Probabilistic Fault Isolation in Embedded Systems Using Training Data

AXEL CORNELL

Masters' Degree Project
Stockholm, Sweden April 2008

XR-EE-RT 2008:019
Abstract

In the heavy vehicle industry customers, laws and increasingly complex processes demand methods of supervising every aspect of a truck. Fault isolation systems are introduced to do just that. In order to assure a sustainable development new types of isolation systems are investigated to substitute the consistency based isolation systems of today.

In this thesis an application of a probabilistic isolation method that ranks possible faults on their likeness of being a fault in the process is implemented and evaluated as a possible future replacement of today’s system. This method bases the isolation on training data collected from measurements on the process and an observation of the process.

The probabilistic isolation method is evaluated on how it performs under different circumstances such as the effect of different amounts of training data and how well it performs if the tests and observations of the process are of varying quality.

Solution to several problems that arise when this method is implemented are also investigated such as how the system handles cases where several faults occur at the same time, what happens if there are missing data in the observations of the system and how to solve problems that involve execution times which is important in embedded systems.

The results that are derived show that this probabilistic isolation system performs well on the process as it is today and that this is a good substitute when developing for future processes. There is however a need for further development of the system such as improved isolation when there are several faults present in the process and questions on how to collect and store the training data still remain to be answered. A full scale implementation would allow for better comparison with the current system and give more information on runtime and storage problems.
Acknowledgements

This master thesis has been carried out at Scania CV AB and in particular the Powertrain Control System Development Diagnosis group. Thanks to everybody for accepting me into the group and for all support throughout the project.

Special thanks to Dan Hallgren and Anna Pernestål at the diagnosis group for sharing their knowledge and being patient as mentors to me during the work on this thesis.

Acknowledgements to Professor Bo Wahlberg at the Automatic Control group at the Royal Institute of Technology in Stockholm for his work and help as examiner of the thesis.

Last, but definitely least, I would like to thank my coworker Anders Sellhammer for his endless patience, hard work and friendship while working on his parallel thesis.
Contents

Abstract 1
Acknowledgements 1
Contents 2
Abbreviations 5
List of Figures 7
List of Tables 9

1 Introduction 10
   1.1 Background 10
   1.2 Objectives 11

2 Fault Isolation 12
   2.1 Fault Diagnosis Problem 12
   2.2 Components and Behavior Modes 13
   2.3 Observations 13
   2.4 Decision Structure 14
   2.5 Consistency Based Diagnosis Systems 14
   2.6 A Probabilistic Approach to Fault Isolation 15
   2.7 Training Data 15
   2.8 Computations of Probabilities 16
      2.8.1 Prior probability 17
      2.8.2 The Likelihood 17
      2.8.3 Normalization Factor 18
      2.8.4 Posterior Probability 18

3 Solving Real Problems 19
   3.1 Methods for Handling Incomplete Observations 19
      3.1.1 Assumption of a Value as a Solution to Lost Data 19
      3.1.2 Marginalization 21
      3.1.3 Elimination of Missing Data 23
      3.1.4 Treating Missing Values As a Third Value 24
3.1.5 Comparison of the Methods for Handling Incomplete Observations ........................................ 25
3.1.6 A Closer Comparison of Marginalization and Elimination of Lost Data .................................. 26
3.2 The Use of Subsystems ............................................. 28
  3.2.1 The Advantages of Subsystems For Storage and Isolation ...................................................... 29
  3.2.2 The Advantages of Subsystems For Collecting Training Data .................................................. 29
3.3 Handling Multiple Modes In the Process Under Diagnosis ............................................................... 29
  3.3.1 Multiple Modes In One Subsystem ............................................................................................ 30
  3.3.2 Known Multiple Modes In a Subsystem ..................................................................................... 31

4 Performance Measures .................................................................................................................... 33
  4.1 Expected Correctness .................................................................................................................... 34
  4.2 Expected Probability of Correct Classification .............................................................................. 35
  4.3 Expected Ratio of the Underlying Probability and the Most Probable Wrong Estimation ............. 35

5 Evaluation of the Diagnosis System ................................................................................................. 37
  5.1 Experimental Setup ...................................................................................................................... 37
  5.2 Amount of Training Data .............................................................................................................. 39
  5.3 Robustness to Probabilities of False Positive Observations and False Negative Observations .... 43
  5.4 Robustness With Respect to the Priors ......................................................................................... 49
  5.5 Evaluation of the Methods for Handling Missing Data ................................................................. 56
  5.6 What Happens If There Is a Double Fault .................................................................................... 62

6 Taking Prior Information Into Account ............................................................................................... 67
  6.1 Weighing the Two Methods ......................................................................................................... 67
  6.2 Evaluation of the Combined Method ............................................................................................. 69
  6.3 Training Data Compensates for Wrong Models .......................................................................... 71

7 Evaluation of the Practical Aspects of an Implementation .................................................................. 76
  7.1 Important Parameters and Data .................................................................................................... 76
    7.1.1 Training Data ......................................................................................................................... 76
    7.1.2 Prior Probabilities .................................................................................................................. 77
    7.1.3 Decision Structure and Assumed Probabilities of False Positive and Negative .................. 77
  7.2 Risks and Difficulties .................................................................................................................... 77
    7.2.1 Wrong Model ....................................................................................................................... 78
    7.2.2 Double Faults ......................................................................................................................... 78
    7.2.3 Collecting Training Data ......................................................................................................... 78
    7.2.4 Storing Training Data ............................................................................................................ 79
    7.2.5 Runtime ............................................................................................................................... 79

3
8 Conclusions, Recommendations and Future Work 80
  8.1 Conclusions ...................................................... 80
  8.2 Recommendations ............................................. 81
  8.3 Future Work ................................................... 81

A Variables and Notation 82

B Probability Rules 83
  B.1 Basic Rules .................................................... 83
  B.2 Bayes Rule ..................................................... 83
  B.3 Marginalization ................................................ 83

Bibliography 84
Abbreviations

ECU  Electronic Control Unit
CDF  Cumulative Distribution Function
CCDF Complementary Cumulative Distribution Function
FTC  Fault Tolerant Control

The used variables are defined in Appendix A
# List of Figures

2.1 Overview of the diagnosis system ............................... 12

3.1 Results to the same isolation with different methods for handling missing data in the observation .......................... 25

5.1 Plot of $\mu$ and $\nu$ for Varying Amounts of Training Data $N$ for Each Mode ......................................................... 40

5.2 Plot of the Probability that the Correct Mode has a Probability Greater than $x$ for Varying Amounts of Training Data $N$ for Each Mode, Also Called the Complementary Cumulative Distribution Function ........................................ 41

5.3 Plot of $\mu$ and $\nu$ for Different Probabilities of False Alarms and Missed Alarms ..................................................... 44

5.4 Plot of the CCDF of the Correct Underlying Modes Probability for Different Probabilities of False Alarms and Missed Alarms ..... 45

5.5 Plot of $\mu$ and $\nu$ for Different Probabilities of False Alarms and Missed Alarms ..................................................... 47

5.6 Plot of the CCDF of the Correct Underlying Modes Probability for Different Probabilities of False Alarms and Missed Alarms ..... 48

5.7 Plot of $\mu$ and $\nu$ for Different Prior Probabilities ............. 50

5.8 Plot of the CCDF of the Correct Underlying Modes Probability for Different Prior Probabilities ..................................... 51

5.9 Plot of $\mu$ and $\nu$ for Different Prior Probabilities ............. 52

5.10 Plot of the CCDF of the Correct Underlying Modes Probability for Different Prior Probabilities ................................. 53

5.11 Plot of $\mu$ and $\nu$ for Prior Probabilities $p_1$ and $p_{10}$ for System 1 ......................................................... 54

5.12 Plot of $\mu$ and $\nu$ for Prior Probabilities $p_1$ and $p_{10}$ for System 1 ......................................................... 54

5.13 Plot of $\mu$ and $\nu$ for Isolating Observations with Missing Data Using Marginalization and Elimination of Missing Data ........ 57

5.14 Plot of the CCDF of the Correct Underlying Modes Probability for Isolating Observations with Missing Data Using Marginalization and Elimination of Missing Data .......................... 58

5.15 Plot of $\mu$ and $\nu$ for Isolating Observations with Missing Data Assuming the Missing Values to Either 0 or 1 ................. 59
List of Tables

2.1 A typical decision structure ........................................ 14
2.2 An Example of Typical Training Data ............................... 16
3.1 An example of a larger decision structure that can be partitioned into two smaller submatrices ........................................ 28
3.2 The decision structure for the process ................................ 31
5.1 The General Setup for the Experiments in Chapter 5 ............ 38
5.2 The First of the Two Decision Structures Used in the Evaluation .................................................. 38
5.3 The Second of the Two Decision Structures Used in the Evaluation .................................................. 38
5.4 Results to Simulations with Varying Amount of Training Data .................................................. 39
5.5 Results from Figure 5.2 Showing the Probability that the Probability of the Correct Modes is Greater Than 70% ............ 42
5.6 The Probabilities of False Positive and False Negative That Are Evaluated For .................................................. 43
5.7 The Probabilities of False Positive and False Negative That Are Evaluated For .................................................. 43
5.8 Results to Simulations with Different Probabilities for False Alarm $p_{fp}$ and Missed Alarm $p_{fn}$ .................................................. 44
5.9 Results from Figure 5.4 Showing the Probability that the Probability of the Correct Modes is Greater Than 70% ............ 46
5.10 The Different Prior Probabilities That Are Simulated for .................................................. 49
5.11 Results to Simulations with Different Prior Probabilities ........ 50
5.12 Results to Simulations with Different Prior Probabilities ........ 51
5.13 Results to Simulations with Different Methods For Handling Missing Data in The Observation for Both $l = 1$ and $l = 2$ Missing Data .................................................. 56
5.14 The Probabilities of False Positive and False Negative Observations That Are Evaluated For .................................................. 62
6.1 Information About the Four Different Observations in the Experiment on the Amount of Needed Training Data ......... 69
6.2 Results from Figure 6.3 Showing the Probability that the Probability of the Correct Modes is Greater Than 70% ............ 71
6.3 The First Decision Structures Modified for Experiment With Wrong Model ............................................. 72
6.4 The Second Decision Structures Modified for Experiment With Wrong Model ............................................. 73
A.1 Variables as we use them in this thesis ................................. 82
Chapter 1

Introduction

1.1 Background

The heavy truck industry is driven mainly by customer demands. Customers demand more and more uptime and longer life cycles of their heavy truck fleets. Any defects must be found and breakdowns quickly repaired, either by fault tolerant control, FTC, or by reparations by a mechanic.

Not only customers demand better performances but law requirements on both a national and international basis. Requirements of lower and lower emissions force the industry to spend large amounts of resources on research and development. Not only to maintain low emissions but also to find faults in the truck so that emission standards can be met and to assure safety on the roads. Today's generation of isolation systems in trucks allow manufacturers to meet the standards of today but newer and more precise isolation systems will allow a sustainable development and to meet future requirements.

As engines become more and more complex processes it becomes more difficult to monitor exactly what goes on in the engine. As an effect new methods of isolating faults in heavy trucks are continuously developed.

In this thesis an application of a probabilistic fault isolation that ranks possible faults on their likeliness of being present faults is implemented and investigated as a possible substitute to today's isolation system. The method bases the isolation on collected training data so that the model that the isolation is based on is created not by engineers but a self-generated model created from collected data of the real process.

Research has been conducted on the subject and this thesis is primarily an application of work in [1].
1.2 Objectives

The objectives with this thesis are:

• An applied version of a probabilistic isolation system based on training data is to be implemented and tested in an embedded system with the purpose to enlighten any problems that are associated with an implementation of this kind.

• An investigation on the probabilistic isolation system should be conducted to test the robustness of the system with respect to different parameters such as amount of training data, quality of tests on the process, two faults occurring simultaneously and missing values from any tests on the process.

• A discussion to show the bottlenecks of an implementation of the isolation system should be given to serve as basis for recommendations for an implementation.

• Recommendations on how an implementation of the probabilistic isolation system can be performed should be given to lay as base to a decision on whether or not to persist development of an isolation system of this kind.
Chapter 2

Fault Isolation

To get an introduction to the content in this thesis the following sections in this chapter will discuss the terms fault diagnosis problem, component, behavior mode, observations, training data, background information and decision structure. The way of thinking in probabilistic terms is also introduced in this chapter. The chapter is based on work in [1].

![Diagram of the diagnosis system](image)

Figure 2.1: Overview of the diagnosis system

2.1 Fault Diagnosis Problem

The idea with fault diagnosis is to detect and isolate any fault or abnormal behavior present in a process under diagnosis. The fault diagnosis procedure can be divided into three parts.

- **Fault detection** – Determination if there are any faults present in the process at the present time.
- **Fault isolation** – Determination of location, kind and time of detection of a fault.
- **Fault estimation** – Determination of the size and time variant behavior of a fault.
In this thesis the focus is on the first two parts, detection and, mainly, isolation.

In order to clarify the fault isolation it is important to know what is meant by a fault. Therefore we define a fault:

**Definition 2.1 (Fault)**

A *fault* is an unpermitted deviation of at least one characteristic property or variable of the process from acceptable, usual, standard or nominal behavior.

### 2.2 Components and Behavior Modes

Components are physical objects in the process under diagnosis such as sensors, actuators, valves, pipes, and cables. All components can function in many different ways, so for describing the status of the component at any given time, the components have a set of behavior modes. In this way every known and possible deviation of the component from 'normal' are connected to a certain behavior mode. The present behavior mode of a component is denoted $M_i = m_i$ where $M_i$ is an arbitrary variable and $m_i$ is its value.

The most common behavior modes is no fault, $nf$. Another common behavior mode is unknown or undefined fault, $uf$, for a component malfunctioning in a previously unknown or undefined way. A component is restricted to one behavior mode at one point in time.

An assignment of behavior modes to all components in the process is called a process behavior mode and will be denoted $M$. The process behavior mode is a set of all modes present in the process. A special notation, $NF$, is also introduced to denote that the whole process is functioning as it should, i.e. no faults are present anywhere in the process.

It is the behavior modes that are present in the process that the isolation of the process is supposed to find.

### 2.3 Observations

An observation is an output from sensors and tests on the process under diagnosis measuring different values on components. These values are translated into binary signals, 0 or 1, depending on whether or not they exceed a given threshold. The isolation system uses the observation to isolate any faults that arise in the process.

In the fault free case the raw values that are read of sensors or tests are designed to have a constant value, generally zero, but due to noise and model errors this cannot be guaranteed. This signal error is the reason that thresholds are needed. If the signal passes the threshold the test gives an alarm and the corresponding value in the observation becomes 1, otherwise the value is 0.

