NTNU – Trondheim
Norwegian University of
Science and Technology

Impact of partial and imperfect testing on reliability assessment of safety instrumented systems

Possible approaches for inclusion of its effects in reliability assessments

Eleojo Samuel Ocheni

Reliability, Availability, Maintainability and Safety (RAMS)
Submission date: June 2015
Supervisor: Mary Ann Lundteigen, IPK

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

RAMS
Reliability, Availability,
Maintainability, and Safety

Impact of partial and imperfect testing on reliability assessment of safety instrumented systems.

Ocheni, Eleojo Samuel

June 2015

MASTER THESIS

Department of Production and Quality Engineering

Norwegian University of Science and Technology

Supervisor: Professor Mary Ann Lundteigen

Co-Supervisor: Fares Innal

Preface

This master's thesis is written during 20 weeks throughout spring 2015 as a fulfilment of one of the prerequisites for the award of master's degree (MSc) in Reliability, Availability, Maintainability and Safety, at the Norwegian University of Science and Technology. This thesis with title 'Impact of partial and imperfect testing on reliability assessment of safety instrumented systems' is written with the guidance of my supervisor professor Mary Ann Lundteigen and co-supervisor Fares Innal at the department of reliability, availability, maintainability and safety (RAMS), faculty of Production and Quality Engineering.

The reader of this thesis is assumed to have some basic knowledge within the field of reliability and should be familiar with the textbook System Reliability Theory: Models, Statistical Methods, and Applications by Rausand and Høyland. Familiarity with the IEC 61508 standard is also key to understanding this work.

I would like to thank my supervisor, Mary Ann Lundteigen for her help, guidance and support with this project. Further thanks to my co-supervisor Fares Innal for his important feedback and input throughout this thesis.

Trondheim, 2015-06-10

Ocheni, Eleojo Samuel

Acknowledgment

My special thanks goes to my supervisor, Professor Mary Ann Lundteigen for the support and mentoring during this project. Her patience, encouragement and coaching has made it possible for the realization of this thesis. I would also like to thank Fares Innal for his unflinching support and contribution throughout this thesis. His guidance with the use of the GRIF software is remarkable.

Finally, I would like to thank my parents, siblings and friends for their moral and psychological support during this period.

Ocheni E. S.

Summary and Conclusions

Testing of safety instrumented systems is vital to ensure they are able to perform the required safety function when the need arises. These tests are carried out at specified time intervals. The verification of the ability of the safety systems to perform as required is carried out by reliability assessment. This is the calculation of how likely it is that the safety instrumented system will function when needed.

In carrying out reliability assessment, proof testing of safety systems is assumed to be perfect which is not always the case in reality. This thesis is important because it looks at how to evaluate this assumption to achieve a realistic estimate since testing is a key factor in reliability calculation. This study identifies the main causes of imperfectness which are classified with the five M-factors namely: Method, Machine, Manpower, Milieu and Material. Based on these, the situations where perfect test may not be realistic with examples are reviewed and documented.

I have studied and compared different ways that the effects of tests can be treated. Three approaches to consider imperfectness of test were identified: the IEC 61508 approach where we consider the proportion (fraction) of dangerous undetected failures that are revealed by the proof test, the probability of detecting a dangerous undetected failure during a given proof test and the PDS method of adding a constant probability of test independent failures. The analysis carried out compared the first and second approach. Based on the analysis, the second approach was proposed to be the most suitable of the first two approaches.

Furthermore, we present different reliability assessment methods for estimating the probability of failure on demand of a safety system. The methods used are: analytical formulas, multi-phase Markov, fault tree approach and Petri net. The principles of application and limitation with each of these approaches are presented in this thesis. In the course of this work, we discovered that some complicated cases and systems can only be analyzed by simulation. Finally, a chemical reactor protection system is used as a case study to demonstrate the principles and methods discussed in this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Reliability is an important aspect of any engineering process. The use of safety instrumented systems (SIS) to provide risk reduction of hazards to acceptable level, make operations safer and more reliable. Testing of these SISs to ensure they are able to perform the intended function when demand arises is therefore a necessity.

1.1 Background

Safety instrumented systems (SISs) are used in different industries to detect the onset of a hazardous event and/or to mitigate the consequences. A SIS is made up of three subsystems namely the input elements (sensors), the logic solver and the final (output) elements. The failure of a SIS could lead to loss of lives, environmental disaster and damage of assets, therefore they should be tested at time intervals to ensure they are able to perform the required safety function if a demand arises. This explains why reliability assessments of SIS is of prominence starting from the design to the operational phase, to ensure they meet a minimum functional specification.

Reliability assessments help to verify that the SIS is performing as required and as specified in the safety requirement specification (SRS). A SIS may operate in low demand mode, high demand mode or continuous mode. The probability of failure on demand (PFD) is used to assess the safety integrity of SISs operating in low demand mode which is when the demand rate is less than once per year. IEC 61508 and IEC 61511 are international standards that ensure functional safety throughout the life cycle of a SIS. The IEC 61508 standard stipulates that SISs which are

operating in low demand mode could have some dangerous failures which are not detected by the automatic diagnostic system (self-tests) therefore should be proof tested. The proof tests are meant to reveal any dangerous undetected failures. Proof tests may be full or partial.

Partial test is a supplement to full test to improve the reliability of the system and reduce losses since it does not require a process shutdown. It is meant to reveal some specific critical failure modes and leave some failures to be latent until a full test is performed which then restores the system to an as-good-as-new condition. A partial test policy is defined by the efficiency of the partial tests and the number or distribution (periodic or non-periodic) of the partial tests in the full test time interval. Partial test is considered as imperfect testing in some cases since it does not reveal all failures but this is intentional. Imperfectness of tests could also be unintentional by unrealistic test scenarios or errors during the test. These facts are clarified in this thesis. The introduction of partial test may improve the safety integrity level rating of the system without hardware changes. With this, the proof test interval can be extended thereby reducing losses due to process shutdown (downtime).

The system's reliability is affected during testing by human errors. Constant testing in the form of partial test may also cause wear and degradation. This thesis presents the different ways that the effects of tests can be treated and how they can be factored into the unavailability calculation in order to achieve a realistic and accurate result. The IEC 61508 approach by the use of proof test coverage factor (PTC) and SINTEF's method of adding a constant contribution due to test imperfectness are considered. The use of analytical formulas, multiphase Markov, fault tree and petri nets for the reliability assessment of system subjected to partial and imperfect testing is presented. The implementation of each method for considering partial/imperfect tests and the limitations associated with them are given.

1.2 Problem situation

Proof tests are of paramount importance in achieving high hardware safety integrity. Regular proof tests are vital for revealing dormant failures in safety-instrumented systems. Many models for quantifying the reliability of safety instrumented systems, do however, assume that the tests are perfect. This is an assumption that may be adequate in many cases, but in other cases

it may lead to overly optimistic estimates about the reliability performance since a proof test differs from a real demand situation and some functions may be impossible to test due to potential damage or wear out of the final elements. More focus is now directed to having realistic rather than theoretical estimates of reliability, in particular from a safety barrier management perspective. The question is "how can the imperfectness of tests be quantified or accounted for in reliability assessment to have a more accurate estimate assuming that the tests are imperfect"?

1.3 Objective

The objective of this master thesis is to identify, document and clarify the use of different strategies for considering the imperfectness of proof tests in reliability models. To achieve this, the following tasks shall be performed:

1. To clarify key concepts in relation to proof testing, such as function test, perfect test, imperfect test, staggered testing and partial testing, and discuss the relationship between these.
2. Identify and describe situations where perfect test may not be realistic.
3. Identify and compare different approaches for how the imperfectness of testing can be included in the reliability modeling, using a literature survey as basis.
4. Identify and discuss possible approaches for determining the test coverage.
5. Compare the different approaches using a case study as basis.
6. Discuss the results in light of areas of future research to overcome some of the challenges and difficulties that you have identified.

1.4 Study approach

In the course of this thesis, different resources and approaches have been used to achieve the objectives stated. The technical report ISO/TR-12489 covers the necessary concepts of testing.

The review of this report helps to gain the required knowledge and understanding associated with testing. Imperfect testing in IEC 61508 (2009) and IEC 61511 (2014) standards is called non-perfect proof test and very little about it is mentioned. The SINTEF's PDS method handbook (2006) has a different approach than the standards, making it interesting to study the different approaches to imperfect testing. Discussions, inputs and recommendations from my supervisors with both practical and theoretical testing experiences are also of great importance.

The software MAPLE is used to generate and calculate the analytical formulas applied in this work in chapters 3 and 4 respectively. This made it possible for different configurations to be considered. The GRIF software by TOTAL is used to simulate different models by using the Markov, Fault tree and Petri net module of the software.

1.5 Limitation

The main focus of this thesis is the analysis of partial and imperfect proof testing of SISs. Much research is carried out to accumulate necessary information needed to write this master's thesis which is to be accomplished within a period of 20 weeks. Chapter 11 of the book "Reliability of safety critical systems: theory and applications by Rausand (2014)" is the basic source used in this report. Other information sources for this work are search engines like OnePetro, scopus, google scholar and Sciencedirect.

The focus of this work is only on SISs working in low demand mode. Some assumptions have been made in some cases but are clearly stated where applicable. The methods in this report are in a simple and concise way and in some cases summarized, therefore for detailed explanation and understanding, the sources are available. This makes it necessary for readers to have a background on system reliability theory.

1.6 Structure of report

Chapter one gives an introduction to the general subject matter. The background to the topic and the problem situation are explained. This chapter also outlines the objectives of this thesis.

Chapter two introduces the concept of failure classification. Definitions and explanations of

terms related to testing are documented here. Partial testing and how to determine the partial test coverage factor is given. The different test strategies play significant role in understanding testing. Finally, the reasons for having imperfect tests are presented here.

Chapter three presents the analytical formulas for calculating the performance of a SIS. This is important in order to see how the full proof test interval τ is used in the formulas. Here, the IEC, ISA, PDS and other authors' formulas for PFD calculation are presented and tabulated for selected configurations.

In chapter four, partial and imperfect testing methods are covered. The IEC 61508 approach by using the proof test coverage (PTC) is introduced as well as the PDS use of probability of test independent failures (P_{TIF}). The Markovian approach is used to disprove the correctness of the IEC 61508 non-perfect proof test formula for a 1oo2 configuration. Finally, some authors' analytical formulas for partial test are presented and compared.

In chapter five, different assessment methods are used to verify the analytical formulas for partial and imperfect testing. Multiphase Markov which is used for periodically tested components is described and the limitations stated. Here, the fault tree approach is also used to model different configurations.

Chapter six presents the dynamic nature of Petri nets. The different modeling alternatives are used to demonstrate their practicability. The use of PN for combined multiple components configuration, CCFs and staggered testing models are given.

A case study is introduced in chapter seven to demonstrate all the discussed reliability assessment methods and concepts.

Finally discussion of results, recommendation and conclusion summary are presented in chapter eight.

Chapter 2

Failure classification and testing Concepts

Proof testing of a SIS is a vital activity for ensuring its ability to respond and act as required when a demand arises. Different discussions, research and analysis is carried out to see how this can work perfectly and how it can be accurately modelled in reliability quantification. This chapter starts by introducing failure classification which is a basis for describing and discussing testing concepts.

2.1 Failure classification

Failure is defined as the termination of the ability of a functional unit to provide a required function or operation of a functional unit in any way other than as required (IEC-61508, 2009). The standard categorizes failures as random hardware failure and systematic failure according to the failure causes. These failure mode classifications are defined:

- Random hardware failure: IEC-61508 (2009) defines these failures as those whose occurrences are random in nature. The failures are caused by natural degradation mechanisms of the hardware which could be due to ageing failures or stress related failures.

- Systematic failures are caused by errors in the specification, design, operation and maintenance phases. These failures can only be rectified by the modification of the design or the manufacturing process, operational procedures, testing, documentation or other relevant factors. Hauge et al. (2013) further splits systematic failures into five categories

namely the software, design related, installation, excessive stress and operational failures.

The failure classification is illustrated in figure 2.1 based on the PDS model which gives a more detailed breakdown of systematic failures.

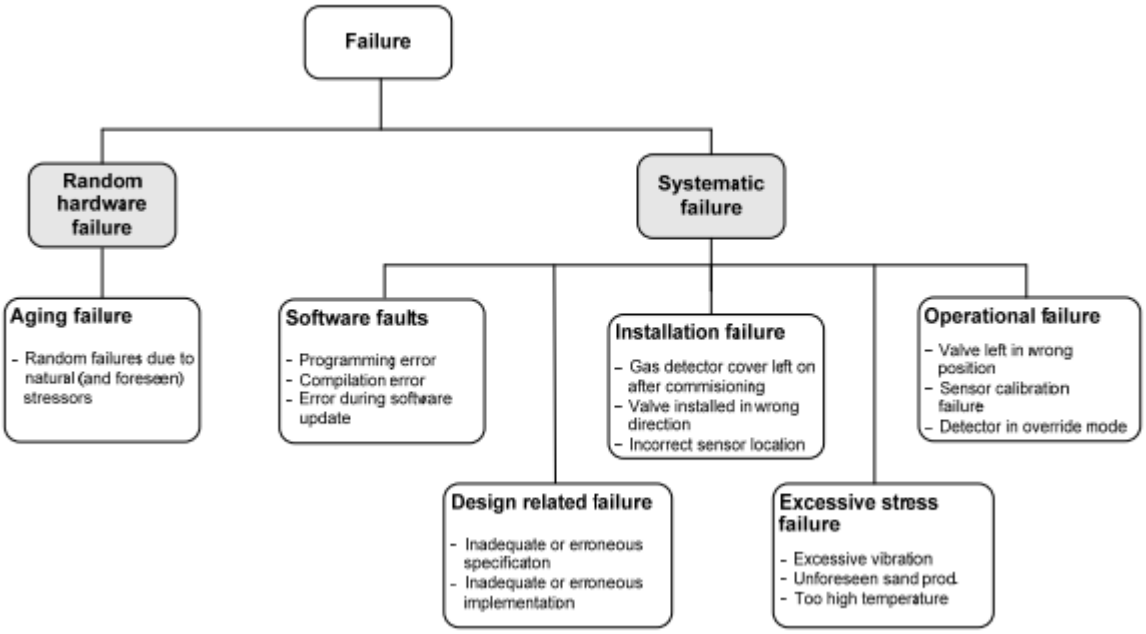


Figure 2.1: Failure classification by causes (adapted from Hauge et al., 2013).

IEC 61508 standard classifies failure modes of random hardware failures as explained below and shown in figure 2.2. They are the dangerous and safe failures.

- Dangerous failure is the failure that causes the SIS not to perform its function upon demand. It is further divided into the dangerous undetected (DU) which is only revealed by proof testing or when a demand occurs represented with a failure rate denoted λ_{DU} and dangerous detected (DD) failures which are revealed automatically by diagnostic testing represented with a failure rate λ_{DD} (IEC-61508, 2009; Rausand and Høyland, 2004).

- Safe failure (S) is a failure that is not dangerous to the system function and this is further classified as safe undetected (SU) and safe detected (SD). These failures are represented by a failure rate λ_{SU} and λ_{SD} respectively. An effect of a safe failure is a spurious trip. A closure of safety valve without a real demand is an example of a spurious trip caused by safe failures (Hauge et al., 2013).

From figure 2.2 the general failure rate of any component is given by λ which is a sum of both safe and dangerous failure rates thus:

$$\lambda = \lambda_D + \lambda_S \tag{2.1}$$

The dangerous and safe failure rates which are expressed in terms of the detected and undetected conditions are given as:

$$\begin{cases} \lambda_D = \lambda_{DU} + \lambda_{DD} \\ \lambda_S = \lambda_{SU} + \lambda_{SD} \end{cases}$$

The logic solvers of modern SISs carry out diagnostic testing during online operation to detect failures of input and output devices by sending signals frequently to confirm the status of the devices. This concept is called diagnostic self testing. Diagnostic coverage (DC) is the fraction of failures that can be revealed by diagnostic self testing. Annex C of IEC 61508 part 2 and IEC 61508 part 6 give the method for calculating diagnostic coverages and examples respectively. DC in relation with the dangerous failure rate is expressed in the equation:

$$DC = \frac{\sum \lambda_{DD}}{\sum \lambda_{DD} + \sum \lambda_{DU}} \tag{2.2}$$

Therefore dangerous failures can be expressed as $\lambda_{DD} = \lambda_D \cdot DC$ and $\lambda_{DU} = (1 - DC)\lambda_D$. Some possible failure modes of a valve are described in Rausand and Høyland (2004) as the following: (i)Failure to close (FTC), (ii) Leakage in closed position (LCP), (iii) Spurious trip (ST) and (iv) Failure to open (FTO).

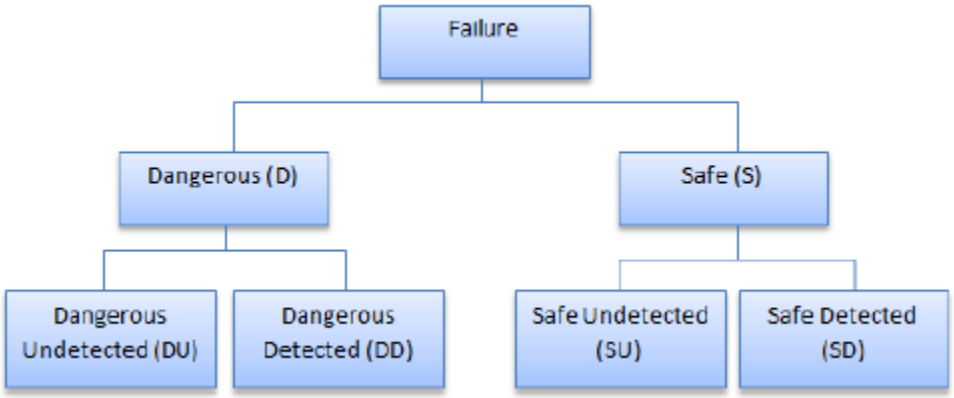


Figure 2.2: Classification of failure mode

2.1.1 Common Cause Failures (CCF)

CCF is defined in IEC-61508 (2009) as failure which occurs as a result of one or more events, causing concurrent failures of two or more separate channels in a multiple channel system therefore leading to system failure. This could be as a result of shock or stress (e.g. temperature, humidity, vibrations) over a certain period of time. For this reason, failure of a component can further be classified into failures due to independent causes and failures due to common causes.

$$\lambda = \lambda^{(I)} + \lambda^{(C)} \tag{2.3}$$

Independent failures are failures that affect a certain component independent of the others whereas CCF is a concurrent failure that affects more than one component in parallel. The beta factor model is a commonly used approach to model CCF. The (β) factor is used to partition the total failure rate into failures due to independent and CCF.

$$\beta = \frac{\lambda^{(C)}}{\lambda} \tag{2.4}$$

Therefore the independent and common-cause failure rates can be expressed in terms of the total channel failure rate λ and the common cause factor β as:

$$\lambda^{(I)} = (1-\beta)\lambda \quad \text{and} \quad \lambda^{(C)} = \beta \cdot \lambda$$

For this project, the expressions below are going to be used to differentiate the dangerous detected and undetected failure rates and classify with respect to independent and CCFs respectively:

$$\lambda_{DU}^{(i)} = (1-\beta)\lambda_{DU} \quad \text{and} \quad \lambda_{DU}^{(c)} = \beta \cdot \lambda_{DU}$$

$$\lambda_{DD}^{(i)} = (1-\beta_D)\lambda_{DD} \quad \text{and} \quad \lambda_{DD}^{(c)} = \beta_D \cdot \lambda_{DD}$$

2.1.2 Influence of Common Cause Failures on testing

Redundancy enhances the performance of SISs but the reliability effect may be reduced if the components are exposed to factors like design errors, operational errors, maintenance errors

and errors during testing. Common cause failures (CCF) is when redundant components fail due to the same cause. The testing team may be the reason for a CCF by mis-calibrations, failure to reset equipment etc. The likelihood of CCFs is less in staggered testing (explained in the next section) since the tests of redundant channels would be at different times therefore less likelihood of replicating the same error and there is a possibility of different test teams carrying out the test (Rausand, 2014).

2.2 Principles of testing

Testing is defined as the execution of a function on a system, subsystem or channel to confirm that its function can be performed according to the stipulated requirements (ISA-TR84.00.03, 2002). A functional safety manual is prepared by the SIS manufacturers which contains guidelines for installation, testing, operation and maintenance of the SIS. During operations, tests may be proof tests, partial tests or diagnostic tests. The following sub sections explain the different categories of tests.

2.2.1 Proof test/Function test

The term proof test is sometimes used interchangeably with function test. While some authors see them as the same, others see them as different and others even use the terms together as functional proof testing. Proof tests are activities performed mainly, to ascertain a specified safety integrity level of a safety system is met. The objective of proof testing is to detect dangerous undetected failures. IEC-61508 (2009) part 4 defines proof test as the "periodic test performed to detect dangerous hidden failures in a safety-related system so that, if necessary, a repair can restore the system to an “as good as new” condition or as close as practical to this condition". Periodic proof testing can contribute to achieving and improving the SIL of the system without making modifications to the safety system design (Torres-Echeverria, 2009). The guiding principles and proof test practices and procedures with examples are explained in HSE-UK (2002) which establishes that the purpose of proof testing is the detection of unrevealed fail-to-danger faults at the time of test. This document contains the content and format of proof testing procedures, planning and scheduling, proof test records, required competence, awareness of

hazard and fianlly, risk and management of change. The HSE UK study justifies that there exists a discord between the need for realistic proof testing and the need to minimise downtime. Despite the fact that this study specifies that the end-to-end test is the ideal practice, it also admits that partial testing is a necessary practice in situations where end-to-end testing is practicable.

The term functional testing as used in IEC-61508 (2009) part 7 means to "reveal failures during the specification and design phases in order to avoid failures during implementation and integration of software and hardware". This consequently means that proof test and functional tests are not the same. In contrast, according to ISA-TR84.00.03 (2002), it is explicitly written that proof tests and functional tests are the same test.

Different authors have worked on identifying proof test intervals. IEC-61511 (2014) mentions that the PFD_{avg} should be used to determine the frequency of proof test. The standards also states that different test frequencies may be used for different parts of the SIS.

2.2.2 Perfect test and Imperfect testing

A perfect test is the ability of the proof test to reveal all DU faults in the component. The assumptions associated with a perfect test are outlined namely:

- The test is performed under similar conditions as the real demand situation.
- The proof test should reveal all DU failures and elements faults that could lead to a DU fault.
- Revealed DU faults are repaired and all channels should be in as-good-as-new condition after repairs.

In most cases, proof tests are considered to be perfect which is not practical. Some factors may affect the test or the test may not cover every aspect which may lead to some DU faults not being revealed. An imperfect proof test will result in a safety function that is not restored to ‘as good as new’ and therefore the probability of failure will increase (IEC-61511, 2014) part 2. Different mathematical expressions have been developed and included in the PFD_{avg} calculation considering the effects of imperfect or partial proof tests of SISs. IEC 61508 refers to imperfect test as non perfect testing.

Bukowski and Van Beurden (2009) classified imperfect testing under two categories: Incomplete and incorrect testing.

- Proof test completeness is here defined as the probability that all dangerous failures are revealed/checked for during a proof test which is a function of the component and the tests that are executed. Based on this definition, the completeness/incompleteness of proof tests can be part of partial or full proof tests. Incomplete proof test therefore has to do with the limitation of the test.

- Proof test correctness indicates the probability that the actual test is correctly executed by the test team as specified and that all existing faults are revealed, repaired and no new problems are introduced during the test. This is therefore seen as a function of the maintenance capabilities and culture at a specific plant site. Incorrect proof test has to do with the limitations of those performing the test.

Impact analyses shows that test completeness has a higher impact on PFD than test correctness (Brissaud et al., 2012; Bukowski and Van Beurden, 2009).

2.2.3 Reasons for having imperfect test

The subsection above mentioned imperfectness of tests and this may be a situation whereby test conditions and procedures are not exactly the same as real demand conditions. Rolén (2007) attributed the reasons for imperfectness of tests to five main factors namely the methods, materials, machines, milieu and manpower. These attributes with their individual characteristics are shown in figure 2.3.

Hauge et al. (2013) gives some typical examples of how some test conditions may not reveal all failures which include but not limited to the following:

- **Partial stroke testing:** This is a planned proof test intended to reveal only some specific failure modes. All failure modes are not revealed but this is intentional.

- **Test buttons on switches:** In this case, test facilities are built in the devices which is used during the proof tests and these facilities may or may not reveal all faults.

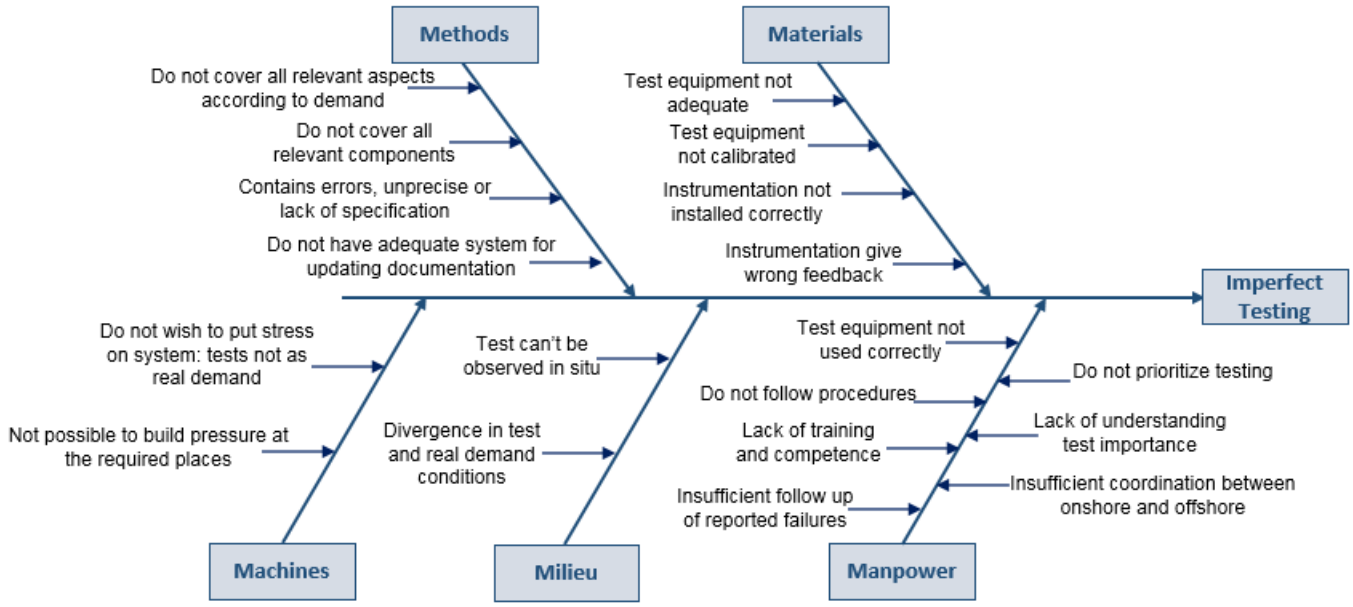


Figure 2.3: Fishbone diagram for causes of imperfect testing (adapted from Rolén, 2007).

