years is shown along the x-axis, with the average probability of failure in that year shown on the y-axis. The range bars show the maximum and minimum average probability found by the simulation within that year. The bars show the average probability across that year.

When this model is applied to the pipework or pressurised ringmain, all the pipework, or pressurised ringmain, in the system is modelled as a unit with this Petri net. If maintenance occurs then it is assumed that all the pipework, or all of the ringmain, in the deluge system is returned to a good state.

The results for the module when applied to the sample pipework data, given in Appendix 3, are given in Figure A5.7. In these results the average probability that the pipework is in the failed state across the year is represented by the bars and the maximum and minimum average value within the year is represented by the range bars. Age related failures of the pipework include those such as scale build-up, corrosion and crack development [191] [192]. Random failure of the pipework includes those due to accidental damage to the pipework. From the input data used to demonstrate this model it is expected that failures due to the age of the pipework will occur with a mean time of approximately 45 years. Routine age-based maintenance is scheduled with a mean value of approximately 33 years and early age-based maintenance is scheduled with a mean value of approximately 20 years. A low probability of failure is expected from these input values, due to the slow aging rate assigned to the model.

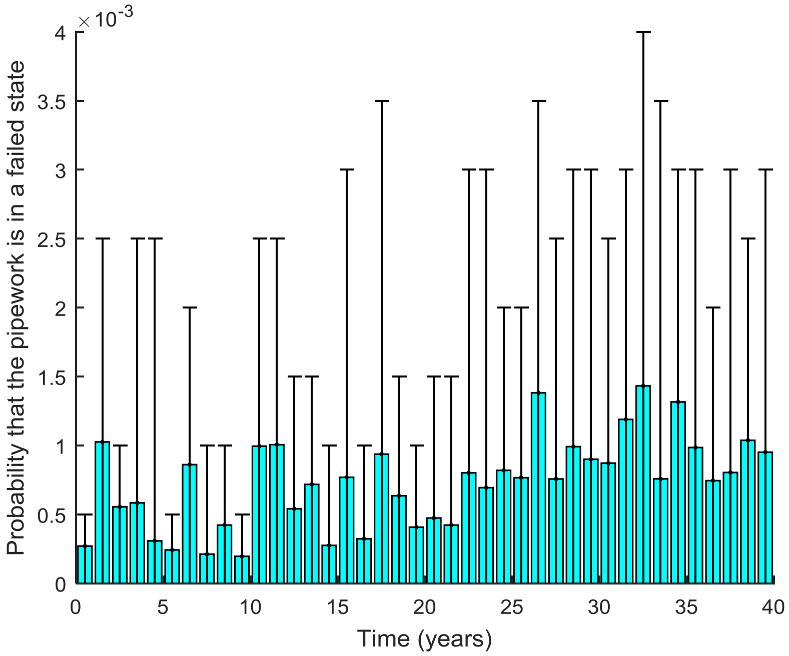


Figure A5.7: The probability that the pipework is in a failed state at each time

The results for the module when applied to the pressurised ringmain data, given in Appendix 3, are given in Figure A5.8. Similarly, to the pipework, aging failures for this component include those such as corrosion, scale build up and crack development. Random failure includes those due to accidental damage or overpressure of the ringmain. From the sample data used to demonstrate this model, failures due to the age of the ringmain are expected with a mean value of approximately 37 years. Routine age-based maintenance is scheduled with a mean value of approximately 24 years and early age-based maintenance is scheduled with a mean value of approximately 14 years. There is also a higher rate of random failures assigned to the ringmain model than the pipework model. From these input values it is expected that there will be a higher probability of failure of the ringmain than the pipework. In the results for the ringmain it is notable that there is a reduction in the probability of failure after the 6-year and 12-year point, these times correspond to the changes in system level maintenance phases. At this point the inspection frequency and system level testing increases and preventative maintenance is enabled, as expected this causes a reduction in the probability that the ringmain is in the failed state.

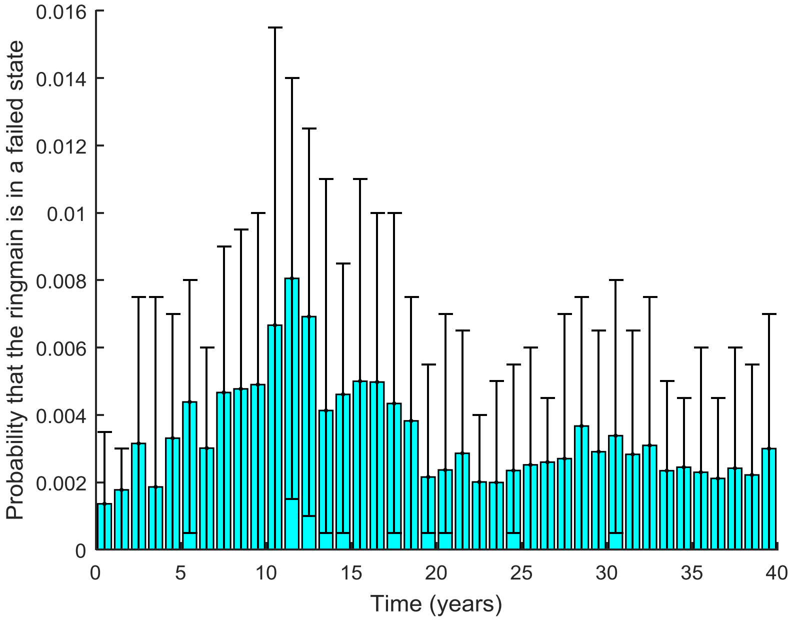


Figure A5.8: The probability that the ringmain is in a failed state at each time

The results for the module when applied to the sample diesel tank data, given in Appendix 3, are given in Figure A5.9. The diesel tank is used to supply diesel to the diesel pump. Here, the bars show the average probability of failure for each year and the range bars show the maximum and minimum average value within each year. A failure of the diesel tank corresponds to any state where the tank failure results in an insufficient supply of diesel from the tank to the diesel water pump. It is assumed in this model that if the diesel tank is functioning correctly then there will be enough diesel to enable the correct functioning of the deluge system, namely, there are no system-level failures due to design flaws such as a diesel tank size that is too small. Random failures include those due to accidental damage leading to a tank leak or insufficient supply of diesel to the tank. Ageing failure include leakage of the tank due to corrosion or cracking.