In this thesis the observation is denoted $X$ and contains as many elements as there are tests, $X = (X_1, X_2, \ldots, X_q)$. Each element in the observation takes the value of either $X_i = 0$ or $X_i = 1$. There is of course a possibility that the values in the observation take more than binary values, i.e. that there
are more than one threshold but this thesis treats the binary case only. The methods in this thesis are however easily adapted to handle more thresholds in the observation.

Figure 2.1 shows an overview of the isolation with the output from the isolation system denoted as candidate.

2.4 Decision Structure

A decision structure, denoted $S$, is used to represent the relationship between the observation from the tests and the different behavior modes. It is displayed as a matrix with $q$ rows and $r$ columns. A row corresponds to a test and a column to a component or fault. Each entry in the decision structure is in other words related to one test and one behavior mode. An $X$ in row $i$ and column $j$ in the decision structure means that test $i$ is affected by a fault in component $j$ and a zero means that no dependencies exists between the test and the component or the fault. Table 2.1 shows an example of a decision structure. In this way all the dependencies between a test and a fault can be illustrated in a visible and intuitive way.

Sometimes the term incidence structure is used, the difference is that incidence structure show all faults that can cause a test to give an alarm while a decision structure only show relationships where the designer knows relationships exists. The designer may also ignore dependencies it they are considered obsolete. It is hence the decision structure that is used.

2.5 Consistency Based Diagnosis Systems

Many diagnosis systems used in industry today are consistency based diagnosis systems. These systems are based on logic reasoning, 'Which mode can be present given the observation?', 'Which mode cannot be present given the observation?'.

The input to such a system is the same as the input to the probabilistic isolation system described in Section 2.6, binary outputs from tests and sensor readings on the process under investigation.

The output is what characterizes the difference between the probabilistic fault isolation and the consistency based isolation system. In consistency based
isolation systems the output is an unranked list of possible faults as opposed to a list with possible faults ranked on their probability of being the underlying mode in the probabilistic isolation.

2.6 A Probabilistic Approach to Fault Isolation

Research is conducted in the area of isolating faults present in a process with a probabilistic approach. This means that behavior modes are ranked on a likeliness to be the underlying mode of the process. This thesis builds on research from [1]. An introduction to this work is given in the following sections. For any further explanations please refer to this work.

2.7 Training Data

While consistency based fault isolation methods are based only on the decision structure, probabilistic fault isolation also uses training data. Training data are real observations collected from the process under diagnosis. The process under diagnosis is set in a certain process behavior mode and training data is collected by reading off the resulting observations. Unfortunately it is not always possible to collect training data for all possible modes, a method for dealing with cases when there is no training data is investigated in the parallel work [2]. However, in this thesis it will be assumed that training data is available. Training data is denoted $D$ and contains the set of all training data samples available.

One of the great benefits of using training data is that it could show dependencies between different faults and observations that are not visible in the decision structure. In the decision structure there will however be unmodeled dependencies that are either not known or ignored. Training data will show these and thus allow for better isolation when trying to figure out the underlying mode of the process. For example if, when collecting training data, the process is in mode $m_1$ which would according to a given decision structure activate an alarm from test $X_1$ but it actually also activates an alarm in $X_2$ then training data will help to show this and takes this into account when alarms are set off in both $X_1$ and $X_2$ and thus makes mode $m_1$ more probable.

To illustrate, an example of how training data can look like if the process contains of 4 different process behavior modes and 5 tests as in the decision structure in Table 2.1. In this example the total amount of training data is 100 (see Table 2.2).

In contrast to the decision structure training data is measured from the process and is not known and cannot be assumed by a system engineer. Training data never contains any guesses or assumptions about the process.
### Table 2.2: An Example of Typical Training Data

<table>
<thead>
<tr>
<th>fault in ( m )</th>
<th>( X = (X_1, X_2, X_3, X_4, X_5) ) ( n_m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_1 )</td>
<td>((1, 0, 0, 1, 0))</td>
</tr>
<tr>
<td>( m_1 )</td>
<td>((1, 0, 0, 0, 0))</td>
</tr>
<tr>
<td>( m_1 )</td>
<td>((0, 0, 0, 1, 0))</td>
</tr>
<tr>
<td>( m_1 )</td>
<td>other values</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>((0, 1, 1, 1, 0))</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>((0, 1, 1, 0, 0))</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>((0, 0, 1, 1, 0))</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>((0, 1, 0, 0, 0))</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>other values</td>
</tr>
<tr>
<td>( m_3 )</td>
<td>((1, 1, 0, 0, 1))</td>
</tr>
<tr>
<td>( m_3 )</td>
<td>((1, 1, 0, 0, 0))</td>
</tr>
<tr>
<td>( m_3 )</td>
<td>((1, 0, 0, 0, 1))</td>
</tr>
<tr>
<td>( m_3 )</td>
<td>((0, 1, 0, 0, 0))</td>
</tr>
<tr>
<td>( m_3 )</td>
<td>other values</td>
</tr>
<tr>
<td>( m_4 )</td>
<td>((0, 0, 1, 0, 1))</td>
</tr>
<tr>
<td>( m_4 )</td>
<td>((0, 0, 0, 0, 1))</td>
</tr>
<tr>
<td>( m_4 )</td>
<td>other values</td>
</tr>
</tbody>
</table>

#### 2.8 Computations of Probabilities

The probabilistic approach to fault isolation calculates the probability that a specific mode is the underlying mode present in the process given information about a process and an observation. This probability is called the posterior probability and is shown in Equation (2.1).

\[
p(M = m \mid X = x, D, I)
\]

where \( I \) denotes the prior information about the process, \( D \) denotes the training data, \( X \) is the observation and \( M \) is the mode for which the posterior probability is calculated.

The procedure of calculating the posterior probability begins by rewriting Equation (2.1) using Bayes Rule stated in Appendix B to obtain

\[
p(M = m \mid X = x, D, I) = \frac{p(X = x \mid M = m, D, I)p(M = m \mid D, I)}{p(X = x \mid D, I)}
\]

Consider the factors on the right hand side of Equation (2.2). The first factor in the numerator, \( p(X = x \mid M = m, D, I) \), is the likelihood, the second, \( p(M = m \mid D, I) \), is the prior probability and the denominator is a normalization constant and is from here on denoted \( \pi_X \). The following sections will discuss each of these factors and are based on [1] and [3].
2.8.1 Prior probability

The prior probability only depends on prior information that is known about the process from earlier experience and assumptions. Based on this prior knowledge the prior probability for different modes can be calculated. This means that the prior probabilities for the different modes are the same as the knowledge for how common a certain mode is in the process. Hence the priors must be given by the system designer so that they are as close to what is believed to be the real values as possible.

The training data collected when the process is manually set to operate in a certain mode, does not provide any further information about how common a mode is. This means that the factor \( p(M = m \mid D, I) \) can be rewritten to \( p(M = m \mid I) \) which is the prior probability.

Example 2.1 (Prior Information and Prior Probability)

Consider a problem where you have an urn with 10 balls. The balls can either be red or blue and you want to determine the probability of picking a red ball. Knowing only this it is hard to say the probability that the first ball you pick is red, \( p(R \mid I_0) \). This because the prior information in this case is only that there are 10 balls of two different colors in the urn. Now consider that you have an urn with 10 balls, where you know that six of them are blue and four of them are red. Now, the challenge of computing the probability \( p(R \mid I_1) = 0.4 \) is no longer that hard.

This means that by knowing more about the problem or the process (more prior information) it is possible to assign a more precise prior probability.

2.8.2 The Likelihood

The likelihood, can be seen as the sampling distribution of the observation \( X \) for a fixed mode \( M \). This is a measure of how common a certain observation \( X \) is when the process is in a specific mode \( M \). Under the condition that the observation vector \( X \) is discrete with \( K \) possible values, such that \( X \in \{1, 2, \ldots, K\} \), the likelihood will be discrete. This discrete likelihood is parameterized by \( K \), for each process behavior mode \( M \). Using marginalization over all possible parameters and the Dirichlet distribution as the prior distribution for the parameters the discrete likelihood becomes:

\[
p(X = x \mid M = m, D, I) = \frac{n^m_x + \alpha^m_x}{N^m + A^m}
\]

where \( n^m_x \) is the amount of samples for \( X = x \) existing in the training data for mode \( m \), \( N^m \) is the total amount of training data for mode \( m \), \( N^m = \sum_x n^m_x \). \( \alpha^m_x \) is the apriori distribution for the observation \( X = x \) in mode \( m \) and \( A^m \) is the amount of possible combination of observation \( X \), \( A^m = \sum_x \alpha^m_x \).
For a more thorough explanation of the derivation of this equation for the likelihood refer to [1] and [8]

The likelihood is not itself a probability, it is a dimensionless numerical function which, when multiplied by a prior probability and a normalization factor, becomes a probability. How the likelihood is computed is explained in the following example.

**Example 2.2 (Likelihood Computation)**

Assume that for mode \( M = m \) there is 100 training data available. The observation vector consists of the result from 5 test which can be either 1 or 0 which leads to the total amount of possible combination \( 2^5 = 32 \). In this example, the a priori information, \( \alpha^m \), equals \( \forall x, m \).

For a specific observation \( X = x \), there is 43 training data of the total 100 that matches the observation \( X \). From this information the likelihood becomes:

\[
p(X = x \mid M = m, D, I) = \frac{n_x^m + \alpha^m}{N^m + A^m} = \frac{43 + 1}{100 + 32} = \frac{1}{3} \approx 0.33
\]  

(2.4)

**2.8.3 Normalization Factor**

The normalization factor is a constant that can be computed using marginalization (Equation (B.4)) over all modes. The reason for the normalization factor is to normalize the calculated values for how common each mode is. By doing this the sum of the values for each mode sums to one and is thereby a probability of how likely each mode is of being the underlying mode. If the normalization factor is not used it is merely a measure of how likely a mode is relatively to the other modes without possibility of saying how common it is in comparison to all modes.

\[
\pi_X = p(X \mid D, I) = \sum_M p(X \mid M, D, I)p(M \mid I)
\]

(2.5)

**2.8.4 Posterior Probability**

By using the information discussed earlier, the posterior probability, Equation (2.1) can be computed. The posterior is the final product that the above calculations have been intended to lead to. This is the final measure of the probability of a mode given a process and an observation. The set of posterior probabilities for all modes is what was previously referred to as the candidate.

\[
p(M = m \mid X = x, D, I) = \frac{p(M = m \mid I) n_x^m + \alpha^m}{\pi_X} \frac{N^m + A^m}{N^m + A^m}
\]

(2.6)
Chapter 3

Solving Real Problems

In this chapter we will discuss the application of new solutions to the fault isolation in diesel engines. . . .

3.1 Methods for Handling Incomplete Observations

When an observation $X$ received from the process under diagnosis does not contain results from all the tests, i.e. a test failed to give a result or not finished performing the test, different approaches can be taken. In this section a discussion of different methods will be discussed, first the possibility of assuming what the missing value was will be investigated, then two methods called marginalization and elimination of missing data are introduced. A final method where a missing value is treated as a value is mentioned but not used in this thesis.

There can be several reasons for a test not to have given a result when the isolation begins, some of these cases are listed below:

- A test might give a result only once, i.e. at startup and then not give any more results after that.

- A test might only be able to give a valid result under certain conditions, for example when something has a certain temperature, pressure or likewise.

- A test might need more time to give a result than for the isolation to run so that when the isolation starts another cycle the test has not given an updated result.

3.1.1 Assumption of a Value as a Solution to Lost Data

The approach of assuming a result when there are missing results means that when you receive an incomplete observation you want to assume a value for
the specific values that are missing. This can be considered to be a very naive approach but it can also be very effective.

This approach is of course the most effective if the assumption is correct but terrible bad if the assumption is wrong. It is therefore important to be sure before assuming a value for this missing data. The usefulness of this method of handling missing data can be debated as it is hard to ever be sure of a missing value. It is however possible to imagine this method to be a good option at the time when a test fails to give a result because it is slower than the isolation and when at the same time the last result was fault free. Thus assuming the last known value of the test. If the assumption is wrong then the next time the test is updated it will give a fault and isolation will be corrected. As a result the correct isolation is only given a time delay.

To assume that the missing test result was a 1 over assuming a 0 is that it might be preferred to assume the worst case scenario.

**Example 3.1 (Assumption of a value in an Incomplete Observation)**

Assume we have the same decision structure as in Table 2.1, and the same training data that is shown in Table 2.2. It is assumed that the priors are \( p(M = m_i | I) \) = 0.25, i.e. the same probability for all modes and that training data was collected with \( N = 100 \) samples for each mode. Each observation is a vector with five elements, each element is a binary test result, a 1 if there is an alarm or a 0 if there is not. This example will illustrate what happens if either a 0 or a 1 is assumed when the observation is missing data for test \( x_3 \).

\[
X = \begin{pmatrix}
    0 \\
    1 \\
    - \\
    0 \\
    0
\end{pmatrix}
\]

If the last known value of \( x_3 \) was 0:

\[
p(M = m_i | X(x_3 = 0), D, I) = \frac{p(M = m_i | I) n_{m_i}^{x_3} + \alpha_k^{m_i}}{\pi_{X(x_3=0)} N^{m_i} + A^{m_i}} =
\]

\[
= \begin{cases}
    1/14 = 0.071 & M = m_1 \\
    5/14 = 0.357 & M = m_2 \\
    7/14 = 0.500 & M = m_3 \\
    1/14 = 0.071 & M = m_4
\end{cases} \quad (3.1)
\]

If the last known value of \( x_3 \) was 1:

\[
p(M = m_i | X(x_3 = 1), D, I) = \frac{p(M = m_i | I) n_{m_i}^{x_3} + \alpha_k^{m_i}}{\pi_{X(x_3=1)} N^{m_i} + A^{m_i}} =
\]

\[
= \begin{cases}
    1/27 = 0.037 & M = m_1 \\
    24/27 = 0.889 & M = m_2 \\
    1/27 = 0.037 & M = m_3 \\
    1/27 = 0.037 & M = m_4
\end{cases} \quad (3.2)
\]
This example shows how different the results of assuming different values to a missing value can be. The results are also showed in Figure 3.1 later in this section.

### 3.1.2 Marginalization

Another method to use for handling missing data in an observation is to use marginalization.

If an observation $X$ is received where value $X_-$ is missing, $X_-$ can take values 0 or 1 and if the rest of the observation that is known is denoted $X_r = x_r$ then the marginalization becomes:

\[
p(M = m \mid X = x, D, I) = p(M = m \mid X_r = x_r, X_- = 0, D, I)p(X_- = 0 \mid X_r = x_r, D, I) + p(M = m \mid X_r = x_r, X_- = 1, D, I)p(X_- = 1 \mid X_r = x_r, D, I)
\]

(3.3)

The value of $p(M = m \mid X_r = x_r, X_2 = x_2, D, I)$ is calculated using the same calculation that is used in the case of a complete observation, Equation (2.6) in Section 2.8.4. The other product, $p(X_- = x_- \mid X_r = x_r, D, I)$, is calculated as:

\[
p(X_- = x_- \mid X_r = x_r, D, I) = \frac{p(X_r = x_r, X_- = x_- \mid D, I)}{p(X_r = x_r \mid D, I)} = \frac{p(X_r = x_r, X_- = x_- \mid D, I)}{\sum_{X_-} p(X_r = x_r, X_- = x_- \mid D, I)}
\]

(3.4)

This is a weighting factor to weigh the probability, depending on whether $X_-$ is 0 or 1, toward the one which is more common in the training data.