- **Transmitters put into test mode and signals injected:** The test mode may have bypassed the original functioning mode of the transmitters and the injection of signals with smart/fieldbus transmitters may be different from real situation hence not revealing all failure modes.

- **Pressure transmitters tested from manifold:** This means that the impulse lines are not tested. In the case of pressure transmitters, the test of such transmitters is by introducing pressure from an external source to see if the PT senses and reacts accordingly as shown in figure 2.4. In a real life, there might be a blockage from where the pressure changes are present, thereby making the tests not 100 per cent perfect as the surrounding factors are not included in the test procedure as depicted by the red circle in the figure.

  Another typical illustration of imperfect proof test of pressure transmitters is that the tests are normally performed after the transmitters have been isolated from the process since pressurizing a pipeline to the preset trip pressure could lead to an unsafe situation. When this test is carried out, DU failures which may be caused by contamination in the pressure sensing lines may not be revealed by the test (Jin et al., 2013).

- **Equipment not tested in normal position:** An example of this is proof testing of gas/fire detectors which could be challenging. The introduction of gas or smoke fumes need to

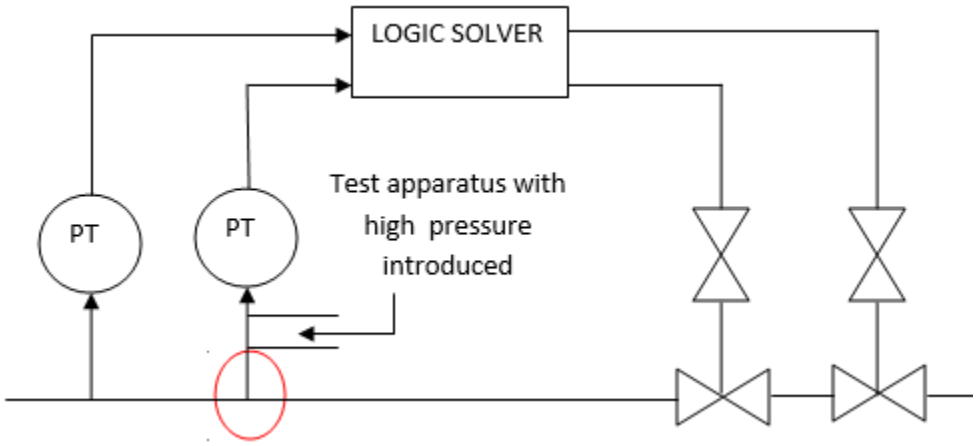


Figure 2.4: Pressure transmitter test illustration

be organized so as to reach the height where these detectors are installed. The use of test gas and smoke fumes which are not exactly the same as the ones used in real situation is a limitation. Also the position or contact it makes with the detectors because of the test apparatus is another factor. There could also be a situation where a change or modification (maybe installation of pipes and flanges in the room or area) of the existing layout, blocks the actual functioning of the detectors but the tests results show perfect working which is different from a real situation.

Figure 2.5 shows how a test gas is applied to a detector to verify its functionality Hauge et al. (2013).

2.2.4 Online and offline testing

Online testing is a test carried out while the process or EUC is operating its normal function. For offline testing, the test is performed while the process or equipment being protected is not in operation. An example of an online test is a pressure compressor operating when a channel (may be a transmitter) of a SIS providing protection a SIF is being tested. The test is considered to be an online test of the transmitter. In as much as online tests does not cause down times, they should not be implemented unnecessarily to compromise the process safety integrity during the test. A risk assessment and cost-benefit evaluation should be carried out on the test equipment and procedure to determine whether the danger of causing an incident due to performing the

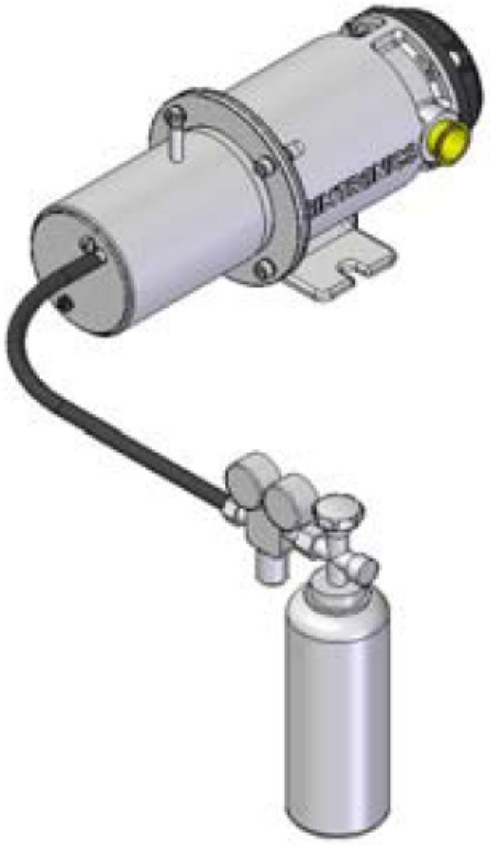


Figure 2.5: Test apparatus for gas detectors

on-line test is greater than the danger of not discovering the failure (ISA-TR84.00.03, 2002).

2.2.5 Full proof test and partial test

Full proof test is a test performed at intervals meant to reveal **all** latent failures of the equipment or component being tested. Proof testing in most cases requires a system shutdown which affects the production and leads to production downtime. A partial proof test is a planned test is implemented to enable extension of the full proof test in order to avoid production loss while still maintaining the integrity of the system. A partial proof test is designed to test one or more specific failure modes of a channel without significantly disturbing the EUC. The HSE-UK (2002) classifies partial testing under two categories namely:

- Testing of system components at different times and frequencies which is called staggered testing.

- Testing of the subsets of functions of single components in the form of measurement simulation or the partial stroking of valves.

2.3 Partial Stroke Testing

Partial stroking of safety valves is a type of partial test to detect failure mode like "failure to close on demand" but it is not possible to detect a failure mode like "leakage in closed position" (Jin and Rausand, 2014). Though partial tests are not as effective as full tests, they have some advantages over full tests. They are less costly and less time consuming and also some safety devices are preferably partially tested in order not to cause degradation or destruction (Brissaud et al., 2012). Figure 2.6 shows how partial stroke testing is carried out on a shutdown valve using two methods: the integrated manually activated PST from the SIS logic solver and the use of vendor package PST equipment. These are highlighted with red in the figure. Failure To Close (FTC) is a

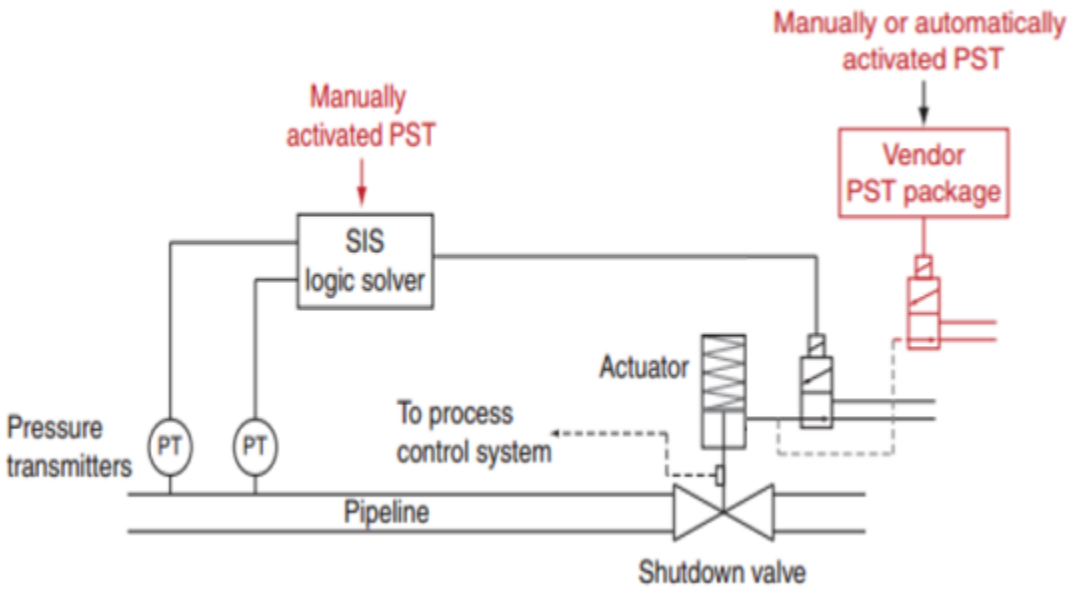


Figure 2.6: PST setup for integrated and separate vendor activation (adapted from Lundteigen and Rausand, 2008a).

common and dangerous failure mode of a shutdown valve. By moving the valve, we can detect this failure. This type of failure can therefore be revealed by what is called Partial Stroke Testing (PST) and this test does not require production shutdown. PST covers a specific failure mode not discovered by diagnostic self-test and it reduces down time and the full proof test interval can be

made longer. Full stroke operation and leakage testing requires a shutdown of process. Partial stroke testing has been introduced to supplement functional testing (Ali et al., 2004; Summers and Zachary, 2000). PST is a way by which a valve is partially opened or closed then returned to its initial position to detect several specific types of DU failures without interrupting the process. The total PFD where a partial test is implemented is given by the equation:

$$PFD_{avg} = PFD_{FT} + PFD_{PT}$$

Figure 2.7 shows the split of the dangerous failure rate λ_D into different failure rates.

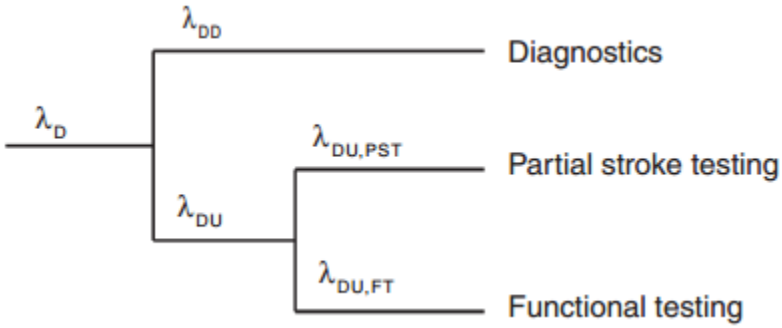


Figure 2.7: Relevant failure rates (adapted from Lundteigen and Rausand, 2007).

(Lundteigen and Rausand, 2008a) describes an approach to finding the test coverage of shutdown valves. The partial test coverage is the proportion of DU failures tested by the partial test and may be expressed be the given equation:

$$\theta_{PST} = \frac{\lambda_{DU,PST}}{\lambda_{DU}} \quad \text{and} \quad \theta_{DC} = \frac{\lambda_{DD}}{\lambda_D} \tag{2.5}$$

where θ_{PST} = Pr (Detect DU failure by PST|DU failure is present) and θ_{DC} is the diagnostic coverage. The PFD of the system in terms of the fraction of DU failures detected by the PST can be expressed as:

$$PFD \approx PFD_{FT} + PFD_{PST} \approx (1-\theta_{PST}) \cdot \frac{\lambda_{DU}\tau_{FT}}{2} + \theta_{PST} \cdot \frac{\lambda_{DU}\tau_{PST}}{2} \tag{2.6}$$

Notice that the $PFD_{DT} = \frac{\lambda_{DD}\tau_{DT}}{2}$ for diagnostic testing is not considered in the formula because diagnostics are performed at short intervals. The PFD considering the PST and FT are shown

in figure 2.8. It shows the PFD reduces with a PST introduced because it reveals and corrects failures within a shorter time interval than the full test. Therefore the SIF's reliability is improved (Lundteigen and Rausand, 2008a).

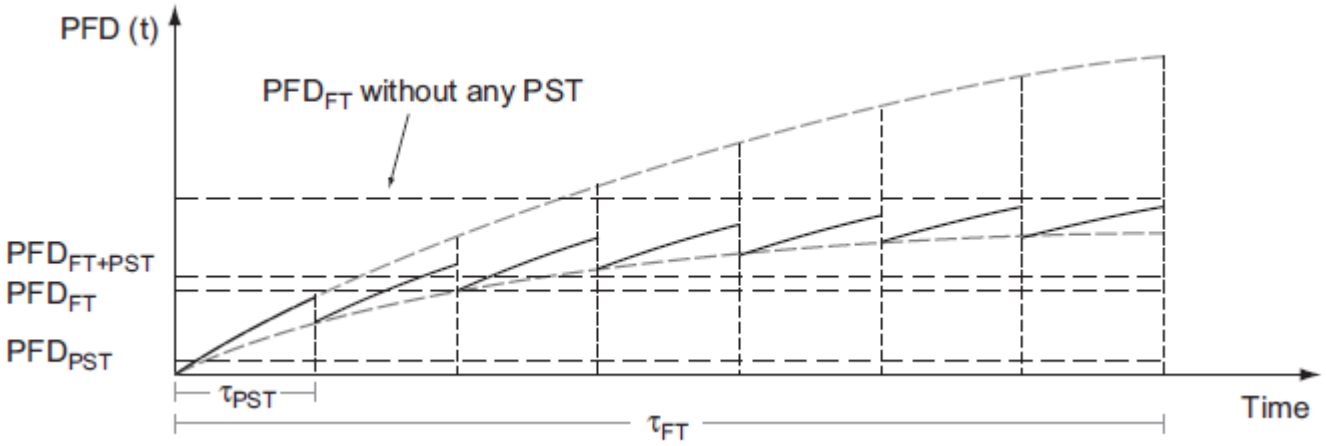


Figure 2.8: Impact of PST on PFD (adapted from Lundteigen and Rausand, 2008a).

2.3.1 Determining the PST coverage

(Lundteigen and Rausand, 2008a) assumes the PST coverage to be a property of individual SIS components rather than a group of components therefore could be expressed as:

$$\theta_{PST} = \frac{Pr(\text{Detect DU failure by PST} \cap \text{DU failure is present})}{Pr(\text{DU failure is present})} \tag{2.7}$$

Further assumptions are that only one failure mode is present at a time even though there could be other DU failure modes represented by $FM_1, FM_2, \cdots, FM_n$ hence the assumption that they are mutually exclusive written as:

$$\theta_{PST} \approx \sum_{i=1}^{n} \frac{Pr(\text{Detect}FM_i|FM_i\text{is present}) \cdot Pr(FM_i\text{is present})}{Pr(\text{DU failure is present})} \tag{2.8}$$

The PST coverage of the DU failure mode FM_i is represented by $\theta_{FM,i} = Pr(\text{Detect}FM_i|FM_i\text{is present})$ and $w_i = \frac{Pr(FM_i\text{is present})}{Pr(\text{DU failure is present})}$ The PST coverage can therefore be expressed as:

$$\theta_{PST} \approx \sum_{i=1}^{n} \theta_{FM,i} \cdot w_i \tag{2.9}$$

This method suggests that $\theta_{FM,i}$ can be determined by evaluating the failure mode's revealability and reliability of the test which can be achieved by expert judgement and checklists respectively. Details of how the PST coverage is determined is explained in 6 steps

- **Step 1:** Becoming familiar with the PST and its implementation.
  This includes getting acquainted with the SIS components operated during a PST, the functional safety requirements of the SIS components like valve closing time, PST initiation and control by dedicated hardware and software, process control system and finally the operational and environmental conditions under which the SIF operates, including fluid characteristics, temperature and pressure.

- **Step 2:** Analyze the PST hardware and software.
  The FMEA analysis is suggested to identify and analyze potential PST hardware and software failures and the effect these failures may have on the PST execution and the SIS. This should be done in collaboration with end users and vendors and it serves as a basis for the checklist.

- **Step 3:** Determine the PST reliability.
  Checklist containing questions which gives credits to the system behavior is used to provide reliable and useful test results. Each question is weighted according to importance. for details of this step refer to Lundteigen and Rausand (2008a).

- **Step 4:** Determine the revealability (per failure mode).
  Deciding whether or not the failure mode may be revealed by the PST. A failure mode may also only be revealed for a portion of the failures in each failure mode. A failure mode that is fully observable is given the revealability factor 100 percent and when not observable at all 0 percent. A failure mode may also be revealable with a certain probability, which is used as the revealability factor.

- **Step 5:** Determine the failure mode weight.
  The weight of failure mode is the fraction the specific failure mode among all failures, shown previously with the equation for w_i. The failure mode weight is determined by expert judgment or by analysis of historical data.

- **Step 6:** Determine the PST coverage.

  The PST coverage θ_{PST} can now be calculated using the formula 5.5 since the values needed have been derived from previous steps.

2.4 Relationship between the types of tests

Though the term proof tests and functional tests may be mixed up or used interchangeably, it is clear from the above sections that they are different. Function test is a test procedure of running or activating a function of the SIF. This does not include other tests like pressure tests (calibration) and leakage tests of valves. Therefore proof tests encompasses both functional tests and leakage tests with other kinds of tests and calibrations. OLF-070 (2004) uses the term functional proof testing to mean the same thing. The guideline's requirement of proof tests is to verify that the entire SIS loop including the sensors, logic solvers and the final elements are working adequately. OLF 070 specifies the aspects which the proof test should cover summarized in complete system functionality. A full proof test and partial proof test may be imperfect if they don't reveal the faults they are designed and expected to reveal.

The figure below shows the relationship between the tests. There are different understand-

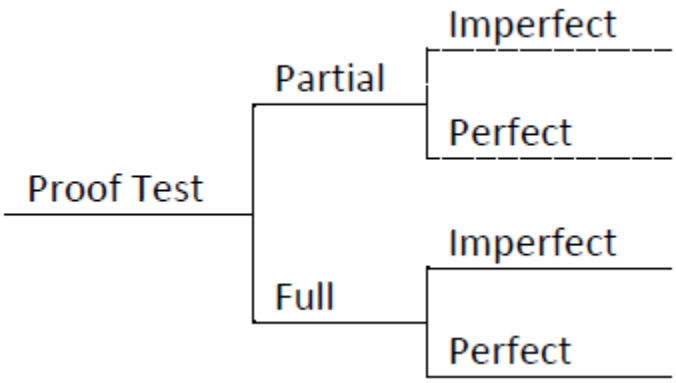


Figure 2.9: Proof test classification (adapted from Rausand, 2014).

ings and contentions on the classification shown in figure 2.9. Some say a partial test is already imperfect as it does not reveal all DU failures. However, the partial test is designed to reveal only some specific failures so it can be imperfect if it does not reveal those failures. Partial from a perspective can be called incomplete functional test.

2.5 Adverse effects of full proof testing and partial testing

In as much as proof test ensures that a safety system is able to function as required upon demand, it also presents some challenges and adverse effects. One of these effects is the down time of the safety system due to full proof tests. Human error is another factor. Lundteigen and Rausand (2008b) mentioned that human error during tests is an additional factor contributing to increase in spurious activations of a SIS. In the differences between diagnostics and functional testing, human interaction during test preparation, execution and restoration has an effect which could be adverse (Lundteigen and Rausand, 2008a). The concepts of human error which could be failure to detect a fault and leaving the component in bad state after test was used in the analysis of optimal test interval. Operator error and the probability of test-caused failure were modelled as constant unavailability and added to the time-dependent unavailability quantification (Lee et al., 1990).

Partial tests on the other hand has its disadvantages. From the definition, only a portion of DU failures are revealed which leaves other dangerous undetected failures which could prevent the system from responding in case of a demand. Secondly, the frequent operation leads to wear and degradation of the system. In case of a valve, there is a potential increase in spurious trip rate since the valve may continue to fail safe position instead of returning to the initial position.

2.6 Test strategies

Test strategies are classified as simultaneous, sequential, staggered and independent tests according to Torres-Echeverria et al. (2009). Proof testing strategies specify the scheduling of proof tests of redundant components with respect to one another. The different strategies are explained the the subsections. A petri net model for the different testing strategies using a 1oo2 subsytem is shown (Liu and Rausand, 2013).

- **Simultaneous testing :** This is when a number N of redundant components are tested at the same time where the time of proof test t is the same for all components. This means that the same number of crews as the components are available during the test. In situation where the safety system must always be in a functioning state, this strategy is not

suitable because the safety system is made unavailable during the test period.

- **Sequential testing :** Sequential testing is a situation where N redundant components are tested one after the other. A second component test can only begin when the first component has been tested and restored to a working state. This sequence repeats for all the components of the subsystem. This strategy enables the SIS to operate in a degraded mode since $n-1$ channels are available when a channel is being tested. (Cepin, 1995; Liu and Rausand, 2013).

- **Staggered testing :** This is a type of sequential testing where the n tests are spread out over the entire test interval. This strategy increases the safety availability of the SIS. IEC 61508 part 6 mentions that If the tests are staggered and adequate procedures implemented, the likelihood of detecting CCFs increases and it is an effective method of reducing the CCF for systems operating in a low demand mode of operation. staggered testing has its adverse effects. The recurrent tests requires extra maintenance management and the cost of testing could increase significantly. An example is a situation where the equipment to be tested is offshore. Hiring the test vessels for different test times means extra cost.

- **Independent testing :** In the case of independent testing, the time of test of the N components are in a random order. There is no specific test schedule between the components (Torres-Echeverria et al., 2009).

Chapter 3

Analytical formulas for performance of SIS

3.1 Analytical approach based on full proof tests

Analytical formulas are used to determine the probability of failure of a SIS. These formulations are only approximations. Different analytical formulas found in the literature for PFD average will be presented in this chapter. The probabilistic performance of a safety function provided by a given SIS is determined by calculating the combination of the performance of the three subsystems (S, LS and FE) as depicted in figure 3.1.

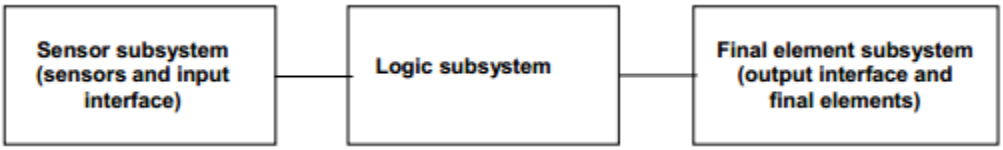


Figure 3.1: Subsystem structure

3.1.1 IEC 61508 approach

The PFD_{avg} for the system is given as the sum of the PFDs for each subsystem expressed in equation 3.1 below:

$$PFD_{sys} = PFD_S + PFD_{LS} + PFD_{FE} \tag{3.1}$$

The main idea of the IEC formulas is to calculate the PFD_{avg} of a voted group (G) of channels as if the group were a single item. The calculation is based on the average dangerous group failure

frequency ($\lambda_{D,G}$) and the group-equivalent mean downtime (t_{GE}). The PFD_{avg} for the group is calculated as:

$$PFD_{avg}^{(G)} = \lambda_{D,G} \cdot t_{GE} \tag{3.2}$$

The reliability block diagram for a 1oo1 architecture of a subsystem considering both DD and DU failures is: The assumptions for the IEC 61508 formulas are listed below:

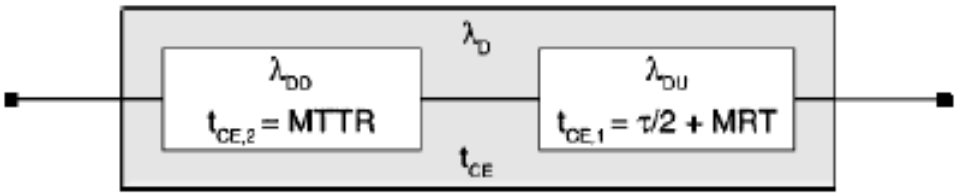


Figure 3.2: 1oo1 RBD

- All failure rates are assumed constant.
- Components are statistically independent.
- Function test coverage is 100 percent.
- All components in an architecture have the same failure rate and diagnostic coverage.
- The function test is at least one order of magnitude greater than the mean repair time (MRT).
- For each subsystem there is a function test interval and MRT.
- Test times are neglected and expected interval of demand is greater than the test interval.
- $e^{-\lambda_{DU} \cdot \tau} \approx 1 - \lambda_{DU} \cdot \tau$ assuming that $\lambda_{DU} \cdot \tau$ is a small enough value ($\lambda_{DU} \cdot \tau \ll 1$).

Based on this understanding, the IEC 61508 formulas for different configurations is given in table 3.1. IEC-61508 (2009) presents detailed derivation of these formulas.

Architecture	PFD_{avg} **according to IEC 61508 Part 6**
1oo1	$(\lambda_{DU}+\lambda_{DD})\,t_{CE}$
1oo2	$2((1-\beta_D)\lambda_{DD}+(1-\beta)\lambda_{DU})^2\,t_{CE}\,t_{GE}+\beta_D\lambda_{DD}MTTR+\beta\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)$
1oo3	$6((1-\beta_D)\lambda_{DD}+(1-\beta)\lambda_{DU})^3\,t_{CE}\,t_{GE}\,t_{G2E}+\beta_D\lambda_{DD}MTTR+\beta\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)$
2oo2	$2(\lambda_{DU}+\lambda_{DD})\,t_{CE}$
2oo3	$6((1-\beta_D)\lambda_{DD}+(1-\beta)\lambda_{DU})^2\,t_{CE}\,t_{GE}+\beta_D\lambda_{DD}MTTR+\beta\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)$

where:

$t_{CE}=\frac{\lambda_{DU}}{\lambda_D}\left(\frac{\tau}{2}+MRT\right)+\frac{\lambda_{DD}}{\lambda_D}MTTR$ and $t_{GE}=\frac{\lambda_{DU}}{\lambda_D}\left(\frac{\tau}{3}+MRT\right)+\frac{\lambda_{DD}}{\lambda_D}MTTR$

$t_{G2E}=\frac{\lambda_{DU}}{\lambda_D}\left(\frac{\tau}{4}+MRT\right)+\frac{\lambda_{DD}}{\lambda_D}MTTR$

MTTR (Mean Time To Repair): Mean time to restore a dangerous detected failure.

MRT (Mean Repair Time): Mean time to repair of dangerous undetected failure.

The standard assumes MTTR=MRT and $\beta_{DU}=\beta$ and $\beta_{DD}=\beta_D$

The standard does not take into account CCF for series architecture like the 2oo2 configuration.

Table 3.1: Analytical formulas based on IEC 61508

3.1.2 ISA approach

ISA-TR84.00.02 (2002) provides the steps in calculating the PFD_{avg} for typical configurations of SIF designed according to ANSI/ISA-84.01-1996. The formulas here include contribution from systematic failures. A difference in the calculation of the PFD with the IEC 61508 is that the IEC standard considers that DD failures of a channel will take the channel to a failed state and it will remain in this state until the component is repaired hence the contribution from DD failures to the PFD during the restoration time. In the ISA standard, the assumption is when a dangerous detected failure occurs, the SIS will take the process to a safe state therefore DD failures are not considered on the PFD calculation (Oliveira and Abramovitch, 2010). Table 3.2 presents the ISA formulas for selected configurations.