From the sample input data used in this model, ageing failures of the diesel tank are expected with a mean value of approximately 19 years. The routine age-based maintenance is scheduled with a mean interval of approximately 10 years and the early age-based maintenance is scheduled with a mean interval of approximately 6 years. It is expected from the sample model inputs that there will be a low probability of failure due to the slow ageing time assigned to this part of the model. In these results a reduction can be seen at approximately 13 years, corresponding to entry into the third system maintenance phase where there is an increase in inspection and system testing.

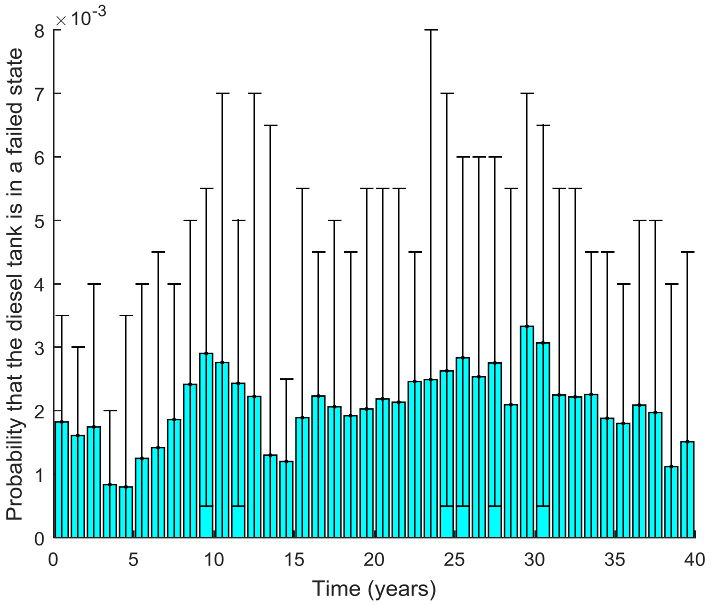


Figure A5.9: The probability that the diesel tank is in a failed state at each time

Type B Component Model

To demonstrate the Type B component model, given in Chapter 5, sample data, given in Appendix 3, has been used to demonstrate the possible results that can be gained from the model. Different data is used for each of the diesel pump, electric pump and jockey pump. The results for the probability of total pump failure are given in Figure A5.10, Figure A5.11 and Figure A5.12 for the electric pump, jockey pump and diesel pump respectively. In the application of this model in Chapter 5, it is assumed that only total pump failures can contribute to a system-level failure. For the models of pump failures, replacement is enabled upon the discovery of a partial failure. Because of this, it is expected that there will be a lower number of pump failures, resulting in a lower probability of failure than if partial failures were not included. Also, a delay in the time that pump failures begin to occur is expected due to this preventative maintenance. It can also be expected that there will be less impact seen across the three maintenance phases on the probability of pump failure due to this constant repair of the pumps before they reach the fully failed state.

From the sample data used as input to the model, it is expected that ageing failures of the electric pump will occur with a mean value of approximately 9 years. Also, routine age-based maintenance is scheduled with a mean interval of approximately 5 years and early age-based maintenance is scheduled with a mean interval of approximately 3 years. Since there is maintenance on partial failure, it is expected that there will be a low level of failure, especially at the earlier stages of the component’s life.

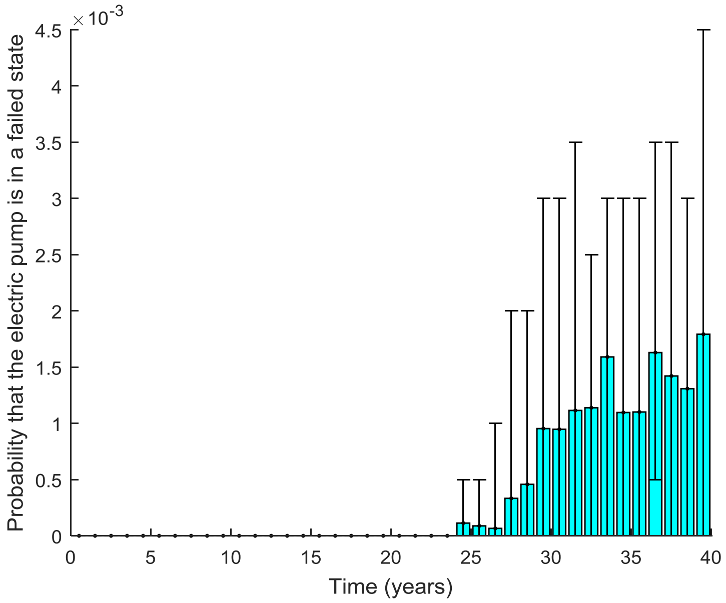


Figure A5.10: The probability that the electric pump is in a failed state at each time

From the sample data used in this model it is expected that the jockey pump will fail due to age with a mean value of approximately 5 years. Routine age-based maintenance is scheduled with a mean interval of approximately 3 years and early age-based maintenance is scheduled with a mean interval of approximately 1 year. There is also a higher rate of random failures assigned to the jockey pump in comparison to the electric pump. From this, it is expected that there will be a higher probability of failure, and that failure will begin to occur at a shorter time.

From the sample data it is expected that the diesel pump will fail due to age with a mean value of approximately 12 years. Routine age-based maintenance is scheduled with a mean interval of approximately 8 years and early age-based maintenance is scheduled with a mean interval of approximately 5 years. There is also a lower rate of random failures assigned to the diesel pump in comparison to the jockey pump, but a similar rate assigned in comparison to the electric pump. From this it is expected that the diesel pump will have the lowest probability of failure in the time period for these results, but that the probability of failure will follow a similar trend to that of the electric pump in this application of the model.