The likelihood $p(X_r = x_r, X_- = x_- \mid M = m, D, I)$, has been calculated in Section 2.8.2. The difference here is that the condition $M = m$ is not given, the calculations are therefore made with training data from all modes,

\[
p(X_r = x_r, X_- = x_- \mid D, I) = \frac{n_{x_r,x_-} + \alpha_{x_r,x_-}}{N + A}
\]

(3.5)

where $n_{x_r,x_-}$ is the number of simultaneous occurrences of $X_r = x_r$ and $X_- = x_-$ in all training data and $\alpha_{x_r,x_-}$ is a parameter chosen by the designer, a prior value of $X_r = x_r$ and $X_- = x_-$. $N$ is the total amount of training data and $A$ is the number of possible combinations. The denominator in Equation (3.4) is the normalization factor and is denoted $\pi^r_{x_r,x_-}$ to differentiate it from the normalization factor in Section 2.8.3 which is calculated as a sum of the prior probability from all different modes for a given $X$ while the normalization factor here is calculated for all different possibilities of the missing value over all training data (i.e. all modes at the same time).
The equation used for marginalization therefore becomes:

\[
p(M = m \mid X = x, D, I) = \sum_{x_3} p(M = m \mid X_r = x_r, X_3 = x_3, D, I) \frac{1}{\pi_{x_r}^{x_3} + \alpha_{x_r}^{x_3}} n_{x_r, x_3} + \alpha_{x_r, x_3} = \sum_{x_3} p(M = m \mid I) \frac{n_{x_r, x_3}^{m} + \alpha_{x_r, x_3}^{m}}{\pi_{x_r}^{x_3} + \alpha_{x_r}^{x_3}} \frac{1}{N^m + A^m} n_{x_r, x_3} + \alpha_{x_r, x_3} = \sum_{x_3} p(M = m \mid I) \frac{n_{x_r, x_3}^{m} + \alpha_{x_r, x_3}^{m}}{N^m + A^m} \frac{1}{\pi_{x_r}^{x_3} + \alpha_{x_r}^{x_3}} n_{x_r, x_3} + \alpha_{x_r, x_3} = \sum_{x_3} p(M = m \mid I) \frac{n_{x_r, x_3}^{m} + \alpha_{x_r, x_3}^{m}}{N^m + A^m} \frac{1}{\pi_{x_r}^{x_3} + \alpha_{x_r}^{x_3}} n_{x_r, x_3} + \alpha_{x_r, x_3}
\]

(3.6)

When more than one value is missing the same method is applied but with one term in this sum for each possible combination of the missing values. This means that for each missing value the number of computations that must be made are doubled. If there are \(n\) missing values then there must be \(2^n\) computations. The easiest solution when creating a function in the diagnosis system that tries every combination is a function that finds each possible combination recursively. It could also be solved in a way where the possible combinations of the missing data has been programmed at forehand. This method is however not as neat as there can be a large number of missing data and coding every combination for every number of missing data can, theoretically, be an infinite work unless it is known how many values that will be missing at the most.

**Example 3.2** (Use of Marginalization in an Incomplete Observation)

The same setup as in Example 3.1 is used. To calculate the probability \(p(M = m \mid X, D, I)\) we use marginalization given in Equation (3.3):

\[
p(M = m \mid X, D, I) = p(M = m \mid X_r = x_r, X_3 = 0, D, I)p(X_3 = 0 \mid X_r = x_r, D, I) + p(M = m \mid X_r = x_r, X_3 = 1, D, I)p(X_3 = 1 \mid X_r = x_r, D, I)
\]

The calculations of \(p(M = m \mid X_r = x_r, X_3 = 0, D, I)\) and \(p(M = m \mid X_r = x_r, X_3 = 1, D, I)\) are made just the same way as for complete calculations. These calculations are already made in Section (3.1.1).

For the calculations of \(p(X_3 = 0 \mid X_r = x_r, D, I)\) and \(p(X_3 = 1 \mid X_r = x_r, D, I)\) we use the Equations (3.4) and (3.5):

\[
p(X_3 = 0, X_r = x_r \mid D, I) = \frac{n_{x_r, x_3=0} + \alpha_{x_r, x_3=0}}{N + A} = \frac{10 + 1}{400 + 32} = \frac{11}{432}
\]

\[
p(X_3 = 1, X_r = x_r \mid D, I) = \frac{n_{x_r, x_3=1} + \alpha_{x_r, x_3=1}}{N + A} = \frac{23 + 1}{400 + 32} = \frac{24}{432}
\]

\[
p(X_3 = 0 \mid X_r = x_r, D, I) = \frac{11}{11 + 24} = \frac{11}{35}
\]

\[
p(X_3 = 1 \mid X_r = x_r, D, I) = \frac{24}{35}
\]

22
The final probabilities become:

\[
p(M = m \mid X = x, D, I) =
\begin{align*}
1/14 \times 11/35 + 1/27 \times 24/35 &= 0.048 & M = m_1 \\
5/14 \times 11/35 + 24/27 \times 24/35 &= 0.722 & M = m_2 \\
7/14 \times 11/35 + 1/27 \times 24/35 &= 0.183 & M = m_3 \\
1/14 \times 11/35 + 1/27 \times 24/35 &= 0.048 & M = m_4
\end{align*}
\]  

(3.7)

The result can also be seen in Figure 3.1 where it is compared against the other methods of handling missing data.

### 3.1.3 Elimination of Missing Data

This approach is similar to the marginalization in Section 3.1.2. Exactly how these methods compare is explained in Section 3.1.6. The best benefit compared to the marginalization is that it is easily solved without recursion while in marginalization when more than one value in the observation is missing recursion simplifies the calculations a lot. The negative thing is that it thought not to give as good result as marginalization. The hope is that the result is not too different from the result calculated using marginalization so that elimination of missing data can be used instead.

The method is as follows: when an observation, \( X \), is received with one or several missing data, a new observation is created that only contains data from the received observation that is not missing, i.e. the elements in the received observation vector that not contain any values are eliminated. Then the probability for this new observation, called \( X_r = x_r \), is then calculated. When the training data for this observation is examined then all observations in the training data for one mode that contain \( x_r \) are grouped together to give one number:

\[
\text{if } x = [x_r, x_-] \\
n_{x_r}^m = \sum_{x_- = 0, 1} n_{x_r}^m
\]

**Example 3.3** (Adding training data)

The training data from Table 2.2 is collected and the observation \( X = (0, 1, -1, 0, 0) \) is received. Assume that \( n_{x_r}^2 \) needs to be calculated. The relevant training data is then:

<table>
<thead>
<tr>
<th>fault in ( m )</th>
<th>( X = (X_1, X_2, X_3, X_4, X_5) )</th>
<th>( n_{x_r}^m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_2 )</td>
<td>((0, 1, 1, 0, 0))</td>
<td>23</td>
</tr>
<tr>
<td>( m_2 )</td>
<td>((0, 1, 0, 0, 0))</td>
<td>4</td>
</tr>
</tbody>
</table>
The calculation of $X_r = x_r$ becomes the sum of all training data that contain the observation $X$ independent of what $x_-$ is:

$$n_{x_r}^2 = \sum_{x_r \subset x} n_{x_r}^2 = n_{x,x_r=0}^2 + n_{x,x_r=1}^2 = 4 + 23 = 27$$

The calculation then becomes:

$$p(M = m \mid X = x, D, I) = \frac{p(M = m \mid I) n_{x_r}^m + \alpha_{x_r}^m}{\pi_{X_r} N^m + A^m} \quad (3.8)$$

To illustrate this the same example is used as in Sections 3.1.2 and 3.1.1. The data is taken from Table 2.2.

**Example 3.4 (Elimination of a missing value)**

The same prerequisites as in Examples 3.1 and 3.2 are used. Using the approach described above the value $X_r$ becomes $X_r = (0, 1, 0, 0)$. It important to know which tests each value corresponds to and it is assumed that this is the case here. The calculation of the probability in Equation (3.8) which gives:

$$\pi_{X_r} = \sum_m p(X \mid M = m, D, I) p(M = m \mid I) = \sum_m \frac{n_{x_r}^m + \alpha_{x_r}^m}{N^m + A^m} p(M = m \mid I) =$$

$$= \frac{1}{4} \frac{(0 + 1) + (27 + 1) + (6 + 1) + (0 + 1)}{100 + 32} = \frac{37}{528} = 7.01 \times 10^{-2}$$

The prior probability is assumed to be the same for all four modes as in the previous examples, i.e. $p(M = m \mid I) = 1/4 = 0.25$. The probabilities then become:

$$p(M = m \mid X = x, D, I) = \frac{p(M = m \mid I) n_{x_r}^m + \alpha_{x_r}^m}{\pi_{X_r} N^m + A^m} =$$

$$= \frac{1/4}{37/(4 \times 132)} \frac{n_{x_r}^m + \alpha_{x_r}^m}{132} = \begin{cases} 1/37 = 0.027 & M = m_1 \\ 28/37 = 0.757 & M = m_2 \\ 7/37 = 0.189 & M = m_3 \\ 1/37 = 0.027 & M = m_4 \end{cases}$$

These results are also shown in Figure 3.1.

### 3.1.4 Treating Missing Values As a Third Value

A last method for handling missing data in the observation is to treat missing data as a value on its own. This could be considered to be the same as adding another possible value to the observation, i.e. going from a binary input to three possible values, 0, 1 and - . This method is discussed in [7] but will not be investigated in this thesis because of the fact that it implies adding another value to the observations.
3.1.5 Comparison of the Methods for Handling Incomplete Observations

![Graphs showing comparison of methods for handling incomplete observations.]

Figure 3.1: Results to the same isolation with different methods for handling missing data in the observation

Three methods for handling the case where an observation is incomplete have been discussed above and the resulting probabilities are all shown in Figure 3.1 where each subfigure is the result of one method for handling incomplete observations. Each method has its advantages and disadvantages.

The assumption of a value would give the correct result if the correct value that is lost is assumed. It is however disastrous if the value that is assumed is wrong. The question then becomes when it is possible to assume the correct value. A possible scenario is to assume the last known value of the test in question. This could be a good solution if it is known that the value that is missing is missing because the test needs more time to complete the test, i.e. if the test is slower than the isolation cycle or if one component has not been used. However, to assume a value is not an option when values are missing for vital components of the engine, in the cases when assuming the wrong value would risk ignoring faults that must be isolated.

Marginalization is the most precise when nothing is known about the received observation. The problem with marginalization is that if several values are missing then marginalization is best solved through recursion which is not the best method in embedded systems. The reason is that in an embedded system there is a limited amount of memory allocated to each application and the use
of recursion can require a large unspecified amount of memory.

The elimination of data is not solved using recursion but it is not thought
give as good a result as marginalization does.

The conclusion is that a mix of the different methods can be used. Assum-
pions of values where it is possible and either marginalization if it is possible or
elimination of the missing value otherwise.

Comparing the results of the common example in these different methods
from Figure 3.1 shows that the results from marginalization and the elimination
of missing data are very similar while the assumption of a value can vary very
much depending on which value that is assumed. This similarity of marga-
ralization and the elimination of lost data causes one to believe that there is some
connection between the two methods. These similarities are therefore further
investigated in Section 3.1.6.

3.1.6 A Closer Comparison of Marginalization and Elim-
ination of Lost Data

The posterior probability for marginalization can be written as:

\[
p(M = m \mid X = x, D, I) = \sum_{x} \frac{p(M = m \mid I) n_{x,x_{-}}^{m} + \alpha_{x,x_{-}}^{m}}{\pi_{x,x_{-}}} \frac{1}{N^{m} + A^{m}} n_{x,x_{-}} + \alpha_{x,x_{-}} \]

(3.9)

and the posterior probability for the elimination of missing data can be
written as:

\[
p(M = m \mid X = x, D, I) = \frac{p(M = m \mid I) n_{x,x_{-}}^{m} + \alpha_{x,x_{-}}^{m}}{\pi_{x,x_{-}}} \frac{n_{x,x_{-}}}{N^{m} + A^{m}}
\]

(3.10)

The case when each test result can take two values (1 or 0) is considered. If
the amount of training data for the observation when the missing value is 1 is
the same as the amount of training data when the missing value is 0, then this
is written as:

\[
n_{x,x_{-}}^{m} = n_{x,x_{-}}^{m} = n_{x,x_{-}}
\]

(3.11)

\[n_{x_{-}}^{m}\] is the sum of training data from all observations that contain \(x_{-}\). If Equation
(3.11) holds then it can be concluded that:

\[
n_{x,x_{-}}^{m} = \frac{1}{2} n_{x_{-}}^{m}
\]

(3.12)

It can also be assumed that, in the same frame of reference, since a received
observation \(X\) with a missing data could be either of two possibilities, both
contained within \(X_{r}\), the values of the a-priori \(\alpha\) are:

\[
\alpha_{x_{-}}^{m} = \alpha_{x_{-}}^{m} = \frac{1}{2} \alpha_{x_{-}}
\]

(3.13)

26
With these relationships assumed it is possible to make a comparison of the normalization factors:

\[
\pi_{x, x-} = \sum_m p(M \mid I) \frac{n^m_{x, x-} + \alpha^m_{x, x-}}{N^m + A^m} = \\
= \sum_m p(M \mid I) \frac{\frac{1}{2} n^m_{x, x-} + \frac{1}{2} \alpha^m_{x, x-}}{N^m + A^m} = \\
= \frac{1}{2} \sum_m p(M \mid I) \frac{n^m_{x, x-} + \alpha^m_{x, x-}}{N^m + A^m} = \frac{1}{2} \pi_x,
\]

The second term in the marginalization Equation (3.3), the weighting factor, becomes 1/2 when the case of Equation (3.11) is fulfilled, i.e.

\[
p(X_- = x_- \mid X_r = x_r, D, I) = \frac{1}{2}
\]

The marginalization then becomes:

\[
p(M = m \mid X = x, D, I) = \\
= \sum_{x_-} \frac{p(M = m \mid I) n^m_{x, x-} + \alpha^m_{x, x-}}{N^m + A^m} \frac{1}{2} = \\
= \frac{1}{2} p(M = m \mid X_r, X_- = 0, D, I) + \frac{1}{2} p(M = m \mid X_r, X_- = 1, D, I) = \\
= \frac{1}{2} p(M = m \mid I) \left( \frac{n^m_{x, x0} + \alpha^m_{x, x0}}{\pi_{x, x0}} + \frac{n^m_{x, x1} + \alpha^m_{x, x1}}{\pi_{x, x1}} \right) = \\
= \frac{1}{2} p(M = m \mid I) \left( \frac{n^m_{x, x0} + \alpha^m_{x, x0}}{\pi_{x, x0}} + \frac{n^m_{x, x1} + \alpha^m_{x, x1}}{\pi_{x, x1}} \right) = \\
= \frac{1}{2} p(M = m \mid I) \frac{n^m_{x, x-} + \alpha^m_{x, x-}}{N^m + A^m} = \\
= \frac{1}{2} N^m + A^m \pi_{x, x-} = \\
= \frac{1}{2} \cdot \frac{1}{2} p(M = m \mid I) \frac{n^m_{x, x0} + \alpha^m_{x, x0}}{\pi_{x, x0}} + \frac{n^m_{x, x1} + \alpha^m_{x, x1}}{\pi_{x, x1}} = \\
= \frac{1}{2} N^m + A^m \pi_{x, x-} = \\
= \frac{1}{2} \cdot \frac{1}{2} \pi_x,
\]

Comparing this result to Equation (3.10) it is now possible to see that the marginalization and the elimination of lost data are always the same under these circumstances, which are:

- The number of training data for for the observation when the missing value is replaced with a 0 is the same as when it is replaced by a 1.
- The a-priori \( \alpha \) is twice as big for the observation when the missing value has been eliminated as for a full length observation.

