OPERATIONAL INDUSTRIAL FAULT DETECTION AND DIAGNOSIS: RAILWAY ACTUATOR CASE STUDIES

by

Joseph Alan Silmon

A thesis submitted to the University of Birmingham for the degree of DOCTOR OF PHILOSOPHY

Department of Electronic, Electrical and Computer Engineering
School of Engineering
University of Birmingham
July 2009

Abstract

Modern railways are required to operate with a high level of safety and reliability. The weakest components are those which have the highest safety requirements and the lowest inherent reliability. Single-throw mechanical actuators, such as powered train doors, trainstops, level crossing barriers and switch actuators (point machines) are a group of components which have these properties.

Preventative maintenance is carried out periodically in order to mitigate the risks of these actuators failing. This is inefficient: a condition-based maintenance approach would reduce costs and the risks to staff. However, this kind of maintenance requires very accurate automatic condition monitoring. Currently, the threshold-based condition monitoring systems installed in pilot schemes around the country do not have enough insight into actuator performance to detect incipient faults. These are hard to spot because their symptoms develop over a long period of time.

It is uneconomical to carry out detailed analysis or modelling, or collect a large amount of training data, for each instance of a large group of assets. Therefore, the solution needed to establish diagnosis rules based on offline analysis, or training data from only one actuator.

This thesis draws on previous work in qualitative trend analysis to build a diagnosis system which uses a combined approach of qualitative and quantitative analysis to transfer the knowledge gathered from one actuator to its fellows in service. The method used has been designed to use straightforward components, so that it can be more easily explained to users.

Two case studies were carried out in order to verify the system's functions. Data were collected from real-life actuators, under simulation of incipient faults. The diagnosis system then operated on the data. The system's performance was almost as good with real-world data as it was with synthetic data.

The system has been a success when operating on the data gathered under laboratory conditions. In the real world, a system such as this could be used to post-process data gathered around the railway network from actuators with local data acquisition equipment. Incipient faults could be detected in the early stages of their development and accurately diagnosed, allowing maintenance effort to be targeted very specifically, saving money, time and exposing staff to fewer hazards.

To all those friends and family members who have loved, supported and tolerated
me over the past four years

Acknowledgements

The author thanks the following parties for their contributions to the project:

The Engineering and Physical Sciences Research Council, for covering the author's course fees and maintenance for the duration of the project

Clive Roberts, for his help, friendship and support as the Project Supervisor

Edd Stewart, for his friendship and assistance with data collection

Sharon Berry, for her administrative support and friendship

Andy Dunn, for his technical support of the practical parts of the project

Rhys Davies and Gareth Webb, for IT support

Phil Neep of GM Rail, for allowing and assisting with experimentation on the point machine training facilities at the GM training school in Rugby

Barry Fox of Angel Trains and Tony Frost of Knight Rail Services, for faciliating the collection of fault simulation data from pneumatic train door actuators on class 442 EMUs

Jim Digby of South West Trains, for facilitating the collection of fault simulation data from pneumatic train door actuators on class 158 DMUs

Network Rail, for facilitating fault simulations on HW switch actuators and for improving the author's understanding of industrial requirements.

Wim Pauwels and his colleagues at Strukton RailInfra B.V. for providing a large set of measured data from point machines

Vitech Corporation, for providing a free academic licence for their CORE systems engineering software

The participants of the EU-funded INNOTRACK project, for networking, support, and allowing data collection from AC switch actuators.