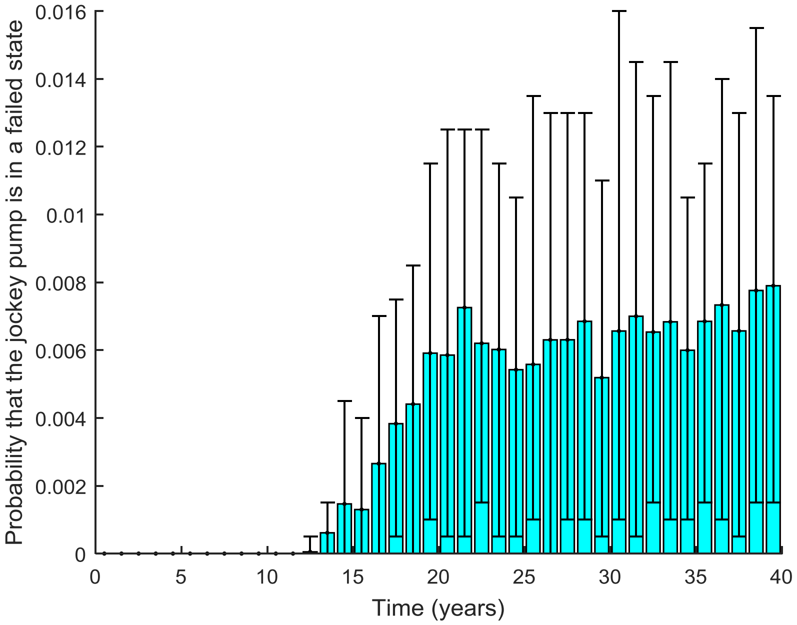


Figure A5.11: The probability that the jockey pump is in a failed state at each time

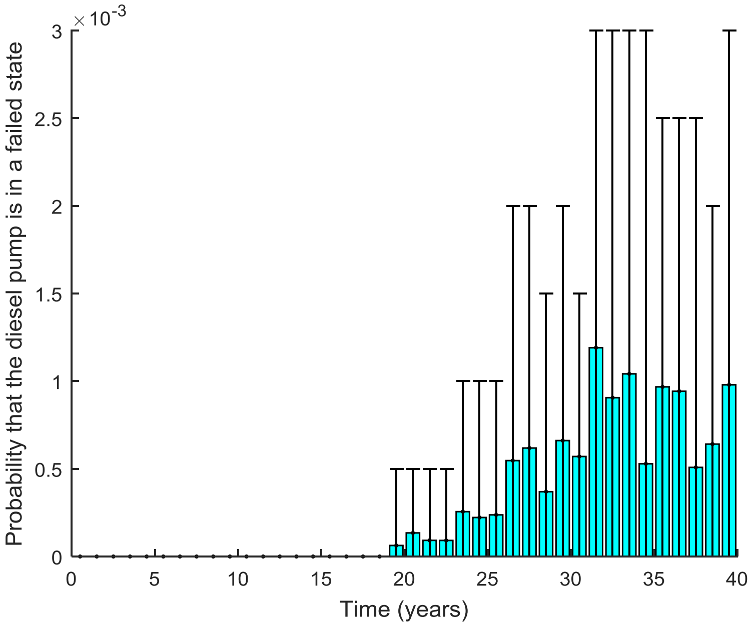


Figure A5.12: The probability that the diesel pump is in a failed state at each time

Type C Component Model

To demonstrate the application of the Type C component model, given in Chapter 5, sample data given in Appendix 3 was used and the model was simulated via Monte Carlo simulation.

Figure A5.13 gives the results of this model applied to the sprinkler head. Here, the bars give the average probability of failure over each year and the range bars give the maximum and minimum average values from the simulation within each year. Random failure of the sprinkler head includes those such as blockages of the strainer or sprinkler head, or accidental damage to the sprinkler head. Sprinkler head failures due to age include failures as a result of corrosion, rusting and mineral build up [192] [191]. Failures due to the ageing of the sprinkler head and strainer are expected with a mean value of approximately 5 years, from the sample input data used in this chapter. Routine age-based maintenance is scheduled with an mean value of approximately 4 years and early age-based maintenance is scheduled with a mean value of approximately 3 years. There is also a relatively high level of random failures assigned to the model. Due to this shorter ageing rate, the high level of failures and the unrevealed nature of a failure it is expected that there will be a higher probability of failure of this component in comparison to other components in the model. Notable in these results is the reduction in the probability of failure at approximately 13 years. This corresponds to the entry of the third maintenance phase, where there is an increase in inspection frequency of the sprinkler head and strainer and in the age-based maintenance of the component.

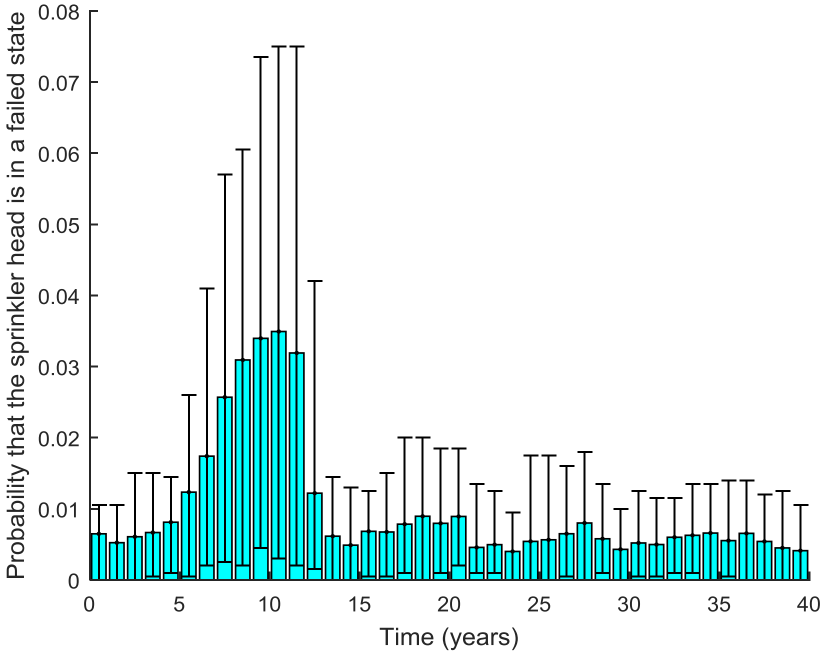


Figure A5.13: The probability that the sprinkler head is in a failed state at each time