It is here obvious that the same holds when there is training data as when there is not. The calculations above show this if \( n = 0 \) and \( N = 0 \) is put into these equations. Of course the first of the two items above is then obliterated.
### 3.2 The Use of Subsystems

When an observation is received the isolation is isolated over the complete process. The complete process consists of hundreds of tests and hundreds of behavior modes. It is obvious that the fault isolation becomes a complex and demanding calculation.

If the structure of the decision structure is examined it is noticed that large sections contain only zeros. This is the result of the fact that most tests test only one or a few behavior modes. Few tests test behavior modes in different parts of the process, i.e., test that tests behavior modes connected to an injector are unlikely to test the brakes on the truck at the same time. These different sections of the decision structure are considered to be independent.

The independent sections of the decision structure can be split into several smaller decision structures that will be called subsystems. So instead of storing the whole decision structure only the subsystems are stored.

Each subsystem is created as small as possible. The requirement is that all behavior modes associated with one component are in the same subsystem and that components that are tested by the same tests as other components are also in the same subsystem.

Table 3.1 shows an example of a decision structure that is divided into two decision structures or subsystems.

#### Table 3.1: An example of a larger decision structure that can be partitioned into two smaller submatrices

<table>
<thead>
<tr>
<th></th>
<th>$m_1$</th>
<th>$m_2$</th>
<th>$m_3$</th>
<th>$m_4$</th>
<th></th>
<th>$m_5$</th>
<th>$m_6$</th>
<th>$m_7$</th>
<th>$m_8$</th>
<th>$m_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$\mathcal{X}$</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_3$</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_4$</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_5$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_6$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
</tr>
<tr>
<td>$x_7$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
</tr>
<tr>
<td>$x_8$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
</tr>
<tr>
<td>$x_9$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\mathcal{X}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$m_1$</th>
<th>$m_2$</th>
<th>$m_3$</th>
<th>$m_4$</th>
<th></th>
<th>$m_5$</th>
<th>$m_6$</th>
<th>$m_7$</th>
<th>$m_8$</th>
<th>$m_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$\mathcal{X}$</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
</tr>
<tr>
<td>$x_3$</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
</tr>
<tr>
<td>$x_4$</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_5$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
<td>$\mathcal{X}$</td>
<td>0</td>
</tr>
</tbody>
</table>
3.2.1 The Advantages of Subsystems For Storage and Isolation

The decision structure in the real process in a truck is very large compared to the examples of decision structures previously used in examples in this thesis. It is therefore not hard to imagine that the storage of training data requires a lot of memory. The use of subsystems allow for a more compact storage as large sections containing only zeros do not have to be stored.

Not only storage is improved using subsystems, the isolation also becomes easier. Instead of comparing the whole observation with as many elements as tests it is possible to take the part of the observation that corresponds to each subsystem and isolate in the specific subsystem only.

This allows for a division of the isolation into two steps. The first step makes a quick check for each subsystem if there has been any test that has given an alarm. If there are no alarms then the isolation is not initiated in that particular subsystem. If one test in the observation has given an alarm then the isolation in the subsystem containing that test is initiated. This will allow for a much faster execution of the isolation.

3.2.2 The Advantages of Subsystems For Collecting Training Data

Another advantage is that collecting training data becomes easier since we can put the truck in several behavior modes at the same time as long as these modes are in different subsystem. This could simplify and shorten the collection of training data. This also gives a better protection against faults when collecting training data, if there are faults in a different subsystem when collecting training data then this will not affect the training data for the specific mode being investigated.

3.3 Handling Multiple Modes In the Process Under Diagnosis

So far in this thesis, the probability computations only handle isolation of a single behavior mode that is present in the process. But when the number of components in a process increases, the probability that more than one behavior mode is present at the same time increases as well. Therefore the need of an approach to isolate the process when more than one behavior mode is present is crucial.

Besides the advantages of arranging the whole process into subsystems that were discussed in Section 3.2 is the advantage of handling multiple modes. In arranging the whole process into subsystems, most of the modes are sorted into different subsystems and the result is that most of the faults are sorted in different subsystems. For these subsystems with only one single fault or behavior mode present in the subsystem the isolation can be made without any
problem. There are however still cases when multiple modes are present in the same subsystem and for these cases the isolation of multiple modes becomes more complicated.

### 3.3.1 Multiple Modes In One Subsystem

If the decision structure only consist of information for single modes and training data is only collected for single modes, the isolation algorithm can only isolate single modes. If the case is that there is more than one mode present, the observation vector will most likely not match with any single mode and only the prior probability is used for the isolation.

A way to solve this problem and make it possible to isolate single or double modes is to compute the probabilities for all possible combinations of double modes as well as for all single modes. When all the probabilities are computed the final probability for each mode can be computed using all the information about single and double modes, e.g. the probability for mode $m_i$ is the sum of the probability for the single mode $m_i$ and no other mode and the probabilities for any double mode containing $m_i$.

For easier computations of all the different probabilities the combination of two behavior modes can be seen as different single modes. The combination of behavior mode $m_i$ and $m_j$ is denoted as $m_{i,j}$ and can be included in the decision structure just like any single behavior mode with the combined information from mode $m_i$ and $m_j$. The total number of modes $r$, single and double, can be computed using Equation (3.16). $k$ is the total number of single modes. This number will increase rapidly as $k$ increases.

$$r = \binom{k}{1} + \binom{k}{2} = \frac{k!}{(k-1)! \cdot 1!} + \frac{k!}{(k-2)! \cdot 2!}$$

Note that if there is possible to have triple modes the numbers of total modes is computed in the same way with the additional of the term for triple modes:

$$\binom{k}{3}$$

It is not hard to realize that when systems grow and contain more and more behavior modes, the number of possible double and triple modes will increase rapidly, e.g. if the number of single modes is ten, the number of single and double modes will be 55 and the number of single, double and triple modes will be 175. This could be a big problem when the process is an embedded system because of the limited availability to memory and restriction in execution time. Section 3.3.2 discusses how this problem is solved in this thesis.

Table 3.2 illustrates how extended decision structure looks like if the structure for the recurring example in this thesis is used.

When the new decision structure for both the single and double modes is complete the isolation can be made as if there only where single modes present. The difference is that the final probability for each mode must be summarized
Table 3.2: The decision structure for the process

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$m_1$</th>
<th>$m_2$</th>
<th>$m_3$</th>
<th>$m_4$</th>
<th>$m_{1,2}$</th>
<th>$m_{1,3}$</th>
<th>$m_{1,4}$</th>
<th>$m_{2,3}$</th>
<th>$m_{2,4}$</th>
<th>$m_{3,4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
<td>0</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
</tr>
<tr>
<td>0</td>
<td>$X$</td>
<td>$X$</td>
<td>0</td>
<td>$X$</td>
<td>$X$</td>
<td>0</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
</tr>
<tr>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>$X$</td>
<td>$X$</td>
<td>0</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
</tr>
<tr>
<td>$X$</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
<td>0</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>$X$</td>
<td>0</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
<td>$X$</td>
</tr>
</tbody>
</table>

afterward the isolation to get the true probabilities for each mode, Equation (3.17).

$$p(M = m_i \mid D, I) = p_{\text{single}}(M = m_i \mid D, I) +$$

$$+ \sum_j p_{\text{double}}(M = m_{i,j} \mid D, I), \quad i \neq j$$

(3.17)

where $p_{\text{single}}$ is the probability for single mode present and $p_{\text{double}}$ is the probability for two modes present at the same time.

In the case that this thesis handles this method becomes extremely inaccurate as this method would not have any training data for any of these modes. This would result in isolations that are very inaccurate as the probability would be based solely on the apriori distribution which becomes the same for all modes and the prior probability which would have to be set as a value at forehand and most likely be the same for all multiple modes. Not being able to isolate different multiple modes for different observations is not a useful isolation. Therefore the following section discusses a possible method for the implementation of double modes.

### 3.3.2 Known Multiple Modes In a Subsystem

One way to avoid this situation with rapidly increasing number of possible multiple modes is to design the process with those multiple modes that are possible or must be isolated according to the designer. The multiple modes that directly contradict each other are not implemented, e.g. the double mode 'oil pressure - too low' and 'oil pressure - too high' is not implemented. This may require a lot of work when setting the parameters for the process. Each possible double mode would have to set at forehand which, if looking at Equation 3.16 can be a lot of matching. Even if the decision to only implement some of the modes is taken there has to be training data for each of these multiple modes. That would mean a lot more training data to collect and store.

By eliminating those combinations that can not be combined and those for which the probability of a combination is nearly zero, the number of multiple modes that are implemented would only result in a small increase of size of the process. When there is only a small increase of size of the process the limitations regarding to the use of memory and execution time are easier to comply with.
than for the method described in Section 3.3.1 but could, as stated, still be a significant increase.

Besides this advantage that the execution time for the isolation decreases by skipping those multiple modes that are not very probable is the advantage that the resulting probability for each mode gets more precise. This is because no probability for a nearly impossible combination \( m_{i,j} \) is computed and added to the resulting probability (Equation (3.17)). The correctness of the probability would also increase as previous calculations have assumed that there are only single modes present in each subsystem. The real world does however contain double modes and because of this this method is a more true model and should, if enough training data is collected and the right prior probabilities are set, give a more correct result.
Chapter 4

Performance Measures

The previous chapters have discussed isolation methods and the solution to some problems that might arise when implemented. In the following chapters these methods will be evaluated to see how they work when applied on models of real processes. Different settings and parameters will be applied to a model and then the effect of changing values will be evaluated. The values that will be changed are the amount of training data, the probabilities of false alarms and missed alarms, and the prior probabilities. The different methods for handling missing data will also be investigated and compared as well as what happens if there is a double fault. The hope is to show what values are the most desirable so that a recommendation of how a system of the sort discussed in this thesis can be as accurate as possible. The goal is to know how well a probabilistic isolation system performs.

In order to decide if a fault isolation system is effective or not, measuring the performance for different setups and variables is needed. In order to do so a definition of an Isolation System must be made so that the definitions of measures of performance can be made. These definitions are taken from [1] and adapted to this thesis.

**Definition 4.1 (Isolation System)**

Let $\mathcal{S}$ be a process with $q$ tests and $r$ behavior modes. Let $M_{\mathcal{S}}$ denote the set of all possible behavior modes. Let $X_{\mathcal{S}}$ be the set of all possible observations. An isolation system is then defined as a function, $\mathcal{F}$, that takes an observation $X = (x_1, x_2, \ldots, x_q) \in X_{\mathcal{S}}$ and maps it onto a vector in the unit $r$-space:

$$\mathcal{F} : X_{\mathcal{S}} \mapsto [0, 1]^r$$

(4.1)

In words, the function $\mathcal{F}$ in Definition 4.1 maps an observation, $X$, to a vector with $r$ elements, each element in the interval $[0, 1]$. Each element is the probability that $X$ is an observation originating from the mode corresponding to that specific element.

A related function $\mathcal{F}_j : X_{\mathcal{S}} \mapsto [0, 1]$ relates an observation $X$ to the behavior
mode $m_j$. Since the observation $X$ must originate from one mode then the sum of all elements in the vector mapped by $\mathcal{F}$ is 1.

$$\sum_{j=1}^{r} \mathcal{F}_j(X = x, M = m_j) = 1$$

The function $\mathcal{F}_j$ can be related to the posterior probability in Section 2.8.4.

$$\mathcal{F}_j(X = x, M = m_j) = p(M = m_j \mid X = x, D, I) \in [0, 1] \quad (4.2)$$

It is now possible to define an optimal isolation system as:

**Definition 4.2 (Optimal Isolation System)**

An optimal isolation system is defined as an isolation system that assigns all probability mass to the correct underlying mode and no probability to the other modes.

$$\mathcal{F}_j(X = x, M = m_j) = \begin{cases} 1 & \text{if } M = m_j \text{ is the true underlying mode} \\ 0 & \text{otherwise} \end{cases} \quad (4.3)$$

\[\square\]

### 4.1 Expected Correctness

The definition of an isolation system allows for the first definition of one of the measures that will be used in the evaluation of the system.

**Definition 4.3 (Expected Correctness)**

The expected correctness is the expected probability assigned to the underlying mode in a process. If the mode of a process is $M = m_j$ and the isolation system $\mathcal{F}$ is used then the expected correctness is defined as,

$$\mu(M = m_j, \mathcal{F}) = E\{\mathcal{F}_j(X_{m_j}, M = m_j)\} \quad (4.4)$$

where $X_{m_j}$ is an observation generated by the process in behavior mode $M = m_j$.

\[\square\]

The optimal value of the expected correctness from Definition 4.3 is $\mu_{opt}(M = m_j, \mathcal{F}) = 1$.

The average expected correctness is defined as the average over all modes of the expected correctness for a process.

$$\bar{\mu}(\mathcal{F}) = \frac{1}{r} \sum_{i=1}^{r} \mu(M = m_i, \mathcal{F}) \quad (4.5)$$

This average expected correctness gives a more general measure of the performance of a system as it implies only one value for the whole isolation system and takes all modes into account.
4.2 Expected Probability of Correct Classification

The definition of expected correctness above, Definition 4.3, is a measure of how much probability is given to the underlying mode of the process. It does not tell anything about weather or not the highest probability is given to the correct mode. It is possible to imagine a case where the value of $\mu$ is relatively low. For example when a process is large, i.e. many tests and behavior modes, each mode will get a small probability of being the correct mode as the probability is divided onto all modes, hence the more modes the less probability. Also if two modes are very similar, or even identical in both training data and decision structure and both equally likely to occur or even similarly then they will both get close to half of the probability that they would have gotten if they were alone in being the possible mode. Therefore another definition is needed to evaluate how well a system isolation is working that is not dependent on the absolute value of the probability that is assigned.