Architecture	PFD_{avg} **according to ISA-TR84.00.02-2002**
1oo1	$\left(\lambda_{DU}\cdot\frac{\tau}{2}\right)+\left(\lambda_F^D\cdot\frac{\tau}{2}\right)\approx\lambda_{DU}\cdot\frac{\tau}{2}$
1oo2	$\left(\lambda_{DU}^2\cdot\frac{\tau^2}{3}\right)+\left(\lambda_{DU}\cdot\lambda_{DD}\cdot MTTR\cdot\tau\right)+\beta\lambda_{DU}\cdot\frac{\tau}{2}$
1oo3	$\left(\lambda_{DU}^3\cdot\frac{\tau^3}{4}\right)+\left(\lambda_{DU}^2\cdot\lambda_{DD}\cdot MTTR\cdot\tau^2\right)+\beta\lambda_{DU}\cdot\frac{\tau}{2}$
2oo2	$\left(\lambda_{DU}\cdot\tau\right)+\left(\beta\lambda_{DU}\cdot\tau\right)$
2oo3	$\left(\lambda_{DU}^2\cdot\tau^2\right)+\left(3\cdot\lambda_{DU}\cdot\lambda_{DD}\cdot MTTR\cdot\tau\right)+\beta\lambda_{DU}\cdot\frac{\tau}{2}$
where: λ_F^D is the dangerous systematic failure rate. τ is the time interval between manual functional tests of the component.	

Table 3.2: Analytical formulas based on ISA-TR84.00.02-2002

3.1.3 The PDS method

SINTEF developed the PDS method based on the IEC 61508 and 61511 principles and it is widely used in the Norwegian petroleum industry. The main differences of this method and the IEC standards in relation to the calculation of PFD is a different CCF model called the multiple beta-factor model. The PDS Beta factor model distinguishes between different types of voting. The configuration is considered in relation to the rate of CCFs and the beta-factor of an MooN voting logic is expressed as $\beta(MooN)=\beta C_{MooN};(M<N)$. The figure below shows the C_{MooN} values for some votings (Hokstad and Corneliussen, 2004).

Voting	1oo2	1oo3	2oo3	1oo4	2oo4	3oo4
C_{MooN}	1.0	0.3	2.4	0.15	0.8	4.0

Figure 3.3: C_{MooN} factors based on system voting logic (adapted from Hokstad and Corneliussen, 2004).

Based on the values in the above figure, the PFD formulas for some common configurations have been derived:

A simplified and generalized form of the formulas given in table 3.3 are provided in table 3.4. The simplification relies on disregarding the contribution of detected failures and the beta factor.

Architecture	PFD_{avg} **according to PDS method**
1oo1	$\lambda_{DU} \cdot \frac{\tau}{2} + \lambda_{DD} \cdot MTTR$
1oo2	$(1-\beta)^2 \lambda_{DU}^2 \cdot \frac{\tau^2}{3} + 2(1-\beta)\lambda_{DD} \cdot \lambda_{DU} \cdot MTTR \cdot \frac{\tau}{2} + \beta\left(\lambda_{DD} \cdot MTTR + \lambda_{DU} \cdot \frac{\tau}{2}\right)$
1oo3	$0.3\left[\beta \cdot \lambda_{DD} \cdot MTTR + \beta\lambda_{DU} \cdot \frac{\tau}{2}\right] + \frac{1}{4}\left[(1-1.7\beta)\lambda_{DU} \cdot \tau\right]^3 + 3(1-1.7\beta)\lambda_{DD} \cdot MTTR \cdot \beta\lambda_{DU} \cdot \frac{\tau}{2}$
2oo2	$(2-\beta)\left(\lambda_{DU} \cdot \frac{\tau}{2}\right) + \beta \cdot \lambda_{DD} \cdot MTTR$
2oo3	$2.4 \cdot \beta\lambda_{DU} \cdot \frac{\tau}{2} + \left[(1-1.7\beta)\lambda_{DU} \cdot \tau\right]^2 + 3(1-1.7\beta)\lambda_{DD} \cdot MTTR \cdot \beta\lambda_{DU} \cdot \frac{\tau}{2}$
SINTEF does not use the normal β factor model for CCF. The coefficient β for CCF is the same for both detected and undetected $\beta_{DU} = \beta_{DD} = \beta$.	

Table 3.3: Analytical formulas based on PDS method (adapted from Hokstad and Corneliussen, 2004)

3.1.4 Other authors approaches

Oliveira and Abramovitch (2010) introduces a generalization of the ISATR84.00.02-2002 PFD equations for application to any KooN architecture especially to systems with higher redundancy. The equation presented for KooN is:

$$\begin{aligned} PFD_{KooN} = {} & C_N^{N-K+1}\left((1-\beta)\lambda_{DU}\right)^{N-K+1}\tau^{N-K}\left(\frac{\tau}{N-K+2} + MRT\right) + C_N^{N-K+1}\left((1-\beta_D)\lambda_{DD}MTTR\right)^{N-K+1} \\ & + \sum_{i=1}^{N-K} C_N^i\left(f_{DU}(i) \times \lambda_{DU}\right)^i \times \tau^{i-1}\left(\frac{\tau}{i+1} + MRT\right) \times C_{N-i}^{N-K+1-i}\left(f_{DD}(N-K+1-i)\right) \\ & \times \left(\lambda_{DD}MTTR\right)^{N-K+1-i} + \beta_{DU}\left(\frac{\tau}{2} + MRT\right) + \beta_D\lambda_{DD}MTTR \end{aligned} \tag{3.3}$$

where the first term and the second term in the equation represent the contributions from n-k+1 DU and DD failures respectively. The third term represents the contributions from all possible combinations of DU and DD failures that add up to n-k+1 failures. The fourth(last) term represents common cause failure (CCFs) contribution for both DU and DD failures.

The functions $f_{DU}(i)$ and $f_{DD}(N-K+1-i)$ are binary functions representing independent failure coefficient or CCFs and the aim is to present the PFD equation in an abbreviated format. The functions are:

Voting	Common cause contribution		Contribution from independent failures
1oo1	-		$\lambda_{DU} \cdot \tau/2$
1oo2	$\beta \cdot \lambda_{DU} \cdot \tau/2$	+	$\left[\lambda_{DU} \cdot \tau\right]^2/3$
2oo2	-		$2 \cdot \lambda_{DU} \cdot \tau/2$
1oo3	$C_{1oo3} \cdot \beta \cdot \lambda_{DU} \cdot \tau/2$	+	$\left[\lambda_{DU} \cdot \tau\right]^3/4$
2oo3	$C_{1oo3} \cdot \beta \cdot \lambda_{DU} \cdot \tau/2$	+	$\left[\lambda_{DU} \cdot \tau\right]^2$
3oo3	-		$3 \cdot \lambda_{DU} \cdot \tau/2$
1ooN; N=2,3,..	$C_{1ooN} \cdot \beta \cdot \lambda_{DU} \cdot \tau/2$	+	$\frac{1}{N+1} \cdot \left[\lambda_{DU} \cdot \tau\right]^N$
MooN, M<N; N=2,3,..	$C_{MooN} \cdot \beta \cdot \lambda_{DU} \cdot \tau/2$	+	$\frac{N!}{(N-M+2)! \cdot (M-1)!} \cdot \left[\lambda_{DU} \cdot \tau\right]^{N-M+1}$
NooN; N=1,2,3,..	-		$N \cdot \lambda_{DU} \cdot \tau/2$

Table 3.4: PDS simplified analytical formulas for PFD_{avg} of KooN architecture (adapted from Hauge et al., 2013).

$$f_{DU}(x) = \begin{cases} 1, & for \ \ x = 1 \\ (1-\beta), & for \ \ x > 1 \end{cases} \quad \text{and} \quad f_{DD}(x) = \begin{cases} 1, & for \ \ x = 1 \\ (1-\beta_D), & for \ \ x > 1 \end{cases}$$

where x is the number of failures of a mode, DU and DD respectively. Table 3.5 shows the PFD formulas for selected configurations based on Oliveira and Abramovitch (2010) generated using MAPLE software.

Innal et al. (2015a) also gives a generic formulation for the PFD_{KooN} considering situations where dangerous detected failures are repaired or not repaired instantaneously. The equation below takes into account the dangerous detected failures ($\lambda_{DD} > 0$) and common cause failures with $\lambda_{DU}^{(c)} = \beta\lambda_{DU}$ and $\lambda_{DD}^{(c)} = \beta_D\lambda_{DD}$. Table 3.6 shows use of this approach, also called the binomial approach and it is based on the following formulas:

$$\begin{aligned} PFD_{KooN} &= C_N^{N-K+1} \times \lambda_D^{(i)N-K+1} \times \prod_{i=1}^{N-K+1} MDT_{1ooi} + \lambda_{DU}^{(c)} \times \left(\frac{\tau}{2} + MRT\right) + \lambda_{DD}^{(c)} \times MTTR \\ &= C_N^{N-K+1} \times \prod_{i=1}^{N-K+1} \left(\lambda_{DU}^{(i)} \times \left(\frac{\tau}{i+1} + MRT\right) + \lambda_{DD}^{(i)} \times MTTR\right) + \lambda_{DU}^{(c)} \\ &\times \left(\frac{\tau}{2} + MRT\right) + \lambda_{DD}^{(c)} \times MTTR \end{aligned} \tag{3.4}$$

Architecture	PFD_{avg} **according to Oliveira and Abramovitch (2010)**
1oo1	$(1-\beta)\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+(1-\beta_D)\lambda_{DD}MTTR+\beta\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+\beta_D\lambda_{DD}MTTR$
1oo2	$(1-\beta)^2\lambda_{DU}^2\tau\left(\frac{\tau}{3}+MRT\right)+(1-\beta_D)^2\lambda_{DD}^2MTTR^2+2\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)\lambda_{DD}MTTR$ $+\beta\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+\beta_D\lambda_{DD}MTTR$
1oo3	$(1-\beta)^3\lambda_{DU}^3\tau^2\left(\frac{\tau}{4}+MRT\right)+(1-\beta_D)^3\lambda_{DD}^3MTTR^3+3\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)(1-\beta_D)^2\lambda_{DD}^2MTTR^2$ $+9(1-\beta)^2\lambda_{DU}^2\tau\left(\frac{\tau}{3}+MRT\right)\lambda_{DD}MTTR+\beta\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+\beta_D\lambda_{DD}MTTR$
2oo2	$2(1-\beta)\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+2(1-\beta_D)\lambda_{DD}MTTR+\beta\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+\beta_D\lambda_{DD}MTTR$
2oo3	$3(1-\beta)^2\lambda_{DU}^2\tau\left(\frac{\tau}{3}+MRT\right)+3(1-\beta_D)^2\lambda_{DD}^2MTTR^2+6\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)\lambda_{DD}MTTR$ $+\beta\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+\beta_D\lambda_{DD}MTTR$
The coefficient β is for CCF of DU failures and β_D is for DD failures. The functions $f_{DU}(x)$ and $f_{DD}(x)$ have been expressed in the equation in terms of $(1-\beta)$ and $(1-\beta_D)$.	

Table 3.5: Selected configurations for formula by Oliveira and Abramovitch (2010)

where $C_n^k=\frac{n!}{(n-k)!k!}$, $\lambda_{DU}^{(i)}=(1-\beta)\lambda_{DU}$ and $\lambda_{DD}^{(i)}=(1-\beta_D)\lambda_{DD}$.

Architecture	PFD_{avg} **according to Innal et al. (2015a)**
1oo1	$(1-\beta)\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+(1-\beta_D)\lambda_{DD}MTTR+\beta\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+\beta_D\lambda_{DD}MTTR$
1oo2	$\left((1-\beta)\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+(1-\beta_D)\lambda_{DD}MTTR\right)\left((1-\beta)\lambda_{DU}\left(\frac{\tau}{3}+MRT\right)\right.$ $\left.+(1-\beta_D)\lambda_{DD}MTTR\right)+\beta\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+\beta_D\lambda_{DD}MTTR$
1oo3	$\left((1-\beta)\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+(1-\beta_D)\lambda_{DD}MTTR\right)\left((1-\beta)\lambda_{DU}\left(\frac{\tau}{3}+MRT\right)\right.$ $\left.+(1-\beta_D)\lambda_{DD}MTTR\right)\left((1-\beta)\lambda_{DU}\left(\frac{\tau}{4}+MRT\right)+(1-\beta_D)\lambda_{DD}MTTR\right)$ $+\beta\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+\beta_D\lambda_{DD}MTTR$
2oo2	$2\cdot(1-\beta)\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+2\cdot(1-\beta_D)\lambda_{DD}MTTR+\beta\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+\beta_D\lambda_{DD}MTTR$
2oo3	$3\cdot\left((1-\beta)\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+(1-\beta_D)\lambda_{DD}MTTR\right)\left((1-\beta)\lambda_{DU}\left(\frac{\tau}{3}+MRT\right)+\right.$ $\left.(1-\beta_D)\lambda_{DD}MTTR\right)+\beta\lambda_{DU}\left(\frac{\tau}{2}+MRT\right)+\beta_D\lambda_{DD}MTTR$
The coefficient β for CCF is the same for both detected and undetected $\beta_{DU}=\beta_{DD}=\beta$.	

Table 3.6: Selected configurations for formula by Innal et al. (2015a)

3.1.5 Summary on the different analytical formulas

The tables in this chapter contain formulas for calculating the PFD_avg for the selected configurations. The formulas are based on almost the same assumptions. We will just mention some main differences:

- The IEC 61508 standard, equation 3.3 and equation 3.4 use the standard beta factor model for common cause failures.
- The PDS method is based on the multiple beta factor.
- Equation 3.4 is a generalization of the IEC 61508 standard. The main difference is on the MDT formulation. Actually, the standard use the complete failure rates, whereas in equation 3.4 independent failure rates are used instead.
- The difference between equations 3.3 and 3.4 is that the second one only considers failure sequences containing DU failures or DU failures ending by a DD failure. The first equation consider all failure sequences except those starting with a DD failure.
- IEC 61508 and equations 3.3 and 3.4, give almost the same results.

Chapter 4

Partial and Imperfect testing

4.1 Analytical formulas for imperfect testing

Testing of SIS is categorized into two different types namely the diagnostic testing also called automatic self test which involves constant sending of signals to detect abnormalities in conditions against the pre-programmed norm of components and functional testing which is carried out at predetermined intervals. Diagnostics and functional testing are meant to reveal dangerous detected and dangerous undetected failures respectively. The fraction of failures detected by diagnostic testing is called diagnostic coverage while the fraction of hidden failures detected during a functional testing is termed proof test coverage (Rausand, 2014). The split of DU failures into two parts based on failures revealed during a proof test is expressed in the equation:

$$\lambda_{DU} = \lambda_{DU}^{(r)} + \lambda_{DU}^{(nr)} \tag{4.1}$$

where $\lambda_{DU}^{(r)}$ is the rate of DU failures that can be revealed during proof testing and $\lambda_{DU}^{(nr)}$ is the rate of DU failure that cannot be revealed by proof testing. The proof test coverage is therefore illustrated by the formula:

$$PTC = \frac{\lambda_{DU}^{(r)}}{\lambda_{DU}} \tag{4.2}$$

The rate of revealed and non-revealed failures expressed in terms of PTC and the DU failure rate is shown below:

$$\lambda_{DU}^{(r)} = PTC \cdot \lambda_{DU} \quad \text{and} \quad \lambda_{DU}^{(nr)} = (1 - PTC) \cdot \lambda_{DU}$$

From equation 3.7 above we can conclude that the time-dependent probability of failure on demand, PFD(t) of a channel can be written as:

$$PFD(t) = PFD^{(r)}(t) + PFD^{(nr)}(t) \tag{4.3}$$

The shape of the time dependent probability graph is shown in figure 4.1

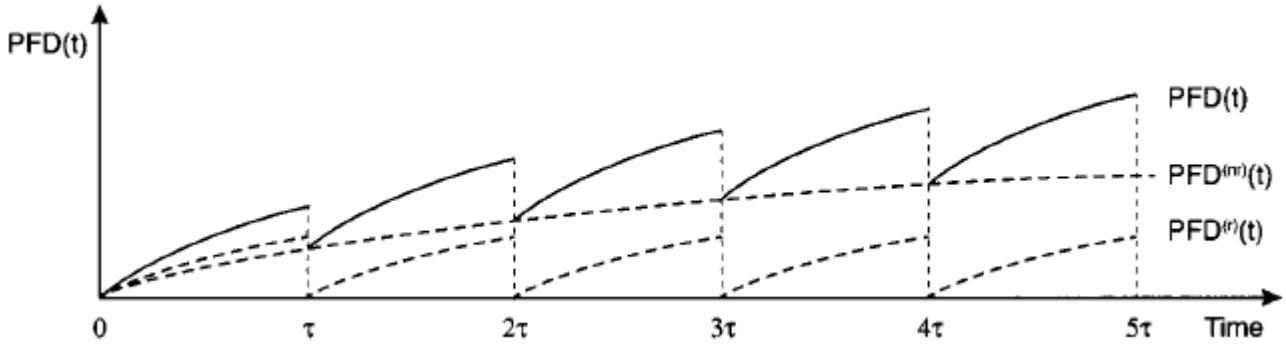


Figure 4.1: The PFD(t) of a channel with imperfect proof-testing (adapted from Rausand, 2014).

4.2 IEC 61508 approach

The standard IEC 61508 gives a brief explanation on how to model effects of non-perfect proof tests. These are situations where faults are detected only when the safety function is required or faults are found during overhaul of the equipment. If the faults are not detected in the normal proof test interval τ with proof test coverage PTC, then the faults are found by the other means stated above represented by (1-PTC) with T as the expected time between demands on the system. These times will influence the downtime hence the PFD.

$$t_{CE} = \frac{\lambda_{DU}(PTC)}{\lambda_D}\left(\frac{\tau}{2} + MRT\right) + \frac{\lambda_{DU}(1-PTC)}{\lambda_D}\left(\frac{T}{2} + MRT\right) + \frac{\lambda_{DD}}{\lambda_D} MTTR \tag{4.4}$$

Therefore the PFD_{avg} given by $\lambda_{D,G} \cdot t_{GE}$ for a 1oo1 channel is:

$$PFD_{avg} \approx PTC \cdot \lambda_{DU}\left(\frac{\tau}{2} + MRT\right) + (1 - PTC) \cdot \lambda_{DU}\left(\frac{T}{2} + MRT\right) + \lambda_{DD} \cdot MTTR \tag{4.5}$$

The PFD_{avg} formula for a 1oo2 configuration is further expressed and clarified in IEC-61508 (2009); Oliveira (2009). The two identical channels with imperfect repair and overhaul is considered to have a D-fault in 5 different ways namely (i) Two DU faults due to CCF, (ii) Two DD faults due to CCF, (iii) Two independent DU revealed faults in same proof test interval, (iv) Two independent non-revealed faults in the overhaul period and (v) One independent DU revealed fault and one independent DU non-revealed fault in the same proof test interval (Rausand, 2014).

The IEC PFD_{avg} for a 1oo2 independent and identical channels is:

$$PFD_{avg} = \lambda_{D,G} \cdot t_{GE} \quad = \quad 2(\lambda_D)^2 \cdot t_{CE} \cdot t_{GE}$$

Integrating the formulas together gives:

$$\begin{aligned} PFD_{avg}^{(1oo2)} = \left[(1-\beta)\lambda_{DU} + (1-\beta_D)\lambda_{DD}\right]^2 \cdot t_{CE} \cdot t_{GE} + PTC \cdot \beta\lambda_{DU}\left(\frac{\tau}{2} + MRT\right) \\ + (1 - PTC) \cdot \beta\lambda_{DU}\left(\frac{T}{2} + MRT\right) + \beta_D\lambda_{DD} \cdot MTTR \end{aligned} \tag{4.6}$$

where $t_{CE} = \frac{\lambda_{DU}^r}{\lambda_D}\left(\frac{\tau}{2} + MRT\right) + \frac{\lambda_{DU}^{nr}}{\lambda_D}\left(\frac{T}{2} + MRT\right) + \frac{\lambda_{DD}}{\lambda_D} MTTR$

and $t_{GE} = \frac{\lambda_{DU}^r}{\lambda_D}\left(\frac{\tau}{3} + MRT\right) + \frac{\lambda_{DU}^{nr}}{\lambda_D}\left(\frac{T}{3} + MRT\right) + \frac{\lambda_{DD}}{\lambda_D} MTTR$

4.3 PDS method

The PDS method handbook uses the term critical safety unavailability (CSU) to quantify the loss of safety. The handbook defines CSU as the probability that a component or system will fail to automatically carry out a successful safety action on the occurrence of a hazardous event.

$$CSU = PFD + DTU + P_{TIF} \tag{4.7}$$

Figure 4.2 shows the various contributors to loss of safety.

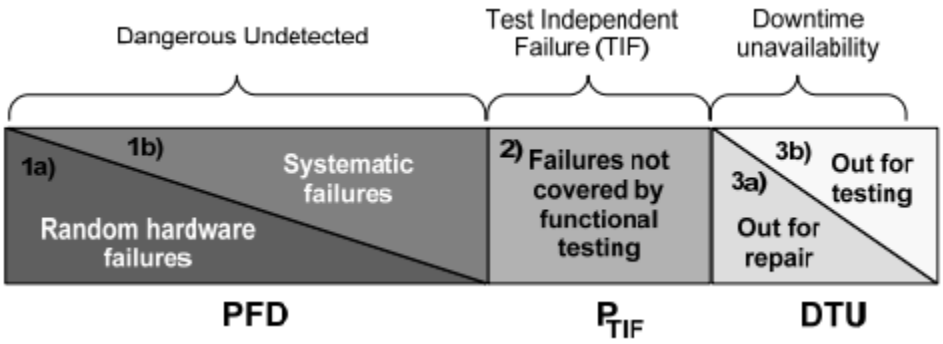


Figure 4.2: Loss of safety contributors (adapted from Hauge et al., 2013).

4.3.1 Probability of Test Independent Failures

Test Independent Failures (TIF) are failures that are not revealed by the proof/functional tests or by automatic self-test but only during a true demand. The PDS accounts for these failures by adding the P_{TIF} which it defines as "the Probability that the component/system will fail to carry out its intended function due to a latent failure not detectable by functional testing "(Hauge et al., 2013).

In the PDS method handbook, P_{TIF} represents the probability of a single component that has just been functionally tested to fail on demand irrespective of the interval of functional testing. The TIF contribution to loss of safety of redundant components voted MooN (M<N) can be calculated with the general formula: $C_{MooN} \cdot \beta \cdot P_{TIF}$ where C_{MooN} are the same as those used to calculate the PFD.

Voting	TIF contribution to CSU for MooN voting
1oo1	P_{TIF}
1002	$\beta \cdot P_{TIF}$
MooN; M < N	$C_{MooN} \cdot \beta \cdot P_{TIF}$
Noon; N = 1,2,...	$N \cdot P_{TIF}$

Table 4.1: Formulas for P_{TIF} for various voting logic (adapted from Hauge et al., 2013).

4.3.2 Incorporating PTC into PFD Formulas

The PDS handbook also suggests modeling the PTC into PFD formulas as an alternative to the P_{TIF} in considering imperfect testing. To achieve this, the rate of DU failures is divided into failures detected during testing with test interval τ and failures not revealed during testing with interval T which could be a complete component overhaul interval or lifetime of the equipment.

The PFD for a 1oo1 voting considering imperfect testing using the PTC is given as:

$$PFD_{1oo1} = PTC \cdot \left(\lambda_{DU} \cdot \frac{\tau}{2}\right) + (1 - PTC) \cdot \left(\lambda_{DU} \cdot \frac{T}{2}\right) \tag{4.8}$$

Generalizing the PFD formula for any NooN configuration using the standard approximation gives:

$$PFD_{NooN} = PTC \cdot \left(N \cdot \lambda_{DU} \cdot \frac{\tau}{2}\right) + (1 - PTC) \cdot \left(N \cdot \lambda_{DU} \cdot \frac{T}{2}\right) \tag{4.9}$$

The common cause contribution for a MooN configuration where a single failure or a CCF is sufficient for the system to fail is given in equation 4.10 below. Note however that consideration of independent failures combinations gives more complex equations.

$$PFD_{MooN} = PTC \cdot \left(C_{MooN} \cdot \beta \cdot \lambda_{DU} \cdot \frac{\tau}{2}\right) + (1 - PTC) \cdot \left(C_{MooN} \cdot \beta \cdot \lambda_{DU} \cdot \frac{T}{2}\right) \tag{4.10}$$

For a 1oo2 configuration, the contribution by a combination of two independent failures detectable during the function test is added to the first term and the combination of two independent failures detectable after time T is added to the second term. Also a contribution from the combination of one failure detectable in during the function test and one failure detectable after time T. The system operates as a 1oo1 system for a period when a single failure occurs which is not detected before time T. As shown in equation 4.11, the second and fourth terms therefore represent the extensions of two independent failures of same type while the last term corresponds to a combination of one failure undetectable the function test and one detectable

(Hauge et al., 2013).

$$PFD_{1oo2} = \beta \cdot PTC \cdot \lambda_{DU} \cdot \frac{\tau}{2} + \frac{(PTC \cdot \lambda_{DU} \cdot \tau)^2}{3} + \beta \cdot (1 - PTC) \cdot \lambda_{DU} \cdot \frac{T}{2} + \frac{\left((1 - PTC) \cdot \lambda_{DU} \cdot T\right)^2}{3} + 2 \cdot \left(PTC \cdot \lambda_{DU} \cdot \frac{\tau}{2}\right)\left((1 - PTC) \cdot \lambda_{DU} \cdot \frac{T}{2}\right) \tag{4.11}$$

For the 2oo3 configuration, the last term in the equation corresponds to one failure that is not detectable during function test but occurring before time and a CCF of the two remaining components, occurring within the same test interval τ but there are also other possibilities to consider. Note that the $C_{2oo3} = 2$.

$$PFD_{2oo3} = 2 \cdot \beta \cdot PTC \cdot \lambda_{DU} \cdot \frac{\tau}{2} + (PTC \cdot \lambda_{DU} \cdot \tau)^2 + 2 \cdot \beta \cdot (1 - PTC) \cdot \lambda_{DU} \cdot \frac{T}{2} + \left((1 - PTC) \cdot \lambda_{DU} \cdot T\right)^2 + 3 \cdot \left((1 - PTC) \cdot \lambda_{DU} \cdot \frac{T}{2}\right)\left(\beta \cdot PTC \cdot \lambda_{DU} \cdot \frac{\tau}{2}\right) \tag{4.12}$$

Regarding the equation 4.12, its last part refers to independent failure which can not be detected by proof test followed by a CCF that can be detected by the proof test. We wonder why the combination starting with failure detected by proof test followed by a CCF that can not be detected by proof test is not considered in the formula. The contribution of this missing combination may be more important than the first one. This contribution is expressed in the equation 4.13 below:

$$3 \cdot \left((PTC) \cdot \lambda_{DU} \cdot \frac{\tau}{2}\right)\left(\beta \cdot (1 - PTC) \cdot \lambda_{DU} \cdot \frac{T}{2}\right) \tag{4.13}$$

4.3.3 Discussion on the use of PTC and P_{TIF}

There are cases where the use of PTC is more appropriate than the use of P_{TIF}. PTC is best appropriate where failures are introduced during operation or develop over time and the standard functional test is not planned to reveal such failures. IEC 61508 standard also suggests using PTC for some cases with imperfect testing. The P_{TIF} is more appropriate to use than the PTC when considering systematic design related failures which have been present from the onset and have constant probability. Hauge and Onshus (2006) contains some P_{TIF} general values assigned to some topside components and these data were based on expert judgment.