Contents

List of Figures

List of Tables

List of mathematical symbols

Symbol	**Meaning**
$\mathcal{A} - \mathcal{I}$	Qualitative state identifiers for episodes and partitions
A	Set of actuators of one particular type, in an asset base
a_t	Actuator in A on which fault simulations are carried out
b	Fault intensity percentage, where 100 % indicates a severity which is on the point of causing failure
c_1	Parameter for Π-shaped membership function generation, equating to the value at which membership on the rising edge is exactly 50 %
c_2	Parameter for Π-shaped membership function generation, equating to the value at which membership on the falling edge is exactly 50 %
$\underline{\mathbf{D}}$	Matrix of filtered data for one actuator operation
E	Set of all possible behaviours in actuators of set A
F	Set of diagnosable actuator conditions
G	Set of observable actuator conditions
f	Member of F, i.e. an actuator condition in the scope of this work
g	Scaling factor for normalising magnitude of units of time and measured variable

Symbol	Meaning
h_1	Parameter for Π-shaped membership function generation, indicating the steepness of the rising transition
h_2	Parameter for Π-shaped membership function generation, indicating the steepness of the falling transition
i	Counter to indicate what condition is currently being considered (i.e. the condition domain)
j	Counter to indicate what measurement repitition is currently being considered
k	Generic counter in the time-domain, indicating what partition or episode is currently being considered
k_e	Counter in the time-domain, indicating the number of the current episode [1]
k_p	Counter in the time-domain, indicating the number of the current partition
κ	Counter for calculation of mean second differential
l	Counter in the domain of measured variables, indicating the number of the parameter currently under consideration
$\underline{\mathbf{M}}$	Matrix containing the measured data from one actuator operation
n_c	Number of common episodes in the profile for a particular variable
n_F	Size of F
n_v	Number of variables/parameters measured by the condition monitoring equipment
n_e	Number of episodes in a profile

[1] This and k_p used only in the function 1.1.5, where counters for both episodes and partitions are required simultaneously

Symbol	Meaning
$\underline{\mathbf{P}}$	Matrix containing the profile for one variable in one actuator operation
R	Number of repetitions in a condition simulation (i.e. the number of operations carried out under that condition)
s	Qualitative state of a partition; takes a value from $\mathcal{A} - \mathcal{I}$
σ	Qualitative state of an episode
T	Sampling period
t_x	The start time of a partition
t_y	The end time of a partition
τ_x	The start time of an episode
τ_y	The end time of an episode
x	The start value of a partition (plain) or episode (with subscript e)
y	The end value of a partition (plain) or episode (with subscript e)

Chapter 1

Introduction

1.1 Railway actuators and reliability

There are several key applications in the railway industry which require reliable, fail-safe mechanical actuators. These include barriers for level crossings, train doors, trainstops and point machines (also known as switch actuators). Each of these applications is safety-critical, so the actuators are designed to fail to a safe position. This often means that an actuator failure causes delays to train services, because it is often impossible for trains to run under the degraded conditions created by a safe or 'right-side' failure. During busy periods, one short delay can often have a greatly detrimental effect to the punctuality of following services and thereby incur financial penalties for train operators and the infrastructure organisation.

Recent accidents at Potters Bar (2002) and Grayrigg (2007) were caused by 'wrong side' actuator failures, where the equipment does not fail to a safe position. The consequences of such failures are much worse than right-side failures; one passenger perished at Grayrigg, whilst at Potters Bar, a total of seven deaths occurred (six passengers and one passer-by, who was killed by falling debris). In the case of the Potters Bar accident, the infrastructure owner and the former infrastructure maintainer jointly accepted financial liability for the accident [1]. The total value of compensation claims is expected to reach several million pounds.

There are, therefore, two incentives for the rail industry to develop a means of detecting faults in actuators before they actually fail:

- To minimise right-side failures and thereby keep the railway running smoothly
- To minimise wrong-side failures and thereby avoid injury and loss of life

By achieving these two goals, the rail industry avoids financial penalties, loss of reputation and the possibility of prosecution for negligence.

1.2 Current maintenance practice on the railways

1.2.1 Prediction of faults

There are three types of fault which can affect railway actuators:

Abrupt - a fault whose symptoms manifest themselves suddenly; in an actuator, the fault can appear after many thousands of apparently fault-free operations

Intermittent - a fault whose symptoms are not always observable; they tend to appear for a few operations and then disappear, making them difficult to isolate

Incipient - a fault which develops gradually over a period of time, with its symptoms gradually becoming stronger

Abrupt faults are difficult to predict, because an actuator can be observed to be working perfectly during one operation, and then to fail completely the next time it is operated. Intermittent faults are similarly difficult to predict, because there are no indicators as to when the fault will appear.