Definition 4.4 (Expected Probability of Correct Classification)
The expected probability of correct classification, $\nu$, is the expected probability that the underlying mode is given the highest probability in the isolation. If the true underlying mode is denoted $m_j$ and

\[
\hat{m} = \max_i \mathcal{F}(X = x, M = m_i)
\]  

then the expected probability of correct classification can be defined as:

\[
\nu(M = m_j, \mathcal{F}) = E\{p(\hat{m} = m_j)\}
\]  

As for the expected correctness an optimal value for the expected probability of correct classification is $p(\hat{m} = m_j \mid \mathcal{F}) = 1$. This means that the underlying mode is always assigned highest probability.

As for the definition of expected correctness, a single value can be calculated to get one value for the expected probability of correct classification. The average expected probability of correct classification over each mode, $\bar{\nu}(\mathcal{F})$:

\[
\bar{\nu}(\mathcal{F}) = \frac{1}{r} \sum_{i=1}^{r} \nu(M = m_i, \mathcal{F})
\]

4.3 Expected Ratio of the Underlying Probability and the Most Probable Wrong Estimation

As a last measure of how well an isolation system performs the ratio between the correct mode and the mode with the highest probability with the correct mode excluded is used.
Definition 4.5 (Ratio of Probability of Correct Mode and Largest Probability of a Wrong Mode)

The ratio of probability of correct mode and largest probability of a wrong mode, $\gamma$, can be expressed as

$$\gamma(m_j, F) = E\left\{ \frac{F(x, m_j)}{\max_i F(x, m_i)} \right\} \quad (4.9)$$

The denominator is the largest probability assigned to a non-present mode,

$$\mathcal{F}^{\max m_j}(x, m_i) = \max_i F(x, m_i), i \neq j$$

$$M = m_j$$

The optimal isolation would be, as stated in Definition 4.1, if the correct underlying mode was given all probability, Equation (4.3). Then the ratio would be undefined, or in some sense go towards $\infty$. If no probability would be given to the correct mode then the ratio would be 0.

The ratio for each mode is calculated separately. In this thesis the measure of the ratio is only measured to see what the maximum expected ratio and minimum ratio are for different modes. The maximum ratio shows the ratio for that mode which has the greatest ratio and the minimum ratio is the ratio for the mode with the smallest ratio. All other ratios are in the linear space between these two values, $[\gamma_{\min}, \gamma_{\max}]$. It is desirable to keep both these values as high as possible. If $\gamma_{\min}$ is smaller than zero then there is one mode that is more likely to give some other mode higher probability when that mode is the true underlying mode.
Chapter 5

Evaluation of the Diagnosis System

5.1 Experimental Setup

In order to make evaluations on a diagnosis system using the methods described in this thesis a model of the process under diagnosis must be constructed. The model must be able to be altered to try different settings for parameters. Running the system once gives one value for the probabilities for the generated observation but in order to get values that are more general such the expected values described above in Sections 4.1, 4.2 and 4.3 more simulations must be made. Monte Carlo simulations are simulations with randomized values that are then used to compute the results. In the simulations that are made in this thesis the observation is generated with a certain probability depending on which mode is the underlying mode. Depending on the test the modes are also randomly generated or decided at forehand how many simulations will be conducted for each mode. The three expected values described above are then calculated for comparison if applicable.

Tests are run with 1000 simulations where the mode is randomly generated with equal distribution among all possible modes. From this mode an observation is generated. The observation is generated from the column in the decision structure related to the underlying mode with a probability of false alarm and a probability of missing an alarm.

Unless otherwise stated the setup of the system is as stated in Table 5.1.

The evaluation of the diagnosis system is conducted on two different models with similar properties to the real process in trucks today. These models are shown by their decision structures in Tables 5.2 and 5.3.
Table 5.1: The General Setup for the Experiments in Chapter 5

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td>400</td>
<td>Amount of training data per mode</td>
</tr>
<tr>
<td>( S )</td>
<td></td>
<td>Decision structure, given in Tables 5.2 and 5.3</td>
</tr>
<tr>
<td>( p_{fp} )</td>
<td>0.01</td>
<td>Probability of false positive,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i.e. probability of having a false alarm from a test</td>
</tr>
<tr>
<td>( p_{fn} )</td>
<td>0.3</td>
<td>Probability of false negative,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i.e. probability of having a test miss an alarm</td>
</tr>
<tr>
<td>( l )</td>
<td>0</td>
<td>The number of missing (lost) values in the observation</td>
</tr>
</tbody>
</table>

Table 5.2: The First of the Two Decision Structures Used in the Evaluation

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( m_1 )</th>
<th>( m_2 )</th>
<th>( m_3 )</th>
<th>( m_4 )</th>
<th>( m_5 )</th>
<th>( m_6 )</th>
<th>( m_7 )</th>
<th>( \text{nf} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.3: The Second of the Two Decision Structures Used in the Evaluation

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( m_1 )</th>
<th>( m_2 )</th>
<th>( m_3 )</th>
<th>( m_4 )</th>
<th>( m_5 )</th>
<th>( m_6 )</th>
<th>( m_7 )</th>
<th>( m_8 )</th>
<th>( m_9 )</th>
<th>( m_{10} )</th>
<th>( m_{11} )</th>
<th>( \text{nf} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( X )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

38
5.2 Amount of Training Data

The method of isolating the underlying mode from an observation by calculating the posterior probability discussed in this thesis is largely dependent training data, i.e. data collected from experiments run on the system. If no training data is collected then the probability depends only on the prior probability, \( p(M = m | I) \), and the a priori, \( \alpha \), both described in Sections 2.8.1 and 2.8.2. In many cases it is hard to specify the prior because everything about the process is not known and the a priori is always the same in this thesis. Therefore the system depends on training data. A parallel thesis [2] describes what happens if there is no training data but in this thesis training data is required for a meaningful isolation.

Setup

The setup of the experiment apart from the amounts of training data is the same as described in Section 5.1. One simulation is run for each of the two systems in Tables 5.2 and 5.3 and for each amount of training data.

The amounts of training data that are tried for are; no training data, 100, 200, 400 and 4000 training data for each mode. The average expected correctness (\( \overline{\mu} \)) and the average expected probability of correct classification (\( \overline{\nu} \)) calculated. For the expected ratio of the underlying probability and the most probable wrong estimation (\( \gamma \)) the highest ratio and the lowest ratio for a mode are collected.

Results and Comments

<table>
<thead>
<tr>
<th>N</th>
<th>System 1</th>
<th>System 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \overline{\mu} )</td>
<td>( \overline{\nu} )</td>
</tr>
<tr>
<td>0</td>
<td>0.125</td>
<td>0.125</td>
</tr>
<tr>
<td>100</td>
<td>0.597</td>
<td>0.765</td>
</tr>
<tr>
<td>200</td>
<td>0.620</td>
<td>0.747</td>
</tr>
<tr>
<td>400</td>
<td>0.662</td>
<td>0.776</td>
</tr>
<tr>
<td>4000</td>
<td>0.701</td>
<td>0.789</td>
</tr>
</tbody>
</table>

The results to these tests are shown in Table 5.4 and in the two Figures 5.1 and 5.2.

Figure 5.1 gives a visible correlation between the amount of training data and the average expected correctness.

The the average expected probability of correct classification also tend to increase with the amount of training data. There is however an exception, when the amount of training data is 100 for each mode then \( \overline{\nu} \) is greater than for when there is 200 training data. One explanation is that when there are a lot of false alarms then the observations for these should be more apparent when
Figure 5.1: Plot of $\bar{\mu}$ and $\bar{\nu}$ for Varying Amounts of Training Data $N$ for Each Mode

there are more training data as the training data would also have these. This means that other modes are more likely to have observations in the training data that are normally not associated with them. When the training data increases these observations become less and less significant and thus $\bar{\nu}$ increases again. This effect is more obvious for the second system since it has more tests that can give false alarms.

A more general observation is however that the amount of training data increases $\bar{\mu}$ more than $\bar{\nu}$ even though both increase with the amount of training data.

Figure 5.2 shows the probability that the correct mode has a probability larger than $x$ in the x-axis for different amounts of training data. The ideal plot would be a straight line at 1 for all $x$ since this would mean that for each simulation the correct mode would be given the probability of 1. The faster the curve descends the lower the probability of the correct mode is. A sudden descend in the slope would mean that a large quantity of the correct modes are given the corresponding $x$ in probability when the estimation is done.

Looking at Figure 5.2 for system 1 and no training data ($N = 0$) a large descent is visible at $x = 0.1$ and a smaller descent at $x = 0.3$. This is because when there is no training data, the probability only depends on the prior $P(M = m | I)$ and on $\alpha$ which is always 1. This prior is the same for all modes except for the mode $n_f$ which is three times as big. The probability vector is therefore always the same, independent of what the observation is. Each mode gets the
Figure 5.2: Plot of the Probability that the Correct Mode has a Probability Greater than x for Varying Amounts of Training Data N for Each Mode, Also Called the Complementary Cumulative Distribution Function.

The probability of 0.1 except for n which gets 0.3. These numbers vary depending on the size of the process and on the priors but if there is no training data the probability does not change with the observations.

Figure 5.2 shows how the results go toward the ideal line described as $N \rightarrow \infty$ as the amount of training data increases. This line is calculated based in the work in [2] and the fact that the real parameters in the model are known. Figure 5.2 also shows that when there is no training data it does not follow this result at all which makes believe that the prior information is not at all enough to isolate the process.

Table 5.5 shows that for this system, if the expected correctness must be created than 70% then, depending on the amount of training data, this can be assured with a certainty from 50% to 70%. The tradeoff is how much training data can be collected and stored which will be decided by the amount of resources allocated to the implementation of this system.

Summary

Training data is the single most important parameter for the isolation method discussed in this thesis. Without training data the isolation only depends on the prior probability which will not give any qualitative result at all but always the same.
Table 5.5: Results from Figure 5.2 Showing the Probability that the Probability of the Correct Modes is Greater Than 70%

<table>
<thead>
<tr>
<th>Training data per mode, N</th>
<th>Probability $p(p(\text{correct mode}) \geq 0.7)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>System 1</td>
</tr>
<tr>
<td>$N = 0$</td>
<td>0%</td>
</tr>
<tr>
<td>$N = 100$</td>
<td>51.5%</td>
</tr>
<tr>
<td>$N = 200$</td>
<td>58.1%</td>
</tr>
<tr>
<td>$N = 400$</td>
<td>66.8%</td>
</tr>
<tr>
<td>$N = 4000$</td>
<td>67.2%</td>
</tr>
<tr>
<td>$N \rightarrow \infty$</td>
<td>72.2%</td>
</tr>
</tbody>
</table>

The tests show the not so surprising result that when there is more training data the results tend to get better. The tests also show that a larger system needs more training data to perform as well as a smaller system.

The amount of training data that is required depends on the general setup and properties of the system. In the following sections 400 training data will be used.

The amount of training data is also a trade off between how much data can be stored and how much data that is needed. There will also be a limit on how much training data that can be collected, some modes are more difficult than others.
5.3 Robustness to Probabilities of False Positive Observations and False Negative Observations

Setup
This section investigates what happens if the probabilities of false positive, $p_{fp}$, and false negative, $p_{fn}$, vary. It is desired to test the system and see how much better it becomes if the probability of false positive and negative are better than 0.01 and 0.3. The amount of training data will be 400 per mode. This has been considered as a reasonable trade off between the amount of training data and the resulting isolation performance from the results in Section 5.2. The probabilities that will be tested for are shown in Table 5.6. These values are chosen this way because of how the values are believed to be in the actual process of the trucks. By 'the way they are' is meant that there is a much higher probability of missing an alarm than a false alarm.

A second set of probabilities that will be tested for are shown in Table 5.7. The reason for these values is not to try to simulate any real process but to see an exaggerated effect of the two different probabilities that are used. The setup for these simulations is the same as for the simulations in Table 5.6.

Table 5.6: The Probabilities of False Positive and False Negative That Are Evaluated For

<table>
<thead>
<tr>
<th>$P_{fp}$</th>
<th>$P_{fn}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.3</td>
</tr>
<tr>
<td>0.001</td>
<td>0.3</td>
</tr>
<tr>
<td>0.01</td>
<td>0.1</td>
</tr>
<tr>
<td>0.01</td>
<td>0.03</td>
</tr>
<tr>
<td>0.001</td>
<td>0.1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.7: The Probabilities of False Positive and False Negative That Are Evaluated For

<table>
<thead>
<tr>
<th>$P_{fp}$</th>
<th>$P_{fn}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>0.3</td>
<td>0</td>
</tr>
<tr>
<td>0.3</td>
<td>0.3</td>
</tr>
</tbody>
</table>

The simulation with the probabilities in Table 5.6, with 0 probability for false positive or negative, is tested for to see what the 'ultimate' result looks like.
Results and Comments

The results for the simulations with the probabilities in Table 5.6 are shown in Table 5.8 and Figures 5.3 and 5.4.

The results for the simulations with the probabilities in Table 5.7 are shown in Figure 5.5 and 5.6.

Table 5.8: Results to Simulations with Different Probabilities for False Alarm $p_{fp}$ and Missed Alarm $p_{fn}$

<table>
<thead>
<tr>
<th>$p_{fp}$</th>
<th>$p_{fn}$</th>
<th>$\bar{\mu}$</th>
<th>$\bar{\nu}$</th>
<th>$\gamma_{\max}$</th>
<th>$\gamma_{\min}$</th>
<th>$\bar{\mu}$</th>
<th>$\bar{\nu}$</th>
<th>$\gamma_{\max}$</th>
<th>$\gamma_{\min}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.3</td>
<td>0.662</td>
<td>0.776</td>
<td>32.5</td>
<td>7.24</td>
<td>0.573</td>
<td>0.702</td>
<td>29.3</td>
<td>0.903</td>
</tr>
<tr>
<td>0.001</td>
<td>0.3</td>
<td>0.752</td>
<td>0.820</td>
<td>69.6</td>
<td>8.90</td>
<td>0.670</td>
<td>0.757</td>
<td>43.3</td>
<td>0.902</td>
</tr>
<tr>
<td>0.01</td>
<td>0.1</td>
<td>0.821</td>
<td>0.911</td>
<td>83.1</td>
<td>15.4</td>
<td>0.721</td>
<td>0.845</td>
<td>101</td>
<td>0.984</td>
</tr>
<tr>
<td>0.01</td>
<td>0.03</td>
<td>0.888</td>
<td>0.954</td>
<td>106</td>
<td>20.3</td>
<td>0.759</td>
<td>0.855</td>
<td>446</td>
<td>0.984</td>
</tr>
<tr>
<td>0.001</td>
<td>0.1</td>
<td>0.898</td>
<td>0.951</td>
<td>110</td>
<td>24.2</td>
<td>0.804</td>
<td>0.863</td>
<td>169</td>
<td>0.984</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.980</td>
<td>1</td>
<td>1203</td>
<td>133</td>
<td>0.891</td>
<td>1</td>
<td>1203</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 5.3: Plot of $\bar{\mu}$ and $\bar{\nu}$ for Different Probabilities of False Alarms and Missed Alarms

From the first set of simulations in Table 5.6, with values that are supposed to be similar to what the real values are believed to be, it is visible that the smaller the probabilities of false alarm and missed alarms become the higher the expected correctness and the expected probability of correct classification.
The values that are used as standard values in this thesis are 0.01 for false positive and 0.3 for false negative. The system is designed in a fashion to make the probability for having a false positive go toward 0 and where the probability false negative range from 0 to 0.5 depending on the type of test. According to engineers a normal value for a false negative is often around 0.1.