Figure A5.14 gives the results of this model when applied to the wiring data given in Appendix 3. Failures of the wiring due to ageing include those such as failures due to corrosion, water ingress or deterioration of the casing. Random failures include those due to accidental damage. From the input data it is expected that failures due to the age of the wiring will occur with a mean value of approximately 35 years, however failures are unrevealed and hence can be present in this model until an inspection is carried out. Routine age-based maintenance is scheduled with a mean value of approximately 35 years and early age-based maintenance is scheduled with a mean value of approximately 21 years. In these results there is an increase in the probability of failure towards 35 years followed by a decrease after 35 years. The increase corresponds to the increase in failures due to the age of the wiring, the decrease corresponds to the start of the preventative age-based maintenance.

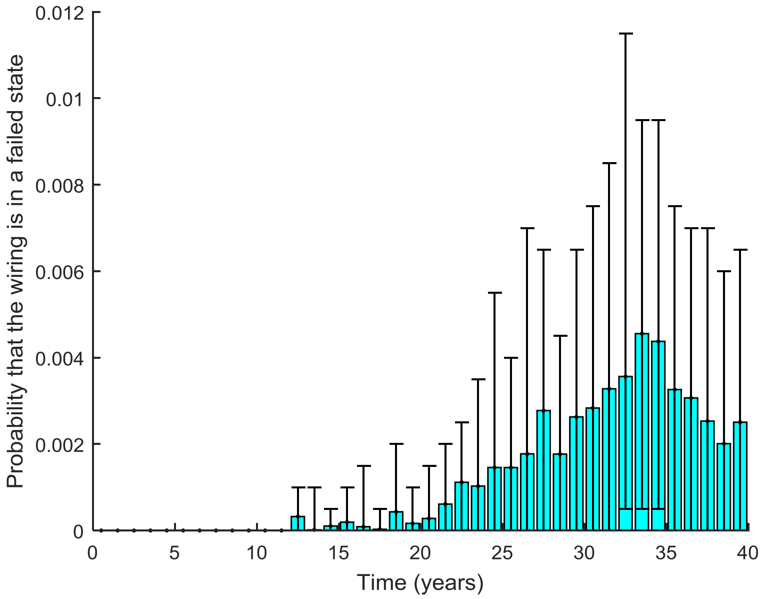


Figure A5.14: The probability that the wiring is in a failed state at each time

Type D Component Model

To demonstrate the model given for Type D components in Chapter 5, sample data was used. This data can be found in Appendix 3.

Figure A5.15 gives the results for the sample isolation valve data, given in Appendix 3. In these results the bars give the average probability of failure over each year and the range bars give the maximum and minimum probability of failure taken from the simulation within each year. This model can be repeated for each isolation valve in the system if required. For the isolation valve, the random failures include those due to human error causing a valve to reside in an incorrect state, for example where the isolation valve has been closed at a previous time and not reopened, and those where the isolation valve has been damaged. Failures due to the age of the components include mechanical failures where the valve is not tight or has failed completely, such as a broken valve stem or rounded operating nut [193] [194]. There are two failure options for the isolation valve. The first is that the isolation valve has failed in the open position and will not contribute to a system-level failure. The second is that the isolation valve has failed in the closed position which can contribute to a system-level failure. In this model, a threshold can be given to classify whether a valve failure is an ‘open’ or ‘closed’ failure. An open failure is defined as a case whereby the failure is insufficient to contribute to the failure of the whole deluge system as it does not inhibit the fluid flow enough. A closed failure is a failure whereby the failure inhibits sufficient fluid flow to contribute to a system failure. Inspection of the valve looks at: the valve operation, reduced flow through the valve when it is in the supposedly open condition, and flow through the valve when it is in the closed position.

From the input parameters used in this sample application of the model it is expected that an isolation valve will fail due to age with a mean time of approximately 6 years. Routine age-based maintenance is scheduled with an interval of approximately 2 years and early age-based maintenance is scheduled with an interval of approximately 1 year. The results show a general increase in the probability of failure towards the 12 year mark, however there are notable decreases present in the 3rd and 13th year, these correspond to the entry times of system maintenance phases, and an increase in inspection, testing and age-based maintenance scheduling.

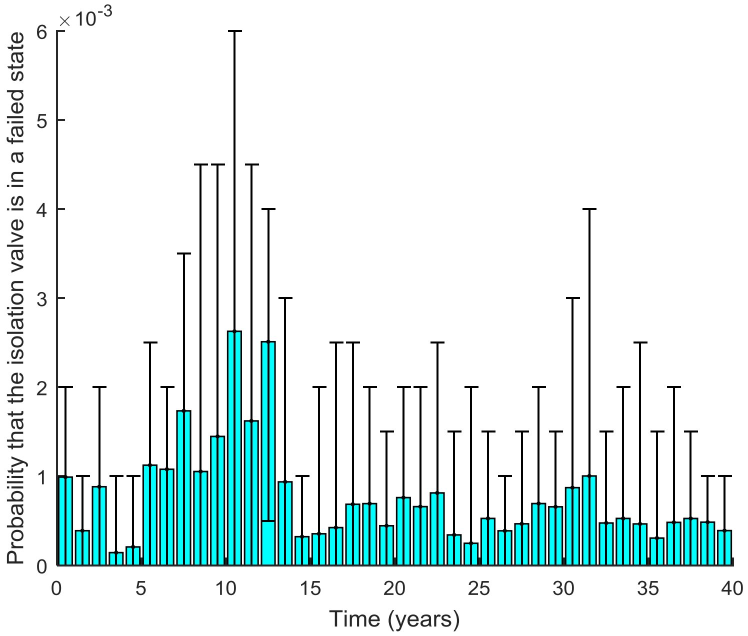


Figure A5.15: The probability that the isolation valve is in a failed state at each time