Incipient faults, however, can be predicted with some accuracy, providing the correct parameters of the actuator are observed for every operation. The development of an incipient fault is shown in figure 1.1, where successive measurements have been plotted as an adjustment fault was gradually introduced on a HW switch actuator.

The waveforms have been coloured from green to red according to how severe the fault was at the time - green plots are from early in the development of the fault, and red plots are later in the development. The reddest waveform corresponds to the maximum severity of the fault which still allows the actuator to complete its operation.

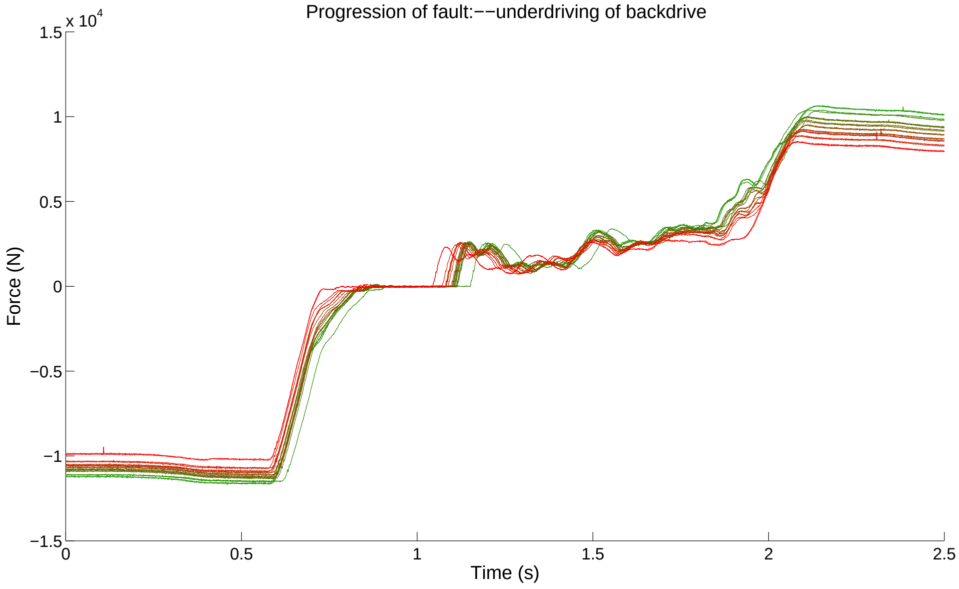

Figure 1.1: Progressive introduction of an adjustment fault into a HW switch actuator

1.2.2 Condition monitoring and diagnosis

When a failure occurs on a railway actuator, the standard practice is for a technician to examine the failed item and use a series of observations to determine the fault.

The technician uses his or her senses of sight and hearing to check for unusual operation, and relies on experience and training to diagnose the fault.

In the case of intermittent faults, this method is unreliable because there is usually a period of time between a failure being reported and the technician examining the actuator. An intermittent fault may have only manifested itself for a few operations and then disappeared, so that when the technician examines the actuator, no fault is found. This wastes time and money, and can expose technicians to risk if they have to work in a dangerous environment (for example, an open railway line where trains are running) in order to carry out the examination.

In response to the clear limitations of a diagnosis approach based on human observations alone, several pilot systems for remote condition monitoring have been introduced on the UK rail network. These systems consist of an array of sensors, a data acquisition device and a communications link to a base station [2]. This allows the actuator to be monitored continuously.

Some incipient faults have been successfully predicted and diagnosed using simple techniques such as triggering alarms when a threshold in a parameter is reached. However, thresholds have a limited use, because sometimes a parameter may take an extreme value as part of normal operation, and thereby breach a threshold when there is no fault present. This creates false alarms which hold up railway operation and reduce the confidence maintainers have in the condition monitoring system. Thresholds must also be tuned to the individual performance of an actuator. This requires lengthy setup times with technicians on site. The ambient temperature at the asset site can have considerable effects on any values measured, for example an asset might work perfectly when adjusted on a spring day of 15°C, but move into a failure mode on a very hot summer's day of 30°C. Equally, the effects of temperature may manifest themselves in measured waveforms as trends away from the normal fault-free data (which would look like a failure mode) but the asset may still be in working order.