The simulation with the values 0.001 for false positive and 0.1 for false positive is considered to be a simulation with good values. The probability of false positive is ten times better than the test with the, for this thesis, standard values and the probability of false negative is three times better. The result on the expected correctness, \( \bar{\mu} \), for the system in Table 5.2 in going from the original values to these improved values is an increase from 0.662 to 0.898 which is an increase of 35.65\%. For the second system in Table 5.3 this increase is even greater, it is 40.31\%. The increases in the expected probability of correct classification, \( \bar{p} \), these values are 22.55\% and 22.93\% for each of the two systems.

These results are very encouraging as this shows that the system will become much better for every increase in certainty of the tests. As a comparison to the results in Section 5.2 where the values for \( p(p\text{(correct mode)} \geq 0.7) \) were collected the corresponding numbers for simulations with the probabilities in Table 5.6 are collected and shown in Table 5.9. The values are read off from Figure 5.4.

If the tests become perfect, i.e. only react when they should then the iso-
Table 5.9: Results from Figure 5.4 Showing the Probability that the Probability of the Correct Modes is Greater Than 70%

<table>
<thead>
<tr>
<th>Probabilities for false positive and negative</th>
<th>Probability $p(p(\text{correct mode}) \geq 0.7)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{fp}$</td>
<td>$p_{fn}$</td>
</tr>
<tr>
<td>0.01</td>
<td>0.3</td>
</tr>
<tr>
<td>0.001</td>
<td>0.3</td>
</tr>
<tr>
<td>0.01</td>
<td>0.1</td>
</tr>
<tr>
<td>0.01</td>
<td>0.03</td>
</tr>
<tr>
<td>0.001</td>
<td>0.1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

ulation becomes perfect in the sense that the correct underlying mode always get the highest probability ($\nu = 1$). The result of the expected correctness for the simulations of system 2 are however not as high as for system 1 where they are 98%. The reason is visible in the decision structure in Table 5.3. Mode $m_4$ and $m_1$ are identical. Since $p_{fp}$ and $p_{fn}$ are both zero the training data is identical for both these modes and therefore if one of them occurs the other will be isolated with the same probability. The result is that they both get close to 50% which lowers the average expected correctness. The expected probability of correct classification is still 100% because if two modes share the highest probability they are both classified as the underlying mode in these simulations.

The second set of simulations in Table 5.7 with results in Figures 5.5 and 5.6 are supposed to give a better view of how false positive affects the results and how false negative affects the results individually.

The first conclusion that can be drawn in from Figure 5.6 and it is that false alarms have a much greater impact on the system than missing alarms. The difference between the curve with 30% false alarms and the curve with 30% of both false alarms and missed alarms is not that big. Especially for the second system. This can be explained by the structure of the system. In the first system there are ten tests and each mode is tested by one or two tests. In the second system there are 17 tests and each test is affected by two or three tests. This means that for the first system there are eight or nine tests that can have false positive alarms for each test while in the second system there are 14 or 15. Every time there is a false positive the probability of isolating the correct mode decreases. The probability false negative is much smaller for both systems since there are not as many tests that are intended to react on each mode. In other words, the more tests there are that can have false positive alarms the more critical is the effect of these false positives.

The curve when there are false negative alarms decreases faster in the lower regions of $x$ than for the the curve where there are many false positive alarms. This effect is explained by the cases when alarms are missed so that no alarm is given. Then the mode that gets most of the probability is $m_f$ which only contains zeros even if it is not the true underlying mode.
In the middle range of $x$ the curve with false alarms decreases faster as the effect of having false positives which gives more probability to the modes which are tested by these alarms. If three modes are likely to be the underlying mode then the expected probability is close to 30% and 25% for four modes. The result is a decrease in the curve at probabilities $x$ in these ranges in the CCDF plot in Figure 5.6.

Figure 5.5: Plot of $\bar{\mu}$ and $\bar{\nu}$ for Different Probabilities of False Alarms and Missed Alarms

**Summary**

The results with the probabilities of false positive and negative used in this thesis are not very good. The real values are however considered to be much better. Simulations for better probabilities show that the result becomes substantially better.

Because of the structure of the two systems described by Tables 5.2 and 5.3 the systems are much more sensitive to false positives. A large probability of false positives has a fatal effect on the isolation.
Figure 5.6: Plot of the CCDF of the Correct Underlying Modes Probability for Different Probabilities of False Alarms and Missed Alarms
5.4 Robustness With Respect to the Priors

Setup

In this section an investigation will be conducted to see how important the prior probability is for the isolation. What happens if the prior is wrong? How much does a wrong assumption change the result of the system?

In the previous simulations the prior has been the same for all modes except for the mode $n_f$. The model with which the process is simulated is however not constructed with this bias to $n_f$. This means that the previous isolations have been conducted with a faulty assumption. This does not affect the results of the previous sections as it is not the effect of the prior that has been investigated and since the prior has been constant for all simulations. In this section the result of this faulty prior will be compared to the actual prior that the model uses.

The investigation is divided into two parts. The first part investigates what happens if one value is wrong, or even if one prior is 0. The second part investigates what happens if the prior is wrong for more than one mode, every second prior is changed by varying amounts.

Table 5.10 shows how the priors are changed. An important comment is that the size of the priors in this table is relative to each other. Normally, since the priors are probabilities, these values would be normalized so that the sum of the priors for all modes sum to one. In this table the priors are displayed in this fashion for simplicity of comparison of their relative size.

<table>
<thead>
<tr>
<th>Prior</th>
<th>Values of the Prior for each mode</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$m_1$ $m_2$ $m_3$ $m_4$ ... $m_{r-1}$ $m_r$</td>
</tr>
<tr>
<td>$p_1$</td>
<td>1 1 1 1 ... 1 1</td>
</tr>
<tr>
<td>$p_2$</td>
<td>1 1 1 1 ... 1 2</td>
</tr>
<tr>
<td>$p_3$</td>
<td>1 1 1 1 ... 1 3</td>
</tr>
<tr>
<td>$p_4$</td>
<td>1 1 1 1 ... 1 0</td>
</tr>
<tr>
<td>$p_5$</td>
<td>2 1 1 1 ... 1 1</td>
</tr>
<tr>
<td>$p_6$</td>
<td>3 1 1 1 ... 1 1</td>
</tr>
<tr>
<td>$p_7$</td>
<td>0 1 1 1 ... 1 1</td>
</tr>
<tr>
<td>$p_8$</td>
<td>1 2 1 2 ... 1 2</td>
</tr>
<tr>
<td>$p_9$</td>
<td>1 3 1 3 ... 1 3</td>
</tr>
<tr>
<td>$p_{10}$</td>
<td>1 10 1 10 ... 1 10</td>
</tr>
</tbody>
</table>

Results and Comments

Figures 5.7 and 5.8 show the results to the seven first tests of different priors shown in the table above ($p_1$, $p_2$, $p_3$, $p_4$, $p_5$, $p_6$ and $p_7$). The first set of priors that were tested for were $p_1$ which is the 'correct', meaning that it is the same that the model used in the simulation uses. The priors $p_4$ and $p_7$ both have one
mode which has prior probability 0. This shows particularly clearly Figure 5.7 for the expected probability of correct classification $\nu$ of system 1 where $\nu$ is visible smaller for these two sets priors. It is less clear for the second system as it has a smaller impact on the probabilities as one mode is a smaller part of the system than one mode in system 1. System 2 is also a more complex system with more dependencies that affect the probabilities as well.

In Section 5.3 where the effect of the probabilities of false positive and negative were investigated one cause of the descent of the curve for the CCDF of the correct mode’s probability in Figure 5.4 was due to false positive alarms that caused $n_f$ to get most of the probability. Figure 5.8 shows this effect as well. When the prior is $\rho_4$ the curve distinguishes itself from the others both at the lower and higher regions of $x$. This is because no probability is falsely given
to the mode \( n_f \) which for prior \( p_4 \) has zero prior probability thus giving more probability to the correct underlying mode.

Apart from the observation that the curve for prior \( p_4 \) distinguishes itself from the other curves in Figure 5.8 the curves look very similar. Figure 5.7 also shows that the variations between the different prior probabilities has little variations. An exception is the prior \( p_4 \) in the first system.

The effect on the first system is greater than on the second system. This difference is related to the structure of the systems. System one has a less complex structure, a change in the prior directly affects the mode for which the prior is changed. Any lost probability mass is likely to go to the mode \( n_f \). The more complex second system experiences less effect in the change of priors as lost probability mass is more likely to go to another mode rather than the \( n_f \) mode thus increasing \( \mu \) and \( \nu \) for that mode and hence causing a smaller effect on the average values.

Table 5.12: Results to Simulations with Different Prior Probabilities

<table>
<thead>
<tr>
<th>Prior</th>
<th>( \bar{\mu} )</th>
<th>( \bar{\nu} )</th>
<th>( \gamma_{max} )</th>
<th>( \gamma_{min} )</th>
<th>( \bar{\mu} )</th>
<th>( \bar{\nu} )</th>
<th>( \gamma_{max} )</th>
<th>( \gamma_{min} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_1 )</td>
<td>0.681</td>
<td>0.810</td>
<td>109</td>
<td>2.58</td>
<td>0.583</td>
<td>0.768</td>
<td>85.5</td>
<td>0.868</td>
</tr>
<tr>
<td>( p_8 )</td>
<td>0.680</td>
<td>0.819</td>
<td>73.8</td>
<td>2.96</td>
<td>0.577</td>
<td>0.745</td>
<td>75.5</td>
<td>0.503</td>
</tr>
<tr>
<td>( p_9 )</td>
<td>0.667</td>
<td>0.808</td>
<td>90.9</td>
<td>2.97</td>
<td>0.568</td>
<td>0.715</td>
<td>84.4</td>
<td>0.321</td>
</tr>
<tr>
<td>( p_{10} )</td>
<td>0.591</td>
<td>0.769</td>
<td>97.9</td>
<td>2.16</td>
<td>0.506</td>
<td>0.707</td>
<td>64.9</td>
<td>0.087</td>
</tr>
</tbody>
</table>

Figure 5.8: Plot of the CCDF of the Correct Underlying Modes Probability for Different Prior Probabilities
Figure 5.9: Plot of $\bar{\mu}$ and $\bar{\nu}$ for Different Prior Probabilities

In the cases where the priors are changed more than for one mode ($p_8$, $p_9$ and $p_{10}$) the results are shown in Table 5.12 and Figures 5.9 and 5.10. A comparison of the results for the set of priors in $p_{10}$ with the 'true' priors in $p_1$ is also done in a more precise mode to mode comparison with results shown in Figures 5.11 and 5.12.

Figure 5.9 shows that the average expected correctness $\bar{\mu}$ and the average expected probability of correct classification $\bar{\nu}$ are both affected negatively, especially when the prior is changed by a lot. One noticeable difference is that the effect on the average expected probability of correct classification seems to be greater on the second system.

Figure 5.10 does not give much more information but it shows how the increasing faulty priors deteriorates the systems performance.

In Figures 5.11 and 5.12 the effect of the extremely faulty set of priors $p_{10}$ is compared to the real set of priors $p_1$. It shows how System 1, Figure 5.11, varies more uniformly and consistent to how the priors are set. The modes with priors that are ten times larger (even numbered modes) get a little more expected correctness compared to when all modes have the same prior probability while those that have smaller prior probabilities (odd numbered modes) lose a lot of probability. The loss of expected probability of correct classification is not nearly as visible for as the loss in expected correctness which is related to the structure of the system. The modes are all independent, no test tests more than one mode.

Looking at the effect on the second system it is immediately obvious that the effect is much stronger than on the first system and not as uniform as for
Figure 5.10: Plot of the CCDF of the Correct Underlying Modes Probability for Different Prior Probabilities

system one. The reason, being the complexity of the system, has been discussed before in how some modes are dependent on others. Some observations might be shared between different modes. The best example might be the effect on modes $m_4$ and $m_{11}$ that have the same structure when looking at the decision structure in Table 5.3. In the simulation with the priors $p_{10}$ mode $m_4$ get a prior that is ten times larger than that of mode $m_{11}$ which results in almost all probability between the two modes being divided in favor of mode $m_4$. Mode $m_{11}$ would have to have ten times more training data for some observation in order to get higher probability for that specific observation.

Summary

To sum up the evaluation of how the isolation method reacts to faulty priors one can say that as long as they do not differ too much from what they should be and if there is enough training data the system handles errors well. One faulty prior does not affect the overall performance too much.

Smaller and simpler systems where one test tests only one mode lose expected correctness when several priors are faulty but still give the most probability to the correct mode thus keeping expected probability of correct classification high while when one mode has a faulty prior the simpler system sees more effect on the expected correctness than the more complex system due to less total effect on larger systems.
Figure 5.11: Plot of $\mu$ and $\nu$ for Prior Probabilities $p_1$ and $p_{10}$ for System 1

Figure 5.12: Plot of $\mu$ and $\nu$ for Prior Probabilities $p_1$ and $p_{10}$ for System 1
Complex systems are however less robust if more than one prior is off due to the competition between modes for the probabilities from the same observations.
5.5 Evaluation of the Methods for Handling Missing Data

In Section 3.1 an introduction to different methods of handling missing values in the observation was given. Two of the methods that were discussed were the methods of Marginalization in Section 3.1.2 and what was called Elimination of Missing Data in Section 3.1.3. Marginalization was believed to be the method which would minimize the expected error from what the 'real' observation would have given. Marginalization was best solved using recursion which should be avoided in embedded systems such as a control system for a diesel engine in large trucks. Therefore the method of elimination was introduced and argued not to give much of a different result compared to the marginalization.