Figure A5.16 gives the result of this model applied to the sample pressure release valve data, given in Appendix 3. The pressure release valve is used to prevent overpressure in the pipework and ringmain. The valve is held closed until the pressure in the ringmain exceeds a certain threshold, whereupon it opens to reduce the pressure. In this model, a false opening of the pressure release valve can result in a system failure by reducing the pressure of the ringmain below the required level. A failure in the ringmain due to overpressure is included in the ringmain component model. Commonly, pressure release valves have a sprung mechanism that allows the valve to open in overpressure situations before returning to the closed position when the pressure drops again. Random failures of the pressure release valve can be due to a jamming of the valve flap or an incorrect recalibration following intervention. Ageing failure can also occur due to the age of the pressure release valve such as a build-up of sediment between the valve flap and the sealing surface, loss of elasticity or rusting of the spring and bending of the valve stem. Inspections check that the valve does not flutter or clatter and that it returns to its original position after overpressure causes it to open. This can be done by a verification device that simulates overpressure without interfering with normal operation [195].

From the distribution governing ageing of the pressure release valve, used in this application of the model, it is expected that failures due to the age of the component will occur with a mean value of approximately 4 years. Routine age-based maintenance is scheduled following an interval with a mean value of approximately 2 years and early age-based maintenance is scheduled following an interval with a mean value of approximately 1.5 years. In comparison to the isolation valve, from the input data, there is also a lower probability that the pressure release valve will fail safe. The results show a higher probability of a hazardous failure for the pressure release valve in comparison to the isolation valve, this can be attributed to the lower probability that the valve will fail safe and a faster ageing rate. These results show a decrease after the 13-year point that corresponds to the increased inspection, testing and preventative maintenance that is associated with entry into the third system level maintenance phase.

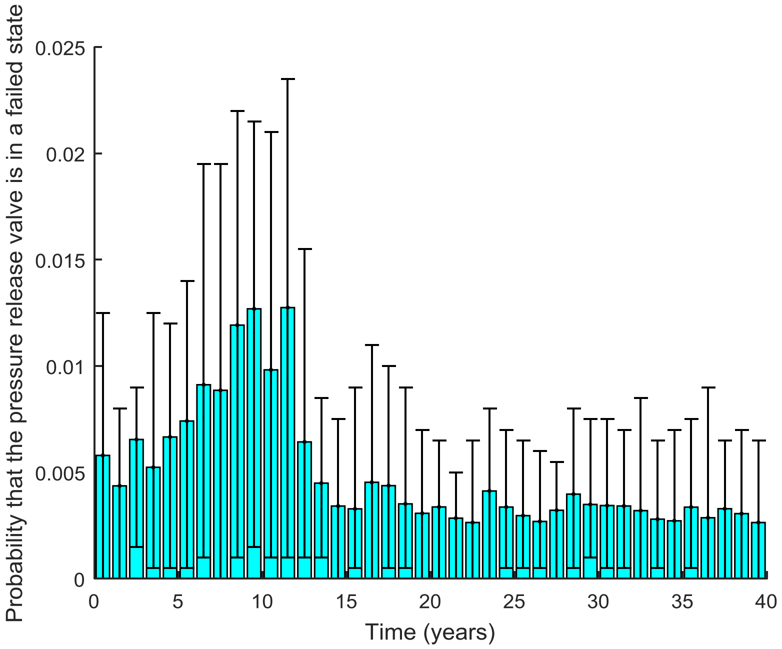


Figure A5.16: The probability that the pressure release valve is in a failed state at each time

Type E Component Model

To demonstrate the model, for Type E components, given in Chapter 5, data given in Appendix 3 was used as input for the model. Figure A5.17 gives the results for the application of this model to the sample data for the deluge valve, found in Appendix 3. The deluge valve separates the pipework containing the pressurized water in the system from the dry pipework leading to the sprinkler heads. The diaphragm of the deluge valve is held in place by a pressure balance between the water closing circuit and the water pressure in the ringmain system. This prevents water from entering the dry pipework. If water leaves the water closing circuit, a pressure difference is created across the diaphragm of the deluge valve which causes the valve to enter the open position, thus allowing water to flow through the system [83]. A false opening of the deluge valve triggers the system to respond as if there is a fire. This can cause costly damage to infrastructure and closure of the station. If the deluge valve fails to open on demand, water cannot flow through the deluge system to the sprinkler heads.

Random failures of the deluge valve include those where the valve becomes stuck and does not return to the closed position following opening, or there is a blockage in the valve. Ageing failures of the deluge valve include those such as damage to the diaphragm or a build-up of sediment within the valve. On failure of the deluge valve, there is a probability associated with it residing in the open or closed position.

From the input data it is expected that failures due to the age of the deluge valve will occur at a mean time of approximately 8 years. Routine age-based maintenance is scheduled at an interval with a mean value of approximately 5 years. Early age-based maintenance is scheduled with an interval with a mean value of approximately 3 years. The results show an increase in the probability of failure as the component age increases, however there is a decrease after the 13-year point. This can be attributed to the increase in inspection, testing and preventative maintenance when the system enters the third system maintenance phase.