Condition monitoring is, for the most part, incorporated in the periodic maintenance regime, which shall be discussed further in the next section.

1.2.3 Preventative maintenance

Right-side actuator failures can, as has already been mentioned, have a big impact on the performance of a railway system. The risk of right-side failure is currently mitigated by a regime of periodic preventative maintenance. Certain components are replaced or maintained at fixed time intervals. For the majority of actuators on the railway network, it is during such maintenance tasks that the condition of the actuator can be ascertained by the maintainer.

The success of preventative maintenance depends on the determination of the correct maintenance interval for each component. If the interval is too large, the risk of failure becomes unacceptable. If the interval is too small, the regime becomes uneconomical. Other risks are also increased if maintenance occurs too often: for lineside equipment, personnel must work in a potentially dangerous area, increasing the risk of accidents; also, there is a greater risk that the intervention may itself cause a fault to develop.

1.3 The design problem

1.3.1 Definitions

This thesis is concerned with the detection and diagnosis of *faults* and, thereby, the prediction of *failures*. *Faults* can be defined as *a deviation between perfect performance and complete failure* [3]. By this definition, then, a piece of equipment may still be able to function with a fault present, but the fault will degrade its performance to a certain degree. A *failure* occurs when the fault is so severe that the equipment is no longer able to function. For example, if a rail switch actuator experiences increased friction because the slide chairs on the switch are ingrained with dirt, then that is always a fault, but it only becomes a failure when the friction is so great that the actuator is unable to drive the switch from one side to the other.

The thesis focuses on monitoring a particular class of device, known as single-throw mechanical equipment (STME). STME has two stable positions but transitions (or *throws*) between the two positions occur in a non-periodic manner. STME usually has a large, non-linear load. The throw time (time taken to move from one stable position to the other) is therefore relatively long and this means that the dynamics of the equipment as a whole are of a comparable timescale to the throw time, unlike electrical switchgear, where the transition occurs much faster [4].

1.3.2 Conclusions from current practice

Section 1.1 leads to a conclusion that there must be a better method of maintaining actuators, where the condition of the actuator determines whether intervention is necessary, rather than a fixed interval of time. This is known as condition-based maintenance.

Condition-based maintenance relies on accurate and comprehensive condition monitoring. Human examination lacks the insight into actuator operation necessary to predict all known faults, and the current methods of remote automatic condition monitoring have potential for improvement, through the development of more sophisticated approaches for analysing the data which is monitored.

1.3.3 Proposal for improvement of remote condition monitoring

Currently, remote condition monitoring for railway actuators consists mostly of diagnosis of abrupt faults and the measurement of key parameters for subsequent time-consuming analysis by trained personnel.

It is proposed that an improvement could be made to this system by automating the monitoring of key parameters, with the aim of detecting incipient faults in the early stages of development, and triggering alarms so that maintenance can be scheduled before the incipient fault develops to the stage where it can cause a failure.

This would improve the reliability of the railways by reducing the number of failures which occur during normal traffic time. There is also potential to improve safety, because wrong-side incipient faults can be detected using the same system. Setup times for monitoring equipment would be reduced because the system would tune itself to the individual performance of actuators.

1.4 Systems engineering approach

Systems engineering is emerging in industry as an effective approach for the development of complex products. The IEEE defines systems engineering, in its standard [5], as :

> *"...an integrated approach to product development, which represents the total technical effort for the following:*
>
> - *Understanding the environments and the related conditions in which the product may be utilised and for which the product should be designed to accommodate*

- *Defining product requirements in terms of functional and performance requirements, quality factors, usability, producibility, supportability, safety, and environmental impacts*
- *Defining the life cycle processes for manufacturing, test, distribution, support, training, and disposal, which are necessary to provide life cycle support for products"*

Depending on the product to be developed, the tasks performed in systems engineering will vary widely, but most system engineering approaches have in common a top-down approach (starting by viewing the system and its performance as a whole), a consideration of the life-cycle of the product, a significant effort to correctly define the initial requirements, and an interdisciplinary approach to the achievement of solutions [6].