Now the two methods are to be compared in simulations on the same two systems that have been used previously in this Chapter, the systems are shown on Page 38.

Setup

Both Marginalization and Elimination of Missing Data were tried with one and two missing data in the observation and compared to the result of having complete observations and isolating the 'normal' way to see how much information that is lost when data in the observation is missing.

The method of assuming a value for the missing data is also investigated in simulations, both for assuming 0 and 1.

The parameters of the system are kept the same as in previous experiments. The probability of false positive and negative are 0.01 and 0.3 respectively. There are 400 training data per mode. The first system in Table 5.2 and the second system in Table 5.3 are used.

Results and Comments

<table>
<thead>
<tr>
<th>Method</th>
<th>System 1</th>
<th>System 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$l$</td>
<td>$\bar{\mu}$</td>
</tr>
<tr>
<td>'Normal'</td>
<td>0</td>
<td>0.662</td>
</tr>
<tr>
<td>Marginalization</td>
<td>1</td>
<td>0.627</td>
</tr>
<tr>
<td>Elimination</td>
<td>1</td>
<td>0.613</td>
</tr>
<tr>
<td>Marginalization</td>
<td>2</td>
<td>0.594</td>
</tr>
<tr>
<td>Elimination</td>
<td>2</td>
<td>0.588</td>
</tr>
<tr>
<td>Assume 0</td>
<td>1</td>
<td>0.627</td>
</tr>
<tr>
<td>Assume 1</td>
<td>1</td>
<td>0.250</td>
</tr>
<tr>
<td>Assume 0</td>
<td>2</td>
<td>0.578</td>
</tr>
<tr>
<td>Assume 1</td>
<td>2</td>
<td>0.154</td>
</tr>
</tbody>
</table>

36
Figure 5.13: Plot of $\mu$ and $\nu$ for Isolating Observations with Missing Data Using Marginalization and Elimination of Missing Data

The results to simulations with marginalization and elimination are shown in Table 5.13 and in Figures 5.13 and 5.14. One observation that can be made in Figure 5.13 is that the result for System 1 is more straightforward compared to System 2. This has to do with the complexity of the systems. System one is a simple system without dependencies between the different modes while System 2 is more ambiguous in the sense that several modes are tested by the same tests.

Looking at Figure 5.13 it is obvious that for system one Marginalization has a better result than the Elimination of Missing Data and that two lost data in the observation has a worse effect than loosing only one.

For system two it looks as if the Elimination of Missing Data has a better effect on the isolation that Marginalization and surprisingly that missing two data in the observation does not affect the isolation in a negative way.

Figure 5.14 shows similar results for both systems. System one has more of a deterioration of the results than system two in going from marginalization to elimination and from one lost data to two lost data in the observation. The second system seems to have much more similar paths for all curves thus indicating that it is more robust to missing data and that either elimination of missing data or marginalization are better methods for handling missing data for system two. This can be explained by the fact that more tests test for more than one mode in the second system so that eliminating the result of one test
Figure 5.14: Plot of the CCDF of the Correct Underlying Modes Probability for Isolating Observations with Missing Data Using Marginalization and Elimination of Missing Data
Figure 5.15: Plot of $\mu$ and $\nu$ for Isolating Observations with Missing Data Assuming the Missing Values to Either 0 or 1
Figure 5.16: Plot of the CCDF of the Correct Underlying Modes Probability for Isolating Observations with Missing Data Assuming the Missing Values to Either 0 or 1
does not compromise the information in the observation as much as it does in System 1.

As for the results of when the missing value is assumed to be either 0 or 1 shown in Figures 5.15 and 5.16 the results are expected. A considerable deterioration of the results is shown, especially for when the value that is assumed is 1. This is because when assuming that a test gave an alarm the risk of creating a false alarm is high. Since false alarms are much less probable than missing alarms this punishes the isolation much more than assuming a 0 and risking missing an alarm. Also, the structures of systems one and two are in such a way that every mode has more tests in each observation that do not test for that specific mode. Therefore, if an observation is missing a value it is more likely that the missing value is a 0.

Figure 5.16 shows how assuming 1 for a missing value completely kills all probabilities. Especially when there are two missing values. The CCDF of the correct modes for when a 0 is assumed follows the CCDF of the curve where no data is missing very well. Especially in System two which is, as concluded earlier, less sensitive to missing data in the observation.

Summary

What was hoped to be the result of this investigation was to show that the use of Elimination of Missing Data does not affect the result too much compared to the use of Marginalization.

In smaller and less complex systems the effect of the amount of missing data and which method that is used to handle this seems to have a greater effect than for greater systems. The explanation is the robustness and complexity of larger systems, each mode is not as dependent on the data from one test as in a smaller system.

The method of assuming a zero also seems to be a good solution to missing data in the observation. The thought that the assumption of a value is made when a missing data goes missing but there has been a previous result. Then the last value that was given can be assumed, especially if the last value was a fault as faults are unlikely to heal them selves.
5.6 What Happens If There Is a Double Fault

Although methods for handling multiple faults in the same subsystem have been discussed the system that has been created and evaluated over is not built to handle them. The probabilities that are calculated are the probability that specific mode is the only underlying mode of the system. This investigation intends to find out what happens if there is a double fault anyway.

Setup

The same systems that are described in Tables 5.2 and 5.3 are used in this simulation. For each observation that is simulated there are two different underlying modes that are randomly generated. Neither of these modes can be the mode \( n_f \).

The average expected correctness is calculated from the probability of both underlying modes. Two new variables are created for this experiment. The maximal average expected correctness, \( \mu_{\text{max}} \), which is the average expected correctness of the mode that of the two underlying modes gets the greater part of the probability mass. The minimal average expected correctness, \( \mu_{\text{min}} \), which is the average expected correctness of the mode that of the two underlying modes gets the smallest part of the probability mass. This will visualize if both underlying modes get high probability.

The definition of a correct classification is in this section defined as the case where one of the two modes is given the highest probability.

The simulations are made for different probabilities of false positive and false negative observations. The values that are used are shown in Table 5.14.

Table 5.14: The Probabilities of False Positive and False Negative Observations That Are Evaluated For

<table>
<thead>
<tr>
<th>( P_{fp} )</th>
<th>( P_{fn} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.3</td>
</tr>
<tr>
<td>0.001</td>
<td>0.1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Results and Comments

The result to the simulations is shown in Figures 5.17, 5.18, 5.19 and 5.20.

Figure 5.17 shows how the average expected correctness decreases as the probability of false observations decrease. The reason for this is that when the isolation relies on training data and training data is collected for single faults then observations from double faults will not look like training data for a single fault unless there are false positive alarms in the training data. No false positive alarms will not allow for matches in the training data unless two different modes look exactly the same.
Figure 5.17: Plot of $\bar{\mu}$ for System One and Two for Isolating Observations with Double Faults and Different Probabilities of False Positive and Negative Observations

Figure 5.18: Plot of $\bar{\nu}$ for System One and Two for Isolating Observations with Double Faults and Different Probabilities of False Positive and Negative Observations
Figure 5.19: Plot of the CCDF of the Correct Underlying Modes, the Most Likely Underlying Modes and the Least Likely Underlying Modes Probabilities for Isolating Observations with Double Faults With $p_{fp} = 0.01$ and $p_{fn} = 0.3$
Figure 5.20: Plot of the CCDF of the Correct Underlying Modes Probability for Isolating Observations with Double Faults for Different Probabilities of False Positive and Negative Alarms

When probabilities for false observations are high the expected correctness for the correct mode with most probability gets about 0.45 average probability in system one while the correct mode gets only close to 0.1. Since 0.1 is about what a mode would get if there was no training data (Table 5.4) this is not a good isolation of both underlying modes.

In system 2 the same thing can be said but the highest average expected correctness is even lower and hence the isolation is even worse for this system.

Figure 5.18 illustrated the cpdf of both the highest probability of the correct modes and the lowest. It shows that the mode that gets the least probability mass never gets more than 25% probability and that the probabilities of having any correct mode isolated with a lot of probability is low.

Figure 5.18 shows how the expected probability of correct classification decreases as the probabilities of false alarms decrease. The reasons are the same as for the decrease in the expected correctness in Figure 5.17 decreases.

The expected probability of correct classification is however surprisingly high for system one with high probabilities of false observations but better probabilities rapidly decreases the expected probability of correct classification to values that are low.
Summary

The isolation when a double fault occurs is very poor as the system is designed to isolate single faults. The high probabilities that do occur for modes isolated when there are double faults get their probability mass because of false negative alarms that occur in the ‘right’ places. If the probabilities of false positive and negative are changed the isolation of double faults gets worse.

In order to be able to isolate double faults training data needs to be collected for specific double faults and isolated as if it is a mode. This is described in Section 3.3.2.
Chapter 6

Taking Prior Information Into Account

The parallel thesis [2] has developed a method for handling the case when there is no training data. The method is based on calculating the apriori $\alpha$ in the likelihood in Section 2.8.2 based on the observation, the decision structure and assumed probabilities of false positive and negative alarms.

The method requires more knowledge about the system than the isolation method in this thesis but does not use any training data. The knowledge about the system does however have to be relatively accurate for a reasonable isolation.

The combination of the two methods will allow training data to compensate for the wrong assumptions about the system in the parallel method without training data from [2] while it at the same time is able to perform isolation for modes where there is no or little training data.

\[
p(X = x \mid M = m, D, I) = \frac{n^m_x + \alpha^m_m}{N^m + A^m} \tag{6.1}
\]

A decision on how to weigh the result from the different methods must be discussed. The method in [2] calculates the apriori $\alpha$ by calculating $\alpha^m_x/A^m$ and then multiplying by $A^m$. The relative size of $A^m$ and $N^m$ is thus what decides the weight between the probability from training data and the probability from the apriori and can be decided by the system designer.

6.1 Weighing the Two Methods

As stated above the value of $A^m$ that is chosen by the system designer is a parameter that decides how to weigh the combination of the two different isolation methods presented in this thesis and the parallel thesis [2].

$A^m$ has previously been chosen as total number of possible observations. For example, in a system with five tests and a binary test results, this number has been chosen as $2^5 = 32$. 

67
When large quantities of training data is available the a-priori should not matter. However, when the amount of training data is small the weight is given more to the a-priori distribution than the distribution of the training data. \( A^m \) should therefore be chosen as a value of what amount of training data that is required to have a reliable isolation. This limit is not an exact limit as the weight will gradually shift from the a-priori to the training data. For example, if \( N^m = 2 \cdot A^m \) then twice as much weight is given to the training data for that mode, but one third of the weight is still given to the a-priori knowledge when the posterior probability is calculated.

A short investigation is conducted to see how much training data that is required for the calculated probabilities to reach acceptable values. The investigation is conducted on the second system described in Table 5.3 which has 17 tests and 2 or 3 tests that test each mode. The investigation is conducted on this larger system as it is the 'worst case' system described in this thesis.

![Graph](image)

**Figure 6.1:** Plot of \( \frac{n^m}{N^m} \) against \( N^m \) for Four Different Observations and the Same Behavior Mode

Simulations are conducted with probabilities of false positive and negative as previously, \( p_{FP} = 0.01 \) and \( p_{FN} = 0.3 \). Figure 6.1 shows how the ratio of training data per observation and total amount of training data per mode, \( \frac{n^m}{N^m} \), of different observations change as the amount of training data increases. Four different observations are used to illustrate that different observations all tend toward the right probability in the end, information about the observations is shown in Table 6.1. All four observations are from the same underlying mode. A solid line in the plots of Figure 6.1 shows the 'ideal' probability which
is calculated with the method for calculating $\alpha^m_X/A^m$ described in [2]. Two dotted lines are also plotted to show the $\pm 10\%$ confidence interval of the ideal probability for each plot.

Table 6.1: Information About the Four Different Observations in the Experiment on the Amount of Needed Training Data

<table>
<thead>
<tr>
<th>Observation</th>
<th>Number of False Positive</th>
<th>Number of False Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$X_2$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$X_3$</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>$X_4$</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

In Figure 6.1 shows that little training data results in wrong assumptions of the real distribution of training data, but as the amount of training data increases it gets fairly accurate. At 400 training data per mode the distribution of the observations for a mode are either within or close to the 10% confidence interval.

Choosing $A^m$ to be 100 gives more weight to the training data when there are more than 100 samples collected per mode but still allows some weight to the a priori distribution when there is considerably more training data. This will also give sufficient weight to the a priori distribution when the training data is low thus compensating for the faulty distribution.

How to choose $A^m$ varies from system to system. It depends on how much training data that is required and, most importantly, how accurate the model used for the calculation of the a priori is. For example if the probabilities of false positive and negative are smaller then the amount of training data does not have to be as large as the distribution of observations won’t be as large.

### 6.2 Evaluation of the Combined Method

#### Setup

The setup for these simulations are the same as for the experiments in Chapter 5, i.e. as described in Table 5.1. The evaluation with the method for calculating the a priori distribution does however require additional setup to the system. For this it is assumed that the decision structures in Tables 5.2 and 5.3 are known. The values of false positive and negative cannot be exactly known but in this experiment the assumed values are the same as the values used in the model. A further study of how wrong assumptions affect the isolation in discussed in [2].

#### Results and Comments

Figures 6.2 and 6.3 shows the results of the combined method and compares it to the equivalent method in this thesis with the same amount of training data and the same probabilities of false positive and negative.
Figure 6.2: Plot of $\mu$ and $\nu$ for the Combination with the Thesis [2]

Figure 6.3: Plot of the CCDF of the Correct Underlying Modes Probability for the Combination with the Method in [2]
Table 6.2: Results from Figure 6.3 Showing the Probability that the Probability of the Correct Modes is Greater Than 70%  

| Training data per mode, N     | Probability  
|------------------------------|---------------  
|                              | $p(p(\text{correct mode}) \geq 0.7)$ |  
|                              | System 1 | System 2 |  
| Only training data          | 66.8%    | 52.3%    |  
| Combined with method from [2] | 69.5%    | 59.9%    |  

Figure 6.2 shows a larger increase in the expected correctness than in the expected probability of correct classification and a general increase in the larger and more complex system described by Table 5.3. The reason is that the amount of training data that was used in this experiment (400 samples per mode) does not allow for an isolation in the second system that is as good as the same amount of training data allows for in the first system. The probabilities that the method with only training data gives are thus further from the result that the ideal isolation would give. The use the method with the calculated apriori can thus improve the result more toward the ideal result for the larger system than for the smaller system.

Figure 6.3 indicates the same result. A larger amelioration in the second system than in the first system.

**Summary**

The combination of the method in this thesis and the method in [2] improves the system considerably. The values in Table 6.2 are higher, has a better performance, than the values achieved with only training data when there was 4000 training data per mode.