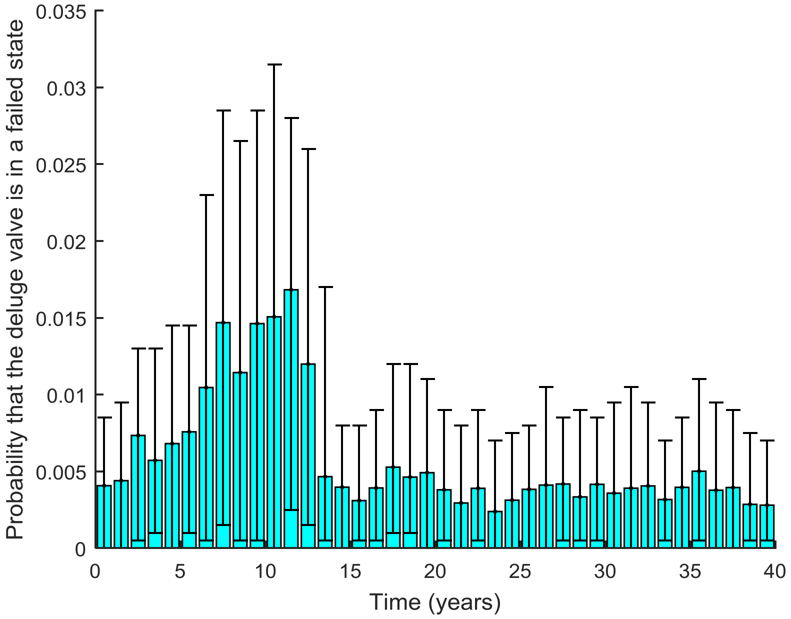


Figure A5.17: The probability that the deluge valve is in a failed state at each time

Figure A5.18 gives the results for the application of this model to the sample data for the solenoid and closing circuit, found in Appendix 3. The deluge system can be automatically initiated. In this case the control box gives a signal to the solenoid valve causing it to de-energize and open. This releases air from a control air circuit reducing the pressure in this circuit. This in turn causes a valmatic release valve to open, draining water from the water closing circuit [196].

Random failures of the solenoid and closing circuits can include: damage to the closing circuits leading to a leak, a false recalibration of the pressure in the circuits following activation, and the solenoid valve not returning to the fully closed position after testing. Failures due to the age of the solenoid and closing circuits can include: a build-up of dirt in the solenoid valve such that it cannot fully close, or the development of cracks in the closing circuit. There are two failure modes for the solenoid and closing circuit in this model. The first is that a failure causes water to leave the water closing circuit, which opens the deluge valve and immediately reveals the failure by falsely activating the deluge system. The second failure mode is that the solenoid does not de-energize, or the valmatic release valve remains closed, on receiving a signal from the control box, which prevents water from flowing through the deluge system when it is needed.

For the sample data used in this application of the model, failures due to the age of the solenoid and water closing circuit are expected with a mean time of approximately 4 years. Routine age-based maintenance is scheduled with a mean interval of approximately 2 years. Early age-based maintenance is scheduled with a mean interval of approximately 1 year. The results show an increase in the probability of component failure up to the 9-year point, followed by a decrease. This decrease corresponds to the routine age-based preventative maintenance, scheduled during the second maintenance phase, followed by the increase in inspection and system testing at the entry of the third system maintenance phase.

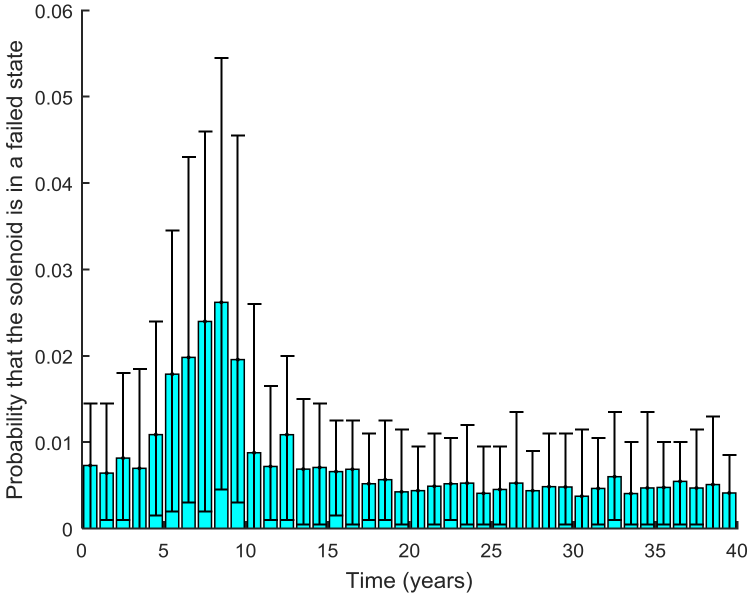


Figure A5.18: The probability that the solenoid is in a failed state at each time

Figure A5.19 gives the results for the application of this model to the sample data for the manual start device, found in Appendix 3. The manual start device is an emergency release valve that can be operated manually to allow water to flow from the water closing circuit to open the deluge valve. Random failures of the emergency release valve include: an accidental operation of the manual start device or a misalignment of the valve following testing. Failure due to ageing of the manual release mechanism include the build-up of debris under the valve or mechanical damage. There are two failure modes for the manual release mechanism. The first occurs when false activation of the system arises due to the surplus opening of the release valve. The second failure mode includes scenarios where the manual release mechanism does not work when required.

From the input data it is expected that there will be failures due to the age of the manual start device with a mean time of approximately 5 years. Routine age-based maintenance is scheduled with a mean time of approximately 3 years and early age-based maintenance is scheduled with a mean time of approximately 2 years. In the results a decrease in the probability of failure can be seen at 13 years, corresponding to an increase in the inspection and testing frequency at this point. In addition, a further decrease can be seen at the 16-year point, corresponding to the preventative maintenance actions scheduled in the second system maintenance phase.