In a research project, systems engineering activities can be used to structure the definition of requirements and the evaluation of previous methods, with a view to novel application or inspiration of an original solution. Although this is an unusual approach for a doctoral research thesis, it was appropriate for the problem posed.

Since the problem posed to the author is highly practical and industrial in nature, it was appropriate to use a design approach which is compatible with industrial needs and implementations. However, the author recognises that research does not produce the best possible results if confined within a dogmatic, linear process. Systems engineering has been used in this thesis to help understand the requirements and provide a logical framework for the thought process, but, in practice, considerable freedom has been exercised during the process of the research. The design process is explained in more detail in appendix A.

1.5 Structure of the thesis

The structure of the remaining chapters of the thesis is outlined below.

Chapter 2 - Requirements Analysis

A set of initial requirements were extracted from the briefing document. These requirements were decomposed into single, testable statements so that each could be tested separately. The resulting set of requirements was used to direct the research and helped define the shape of the solution.

Chapter 3 - Method research and evaluation

A comprehensive literature review was carried out in three main sections: first, a general survey of fault diagnosis; second, fault diagnosis as specifically applied to the railways; and third, a review of a series of papers proposing diagnosis methods for a benchmark actuator. The conclusions of this review led to the formation of some generalisations for the structure and operation of railway condition monitoring. Several approaches from the literature search were evaluated in more detail and measured against the requirements.

Chapter 4 - Functional Design

The system to be designed was first represented as a black box, executing the function required with given inputs and outputs. A decomposition was carried out, resulting in a functional representation for each part of the system, at the lowest possible level. This design was then expressed in terms of mathematical functions.

Chapter 5 - Case study I: Pneumatic door actuator

The first case study was carried out on pneumatic actuators for swing-plug train doors. Two faults were simulated on each door of a single vehicle of a class 158 diesel multiple-unit. Pressure and displacement were recorded. The system was able to detect and diagnose the gradual introduction of the faults, but behaved less reliably when the fault effects were proportionally smaller, compared to the general scale of the measured waveforms. One notable feature of this case study is that each actuator was in a very similar setting - because all four doors were identical. The author gratefully acknowledges the cooperation of South West Trains in this case study.

Chapter 6 - Case study II: HW switch actuator

The second case study examined faults on HW switch actuators, used throughout the UK rail network. Several faults were simulated on switches at two Network Rail training schools. Data were recorded as faults were gradually introduced. The results were positive and show a clear ability to detect the onset of faults at an early stage of development. The data in this case study were more challenging to the system and identified some weak points for further work. In contrast to case study I, the actuators were installed with varying loads, making the diagnosis task more difficult. The author gratefully acknowledges the cooperation of Network Rail in this case study.

Chapter 7 - Conclusions

In this section, the results of the case studies are critically reviewed. It is fair to conclude that the system performs its required function and is very sensitive to the early onset of faults. It cannot be used on its own, however, and would be ideal as a plug-in to a larger fault diagnosis system. Further work needs to focus on making diagnosis more accurate, and, in particular, achieving a higher quality of qualitative trend analysis.

Appendix A - Details of the design process

The systems engineering approach was formalised into distinct phases, each of which is represented by a flow diagram. Systems engineering requires iteration, which was incorporated into each phase as required. Phase I was the collection of requirements and initial literature search. Phase II was the functional definition of the system according to the requirements. The system was implemented in phase III and tested and verified in phase IV.

Appendix B - Mathematical notation

Some mathematical notation conventions were adapted to represent certain concepts within this thesis. This enabled simple representation of subsets of matrices and element-wise numerical operations.