The greatest use of this method is when there is little or no training data which is the case for some modes. Also at this point in time there is no training data at all collected thus this combination will allow to implement the isolation system and then add training data later.

### 6.3 Training Data Compensates for Wrong Models

The method in [2] is a perfect method if any models and assumptions are correct. The problem with this method is that if the model is not correct then the result will also be incorrect. Combining the method in [2] with the method in this thesis will allow to correct for wrong assumptions. The following simulations are conducted to see how wrong assumptions can be corrected by the use of training data.
Setup

The same setup and models as in previous simulations are used in this experiment. The real models that are used are the same two systems that are displayed in Tables 5.2 and 5.3. The same probabilities of false positive and negative are used, $p_{fp} = 0.01$ and $p_{fp} = 0.3$. The assumed values needed for the combination with [2] are the same but the decision structure that these calculations are based on are showed in Tables 6.3 and 6.4. The boxed values shows which values that are wrong with respect to the real model with which data is generated.

Three sets of simulations are made for both systems. One simulation without training data and with the correct model known, one simulation without training data and with the wrong model and one third simulation with the wrong model and 400 training data per mode.

Table 6.3: The First Decision Structures Modified for Experiment With Wrong Model

<table>
<thead>
<tr>
<th></th>
<th>$m_1$</th>
<th>$m_2$</th>
<th>$m_3$</th>
<th>$m_4$</th>
<th>$m_5$</th>
<th>$m_6$</th>
<th>$m_7$</th>
<th>$nf$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>$\Box$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$\Box$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\Box$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_4$</td>
<td>0</td>
<td>$\Box$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_5$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\Box$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_6$</td>
<td>0</td>
<td>0</td>
<td>$\Box$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_7$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\Box$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_8$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\Box$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_9$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\Box$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\Box$</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 6.4: The Second Decision Structures Modified for Experiment With Wrong Model

<table>
<thead>
<tr>
<th></th>
<th>$m_1$</th>
<th>$m_2$</th>
<th>$m_3$</th>
<th>$m_4$</th>
<th>$m_5$</th>
<th>$m_6$</th>
<th>$m_7$</th>
<th>$m_8$</th>
<th>$m_9$</th>
<th>$m_{10}$</th>
<th>$m_{11}$</th>
<th>$n_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_3$</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
</tr>
<tr>
<td>$x_5$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_6$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_7$</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_8$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_9$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
</tr>
<tr>
<td>$x_{11}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
</tr>
<tr>
<td>$x_{12}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_{13}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_{14}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_{15}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
</tr>
<tr>
<td>$x_{16}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
</tr>
<tr>
<td>$x_{17}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$X$</td>
<td>0</td>
</tr>
</tbody>
</table>
Results and Comments

Figure 6.4: Plot of $\bar{\mu}$ and $\bar{\nu}$ Showing How Training Data Corrects Faulty Knowledge

Figures 6.4 and 6.5 show the achieved results.

Figure 6.4 show that for system one both the expected correctness and the expected probability of correct classification are increased when adding training data. For the second, larger, system the expected correctness is increased but the expected probability of correct classification remains at the lower level achieved without training data. There are several reasons for this, but the most obvious is because 400 training data per mode is not as effective for this larger system as for the smaller system.

Figure 6.5 also shows how adding training data to a faulty model approaches the result to the correct model.

Summary

This experiment shows that adding training data to the isolation of a process increases the accuracy and robustness to human errors of the isolation. This is the main purpose and greatest advantage of the isolation system developed and implemented in this thesis.
Figure 6.5: Plot of the CCDF of the Correct Underlying Modes Probability Showing How Training Data Corrects Faulty Knowledge
Chapter 7

Evaluation of the Practical Aspects of an Implementation

7.1 Important Parameters and Data

In order for the isolation system in this thesis to work properly there are certain parameters that must be known. In this section these parameters are discussed. Some of these parameters already exist while others have to be fixed or collected.

\[
p(M = m \mid X = x, D, I) = \frac{p(M = m \mid I) \ n_x^m + \alpha_x^m}{\pi_X N^m + A^m}
\]  

(7.1)

Equation (7.1), the posterior probability, taken from Section 2.8.4 shows what is needed for the isolation. The prior probability for the modes, the training data and any parameter that must be known for the computation of the apriori distribution of the observation if the combination with the method in [2] is used.

7.1.1 Training Data

The most important data in the isolation described in this thesis is the training data. If there is no training data the isolation does not depend on the observation but always gives the same result to the isolation. Today no training data is available and therefore has to be collected. A reasonable amount of training data must be collected for each mode. The amount that must be collected depends on how the process is constructed. If the probabilities of false positive and negative alarms are low the amount of training data that must be collected is smaller because the spread of observations becomes smaller.

When collecting training data it is very important to know that the process is only in the behavior mode that data is being collected for. Otherwise the training data is data for another mode. This has partially been solved as training data is collected for each subsystem as described in Section 3.2 and a fault in another subsystem does not change training data.
7.1.2 Prior Probabilities

The prior probability, \( p(M = m \mid I) \), is a parameter that is required in the calculation of the posterior probability in Equation (7.1). It defines how probable a mode is compared to other modes. This value has to be set by the system designer and therefore it might be a source of error.

It is argued that if this value is set too high for a mode then, even if this is the true prior, when another mode occurs the mode with the high prior gets the highest probability. Because of this the prior should not be set so that it varies too much from mode to mode.

If the prior is unknown, setting the prior to be the same for all modes should still give a reasonable isolation.

7.1.3 Decision Structure and Assumed Probabilities of False Positive and Negative

The nice thing about the method with training data is that the relationship between modes and tests does not have to be known at beforehand. With enough training data these relationships will appear by themselves, no decision structure is needed.

If the combination with the method in [2] is used there are additional parameters that must be known. The decision structure and the probabilities of false negative or false positive alarms for each element in the decision structure must be known to a certain precision.

The decision structure is the only parameter that is known today. How close to the incidence structure that the decision structure is should be investigated but it should be good enough as it is today if the combined isolation method with training data is applied.

[2] discusses how the decision structure and the probabilities of false positive and negative can be stored as one matrix since there is only one value of either false positive if there is a 0 in the decision structure or false negative if there is a \( X \).

To know the exact values of false positive and negative is impossible, especially since there are thousands of elements in the decision structure of the full-scale process of an engine in a truck. If a standard value is estimated for all element in the decision structure, one for each 0 and one for each \( X \). Then the value of certain elements can be changed if more precise values are known.

The idea is that when a new test is implemented these values are decided from the beginning with the possibility to adjust them later.

7.2 Risks and Difficulties

Before implementing an isolation system such as the one discussed in this thesis the risks and difficulties of an implementation must be known and investigated so that a judgment about the possibility of an implementation can be made. It
is important to solve problems that remain and know where problems may arise when implementing a new system.

7.2.1 Wrong Model

If training data is collected the isolation is based on the true model. But training data cannot be collected for all modes so the use of the combined method with [2] must be implemented. This is great as long as the model is the correct model. If there are errors in the model these errors will cause error in the isolation. Serious errors in the isolation can be dangerous, for both truck and driver.

If the combined method is implemented the prior and the training data could contradict each other if the model is wrong and cause the probability mass to be shared between several modes.

To avoid this problem collecting training data and verifying the model are the only solutions.

7.2.2 Double Faults

The isolation calculates the posterior probability based on the knowledge that there is one and only one behavior mode present in the process. The division of the decision structure into subsystems, Section 3.2, solved the problem of double faults in different subsystems. The problem is when double faults occur within the same subsystems that this becomes a problem.

Double faults that are caused as an effect of a first fault causing another then this will show up in the training data. Two faults that occur simultaneously and independent of each other must however be implemented as a separate behavior mode in the decision structure.

7.2.3 Collecting Training Data

The method in this thesis is based on training data. Training data must be collected with a considerable amount for each behavior mode.

One difficulty with this method is that training data cannot be collected for all behavior modes. It is impossible to manually set the process in some modes for the collection of training data.

Collecting training data is also an extensive process. It will take time before training data exists in such a quantity that this isolation system works as intended.

Isolating when there is training data for some modes and none for others requires fine tuning in how to weigh the method using training data and the method based on the calculation of the a priori.

If each truck must have different training data then the process of collecting training data could turn out to be too difficult. Hopefully training data is the same, at least the majority, for different trucks.
7.2.4 Storing Training Data

Training data is stored as described in Section 2.7 and in particular Table 2.2. Each observation that occurs for a mode is stored as well as how many times it occurred.

The amount of storage needed depends on how large the spread of observations is for each mode. Each different observation must be stored once and then a number indicating how many times that observation occurred. In other words, if only one observation occurs for a mode it does not matter how many training data is collected as this will only change the value indicating the amount.

The maximal amount of training data that must be stored can theoretically be very large. Worst case is when every possible observation occurs in the collection of data as this would mean $2^q$ different observations to store where $q$ is the number of tests in that subsystem. Increasing the amount of training data does not imply a larger storage requirement.

7.2.5 Runtime

Compared to consistency based isolation systems the method in this thesis performs more computations. This implies longer time cycle for one isolation. Since the system is limited in terms of runtime this could become a concern. In order to save runtime it is possible to skip isolations over subsystems where the observation contain only zeros, i.e. no alarms occurred.
Chapter 8

Conclusions,
Recommendations and Future Work

8.1 Conclusions

• An implementation of a stripped down version of a probabilistic isolation system based on training data has been implemented in a control system for the engine of a truck. This implementation together with an artificial model built in a computer environment has served as a basis for tests and experiments.

• An investigation of the robustness of the probabilistic isolation system has been conducted on a model of the process in Chapter 5 where parameters such as the quality of the tests, the amount of training data and multiple faults at the same time is experimented with.

• Difficulties with the probabilistic isolation method have been discussed to show where problems are likely to arise when an implementation of an isolation system of the kind is implemented in Chapter 7. These difficulties are mentioned in Chapter 7 and comprise topics such as collecting and storing training data and optimizing runtime.

• Recommendations and discussion of a possible implementation of the probabilistic isolation system as well as a recommendation to further work is given Sections 8.2 and 8.3
8.2 Recommendations

The effectiveness of the probabilistic isolation system cannot be denied and is therefore recommended as an isolation system to continue further development. There are however questions remaining that should be solved before a full scale implementation is possible. These questions include topics such as how to find a good isolation of double faults that find all underlying modes, how to make sure that the runtime of this more complex isolation system does not exceed values that are acceptable and the question of how to collect and store training data in an effective way. If the combination with the work in [2] is also used the different parameters needed for the calculations needed this method must also be found with a certain precision.

This thesis shows that an implementation of this system is something that should be continued with the knowledge that more work has to be done. A more detailed recommendation on what future work should be is given in Section 8.3.

8.3 Future Work

As the recommendation states that more work is needed before a full scale adaptation can be made there is much more work to be done.

A full scale implementation must be performed so that an evaluation in contrast with the isolation system of today can be performed to get values and measurements on increased runtime and isolation accuracy can be performed. Possibly as another master’s thesis by an engineering student with a background in computer science as programming skills and knowledge about embedded systems are useful.

Solutions to the remaining problems can be researched and tested in an implemented version of the system. These problems include the collection and storage of training data, solution to double faults in the same subsystem and derivation of any values needed for an implementation of the conclusion and such as probabilities that allow for an isolation of modes without training data.
Appendix A

Variables and Notation

<table>
<thead>
<tr>
<th>Variable</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>The system behavioral mode, describes the modes of all components, $M = (M_1, M_2, \ldots, M_r)$</td>
</tr>
<tr>
<td>$M_i$</td>
<td>Variable describing the mode of a component, $M_i = m_i$</td>
</tr>
<tr>
<td>$m$</td>
<td>A specific system behavioral mode, $M = m = (m_1, m_2, \ldots, m_r)$</td>
</tr>
<tr>
<td>$m_i$</td>
<td>Specific mode of a component. $m_i \in {f_1, f_2, f_3, \ldots, nfu}$</td>
</tr>
<tr>
<td>$M_{ij}$</td>
<td>A combination of two modes.</td>
</tr>
<tr>
<td>$f_i$</td>
<td>notation for a fault</td>
</tr>
<tr>
<td>$nf$</td>
<td>Behavior mode describing &quot;no fault&quot; in one specified component</td>
</tr>
<tr>
<td>$uf$</td>
<td>Behavior mode describing &quot;unknown fault&quot; in one specified component</td>
</tr>
<tr>
<td>$M_S$</td>
<td>The set of all possible system behavior modes</td>
</tr>
<tr>
<td>$X$</td>
<td>An observation $X = (X_1, X_2, X_3, \ldots, X_q)$</td>
</tr>
<tr>
<td>$X_i$</td>
<td>A specific testresult in the observation $X_i = x_i$</td>
</tr>
<tr>
<td>$x_i$</td>
<td>The specific value of a testresult, $x_i \in {0, 1}$ in the binary case</td>
</tr>
<tr>
<td>$X_S$</td>
<td>The set of all possible observations</td>
</tr>
<tr>
<td>$-\ 0\ 1$</td>
<td>Notation for a test results that has not given a value</td>
</tr>
<tr>
<td>$q$</td>
<td>Number of tests</td>
</tr>
<tr>
<td>$r$</td>
<td>Number of behavior modes</td>
</tr>
<tr>
<td>$S$</td>
<td>Notation for the decision structure</td>
</tr>
<tr>
<td>$\mathcal{X}$</td>
<td>Denotes a dependency in the decision structure</td>
</tr>
<tr>
<td>$I$</td>
<td>Background information</td>
</tr>
<tr>
<td>$D$</td>
<td>Training data</td>
</tr>
<tr>
<td>$NF$</td>
<td>System behavior mode describing whole system as &quot;no fault&quot;</td>
</tr>
</tbody>
</table>
Appendix B

Probability Rules

In this thesis it is assumed that the reader has been exposed to the basics of probability. Nevertheless, in this appendix there will be a short recapitulation of some notations.

B.1 Basic Rules

\[ p(AB) = p(A)p(B \mid A) = p(B)p(A \mid B) \]  \hspace{1cm} \text{(product rule)} \hspace{1cm} (B.1)

Where \( A \) and \( B \) are two separate events.

\[ p(A) + p(\bar{A}) = 1 \]  \hspace{1cm} \text{(sum rule)} \hspace{1cm} (B.2)

Where \( \bar{A} \) is the complement of \( A \).

B.2 Bayes Rule

Rewriting Equation (B.1) we can derive Bayes Rule which states:

\[ p(A \mid B) = \frac{p(B \mid A)p(A)}{p(B)} \]  \hspace{1cm} \text{(Bayes rule)} \hspace{1cm} (B.3)

B.3 Marginalization

\[ p(A) = \sum_i p(AB_i) = \sum_i p(A \mid B_i)p(B_i) \]  \hspace{1cm} \text{(Marginalization)} \hspace{1cm} (B.4)
Bibliography