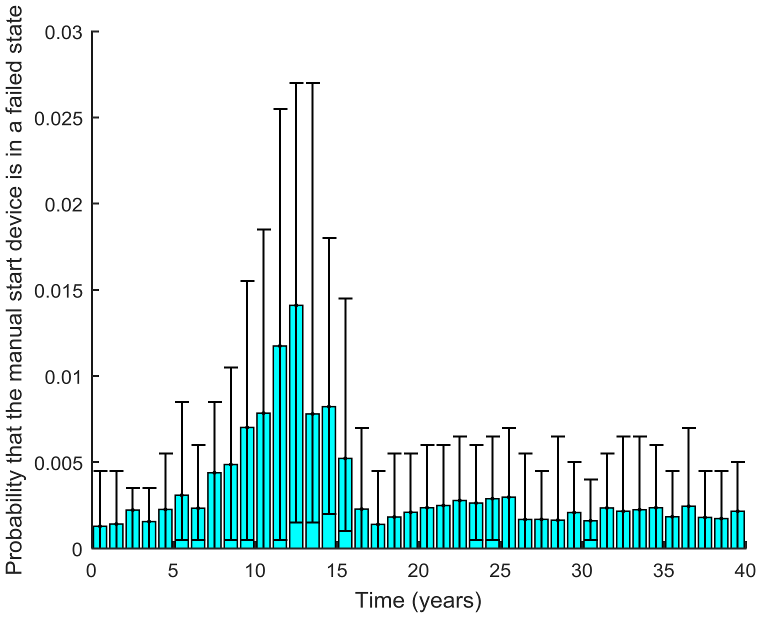


Figure A5.19: The probability that the manual start device is in a failed state at each time

Type F Component Model

To demonstrate the model, for Type F components, given in Chapter 5, data given in Appendix 3 was used as input. The results for each type of population of components is modelled by a repeated unit of this model. In this application of the model, there are two zones in the station, each with a different detection circuit. The first is a public zone where heat detectors are present. The second is a non-public zone where smoke detectors are present. It is assumed that the detectors are spaced such that if there is a failure of one detection unit then the fire can be detected by a second unit in a nearby location. Random failures in the detectors can occur at any point due to accidental damage and age-related failures, include those due to increased sensitivity and dust or dirt accumulation [192]. Figure A5.20 gives the results for this model with the smoke detector sample data, as given in Appendix 3. Figure A5.21 gives the results for this model with the heat detector sample data, as given in Appendix 3.

From the input data used in this model it is expected that the smoke detector will fail due to age with a mean time of approximately 10 years. Routine age-based maintenance is scheduled with a mean time of approximately 8 years. Early age-based maintenance is scheduled with a mean time of approximately 5 years. From the input data used in this model it is expected that the heat detector will fail due to age with a mean time of approximately 13 years. Routine age-based maintenance is scheduled with a mean time of approximately 9 years. Early age-based maintenance is scheduled with a mean time of approximately 6 years. The results for the heat and smoke detectors both show a decrease in the probability of failure at the 3-year point corresponding to entry to the second maintenance phase, and the corresponding increase in inspection frequency. Preventative maintenance, scheduled in the second system level maintenance phase can occur at approximately 8 years for the smoke detector, and at approximately 9 years for the heat detector. Further second phase and third phase preventative maintenance is scheduled from this point onwards. The results show a levelling of the probability of failure at these points followed by a decrease towards a lower more consistent probability of failure, despite the ageing of the system. This can be attributed to the preventative maintenance actions on the components.

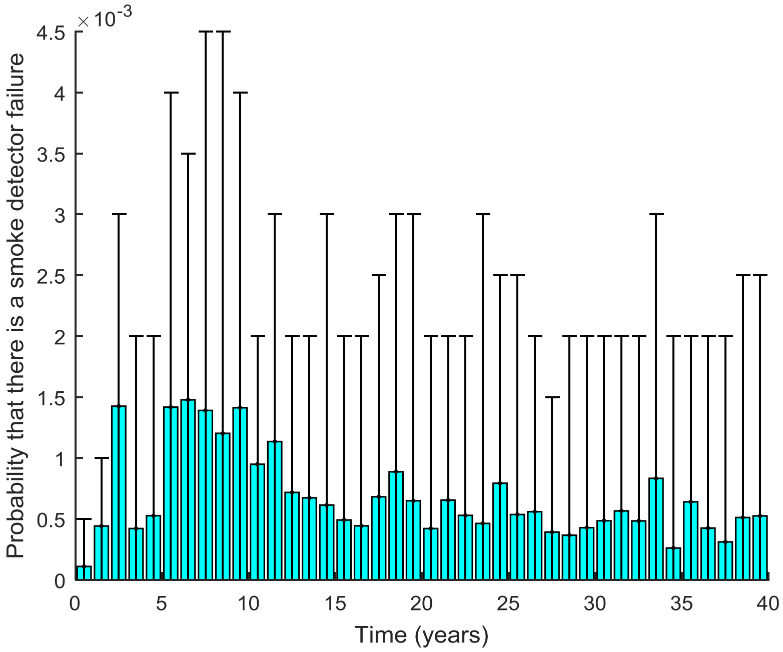


Figure A5.20: Probability that there is a smoke detector failure at each time

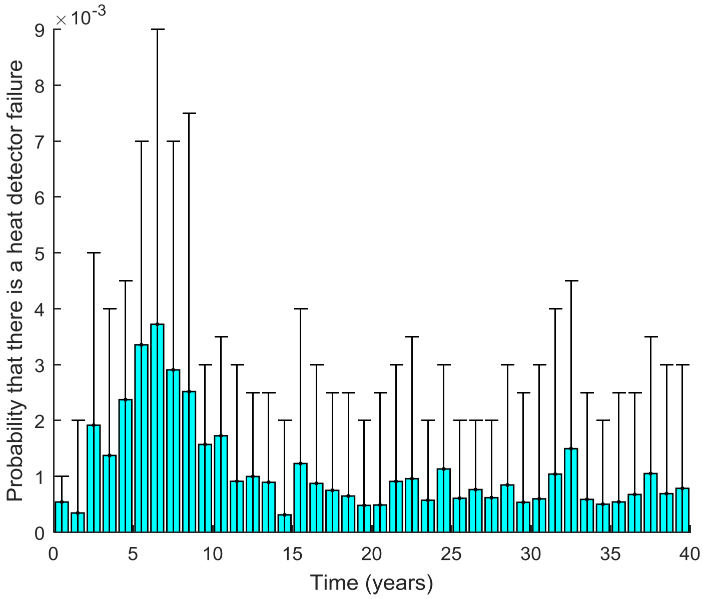


Figure A5.21: The probability that there is a heat detector failure at each time

Figure A5.22 gives the results for this model when applied to the sample call point data, given in Appendix 3. In this application of the model there is one set of call points in the first zone and one set of call points in the second zone. This Petri net models the group of call points in one zone and is repeated for each zone. It is assumed here that there is no difference between the probabilities of failure in each zone and so the results of the simulation are simply duplicated when combined via a Fault Tree structure. However, if data is available then this Petri net can be simulated with different data for each zone.