There are some cases where neither the P_{TIF} nor the PTC are suitable for modeling imperfect

testing. Examples are human error in between tests during operation which are detected during the next test and factored as part of the λ_{DU} (Hauge et al., 2013).

4.4 The Markovian approach to partial testing

The Markov model helps to catch the dynamism of a system under study. The multi-phase Markov model as explained in detail in chapter 5 is more suitable for modeling the behaviour of periodically tested systems. This is applied and result compared with other formulas for the 1oo2 configuration. The behaviour of two identical channels working in redundancy and subjected to periodic partial and full tests is shown by the model in figure 4.3. CCF is not considered in this model for simplicity. In addition, the sate 4 is only replicated for clarity purpose (Innal et al., 2015b).

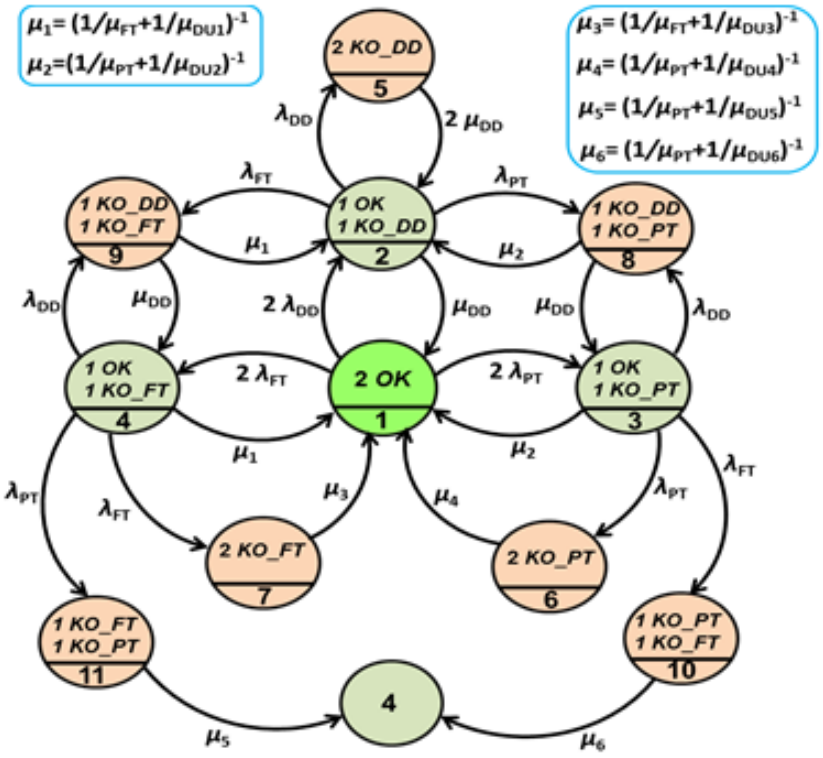


Figure 4.3: Approximate Markov model for 1oo2 system based on multi-phase Markov.

Some of the transitions in figure 4.3 like μ_{DU3} (state 7 to state 1), μ_{DU4} (state 6 to state 1), μ_{DU5} (state 11 to state 4) and μ_{DU6} (state 10 to state 4) are related to specific failure sequences

as depicted in figure 4.4. The repair time are neglected in order to only compute the unrevealed sojourn times. The quantities μ_{FT} and μ_{PT} are respectively $1/MRT_{FT}$ and $1/MRT_{PT}$. MRT_{FT} and MRT_{PT} are respectively the mean restoration time for failures revealed by full tests and failures revealed by partial test. μ_1 is the reciprocal of the unrevealed sojourn time (in state 4:1 FT failure) and repair time due to failures that detected by full tests: $\mu_1 = 1/[\tau/2 + MRT_{FT}]$.

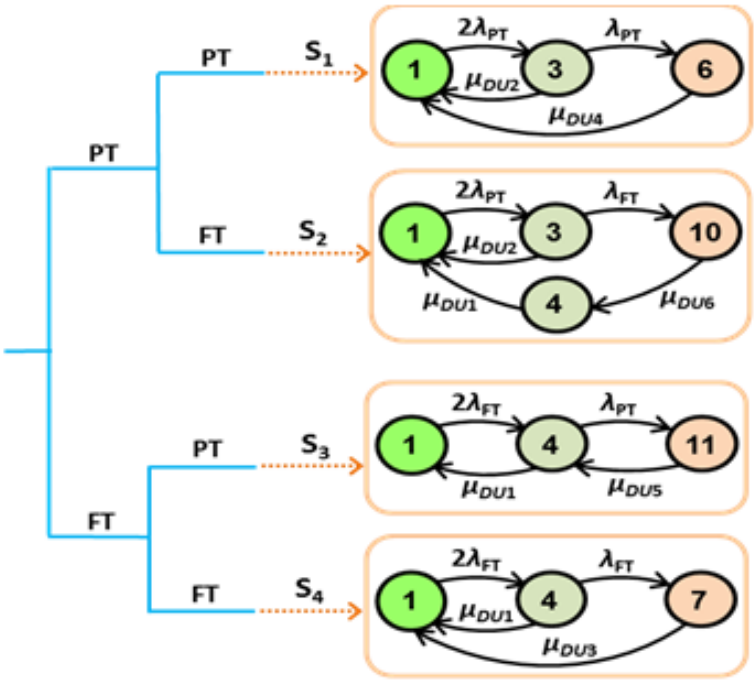


Figure 4.4: Undetected failure sequences for 1oo2 system.

S1,S2,S3,S4 are the various failure sequences and are summarized below:

- **Sequence 1 (μ_{DU4}):** This quantity is the reciprocal of the mean sojourn time in state 6 (two successive PT failures) and the relationship is shown in figure 4.3.

- **Sequence 2 (μ_{DU6}):** This quantity is the reciprocal of the mean sojourn time in state 10 (PT failure is observed first, and then a FT failure takes place).

- **Sequence 3 (μ_{DU5}):** This quantity is the reciprocal of the mean sojourn time in state 11 (FT failure is observed first, and then a PT failure takes place). For this scenario, the establishment of the analytical solution from the corresponding multi-phase Markov model is quite difficult. Hence, in the following we will adopt a different approach. Actually,

the mean unavailability to the combination of FT and PT failures (whatever their order of occurrence: states 10 and 11) could be easily obtained.

- **Sequence 4 (μ_{DU3}):** This quantity is the reciprocal of the mean sojourn time in state 7 (two successive FT failures).

The complete process of these repair rates is given in (Innal et al., 2015b). The PFD_{avg} for the 1oo2 system based on the Markov model in figure 4.3 can be calculated as:

$$PFD_{avg}^{1oo2} = \sum_{i=5}^{11} P_i(\infty) \tag{4.14}$$

The steady-state probabilities based on substitutions and approximations for the different states are given as:

$$P_1(\infty) \approx 1; P_2(\infty) \approx \frac{2\lambda_{DD}}{\mu_{DD}}; P_3(\infty) \approx \frac{2\lambda_{PT}}{\mu_2}; P_4(\infty) \approx \frac{2\lambda_{FT}}{\mu_1}; P_5(\infty) \approx \frac{2\lambda_{DD}^2}{2\mu_{DD}^2}; P_6(\infty) \approx \frac{2\lambda_{PT}^2}{\mu_2\mu_4};$$
$$P_7(\infty) \approx \frac{2\lambda_{FT}^2}{\mu_1\mu_3}; P_8(\infty) \approx \frac{2\lambda_{PT}\lambda_{DD}}{\mu_2\mu_{DD}}; P_9(\infty) \approx \frac{2\lambda_{FT}\lambda_{DD}}{\mu_1\mu_{DD}}; P_{10}(\infty) \approx \frac{2\lambda_{PT}\lambda_{FT}}{\mu_2\mu_6}; P_{11}(\infty) \approx \frac{2\lambda_{FT}\lambda_{PT}}{\mu_1\mu_5};$$

Therefore the PFD_{avg} can be calculated thus:

$$PFD_{avg}^{1oo2} = \frac{2\lambda_{PT}}{\mu_1} \cdot \left(\frac{\lambda_{FT}}{\mu_6} + \frac{\lambda_{PT}}{\mu_4} + \frac{\lambda_{DD}}{\mu_{DD}}\right) + \frac{2\lambda_{FT}}{\mu_1} \cdot \left(\frac{\lambda_{FT}}{\mu_3} + \frac{\lambda_{PT}}{\mu_5} + \frac{\lambda_{DD}}{\mu_{DD}}\right) + \frac{\lambda_{DD}^2}{\mu_{DD}^2} \tag{4.15}$$

The mean downtime MDT_{1oo2} is given by $PFD_{avg}^{1oo2}/PFH_{1oo2}$ which is expressed as:

$$MDT_{1oo2} \approx \frac{P_3(\infty)\left(\frac{\lambda_{FT}}{\mu_6} + \frac{\lambda_{PT}}{\mu_4} + \frac{\lambda_{DD}}{\mu_{DD}}\right) + P_4(\infty)\left(\frac{\lambda_{FT}}{\mu_3} + \frac{\lambda_{PT}}{\mu_5} + \frac{\lambda_{DD}}{\mu_{DD}}\right)}{\lambda_D\left[P_3(\infty) + P_4(\infty)\right]} \tag{4.16}$$

The MDT_{1oo2} given in Oliveira (2009); Rausand (2014); IEC-61508 (2009) expressed in equation in section 4.2 above can be re-written as:

$$MDT_{1oo2} \approx \frac{1}{\lambda_D}\left(\frac{\lambda_{FT}}{\mu_3} + \frac{\lambda_{PT}}{\mu_4} + \frac{\lambda_{DD}}{\mu_{DD}}\right) \tag{4.17}$$

Equations 4.15 and 4.16 would be the same if $\mu_5 = \mu_4$ and $\mu_6 = \mu_3$. Therefore in the PFD_{avg}^{1oo2}

formula given as $2\lambda_D^2 MDT_{1oo1} MDT_{1oo2}$ where MDT_{1oo1} and MDT_{1oo2} are given as t_{CE} and t_{GE} respectively, only the MDT_{1oo2} is incorrect as it is given by equation 4.16. This clearly proves that the formula for 1oo2 expressed in IEC-61508 (2009); Rausand (2014), the PDS approach in Hauge and Onshus (2006) and the generalization formula in Oliveira (2009) are only partially correct and results will be conservative as they do not consider in detail all possible failure scenarios in the process. However the consideration of CCF events would attenuate this conservativeness.

4.5 Other authors approaches to partial tests

Partial tests have been considered and included in some of the PFD calculation equations given by different authors. This approach considers a number of partial tests within a full proof test interval. These partial tests could be periodic or non-periodic, however the periodic partial test is the most commonly assumed and considered.

- Brissaud et al. (2012) introduces non-approximate equations for PFD assessment for an MooN architecture subject to partial and full tests. Here the partial test may occur at any time instants within a full test time interval. The RBD below is used to derive the analytical expression of the PFD_{avg}. CCFs are not considered in the equation. The behavior of any element constituting the MooN system is given by the RBD of figure 4.5.

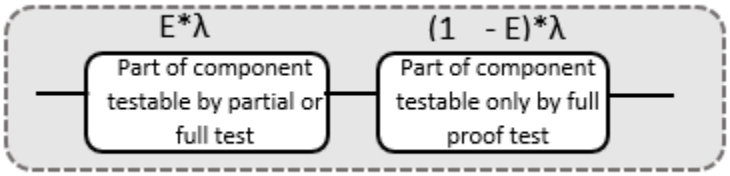


Figure 4.5: RBD of any component of a system subjected to partial tests.

The survival function equation from the RBD is:

$$R(t) = e^{-E\cdot\lambda\cdot(t-t_{i-1})} \cdot e^{-(1-E)\lambda\cdot t} \quad \text{for} \quad t_{i-1} \leqslant t < t_i$$

$$R(t) = e^{E\cdot\lambda\cdot t_{i-1}} \cdot e^{-\lambda\cdot t} \quad \text{for} \quad t_{i-1} \leqslant t < t_i \tag{4.18}$$

where E = θ (partial test coverage) and t_i is the time instant to execute the ith partial test.

The equation for the PFD_{avg} considering partial tests derived from mathematical operations using Fubini's theorem is therefore given as:

$$PFD_{avg}^{(p)} = 1 - \sum_{x=M}^{N} \left[S(M,N,x) \cdot \frac{1 - e^{-x \cdot \lambda \cdot T_0}}{x \cdot \lambda \cdot T_0} \cdot \frac{1}{n} \cdot \sum_{i=1}^{n} \left[e^{-x \cdot (1-E) \cdot \lambda \cdot (i-1) \cdot T_0} \right] \right] \tag{4.19}$$

where $S(M,N,x)$ is a combination of failures of the components from M to N represented by the equation:

$$S(M,N,x) = \sum_{k=M}^{x} \left[\binom{N}{x} \cdot \binom{x}{k} \cdot (-1)^{x-k} \right]$$

for x = M,...,N and periodic partial test period $T_0 = \tau / n$ and τ is the full test time interval. E in formula 4.19 is the efficiency of partial tests which represents the test coverage denoted by θ in most other cases.

- Jin and Rausand (2014) shows how partial tests affect the reliability of a SIS. The formulas can be applied to both periodic and non-periodic partial tests and include both partial and full proof testing. CCF is not considered in the selected case. The figure in 4.6 shows the RBD for a KooN system subject to partial testing where λ_a are type 'a' failure rates revealed by partial testing such that $\lambda_a = \theta\lambda$ and λ_b are type 'b' failure rates not revealed by partial testing such that $\lambda_b = (1-\theta)\lambda$. The corresponding equation is given as:

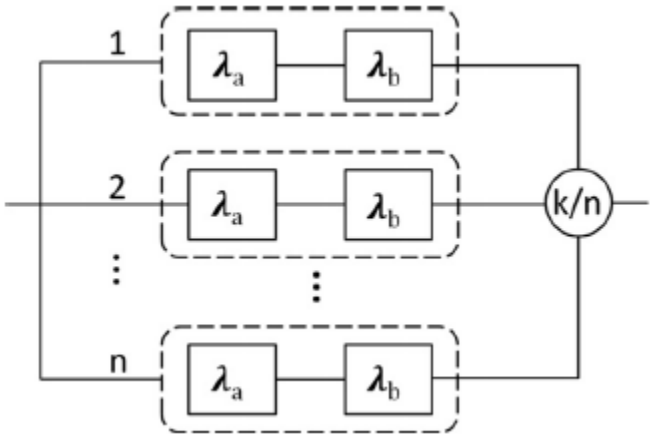


Figure 4.6: RBD of a koon system subject to partial tests. (adapted from Jin and Rausand, 2014).

$$PFD_{avg}^{(p)} \approx \frac{1}{m}\sum_{i=1}^{m}\sum_{j=0}^{n-k}\binom{n}{j}\left((i-1)\lambda_b\widetilde{\tau}\right)^j \cdot \frac{(n-j)!(\lambda\widetilde{\tau})^{n-j-k+1}}{(n-j-k+2)!(k-1)!} + \frac{1}{m}\sum_{i=1}^{m}\sum_{j=n-k+1}^{n}\binom{n}{j}\left((i-1)\lambda_b\widetilde{\tau}\right)^j \tag{4.20}$$

where $\widetilde{\tau}$ is the partial test interval such that $\widetilde{\tau} = \tau/m$. τ is the full proof test interval and m partial tests are performed in τ. The letter j is the number of type b failures at t_{i-1} and j > n - k.

- Chebila and Innal (2015) gives generalized analytical expressions for PFD and PFH taking partial stroke testing into account similar to Hui Jin's approach.

$$PFD_{KooN} \approx \frac{\binom{n}{n-k+1}}{(n-k+2)\cdot\lambda_{DU}\cdot m\cdot T_{ST}} \cdot \sum_{j=0}^{m-1}\left[\left(\left(\lambda_{DU}+\lambda_{PT}\cdot j\right)\cdot T_{ST}\right)^{n-k+2} - \left(\lambda_{PT}\cdot j\cdot T_{ST}\right)^{n-k+2}\right] \tag{4.21}$$

where T_{ST} is the partial test interval. The relationship with the proof test interval is $T_1 = m\cdot T_{ST}$. λ_{PT} is the proof test failure rate.

4.6 Partial tests impact on PFD calculation

A table with results based on the different general formulas in section 4.5 is presented. The results are compared to see the validity of each method proposed. The relationship between the test interval and m is $\tau_{PT} = \tau/m$. Therefore, when m = 1, there is no partial test or the partial test is equal to zero hence the full proof test and when m=4, then the partial test is performed 4 times within the test interval. If the test interval is 1year (8760 hrs) then the partial test is implemented every 3 months (2190 hrs). The inspection of table 4.2 shows that the different sets of results are close to each other. In addition, we can see that when the partial test is implemented, the PFD of the system reduces. This shows the importance of implementing partial tests and the positive impact it has on the PFD of a system.

KooN	PST Strategy	Without CCF		
		Eq. 4.19	Eq. 4.20	Eq. 4.21
1oo1	m = 1	1.09E-2	1.10E-2	1.10E-2
	m = 4	6.00E-3	6.02E-3	6.02E-3
	m = 12	4.91E-3	4.93E-3	4.93E-3
1oo2	m = 1	1.57E-4	1.60E-4	1.60E-4
	m = 4	4.44E-5	4.48E-5	4.48E-5
	m = 12	3.07E-5	3.09E-5	3.09E-5
1oo3	m = 1	2.58E-6	2.63E-6	2.63E-6
	m = 4	3.75E-7	3.72E-7	3.72E-7
	m = 12	1.76E-7	2.18E-7	2.18E-7
2oo2	m = 1	2.16E-2	2.19E-2	2.19E-2
	m = 4	1.20E-2	1.21E-2	1.20E-2
	m = 12	9.79E-3	9.88E-3	9.86E-3
2oo3	m = 1	4.67E-4	4.80E-4	4.80E-4
	m = 4	1.32E-4	1.34E-4	1.34E-4
	m = 12	9.17E-5	9.29E-5	9.27E-5

Table 4.2: PFD_{avg} of selected configurations for Partial test formulas comparison

Chapter 5

Verification of Analytical formulas for testing

5.1 Phased Markov for periodically tested components

Reliability assessment methods like the RBD and FTA only consider a system's functioning and failed state. To be able to model other states of a system like the degraded state, Markov analysis is ideal. Markov analysis considers different system states, transition between the states and the rate at which the transitions occur. The Markov property is a stochastic process where the future state only depends on the present, and not the past.

$$Pr(X(t+s) = j \mid X(t) = i, X(u) = i, X(u) = x(u), 0 \leq u < s) \;=\; Pr(X(t+s) = j \mid X(s) = i$$

The standard Markov chain has some limitations which makes it difficult to correctly model the behavior of periodically and partially tested SIS with several periods. This test behavior requires the use of a multi-phase Markov Chain (Dutuit et al., 2008; Mechri et al., 2013). IEC 61508 suggests a multi-phased Markovian approach to model such systems. Multi-phase Markov chains are appropriate for modeling changes in structure of the states at known instants or when the state of some parts of the system are known at some instants. The latter case is proof testing which creates a new phase in the Markov chain evolution. Figure 5.1 shows that the normal Markov process on the upper part of the figure represents a single component which can fail

(W to DU) or under repair (R to W). Repairs can not be started within a test interval, therefore no transition from DU to R (IEC-61508, 2009). When a test is performed, a repair is started if a failure has occurred (DU to R), the component remains working if it was in a good functioning state (W to W) and in the very hypothetical case that a repair started at the previous test is not finished, remains under repair (R to R). The linking matrix may be used to calculate the initial conditions at the beginning of state $i+1$ from the probabilities of the states at the end of test i. Refer to IEC-61508 (2009); Rausand (2014) for the mathematical equations and representations. The linking matrices represent the states before and after each test.

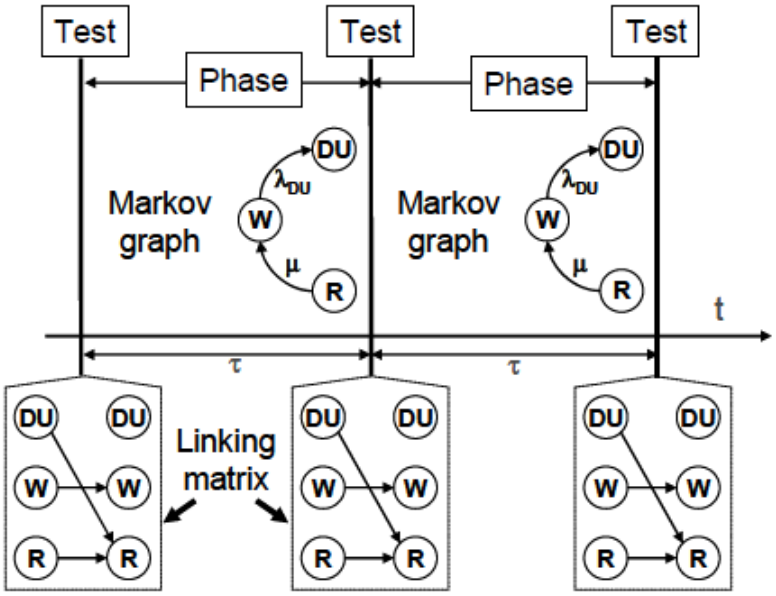


Figure 5.1: Principle of the multiphase Markovian modeling (adapted from IEC-61508, 2009) part 6.

5.1.1 Application of multi-phase Markov by analysis of a 1oo1 system

A component which is tested periodically may have three states due to DU failures: working, DU failure and repair. In addition to the continuous time Markov model between test times, one has to be able to consider the tests ability to detect failure just at the starting of that test. This may be used to consider imperfect repairs and maintenance.

The Markov model in figure 5.2 has four states. State 1 is a fully working state; State 2 has detected failure with rate λ_{DD} and the repair rate of the DD failure is given as μ_{DD}; State 3

represents an undetected failure occurring with rate λ_{DU}; State 4 represents the repair of the unrevealed failure mode after being detected by test. States 4 and 2 could be the same but have been represented separately since repair times of DD and DU failures could be different. Note that the transition between states 3 and 4 is a deterministic and instantaneous one which only reflects the failure detection when a test is performed.

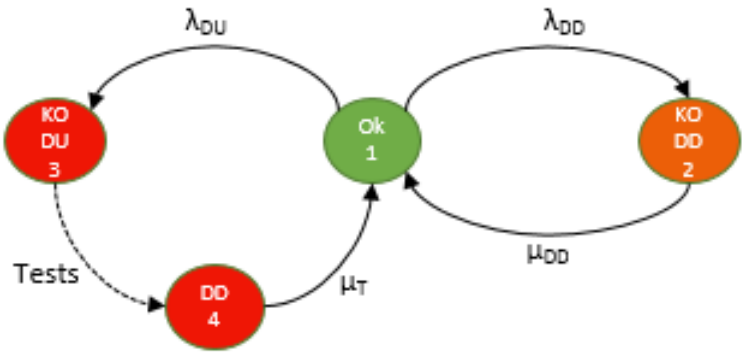


Figure 5.2: 1oo1 Markov model for a periodically tested element.

The behavior of elements tested periodically over several test periods can be correctly rendered by a multi-phase Markov model (regenerative Markov process) meaning that between each two consecutive tests (phase), the behavior is given by a classical Markov model as shown below (Mechri et al., 2013):

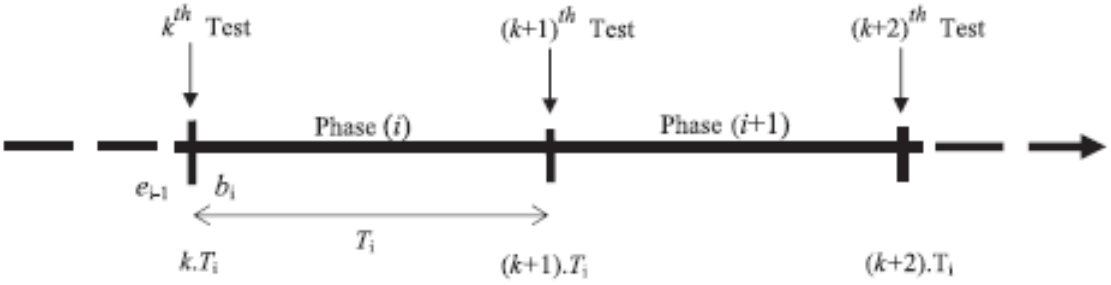


Figure 5.3: Principle of multi-phase Markov modeling.