Random failures of the call point can include those caused by accidental damage or vandalism. Failures can also occur due to the age of the call points, such as those due to water ingress or dust and dirt accumulation [197]. From the input data it is expected that there will be failures due to the age of the call points with a mean time of approximately 7 years. Routine age-based maintenance of the call points is scheduled following an interval with a mean time of approximately 3 years and early age-based maintenance of the call points is scheduled with a mean time of approximately 2 years. There is also a higher rate of random failures assigned to the call points, when compared to the rate assigned to the smoke and heat detectors. The results show a decrease at the 3-year point, following the entry to the second system level maintenance phase and the associated increase in inspection frequency. The results also show a decrease following the 6-year point, this corresponds to the initialisation of the age-based preventative maintenance scheduled in the second system level maintenance phase. The higher and more consistent probability of failure of the call points can be attributed to the higher random failure rate of the component.

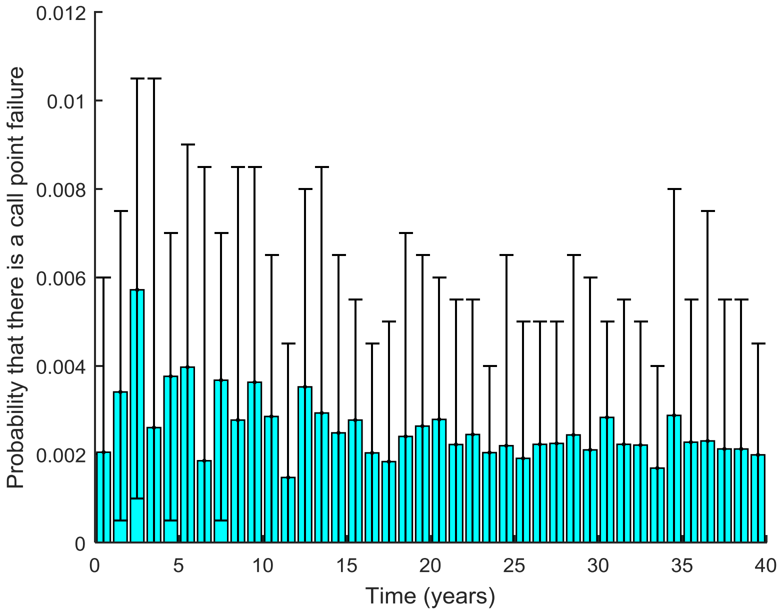


Figure A5.22: The probability that there is a call point failure at each time

Appendix 6

This appendix gives the distributions and parameters for each of the models in the fourth example in Chapter 7. Here, normal distribution parameters are given in order of $N(\mu,\sigma)$, Weibull distribution parameters are given in order of $Wei(\eta,\beta)$ Where a question mark is given for a parameter, this denotes a parameter that is updated within the example.

Reference Petri Net		
Transition	**Distribution**	**Parameters**
t1	None	Fires Instantaneously
t2	None	Fires Instantaneously
t3	None	Fires Instantaneously
t4	2-Parameter Weibull	30,2
t5	2-Parameter Weibull	5,1
t6	None	Fires Instantaneously
t7	None	Fires Instantaneously
t8	Normal	0.001,0.0001
t9	Normal	6,0.5
t10	2-Parameter Weibull	12,1
t11	2-Parameter Weibull	3,1

t12	None	Fires Instantaneously
t13	None	Fires Instantaneously
t14	Normal	0.001,0.0001
t15	Normal	6,0.5
t16	None	Fires Instantaneously
t17	None	Fires Instantaneously
t18	Normal	0.001,0.0001
t19	Normal	0.001,0.0001
	Normal	0.001,0.0001
t20	Normal	0.001,0.0001
t21	Normal	0.001,0.0001
t22	Normal	0.001,0.0001
t23	None	Fires Instantaneously
t24	None	Fires Instantaneously
t25	Normal	1, 0.25
t26	Normal	0.001,0.0001
t27	Probability	p=0.2
t28	Normal	0.001,0.0001
t29	Normal	0.001,0.0001
Reduced Petri net 1		
t1	Normal	?,20
t2	Normal	?,1
Reduced Petri net 2		
t1	None	Fires Instantaneously
t2	None	Fires Instantaneously
t3	None	Fires Instantaneously
t4	Normal	50,2
t5	Normal	?,1
t6	Normal	40,1
t7	Normal	?,1
Reduced Petri net 3		
t1	None	Fires Instantaneously
t2	None	Fires Instantaneously
t3	None	Fires Instantaneously
t4	2-Parameter Weibull	30,2
t5	2-Parameter Weibull	5,1
t6	Normal	?, 1

t7	Normal	?, 1
t8	2-Parameter Weibull	12,1
t9	2-Parameter Weibull	3,1
t10	Normal	?, 1
t11	Normal	?, 1
Reduced Petri net 4		
t1	None	Fires Instantaneously
t2	None	Fires Instantaneously
t3	None	Fires Instantaneously
t4	2-Parameter Weibull	30,2
t5	2-Parameter Weibull	5,1
t6	None	Fires Instantaneously
t7	Normal	?,1
t8	Normal	0.001,0.0001
t9	Normal	6,0.5
t10	2-Parameter Weibull	12,1
t11	2-Parameter Weibull	3,1
t12	None	Fires Instantaneously
t13	Normal	?,1
t14	Normal	0.001,0.0001
t15	Normal	6,0.5
t16	Normal	?,1
t17	Normal	?,1
t18	None	Fires Instantaneously

Table A6.1: Data for each of the Petri net models in the fourth example of Chapter 7.