This means that the probabilities of the states at the beginning (b_i) of the phase i are deduced from the probabilities obtained at the end e_{i-1} of the phase *i-1*. The state probabilities at the beginning of each phase for the model above are: $P_1(b_i) = P_1(e_{i-1})$ (the component is operational before the test and stays operational after the test); $P_2(b_i) = P_2(e_{i-1})$ (when the test starts, the component is already under repair and stays in that condition); $P_3(b_i) = 0$ (due to

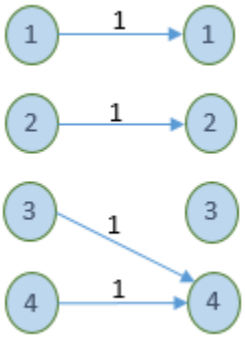


Figure 5.4: Passage matrix at the $(K+1)\cdot T_i$.

the test, the failure is no longer hidden and therefore the probability of being in state 3 after the test is 0); $P_4(b_i) = P_4(e_{i-1}) + P_3(e_{i-1})$ (the component is under repair after the test). This probability redistribution procedure can be reflected by a passage matrix M between two contiguous phases (phase *i* to phase *i+1*) that specifies the probability that the state *j* at the end of phase *i* will give to the state *k* at the beginning of phase *i+1*. For the model given in figure 5.2, we get:

$$P(b_i) = P_(e_{i-1}) \cdot M = [P_1(e_{i-1}) \quad P_2(e_{i-1}) \quad P_3(e_{i-1}) \quad P_4(e_{i-1})] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.1}$$

where the passage (transition) matrix M is used only at each time KTi such that the sum of each row is equal to one and all the coefficients m_{ij} are equal to or greater than zero. The time dependent states probabilities by applying Kolmogorov's equation:

$$\frac{dP(t)}{dt} = P(t) \cdot A \tag{5.2}$$

where the transition rate matrix based on the Markov model in figure 5.2 is given below:

$$A = \begin{bmatrix} -(\lambda_{DD} + \lambda_{DU}) & \lambda_{DD} & \lambda_{DU} & 0 \\ \mu_{DD} & -\mu_{DD} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \mu_T & 0 & 0 & -\mu_T \end{bmatrix}$$

Given the state probabilities at the beginning of each phase i $P(b_i)$, solving equation 5.2 using the exponential method gives:

$$P(t) = P(b_i) \cdot e^{A \cdot t} \tag{5.3}$$

By recurrence of equation 5.1 and 5.3, the state probabilities at the beginning of each phase i are given by equation 5.4

$$P(b_i) = P(0) \cdot \left(e^{A \cdot T} \cdot M\right)^{i-1} \tag{5.4}$$

And finally by merging equations 5.3 and 5.4, we get

$$P(t) = P(0) \cdot \left(e^{A \cdot T} \cdot M\right)^{i-1} \cdot e^{A \cdot [t-(i-1) \cdot T]} \tag{5.5}$$

Where t>0

$i = lnt(t/T) + 1$ (lnt(x) gives the integer part of x)

$P(0) = P(b_1) = [1\ 0\ 0\ 0]$.

The $PFD(t)$ for a periodically tested component based on this 1oo1 architecture represented in the Markov model is give n as:

$$\begin{aligned} PFD(t) = P_2(t) + P_3(t) + P_4(t) \ &= \ P(t) \cdot \mathbf{F} \\ &= P(0) \cdot \left(e^{A \cdot T} \cdot M\right)^{i-1} \cdot e^{A \cdot [t-(i-1) \cdot T]} \cdot \mathbf{F} \end{aligned} \tag{5.6}$$

where **F** is a column vector used to sum up the failed state probabilities:

$$F = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

5.1.2 Multi-phase Markov result comparison with partial test consideration

To be able to compare the Markov model with the analytical formulas, the Markov model for a 1oo1 element subject to partial and full test is shown in figure 5.5 and the analysis is based on the principles explained in the first subsection. State 1 represents the functioning state of

the system. If there is a DD failure, the system goes to state 2. State 3 represents a DU failure detectable by partial test and state 4 represents a DU failure detectable by full proof test. State 5 is when the DU failure in state 3 is detected during the partial test before being repaired to a functioning state (1), likewise state 6 for the failure detection by full test.

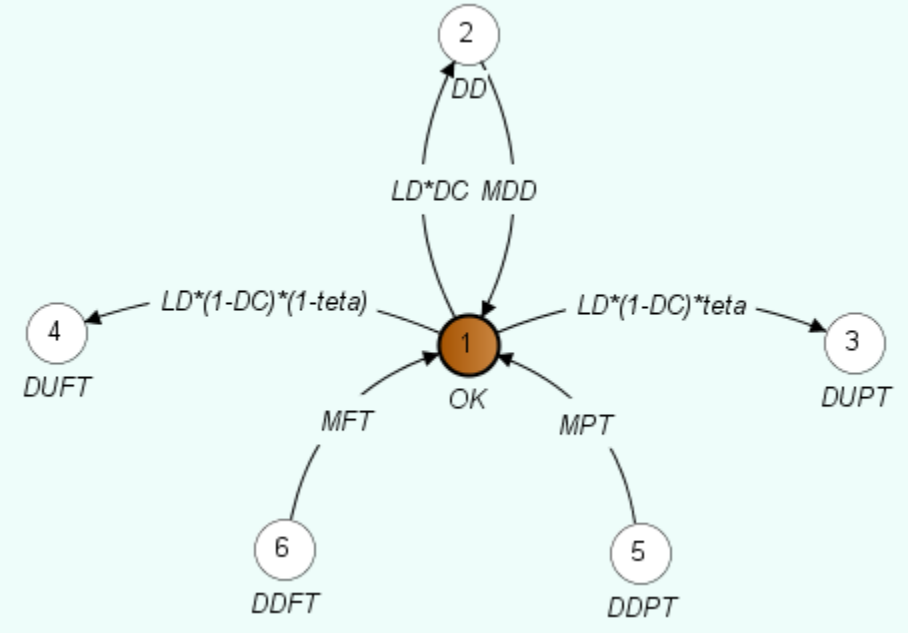
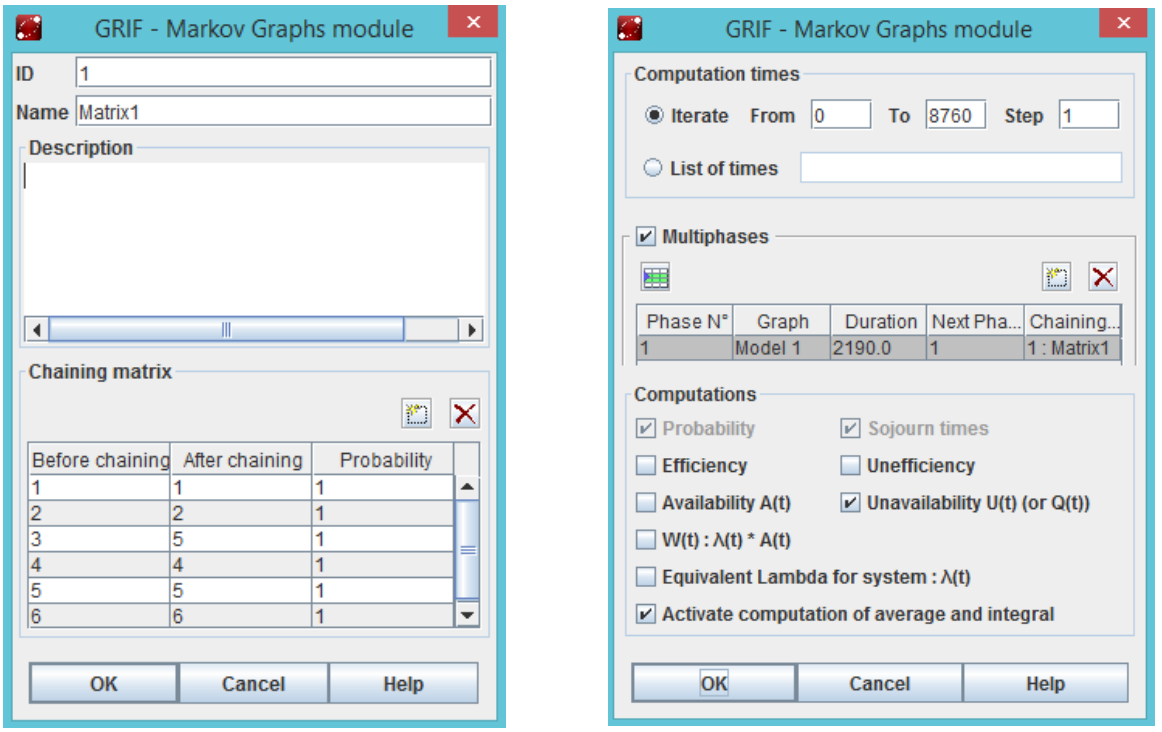


Figure 5.5: Markov model for 1oo1 subject to PT and FT.

The tables for the transition matrix and the settings to achieve the partial tests over the first full test interval are shown in figure 5.6. The figure in 5.6a shows the changing matrix that depicts the transition from each state in the test phases with the probability 1. The computation settings interface in figure 5.6b allows the specification of the multiphase function for partial test and full test interval.

Based on the model description, the unavailability of the system *(PFD(t))* is the sum of the probabilities of being in states 2 to 6. The calculations are performed thanks to the Markov graph module of the GRIF software. The result is represented graphically in figure 5.7. This means for the Markov model, the PFD_{avg} of a 1oo1 system subjected to partial and full test with the inclusion of DD failure is **5.995E-3** where the partial test is set at 2190 hours (3 months). The result from the analytical formulas for m=4 are 6.00E-3 and 6.02E-3 according to different authors which is very close to 5.995E-3 from the multiphase Markov approach. Note that DD failure has not been considered in the analytical approach.



(a) Transition matrix

(b) Computation settings interface

Figure 5.6: Representation of settings from the GRIF Markov module

5.1.3 Limitation of the Markov approach

In as much as the Markov model allows for dynamic modeling of systems, there are limitations to its use for modeling. Higher system configurations and possibly other dynamic considerations will result in a more complex model due to the states explosion (Kumar et al., 2008; Zhang et al., 2008). For instance consideration of partial and full tests for a 2oo4 system will give a complex model with so many states. Consequently the calculation and computation require extra time and resources.

5.2 Use of Fault tree

Fault tree models are used to calculate the PFD_{avg} of systems and components which in accordance with the IEC 61508 standard, distributions for periodically tested components are introduced. Rausand and Høyland (2004) defines a fault tree as a top-down logic diagram that illustrates the interrelationships between a potential hazardous event in a system and the causes of

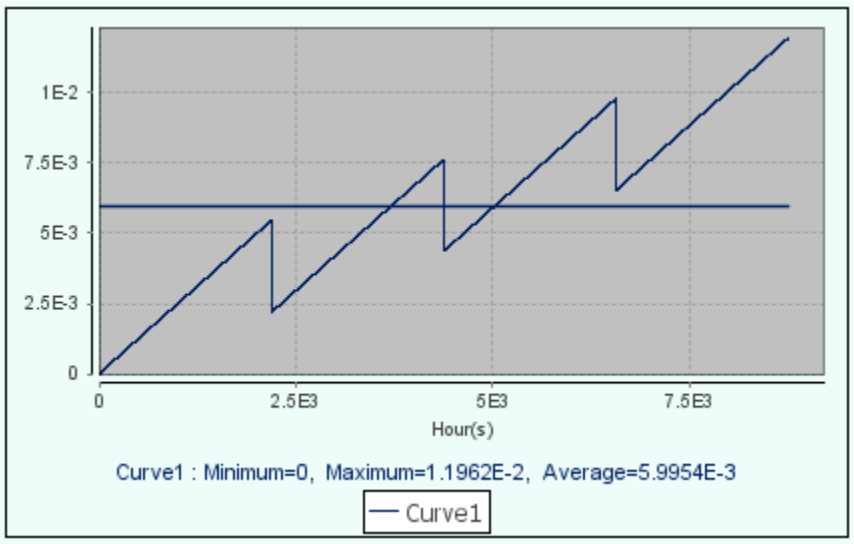


Figure 5.7: System unavailability graph for 1oo1 Markov model.

this event. A fault tree analysis is a model used to analyze how systems can fail. Most fault tree assessment tools calculate the unavailability of top event by using the minimal cut sets (MCS).

For large systems however, the MCS are very many due to the large fault tree therefore cutoffs are applied to focus on the most important MCS. The aim of using the cutoffs is to disregard MCS that have very low (negligible) probabilities. For systems with periodically tested components the probabilities of MCS evolve periodically over time such that the set of MCS at time t_1 may be completely different from the MCS at time t_2. The Binary Decision Diagram (BDD) approach is used instead of recomputing the MCS at intervals (time consuming). Rauzy (2008) gives a systematic approach on how the BDD is implemented. The BDD is computed once for all then the system unavailability is assessed in linear time with respect to the size of the BDD (Dutuit et al., 2008). The GRIF software uses the BDD approach for its computation.

As stated in the previous section, the Markov model for higher configurations will result in a more complex model due to the states explosion. For instance consideration of partial and full tests will give a complex model but the fault tree approach is more accommodating as extra components or consideration only require extra gates or additional basic events. This is the main advantage of the fault tree approach over the Markov. The illustration of this concept is shown in figure 5.8. The fault tree driven Markov models simplifies the quantification by combining simple Markov models related to basic events and the fault tree straightforward calculation process (MCS or BDD techniques). For instance the time dependent failure probability for

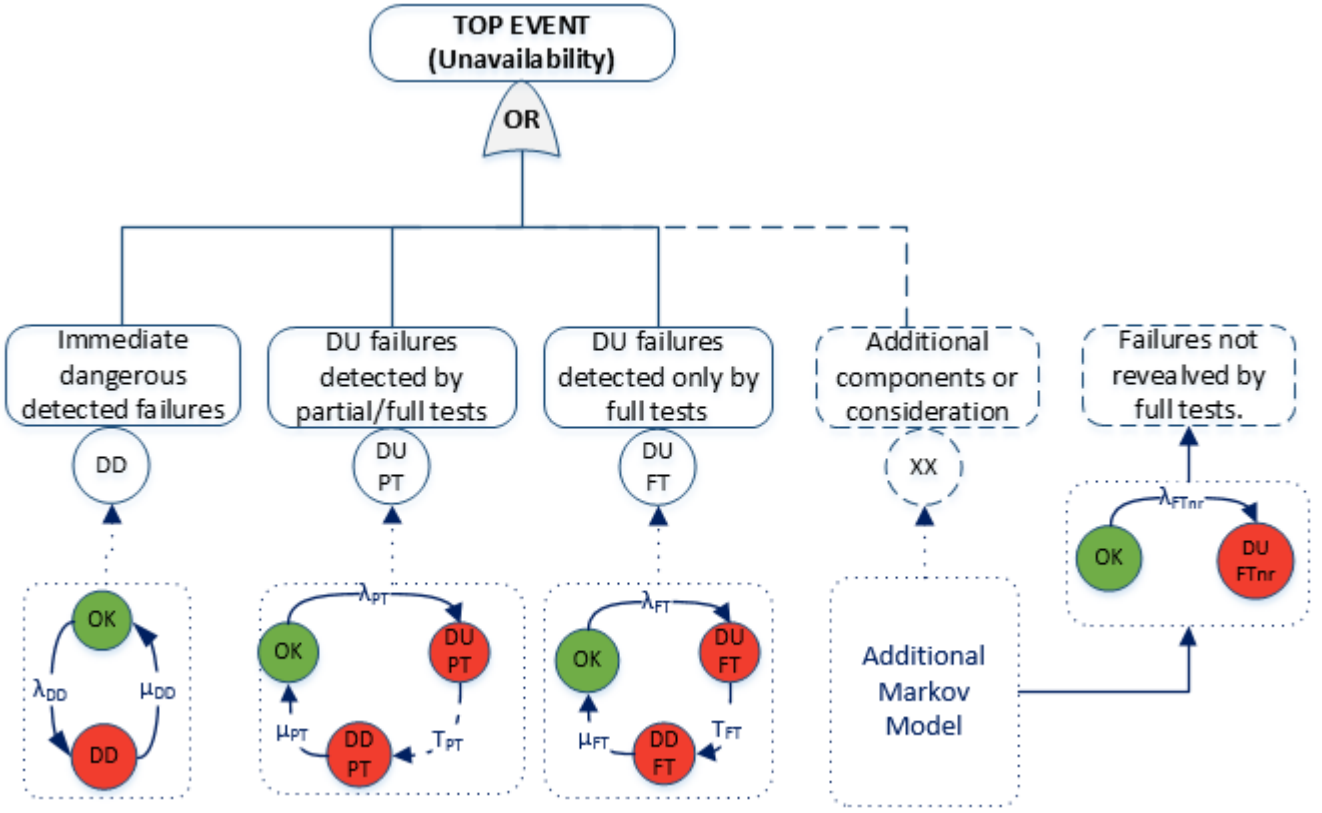


Figure 5.8: Modeling concept using Fault Tree.

one element subject to DD, PT and FT failure modes based on the MCS is given hereafter:

$$PFD(t) = P(DD \cup PT \cup FT \cup XX) = 1 - P(\overline{DD}) \cdot P(\overline{PT}) \cdot P(\overline{FT}) \cdot P(\overline{XX}) \tag{5.7}$$

where $P(DD) = \frac{\lambda_{DD}}{\lambda_{DD}+\mu_{DD}} \cdot \left[1 - e^{-(\lambda_{DD}+\mu_{DD})\cdot t}\right]$; $P(PT)$ and $P(FT)$ will be calculated according to the procedures in section 5.1; $P(XX)$ to be calculated using an appropriate formula depending on the additional case being considered. The fault tree model therefore has a less complex quantification method compared with a Markov model where higher KooN configurations lead to complex Markov diagrams and hence complicated calculation.

For the imperfectness of test to be considered using a fault tree, It can be modeled as an additional basic event with a transition shown and depicted in the figure (5.8) as λ_{FTnr} which represents failure not revealed by the full test. The can be expressed in the formula below:

$$\lambda_{FTnr} = \lambda_D \cdot (1 - DC) \cdot (1 - \theta) \cdot (1 - \xi) \tag{5.8}$$

where DC is the diagnostic coverage factor, θ is the partial test coverage and ξ is the full proof test coverage factor.

5.2.1 Fault tree result comparison with the partial test model

The use of fault tree described in the previous subsection is implemented in the GRIF fault tree module. As can be seen in figure 5.9 the consideration of different KooN configurations can easily be modeled by adding an additional branch. Common cause failures (CCFs) can also be considered by using an "AND" gate at the beginning of the fault tree and then indicating the independent and common cause parts. The fault tree here is an example for a 1oo2 or 2oo2 system. The GRIF software is built to implement different cases and distributions for the basic

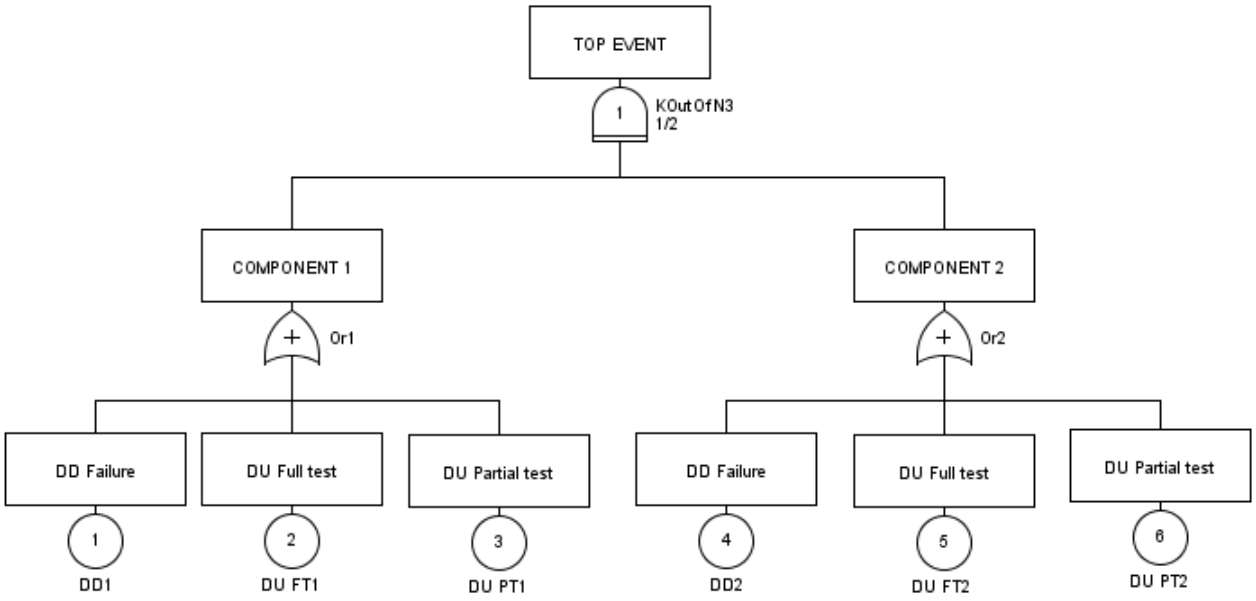


Figure 5.9: Fault tree for a 1oo2 system.

events. In this case the basic events PT and FT which are subjected to tests can be modeled in GRIF by the extended periodic test which gives the possibility to indicate the test and repair parameters. This fact is shown in figure 5.10.

The fault tree model for partial and full test consideration is used and results are compared with the analytical formulas. The table below gives the results of different configurations from the fault tree approach and the three analytical formulas. The results induced by the fault tree are very close to other analytical results. This may be regarded as a mutual validation of the

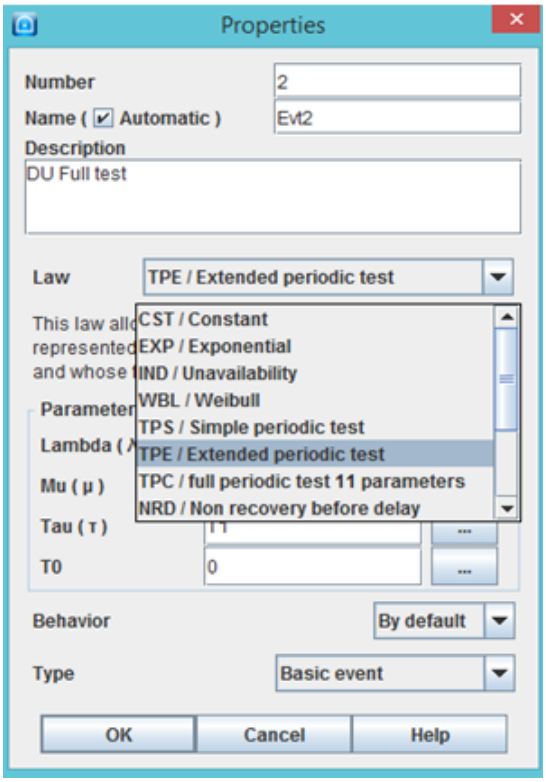


Figure 5.10: Basic event setup properties in GRIF.

different used approaches.

KooN	PST Strategy	Without CCF			
		FTA	Eq. 4.19	Eq. 4.20	Eq. 4.21
1oo1	m = 4	6.00E-3	6.00E-3	6.02E-3	6.02E-3
1oo2	m = 4	4.44E-5	4.44E-5	4.48E-5	4.48E-5
1oo3	m = 4	3.67E-7	3.75E-7	3.72E-7	3.72E-7
2oo2	m = 4	1.20E-2	1.20E-2	1.21E-2	1.20E-2
2oo3	m = 4	1.32E-4	1.32E-4	1.34E-4	1.34E-4

Table 5.1: PFD_{avg} comparison of FTA and analytical formulas

Chapter 6

Petri Net

6.1 Introduction to Petri Nets

The Petri net modeling approach was introduced by Carl Adam Petri and IEC 61508 part 6 suggest it as a suitable approach for reliability analysis. IEC standard 62551 defines the terminologies and gives requirements for the use of Petri net modeling in reliability analysis. The Petri net approach is similar to the Markov approach based on the possible system states and how they change when events occur (Rausand, 2014).

6.1.1 Concepts of Petri nets

A typical Petri net model consists of places represented by circles and transitions represented by a bar or rectangle. The directed arcs are arrows that connect the places and the transitions. The places, transitions and the oriented arcs are the static part of a PN. Tokens represented by black bullets are assigned to places to indicate the status/properties of the place. A place is considered active if it contains a token but it some cases may need more than one token to be considered active. Marking is a term used to describe the distribution of tokens in the places of the PN model. Firing a transition enables the movement of a token from place to place (David and Alla, 2010). Transitions could be guarded by deterministic values (e.g. delay); stochastic variables (e.g. random value according to a probability distribution); conditional statements (predicates, guards) and arc constrains (e.g., weights or inhibitors) (Aguilar Martinez, 2014). An inhibitor arc

is used to prevent a transition from being enabled. The weight (multiplicity) is a digit assigned to each arc to represent the number of tokens the arc can deliver at a time. For a transition to be enabled, the number of tokens in the input place must be equal to or greater than the weight of the input arc. A weight of 1 is a default and normally not written in a PN model but weight of 2 or more are indicated by an integer (Liu and Rausand, 2013). Calculations from Petri Nets are based on Monte Carlo Simulation. A stochastic Petri net is a PN that the transitions use probabilistic delays and the transition is only enabled after the delay is over. Each transition may or may not have a memory. Transition with a memory generates a probabilistic delay as well, but when transition with lower delay fires, it maintains its current value of delay (Grunt and Briš, 2015). The figure in 6.1 shows the described concepts above.

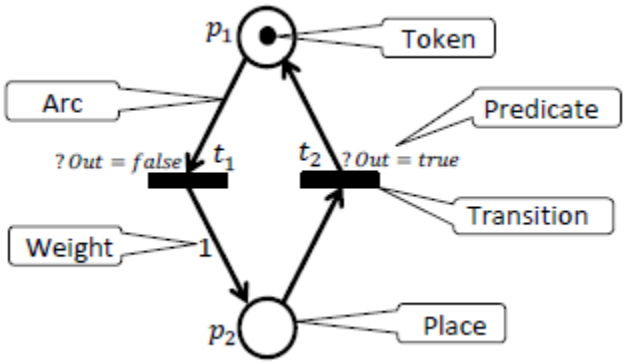


Figure 6.1: A simple Petri Net showing the main graphical elements.

Petri Nets driven by Virtual RBDs

RBDs can be used as a basis for drawing a PN model by drawing a PN for a single component represented by each block of the RBD. The PN model for the behavior of the system will be the combination of the group of all single PNs. The main target of the RBD driven PN are safety systems which mainly the functioning and failed states are considered Signoret et al. (2013).

6.1.2 Petri net models for selected cases

This section demonstrates the methodical use of PNs for modeling the behavior of systems components.

1. **Component with detected failure.** This case shown in figure 6.2 has a token in P_1 which indicates that the component is in a functioning state. The transition T_1 is fired if there is a DD failure and the token in P_1 is moved to P_2. When a repair is carried out, the transition T_2 is fired and the token is moved back to P_1. Note that the transition may be immediate (delay=0) or may take some time to repair represented by a repair rate $\mu = 1/MTTR$.

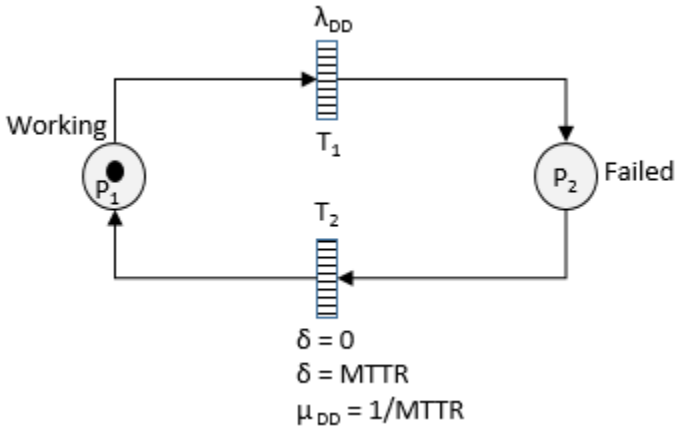


Figure 6.2: A Petri Net model for a failed and repaired component.

2. **Component with detected failure and repair resources limitation**. This case illustrated in figure 6.3 considers the availability of repair team before the repair is carried out. T_2 in this case is only fired if the condition for the repair is met. The condition here checks that the number of repair team (RN) is greater or equal to the required repair number (RRN) before the repair can commence. This shows the flexibility and dynamic nature of PN models.

3. **Component with undetected failure and instantaneous test.** This case in figure 6.4 describes an element with a DU failure subject to proof testing. In this case, it is assumed that the test is perfect and the test is carried out instantaneously (test duration = 0). When there is a failure, the token is moved to P_2 and repair can only be initiated if the failure is detected during test ($\delta = \tau$) when the transition T_4 is enabled and token moved from P_4 to P_5. The guards ?#4=0 and ?#2=0 indicated at T_2 and T_5 respectively mean that the token in P_4 be equal to 0 before the transition T_2 be enabled and the token in P_2 be =0 for the repair to take place and token back to P_1 the working state before the transition T_5 be enabled

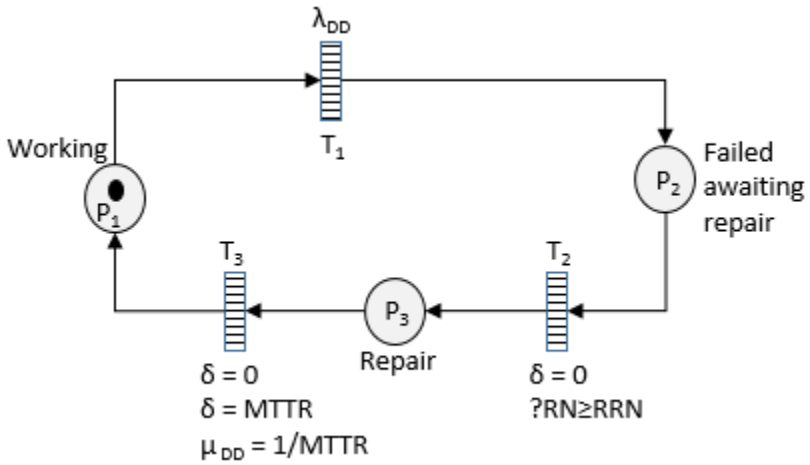


Figure 6.3: A PN model with inclusion of repair resources.

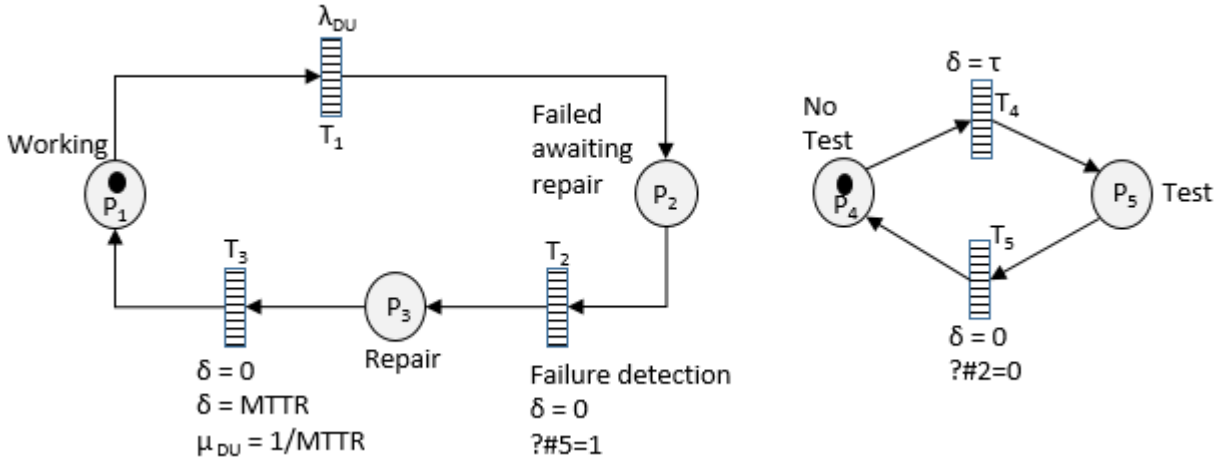


Figure 6.4: PN model for a DU failure subject to testing.

so that the token is moved from P_5 back to P_4 for a new test period to start. The inhibitor arc described in the previous section can also be used to achieve same aim of the guards.

4. **Component with undetected failure and non-instantaneous test.** This case considers a situation whereby the actual time of performing the test is not instantaneous ($T_D \neq 0$). The dynamics is same as the third case except for inclusion of a place representing the test period (time of performing the test). As shown in 6.5 when there is a failure the token is in P_2. When it is time for test, the transition T_5 is fired for token in P_5 to be moved to P_6 and the transition T_2 confirms this and the test is initiated at P_3. The test takes some time (T_D) which is not negligible. When the test is completed, repair is then carried out.

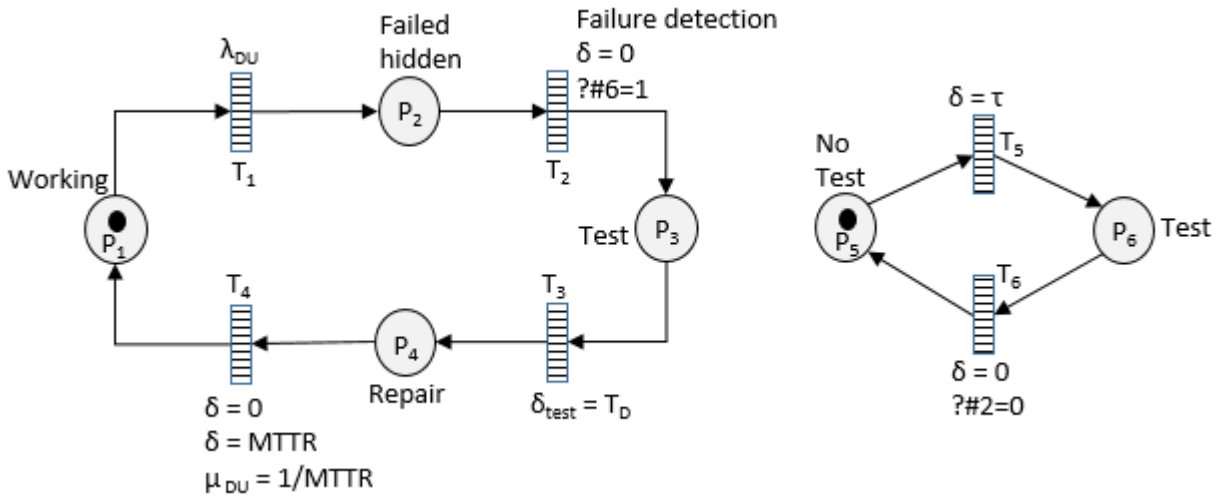


Figure 6.5: PN model for Non-instantaneous test time.

5. **Component with undetected failure, non-instantaneous test and available during test.** This case is similar to the fourth one except that the component becomes unavailable when the test is performed (For instance, the element being tested needs to be disconnected). This fact is characterized by transition T_5. Obviously as soon as the test is finished (firing of transition T_6) the element goes back to P_1 without any repair action. Figure 6.6 shows the inclusion of the availability consideration in the model.

6. **Component with undetected failure subject to partial and full tests.** This case is a model that is subject to both partial test failure and full test failures revealed during the partial and full tests respectively. Figure 6.7 shows the dynamics of this system. When there is a failure, the transitions T_1 or T_5 is fired to places P_2 or P_5 to denote a partial test failure ($\lambda_{PT} = \theta \cdot \lambda_{DU}$) or a full test failure ($\lambda_{FT} = (1-\theta) \cdot \lambda_{DU}$) is present respectively. During partial test the transition T_9 is fired and the token moves from P_8 to P_9. T_2 is enabled if it checks that there is no token in P_8 and then test is initiated. The cycle goes on as explained in the fourth case. The same process is repeated during the full test interval.

7. **Consideration of imperfectness of proof tests.** Actually, there are two possibilities to consider the imperfectness of proof tests.

   - The first option is to split the failures which are not detected by full tests (λ_{FT}) in two portions according to the test coverage factor denoted ξ: failures that are de-

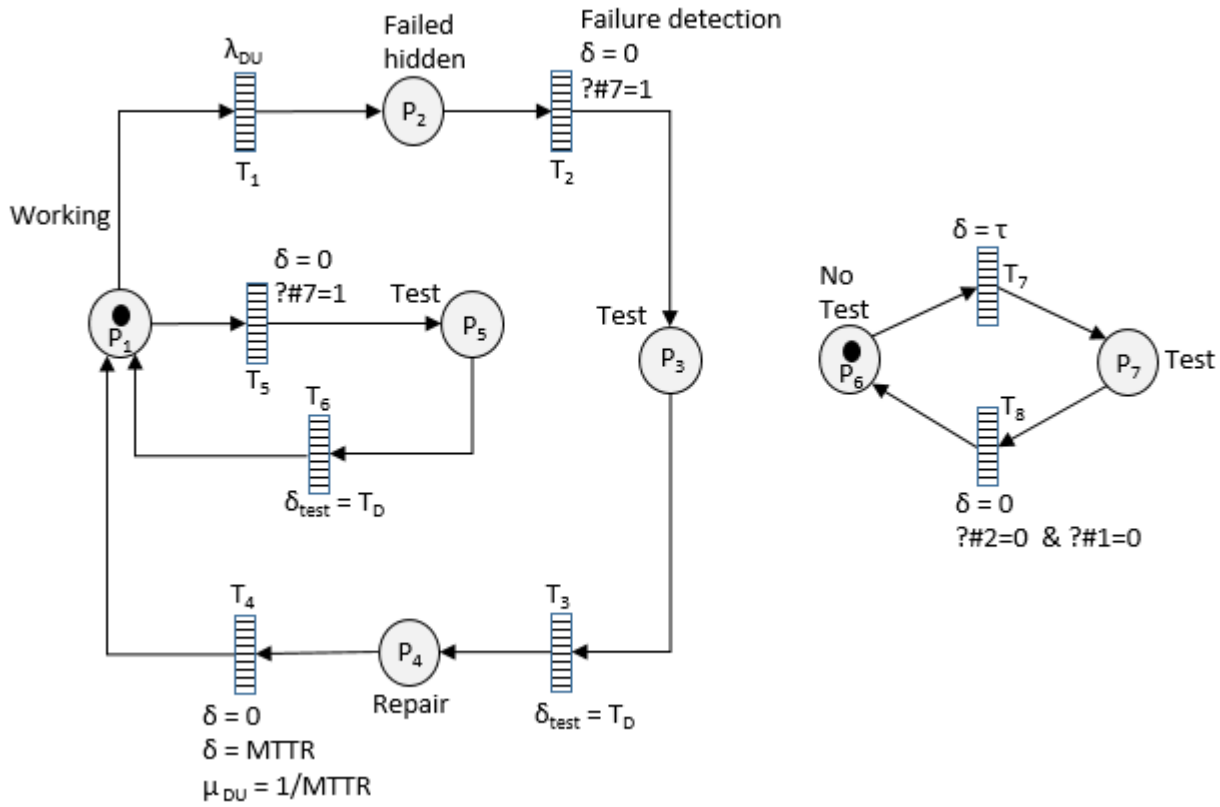


Figure 6.6: PN model for availability consideration of component.

tected by full proof tests ($\lambda_{FT1} = \xi \cdot \lambda_{FT}$) and failures that are never revealed by these tests $\lambda_{FT2} = (1-\xi) \cdot \lambda_{FT}$. This is depicted in figure 6.8 by the transitions T_5 and T_13 respectively.

- The second option is to consider test coverage factor as a probability of detecting failures. This means the full test can be successful in detecting all failures with a probability = σ. In addition, if the test does not detect the failures with the probability $1-\sigma$, these failures may be detected in the next full test. This concept is shown in figure 6.9 where the transition T_7 is the probabilistic transition with two possible outcomes.

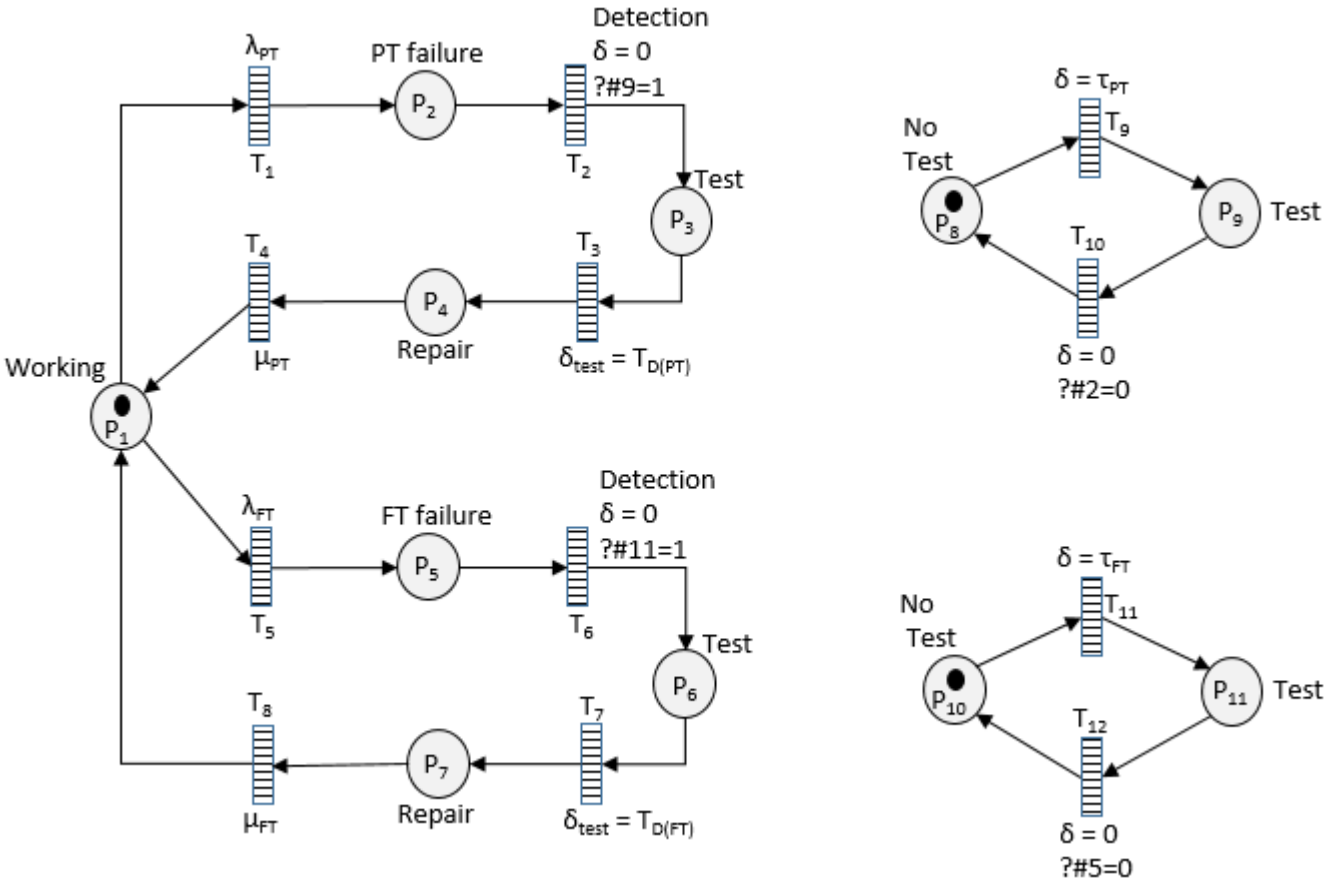


Figure 6.7: A Petri Net model for both partial and full tests.

8. **Consideration of failure due to tests.** In the course of performing the tests, there is a probability that a failure might occur due to the test itself. The probability of failure due to test is classically designated by γ. This factor can for example be added to the PN model of figure 6.9 as shown in figure 6.10.

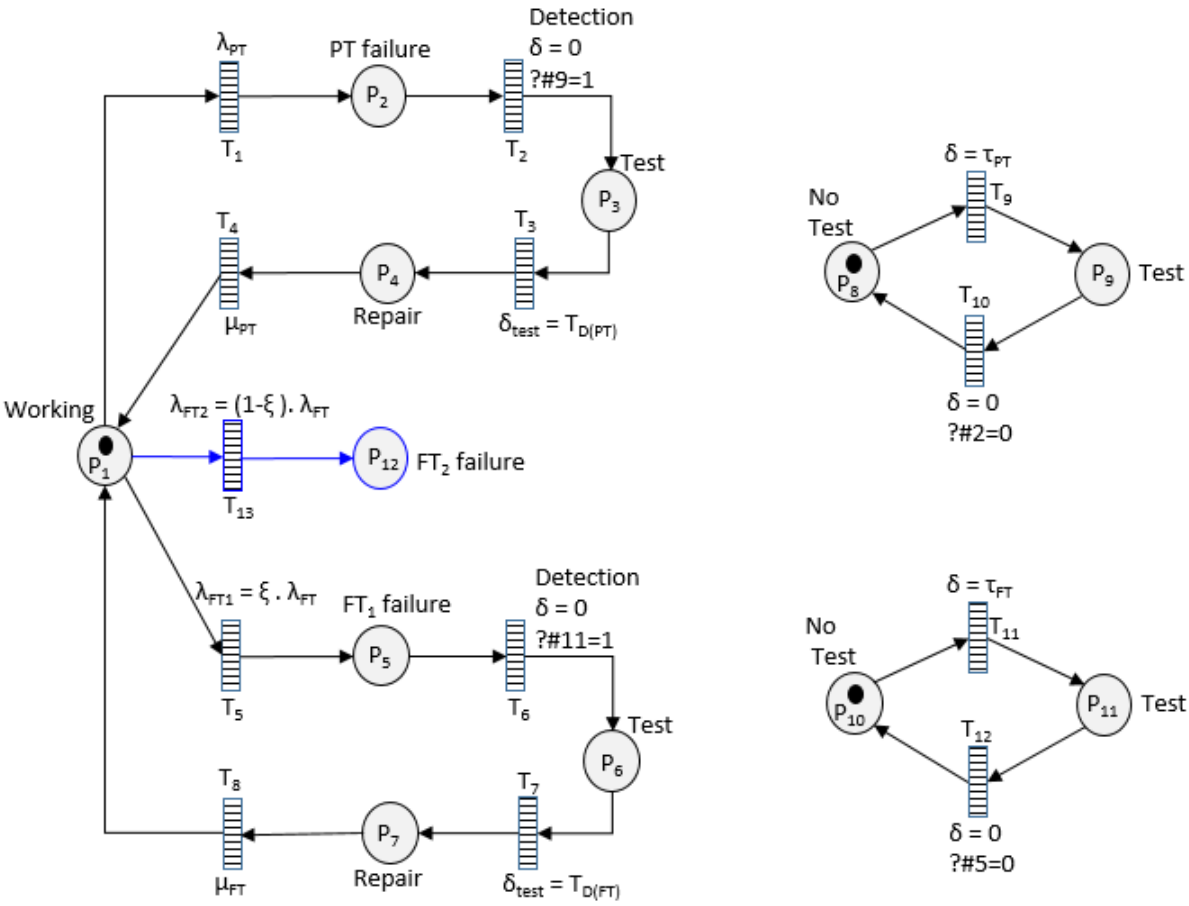


Figure 6.8: A PN model with imperfect test as proportion of unrevealed failures.

6.2 Combining elements behavior with Petri net

So far, we have modeled the behavior of a single element. In order to study more complicated system, one needs to combine behaviors for different elements. In the following, we will discuss three issues related to that combining, namely: common cause failures (CCFs), staggered tests and the proper combining of the elements regarding to the system configuration.

6.2.1 Proper combining

This issue can be dealt with by creating separate Petri net. Let us consider the case of a system made up of 3 elements as depicted in figure 6.11. We assume that each of the 3 elements is modeled according to a given model as given in the previous section. The place 1 in each of the models relates to the working condition. The conditions (Guards) related to the transitions can

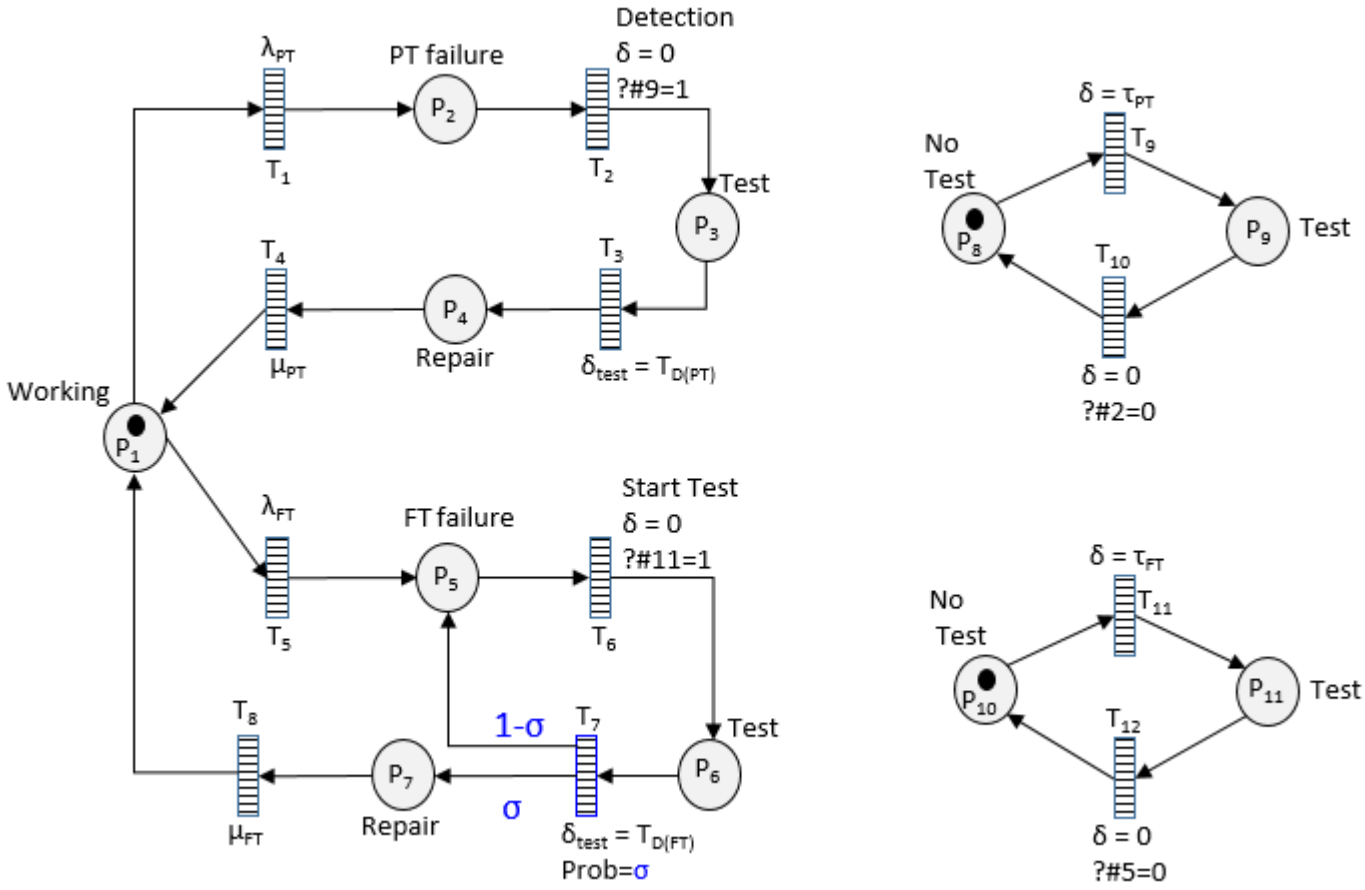


Figure 6.9: A PN of imperfect test as probability of not revealing failures.

simply be derived from the structure function of the system.

6.2.2 Staggered test

To implement staggered test for n components, we only have to change the first time of test for the whole elements constituting the system. For instance, for a uniformly distributed test, the first test interval for element i can be defined as: $i \cdot \tau / n$. The test strategy can now be implemented in place of the simultaneous test strategy used in the previous cases. This is shown in figure 6.12

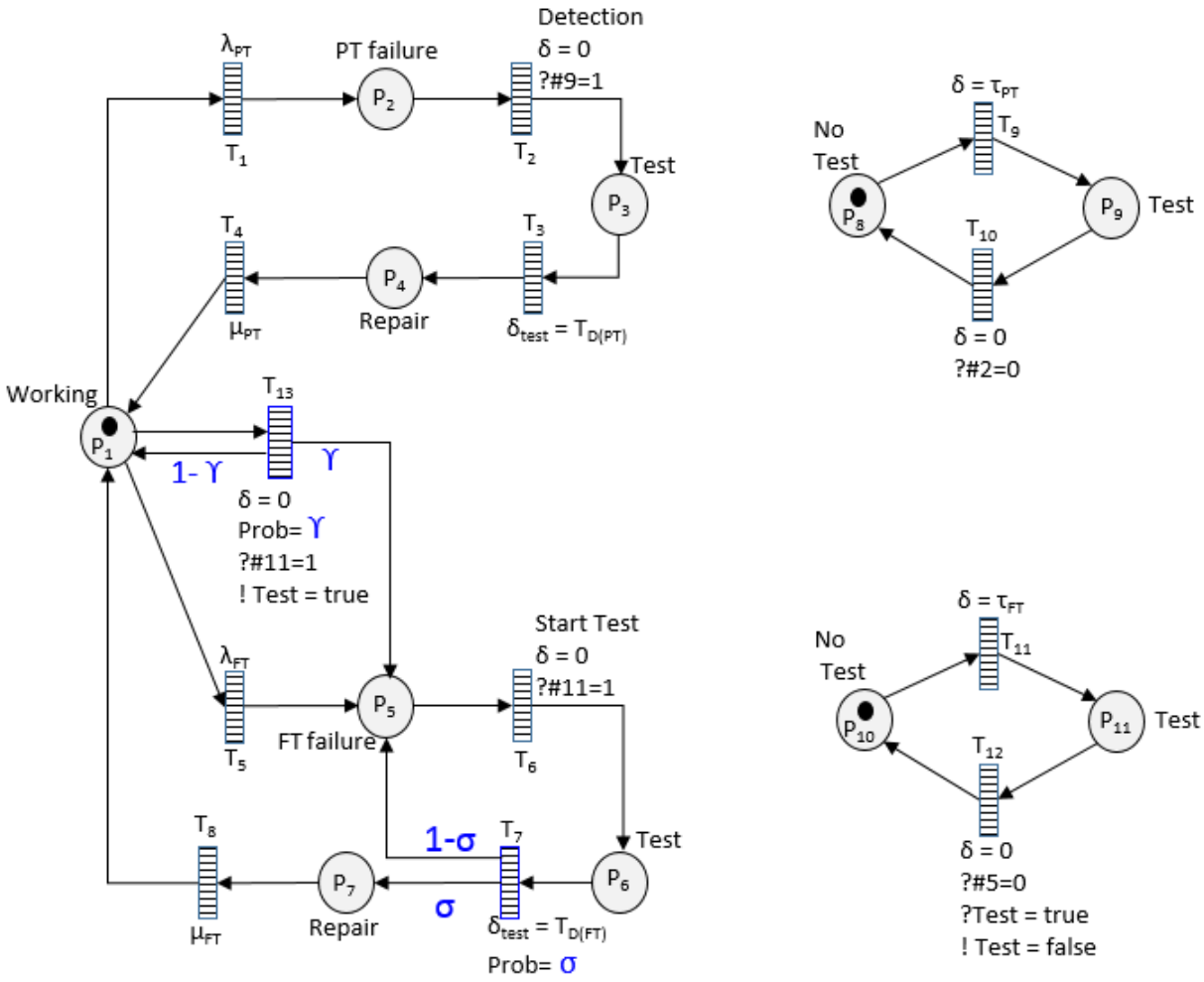


Figure 6.10: A PN model with probability of test induced failure.

6.2.3 Common cause failures (CCF) contribution

As mentioned is chapter 3, the CCF contribution to the PFD is significant and therefore must be considered. For this end, we can add Petri net as shown in figure 6.13 where two of the failure modes (PT and FT) are accounted for. However, additional common cause consideration for other failure modes (eg. DD failure) can be added to the model. The CCF effects are assigned to the individual behavior by adding new failure transitions as depicted in figure 6.13.

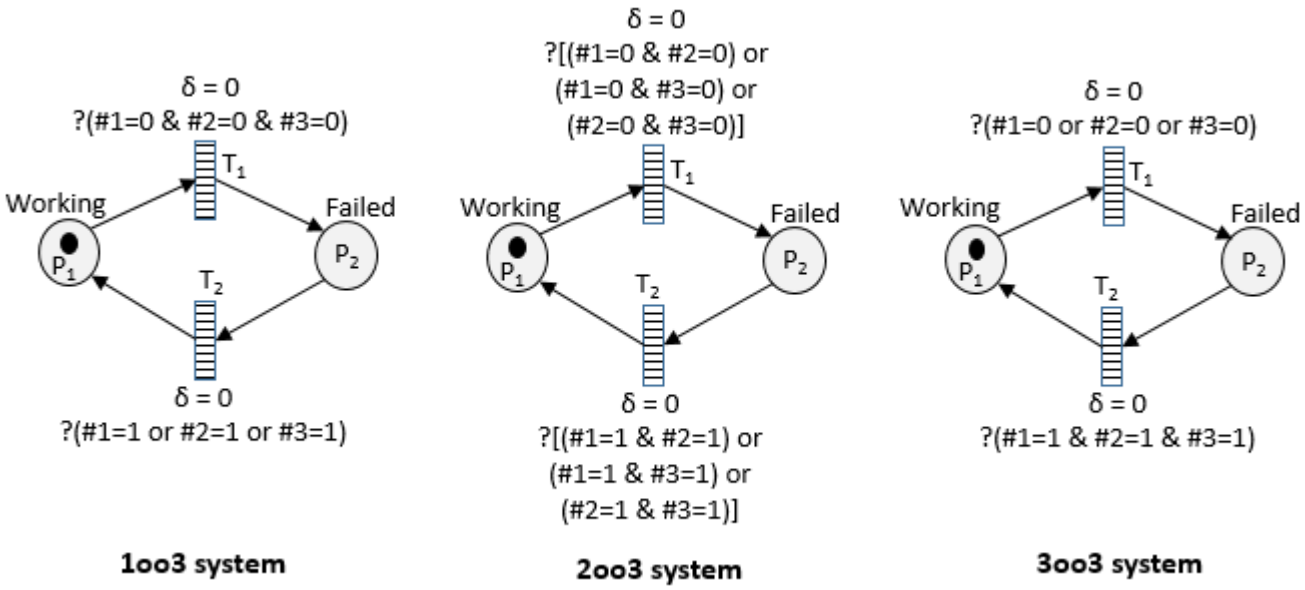


Figure 6.11: PN models for multiple components of different configurations.

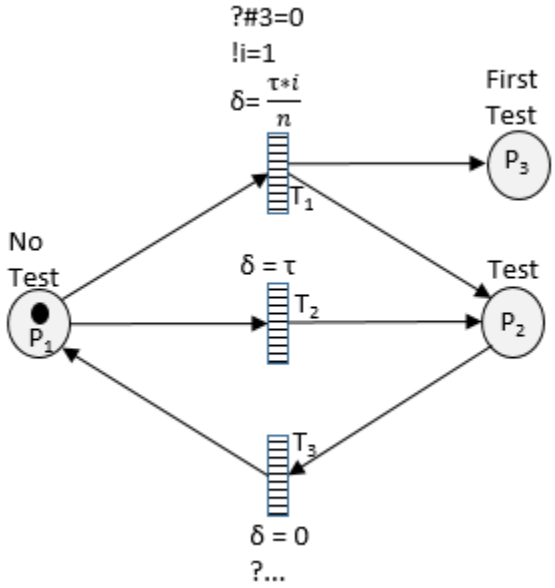


Figure 6.12: PN model for a uniformly distributed staggered tests.

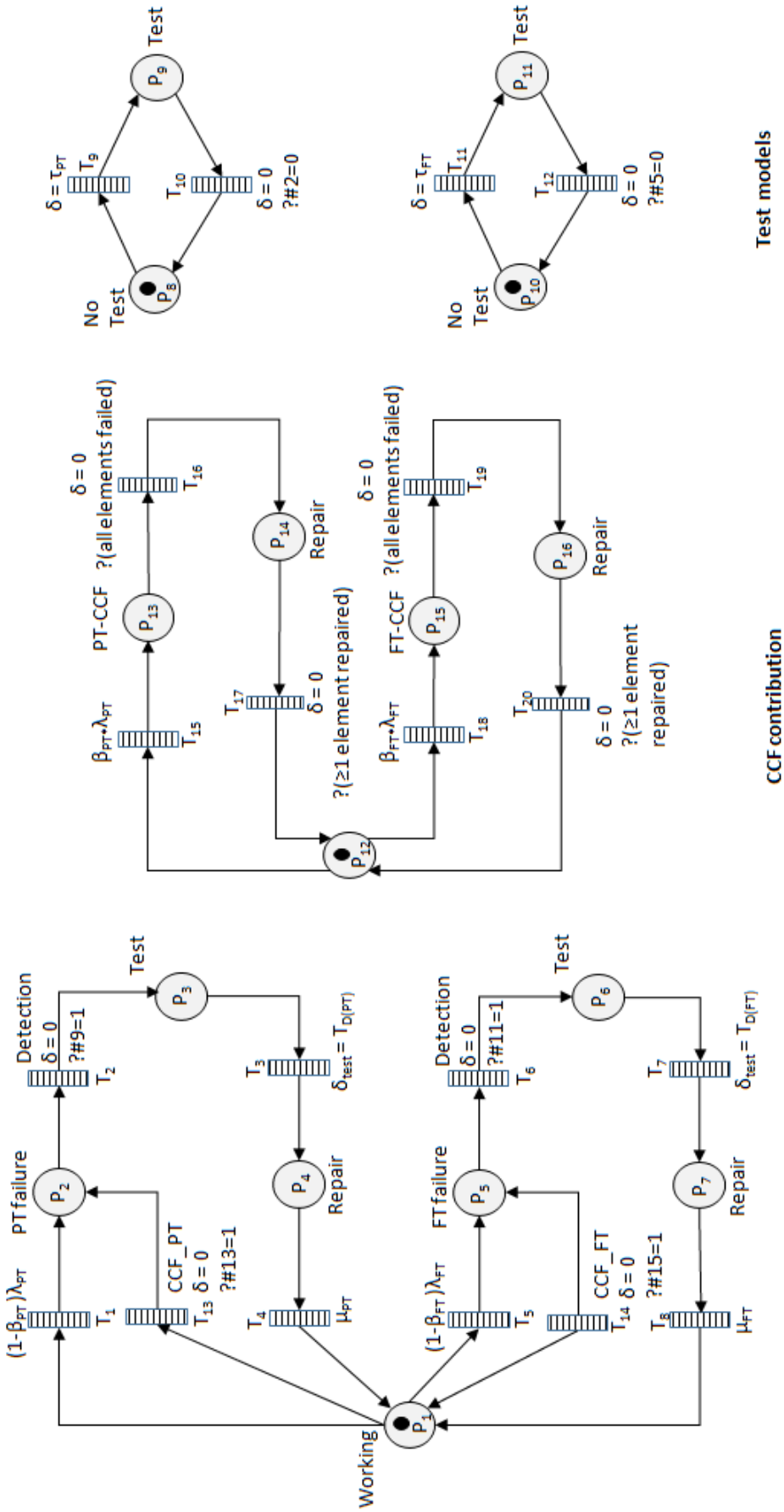


Figure 6.13: PN model for CCF consideration.

Chapter 7

Case study

7.1 Chemical reactor protection system (CRPS)

This chapter presents the case study and uses the reliability assessment methods described in the previous chapters to demonstrate the effect of partial and imperfect testing on the reliability of a SIS.

7.1.1 System description

In this section, the case study described below taken from Torres-Echeverria et al. (2009) is used to illustrate the use of fault tree driven multi-phase Markov and Petri nets approaches described in the previous chapter. The case study is a protection system designed and implemented to control high temperature and pressure of a chemical reactor. The system is composed of four subsystems namely:Temperature transmitter (TT), pressure transmitter (PT), logic solver (LS) and the final control element (FC). When high pressure or temperature is detected by the transmitters, the system should shut the supply source to the reactor in order to prevent explosion of reactor. The structure of the system is shown in figure 7.1 below. Each subsystem is parallel redundant. The sensor layer is made up of two transmitters:Temperature transmitter (TT) and pressure transmitter (PT), structured in 1oo2 architecture. The Logic Solver layer (LS) structured in 1oo3 architecture and the final control (FC) layer is structured in 1oo3 architecture, made up of three valves (Torres-Echeverria et al., 2009; Mechri et al., 2015). The reliability block diagram

is shown in figure 7.2. Different cases will be studied in the next subsection.

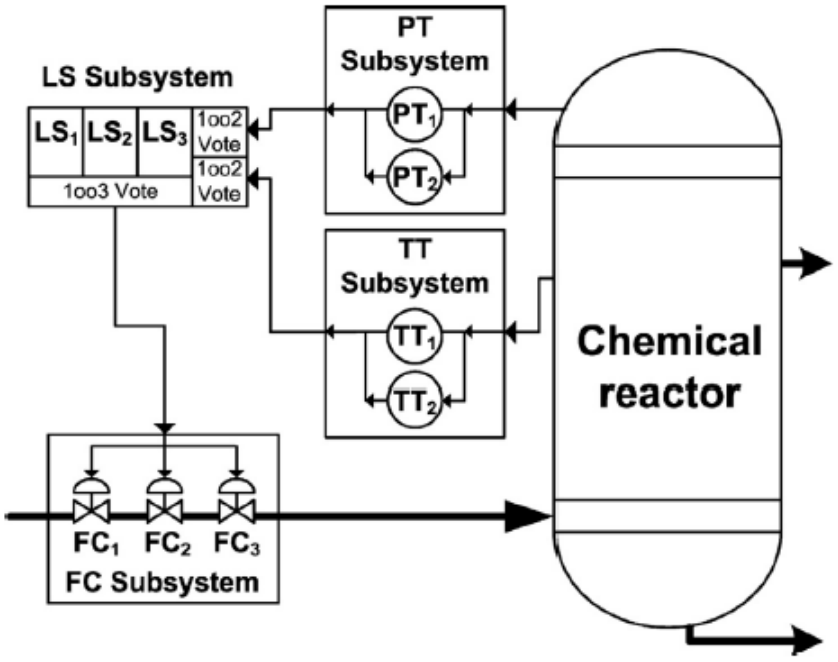


Figure 7.1: High integrity protection system of a chemical reactor (CRPS)

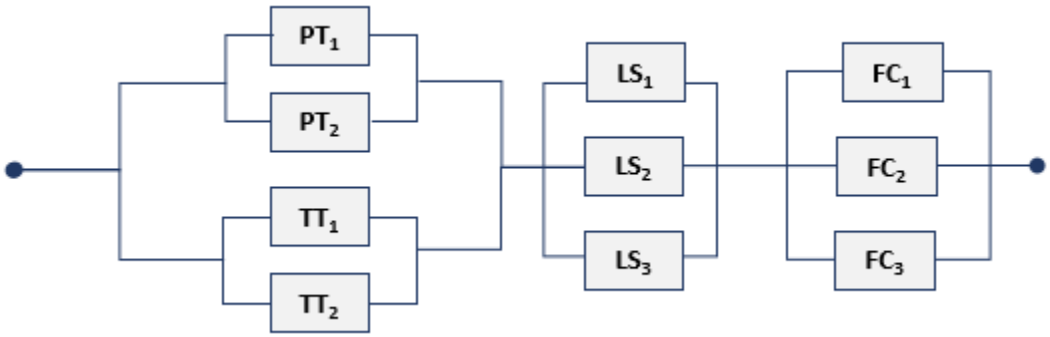


Figure 7.2: CRPS reliability block diagram.

7.1.2 Comparison and discussion of results

Different reliability data are gathered and presented in table 7.1. Note that the failure rates and their respective diagnostic coverages are taken from (Torres-Echeverria et al., 2009). The other parameters are only suggested values.

Parameters	LS (Safety PLC)	PT (Electronic)	TT (Switch)	FC (Air operated valve)
$\lambda_D(1E-6/h)$	0.032	1.90	7.60	3.35
DC (%)	81.25	51.1	10	25
$\tau(h)$	8760	4380	4380	4380
MTTR=MRT (h)	8	8	8	8
Other parameters common to all components:				
$\beta = 0.1$ and $\beta_D = 0.05$				
Proof test coverage $\xi = 0.8$				
Probability of detecting failure $\sigma = 0.8$				

Table 7.1: Parameters for the system analysis

For formula comparison purposes where applicable (case 1 and case 2), the formula given in Chebila and Innal (2015) is used and given in equation 7.1:

$$PFD_{KooN} \approx \frac{\binom{n}{n-k+1}}{(n-k+2)\cdot\lambda_{DU}^{(i)}\cdot m\cdot T_{ST}}\cdot\sum_{j=0}^{m-1}\left[\left(\frac{\lambda_{DD}^{(i)}}{\lambda_{DD}^{(i)}+\mu_{DD}}+\left(\lambda_{DU}^{(i)}+\lambda_{PT}^{(i)}\cdot j\right)\cdot T_{ST}\right)^{n-k+2}\right.$$
$$\left.-\left(\frac{\lambda_{DD}^{(i)}}{\lambda_{DD}^{(i)}+\mu_{DD}}+\lambda_{PT}^{(i)}\cdot j\cdot T_{ST}\right)^{n-k+2}\right]+\frac{\lambda_{DD}^{(CCF)}}{\lambda_{DD}^{(CCF)}+\mu_{DD}}+\lambda_{ST}^{(CCF)}\cdot\frac{T_{ST}}{2}+\lambda_{PT}^{(CCF)}\cdot\frac{T_1}{2} \tag{7.1}$$

The fault tree and Petri net models are built in GRIF. CCFs are considered. The value 0.8 for ξ was used both for the proportion and for the probability σ. The robust nature of the GRIF software allowed for the consideration of staggered tests and uncertain parameters that follow uniform distribution. The two options (1 and 2) mentioned in the cases mean:

- **Option 1:** The first option is to split the failures which are not detected by the full proof tests (λ_{FT}) in two portions according to the test coverage factor denoted ξ: failures that are detected by full proof tests ($\lambda_{FT1} = \xi\cdot\lambda_{FT}$) and failures that are never revealed by these tests $\lambda_{FT2} = (1-\xi)\cdot\lambda_{FT}$.

- **Option 2:** The second option is to consider test coverage factor as a probability of detecting failures σ. This means the full test can be successful in detecting all failures with a probability = σ. In addition, if the test does not detect the failures with the probability $1-\sigma$, these failures may be detected in the next full test.

The FT-MPM and MC-PN in the table mean fault tree driven multi-phase Markov and Monte Carlo Petri net respectively.

Cases	Description	FT-MPM	Formula	MC-PN
Case 1	$\theta = 0$; $and\ \xi = 1$: This is a case where the proof test is perfect, no partial tests are considered and no failure possibility due to the test itself	5.537E-4	5.558E-4	5.556E-4
Case 2	$\theta = 0$; $\xi < 1$ (option 1 is used for test imperfectness): This is a case where the proof test is imperfect, no partial tests are considered and no failure possibility due to the test itself. Note that the option 1 for full test imperfectness is used: the undetected failure rate is split in two portions. The values of ξ are provided in the table	5.060E-3	5.136E-3	4.890E-3
Case 3	$\theta = 0$; $\sigma < 1$ (option 2 is used for test imperfectness): This is the same as case 2 except the option 2 for full test imperfectness is used: σ is used as probability of detecting a failure during the test.	8.239E-4	-	8.221E-4
Case 4	same as case 3 (σ as a probability) but with θ = 0.5 for final elements	4.614E-4	-	4.579E-4
	a sub case with $\theta = 0.7$ for final elements	3.164E-4	-	3.142E-4
Case 5	same as case 2 (ξ as a proportion) but with θ = 0.5 for final elements	2.578E-3	-	2.579E-3
	a sub case with $\theta = 0.7$ for final elements	1.617E-3	-	1.587E-3
Case 6	same as case 3 but with staggered tests for final elements	2.816E-4	-	2.793E-4
Case 7	same as case 4 but θ (for all elements) and σ (for valves) are considered as uncertain parameters following uniform distributions: $\theta = unif(0.4, 0.6)$; $\sigma = unif(0.7, 0.9)$. The uncertainty propagation is carried out using Monte Carlo technique	4.648E-4	-	4.560E-4

Table 7.2: Different cases with their related PFD_{avg} values by different methods

The cases and results from the different methods presented in table 7.2 show that the results from the different approaches give very close results. For Petri net simulation, $1E+6$ trials have been performed. When we compare case 1 (perfect proof test) and case 2 (imperfect proof test), we see an increase in the PFD of the system meaning that consideration of imperfectness of tests could affect the SIL. It is the same instance when case 1 is compared with case 3. The results of cases 2 and 3 show that considering σ as a probability (case 3) reaches a steady state hence gives a better $PFD(t)$. This fact is further clarified in the graph

Figure 7.3 depicts the $PFD(t)$ for the comparison of cases 2 and 3. It shows that the $PFD(t)$ related to case 2 increases continuously and consequently the PFD_{avg} keeps on increasing. The $PFD(t)$ for case 3 reaches a periodic steady state very quickly and hence the PFD_{avg} becomes constant irrespective of the observation period. Based on these results and the description in the literature review, we believe that the second option is more appropriate to characterize proof test imperfectness. Indeed, according to full test procedures and conditions, undetected failures may not be detected in a given test but they can be detected during the next test because of possible changes in the test conditions. This can be justified with the concept of test completeness and correctness mentioned in chapter two.

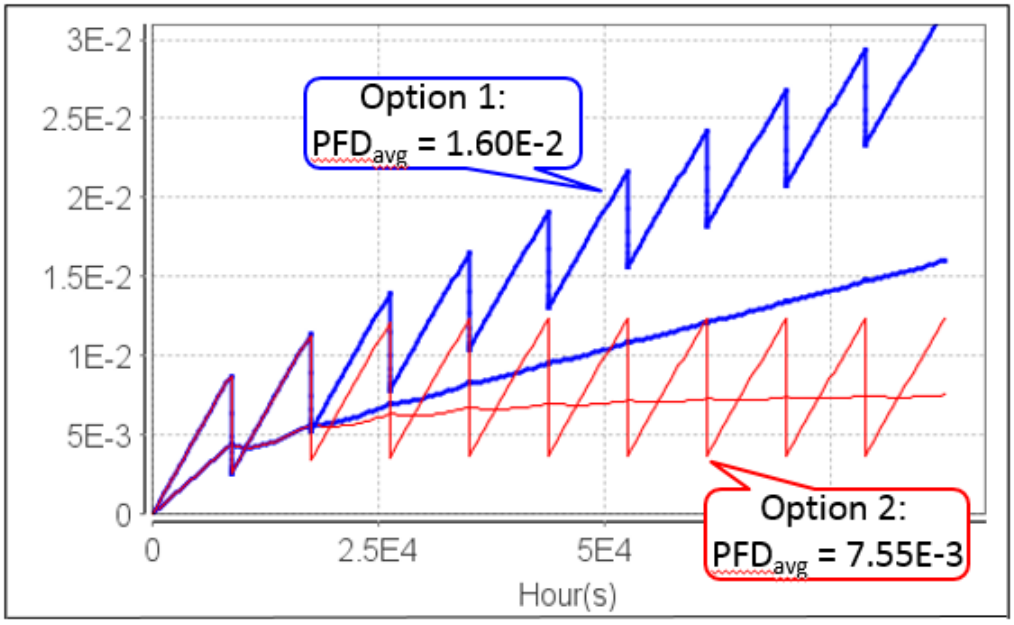


Figure 7.3: $PFD(t)$ related to the imperfectness options: $\xi = 0.7$, $\sigma = 0.7$, $\lambda_{FT} = 1E-6$, $\tau = 1yr$ and observation period of $10yrs$.

Case 4 includes implementation of partial testing in addition to case 3 and we can see the reduction in the PFD_{avg}. Note that the partial test introduced is only on the final elements but still showed a reasonable decrease in the $PFD(t)$ when compared with the results in case 3. An increase of the partial test coverage factor θ shows a further decrease in the PFD_{avg}. This emphasizes the importance of determining accurate test coverages. Case 5 is similar to case 4 but here, the partial test coverage is introduced with the option of ξ as a proportion of unrevealed failures. When this case is compared to case 2, we also see a decrease in the result by the introduction of partial test and a further decrease by an increase the partial test coverage factor θ. The partial test interval used for these cases is 1 month (730 hours). It is worth noting that in case 5, even with $\theta = 0.7$, the corresponding PFD_{avg} is still higher compared to cases 3 ($\theta = 0$) and 4 ($\theta = 0.5$ & 0.7). This shows that option 1 for test imperfectness is very pessimistic compared to the option 2. Note that for this case, there is no available analytical formula in the literature except for a 1oo1 system (Rolén, 2007).

In case 6, a staggered test strategy is implemented. This was done by changing the time of first test as explained with the Petri net model in the previous chapter. When we compare this case with case 3, the results show a decrease meaning that staggered tests have positive impact on the PFD_{avg} of the system. This justifies the fact that staggered tests reduces the possibility of CCFs mentioned in chapter 2. However, there might be some disadvantages with implementing staggered tests like extra costs and resources that may be involved.

The last case which considers the θ and σ as uncertain parameters following a uniform distribution gives results close to case 4. When uncertainties are considered, we can have lower results as in the case of the fault tree approach of case 7 compared with case 4 with $\theta = 0.5$. This is however not the case with the PN result. The reason may be the number of iterations carried out. 10000 iterations was used for the fault tree and only 1000 for the Petri net due to simulation time. Other correct distributions for any parameters could be used like the lognormal. The graph in figure 7.4 shows the propagation of the result with the upper and lower boundaries.

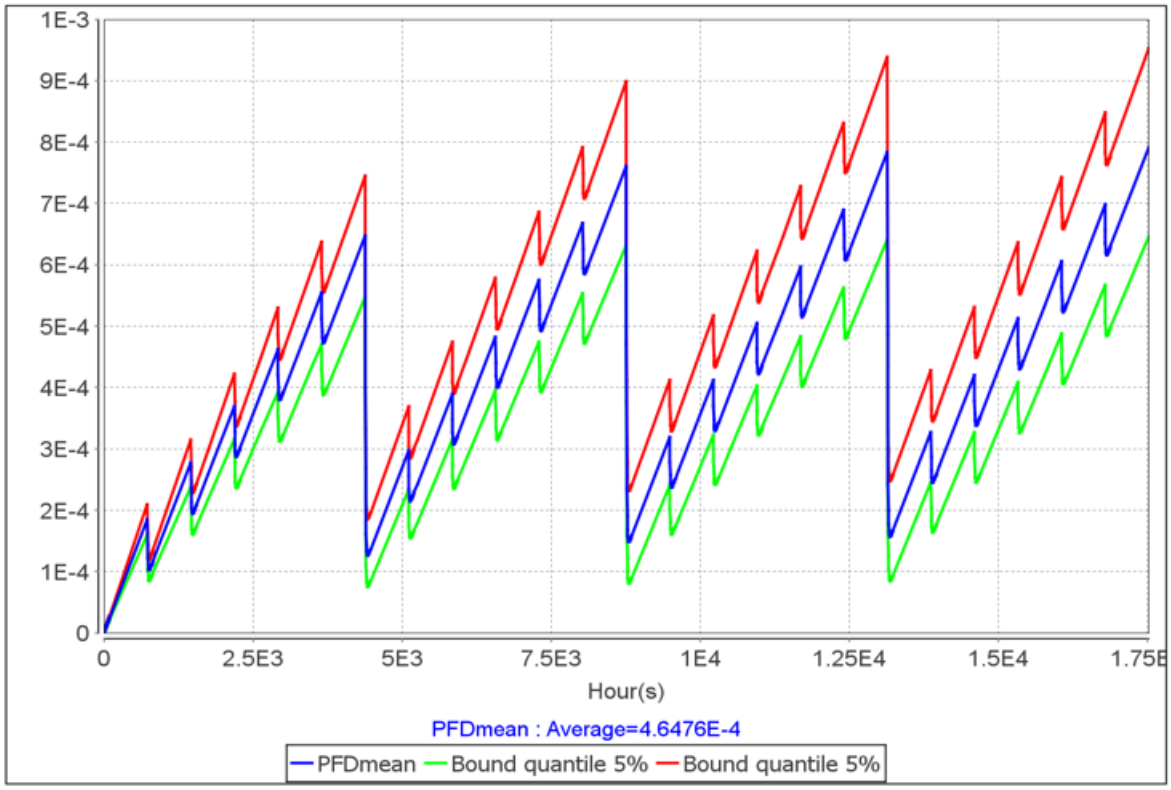


Figure 7.4: $PFD(t)$ graph for uniformly distributed parameters with uncertainties.

Chapter 8

Discussion and Conclusion

8.1 Discussion

In performing reliability assessments, many assumptions are made as regards the system being analyzed and the conditions in which they operate. Results from the assessments influence the design of a SIS to a large extent therefore, care should be taken to incorporate the assumptions and uncertainties into the assessment, so that results are not misinterpreted.

In this thesis, we looked at three approaches to consider imperfectness of tests. First is the IEC 61508 approach where the proof test coverage (PTC) is considered as a proportion (fraction) of DU failures that are revealed by the proof test. This is also represented by the Greek letter ξ. Secondly, we analysed imperfectness as the probability of detecting a DU failure during a given proof test. The Greek alphabet σ was used to denote this probability. The third approach considered is the PDS method which introduces the addition of a constant (probability of test independent failures (P_{TIF})) for modeling the effect of imperfect proof tests due to systematic failures.

Furthermore, different methods of performing reliability assessments were used. We looked at the use of analytical formulas. The cases analyzed in chapter seven reveals there are some complicated cases where analytical formulas are not available. In chapter five, we used the multi-phase Markov approach and discussed the limitation regarding states explosion with complex models (e.g. higher configuration) and other failure considerations. This thesis showed that the fault tree approach whereby some basic events could be modelled using multi-phase Markov

(hence called fault tree driven multi-phase Markov) is a better approach compared to the two earlier stated approaches. The flexibility and dynamism of the Petri net is shown in chapter six. Here, different cases of how a PN model can be used were given. The case study is not a complex one hence did not demonstrate how the PN approach is further used in complicated cases where other methods are not feasible.

8.2 Conclusion

Having analyzed two options of considering the proof test coverage: as a proportion (fraction) of DU failures that are revealed by the proof test ξ (the test does not cover all possible failures: inadequate test method) and as a probability of detecting a DU failure during a given proof test σ (test does not detect all the failures: unsuccessful test), we have shown that the use of the latter is more appropriate. This is because undetected failures which are detected in a given proof test may be detected during the next test because of possible changes in the test procedures and conditions. The justifies the concept of test completeness and correctness described in chapter two. The graph illustrating this fact is shown in chapter seven, figure 7.3. It is worth noting that from the cases presented in table 7.2 in chapter seven, case 5 even with $\theta = 0.7$, the corresponding PFD_{avg} is still higher compared to cases 3 ($\theta = 0$) and 4 ($\theta = 0.5\&0.7$). This shows that option 1 for test imperfectness is very pessimistic compared to the option 2 therefore validates our point that option 2 is more suitable. Note however that there may be cases where a combination of two of the approaches may be more suitable.

When considering different complicated maintenance strategies (e.g. repair strategies and resource availability) and system reconfiguration following a failure, only simulation will be the suitable approach. This is to enable these factors to be considered in the modeling of the system failure probability. A typical example of this is a degradation with repair process due to partial tests. This can not be calculated except simulated using Petri net. Some of the cases in table 7.2 in chapter seven prove this fact.

This thesis has made it obvious that it is important to critically appraise and evaluate the assumptions made when performing reliability calculations. The case of imperfectness of tests justifies that disregarding the estimation of non-testable (non-detectable) failures could lead to

an inaccurate PFD result. With the increase in the use of SIS for risk reduction, improvement in the quality of the reliability calculations is important.

8.3 Recommendations for further work

In the course of this master's thesis, some interesting aspects of partial and imperfect testing which more clarification and better understanding are needed were unveiled.

- The procedures for determination of the partial stroke test coverage factor θ suggested in Lundteigen and Rausand (2008a) needs to be generalized to be applicable to different equipment.

- Adequate procedures for finding a realistic value of the full proof test coverage factor ξ or PTC as proportion of non-testable or non-detectable failures should be researched. In a case where the probabilistic value is considered, the appropriate probability value needs to be determined.

- There are some case cases where the partial tests results in degradation of components. An example is a case of a ball valve being partial tested by rotating to say 20 %, which could cause wear and building up of sediments which may affect the full closing of the valve. How can this degradation factor be considered in partial test calculation?

Bibliography

Aguilar Martinez, W. A. (2014). Methods for determining pfd/sil for workover control systems with short test-intervals and imperfect testing.

Ali, R., Goble, W., and Business, F. (2004). Smart positioners to predict health of esd valves. In *ANNUAL SYMPOSIUM ON INSTRUMENTATION FOR THE PROCESS INDUSTRIES*, volume 59, pages 29–38. INSTRUMENT SOCIETY OF AMERICA.

Brissaud, F., Barros, A., and Bérenguer, C. (2012). Probability of failure on demand of safety systems: impact of partial test distribution. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, page 1748006X12448142.

Bukowski, J. V. and Van Beurden, I. (2009). Impact of proof test effectiveness on safety instrumented system performance. In *Reliability and Maintainability Symposium, 2009. RAMS 2009. Annual*, pages 157–163. IEEE.

Cepin, M. (1995). Sequential versus staggered testing towards dynamic psa.

Chebila, M. and Innal, F. (2015). Generalized analytical expressions for safety instrumented systems' performance measures: Pfd avg and pfh. *Journal of Loss Prevention in the Process Industries*, 34:167–176.

David, R. and Alla, H. (2010). *Discrete, continuous, and hybrid Petri nets*. Springer Science & Business Media.

Dutuit, Y., Innal, F., Rauzy, A., and Signoret, J.-P. (2008). Probabilistic assessments in relationship with safety integrity levels by using fault trees. *Reliability Engineering & System Safety*, 93(12):1867–1876.

Grunt, O. and Briš, R. (2015). Spn as a tool for risk modeling of fires in process industries. *Journal of Loss Prevention in the Process Industries*.

Hauge, S., Lundteigen, M. A., Hokstad, P., and Håbrekke, S. (2013). Reliability prediction method for safety instrumented systems–pds method handbook, 2013 edition. *SINTEF report STF50 A*, 6031.

Hauge, S. and Onshus, T. (2006). *Reliability Data for Safety Instrumented Systems: PDS Data Handbook*. Sintef.

Hokstad, P. and Corneliussen, K. (2004). Loss of safety assessment and the iec 61508 standard. *Reliability Engineering & System Safety*, 83(1):111–120.

HSE-UK (2002). Principles for proof testing of safety instrumented systems in the chemical industry.

IEC-61508 (2009). Functional safety of electrical/electronic/programmable electronic safety-related systems: Parts 1 - 7.

IEC-61511 (2014). Functional safety—safety instrumented systems for the process industry, parts 1 - 3.

Innal, F., Dutuit, Y., and Chebila, M. (2015a). Safety and operational integrity evaluation and design optimization of safety instrumented systems. *Reliability Engineering & System Safety*, 134:32–50.

Innal, F., Liu, Y., Lundteigen, M., and Barros, A. (2015b). Pfdavg and pfh formulas for sis subject to partial and full periodic tests. *Reliability Engineering & System Safety*.

ISA-TR84.00.02 (2002). Safety instrumented functions sif—safety integrity level (sil) evaluation technique: Determining the sil of a sif via markov analysis.

ISA-TR84.00.03 (2002). Guidance for testing of process sector safety instrumented functions (sif) implemented as or within safety instrumented systems (sis).

Jin, H., Lundteigen, M. A., and Rausand, M. (2013). New pfh-formulas for k-out-of-n: F-systems. *Reliability Engineering & System Safety*, 111:112–118.

Jin, H. and Rausand, M. (2014). Reliability of safety-instrumented systems subject to partial testing and common-cause failures. *Reliability Engineering & System Safety*, 121:146–151.

Kumar, M., Verma, A. K., and Srividya, A. (2008). Modeling demand rate and imperfect proof-test and analysis of their effect on system safety. *Reliability Engineering & System Safety*, 93(11):1720–1729.

Lee, J. H., Chang, S. H., Yoon, W. H., and Hong, S. Y. (1990). Optimal test interval modeling of the nuclear safety system using the inherent unavailability and human error. *Nuclear Engineering and Design*, 122(1):339–348.

Liu, Y. and Rausand, M. (2013). Reliability effects of test strategies on safety-instrumented systems in different demand modes. *Reliability Engineering & System Safety*, 119:235–243.

Lundteigen, M. A. and Rausand, M. (2007). The effect of partial stroke testing on the reliability of safety valves. *ESREL'07*.

Lundteigen, M. A. and Rausand, M. (2008a). Partial stroke testing of process shutdown valves: How to determine the test coverage. *Journal of Loss Prevention in the Process Industries*, 21(6):579–588.

Lundteigen, M. A. and Rausand, M. (2008b). Spurious activation of safety instrumented systems in the oil and gas industry: Basic concepts and formulas. *Reliability Engineering and System Safety*, 93:1208–1217.

Mechri, W., Simon, C., and BenOthman, K. (2015). Switching markov chains for a holistic modeling of sis unavailability. *Reliability Engineering & System Safety*, 133:212–222.

Mechri, W., Simon, C., Bicking, F., and Othman, K. B. (2013). Fuzzy multiphase markov chains to handle uncertainties in safety systems performance assessment. *Journal of Loss Prevention in the Process Industries*, 26(4):594–604.

OLF-070 (2004). Norwegian oil and gas association application of iec 61508 and iec 61511 in the norwegian petroleum industry.

Oliveira, L. F. (2009). Pfd of higher-order configurations of sis with partial stroke testing capability. pages 1919–1928.

Oliveira, L. F. and Abramovitch, R. N. (2010). Extension of isa tr84. 00.02 pfd equations to koon architectures. *Reliability Engineering & System Safety*, 95(7):707–715.

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

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

Rauzy, A. (2008). Binary decision diagrams for reliability studies. In *Handbook of Performability Engineering*, pages 381–396. Springer.

Rolén, H. (2007). *Partial and imperfect testing of safety instrumented functions*. PhD thesis, Master thesis at NTNU.

Signoret, J.-P., Dutuit, Y., Cacheux, P.-J., Folleau, C., Collas, S., and Thomas, P. (2013). Make your petri nets understandable: Reliability block diagrams driven petri nets. *Reliability Engineering & System Safety*, 113:61–75.

Summers, A. and Zachary, B. (2000). Partial-stroke testing of safety block valves. *Control Engineering*, 47(12):87–89.

Torres-Echeverria, A., Martorell, S., and Thompson, H. (2009). Modelling and optimization of proof testing policies for safety instrumented systems. *Reliability Engineering & System Safety*, 94(4):838–854.

Torres-Echeverria, A. C. (2009). Modelling and optimization of safety instrumented systems based on dependability and cost measures.

Zhang, T., Wang, Y., and Xie, M. (2008). Analysis of the performance of safety-critical systems with diagnosis and periodic inspection. In *Reliability and Maintainability Symposium, 2008. RAMS 2008. Annual*, pages 143–148. IEEE.

Appendix A

Acronyms

CCF Common cause failure

DC Diagnostic coverage

EUC Equipment under control

FTA Fault tree analysis

MRT Mean restoration time

MTTR Mean time to repair

PFD Probability of failure on demand

PTC Proof test coverage

RBD Reliability block diagram

SIL Safety integrity level

SIS Safety instrumented system

SRS Safety requirement specification

Appendix B

Fault tree of the case study

The fault tree of the case study is a big one therefore has been divided into sub groups. The group shown here is for final control element (FC). The fault tree for the independent and CCFs for the other components (PT, TT and LS) are modeled the same way as the FC except that no partial tests are considered.

B.1 General part of the fault tree of the case study

This fault tree is shown in figure B.1

B.2 Fault tree for independent failures of the FC

This fault tree is shown in figure B.2

B.3 Fault tree of CCF for the final control (FC) element

This fault tree is shown in figure B.3

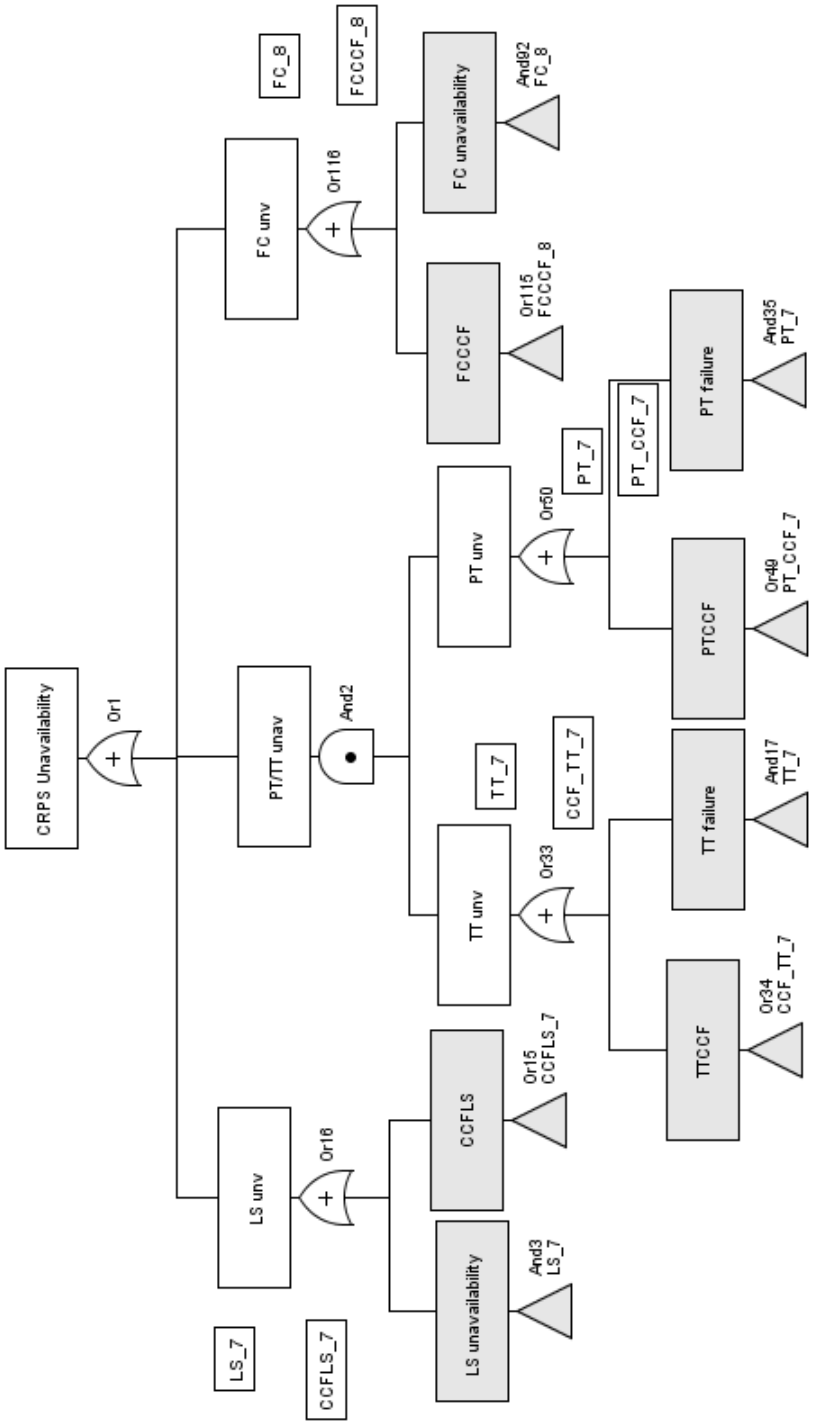


Figure B.1: General part of fault tree of the case study.

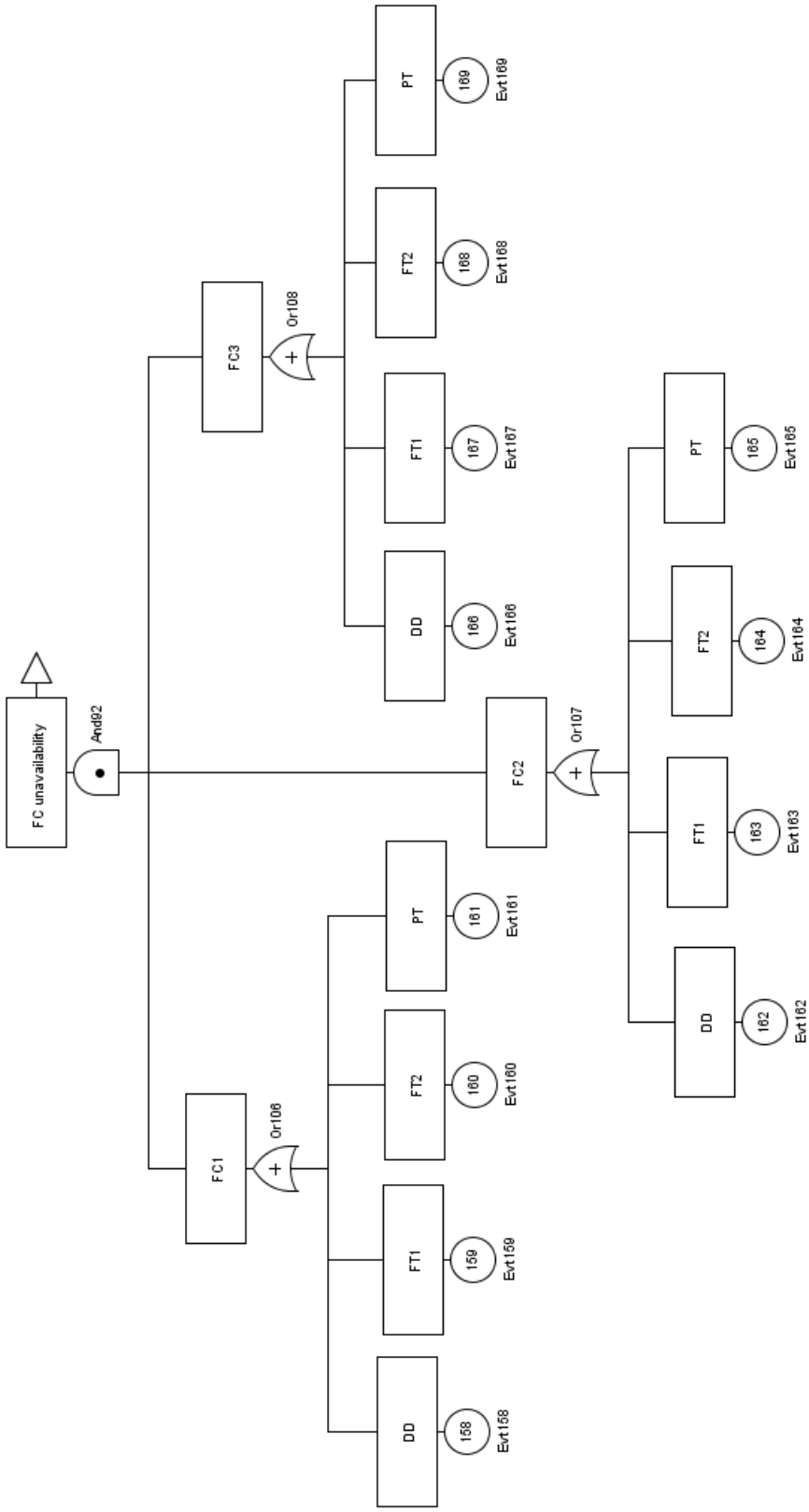


Figure B.2: Fault tree for independent failures of the FC.

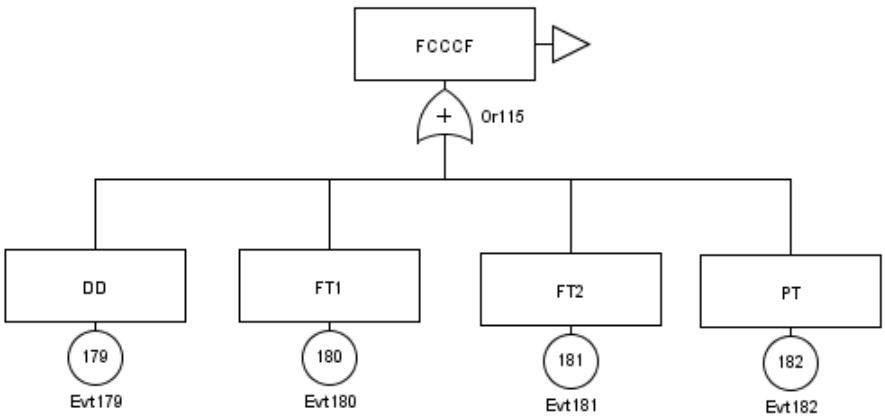


Figure B.3: Fault tree of CCF for the final control (FC) element.

Appendix C

Petri net model of the case study

The Petri net model of the case study was designed and simulated using the GRIF software. The picture in C.1 shows the representation for only two components (considering DD, FT1, FT2 failures), the test model and the common cause model. Note that this is only page 1 of 12 pages of the complete model. PT failure is further considered when modelling the final control elements.

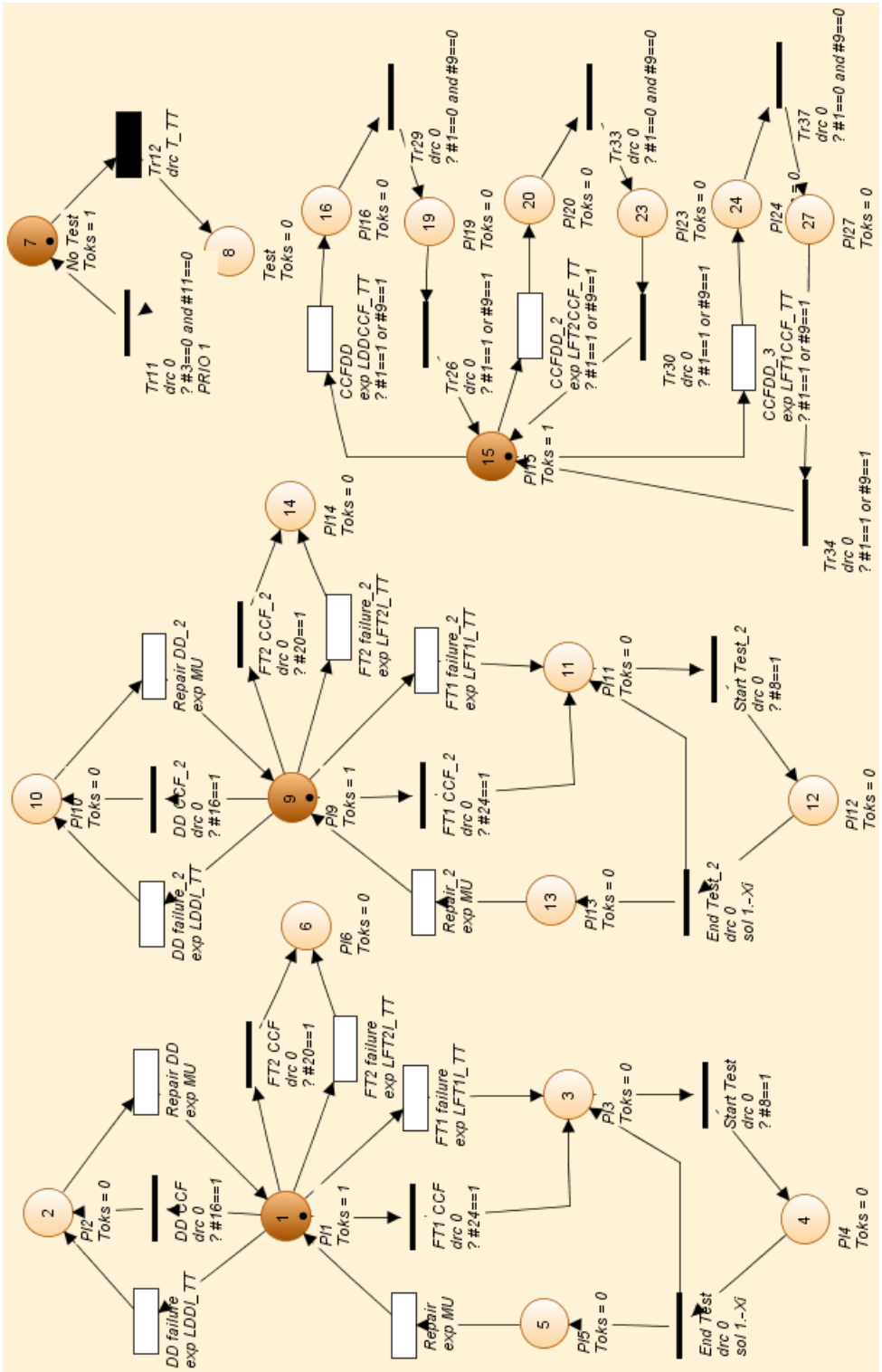


Figure C.1: Part of PN model the CRPS system.

Appendix D

The use of MAPLE software

A sample of the use of MAPLE for formula generation for different configuration is shown here. This is also used for calculations where necessary. Figure D.1 shows the Oliveira's formula generation code used in chapter 3. Similar codes have been used for formulas of different authors.

Oliveira's formula

```
> restart;
  k := 1;
  n := 3;
  S := 0;

  for i from 1 to n − k do

  S := S + (binomial(n, i)·(piecewise(i = 1, 1, i > 1, (1 − beta)))·lambda[DU])^i·τ^(i−1)·(tau/(i + 1) + MRT)·binomial(n − i, n − k + 1 − i)·((piecewise(n − k + 1 − i = 1, 1, n
      − k + 1 − i > 1, (1 − beta[D])))·lambda[DD]·MTTR)^(n − k + 1 − i)
  end do;
  PFDavg[koon] := S + binomial(n, n − k + 1)·((1 − beta)·lambda[DU])^(n − k + 1)·τ^(n − k)·(tau/(n − k + 2) + MRT) + binomial(n, n − k + 1)·((1 − beta[D])·lambda[DD]
      ·MTTR)^(n − k + 1) + beta·lambda[DU]·(tau/2 + MRT) + beta[D]·lambda[DD]·MTTR;
  evalf(PFDavg[koon])
```

$$k := 1$$

$$n := 3$$

$$S := 0$$

$$S := 3\,\lambda_{DU}\left(\frac{1}{2}\tau + MRT\right)\left(1-\beta_D\right)^2\lambda_{DD}^2 MTTR^2$$

$$S := 3\,\lambda_{DU}\left(\frac{1}{2}\tau + MRT\right)\left(1-\beta_D\right)^2\lambda_{DD}^2 MTTR^2 + 9\left(1-\beta\right)^2\lambda_{DU}^2\tau\left(\frac{1}{3}\tau + MRT\right)\lambda_{DD} MTTR$$

$$PFDavg_{koon} := 3\,\lambda_{DU}\left(\frac{1}{2}\tau + MRT\right)\left(1-\beta_D\right)^2\lambda_{DD}^2 MTTR^2 + 9\left(1-\beta\right)^2\lambda_{DU}^2\tau\left(\frac{1}{3}\tau + MRT\right)\lambda_{DD} MTTR + \left(1-\beta\right)^3\lambda_{DU}^3\tau^2\left(\frac{1}{4}\tau + MRT\right) + \left(1-\beta_D\right)^3\lambda_{DD}^3 MTTR^3$$
$$+ \beta\,\lambda_{DU}\left(\frac{1}{2}\tau + MRT\right) + \beta_D\lambda_{DD} MTTR$$

Figure D.1: Formula generation using MAPLE