Appendix C - Results graphs from case study I

The complete output of the system from case study I is plotted here in graphs of the fault strength against severity of the simulated fault.

Appendix D - Results graphs from case study II

The complete output of the system from case study II is plotted here in graphs of the fault strength against severity of the simulated fault.

1.5.1 Extra-curricular work

In addition to the two case studies included in the thesis, some work has been carried out using the system developed. The INNOTRACK European project for innovative track systems includes a sub-project based on condition monitoring for switch systems. As part of this work, fault simulations were carried out on an AC switch actuator. There is limited scope to test the system with data from only one actuator, but this instance was more relevant to the requirements of the European partners. The results of the use of the system developed here were included in INNOTRACK deliverable 3.3.4 along with much of the method evaluation presented here.

Chapter 2

Requirements analysis

2.1 Initial requirements

The first stage of requirements analysis is to extract an initial set of originating requirements from appropriate source documents. For this thesis, there is a single pertinent source document which is the thesis brief [7]. The text of this document has been used (in modified form agreed with the thesis supervisor) to form a single, top-level requirement from which all others shall be drawn. This top-level requirement is quoted below:

> The thesis shall develop a novel, intuitive and straightforward method for the detection and diagnosis of faults in actuators (Single-Throw Mechanical Equipment). A selection of railway case studies shall be used to demonstrate the generic nature of the methodology developed and the applicability of widespread usage. The actuators which could be used are listed below:
>
> - Westinghouse M63 electric railway switch actuator
> - ALSTOM HW electric railway switch actuator

- Smiths Industries Clamp Lock hydraulic railway switch actuator
- Various brands of London Underground electro-pneumatic trainstop
- Vapor-Stone pneumatic train door actuator
- Vapor-Stone electric train door actuator

A practical implementation, which can be deployed on the railway for demonstration, is desirable, but a sufficiently rigorous lab-based demonstration of the functioning system shall be considered sufficient to prove the concept. In either case, the system shall incorporate a suitable interface to the user for the operation of the machine and the presentation of the results.

2.2 Requirement decomposition

The top-level requirement clearly states what the thesis is to deliver. The purpose of requirements decomposition is to transform this large requirement into a set of individual testable statements which define in more detail what the thesis is to achieve [6]. These statements can then be used as a benchmark to assess the success of the resulting system. Domain knowledge can be an additional input to this process, as it influences the structure of the decomposition and the text of the resulting requirements.

The following requirements were derived from the top-level requirement:

1. A system shall be developed to detect and correctly diagnose present and/or emerging faults. The scope of this function shall be limited to faults which can be reliably simulated on a working actuator.
2. A doctoral thesis shall be submitted to the University of Birmingham,

containing an account of the design process, the method developed, and the results of implementation. The submission shall be made no later than 25th September 2009 (in order to comply with regulations from the funding source, the EPSRC).

3. The method developed shall be demonstrated to solve the design problem by constructing a system in software, testing it with simulated inputs (from measurements made elsewhere) or measurements from a lab-based actuator, and recording the results. The simulated inputs shall have diverse sources, in order to show that the system implements a generic algorithm. If a practical implementation is pursued, this requirement shall still be fulfilled as a simulation step.

4. The method developed shall be demonstrated to solve the design problem by constructing a system in physical form, deploying it on live actuators and recording the results. The actuators used shall be diverse in nature, in order to show that the system implements a generic algorithm.

5. The system shall possess a significant measure of novelty. This could be in the form of new methods developed to fulfil certain system functions or a novel implementation of existing methods.

6. The system shall be able to perform its specified function for at least 3 types of actuator from the following set:

   - Electric railway switch actuators e.g. Westinghouse M63, Alstom HW, Siemens S700 series
   - Smiths Industries Clamp Lock hydraulic railway switch actuator
   - Electro-pneumatic trainstop
   - Pneumatic train door actuators e.g. Vapor-Stone, Kiekert
   - Electric train door actuators e.g. IFE, Faiveley
   - Level crossing barriers

7. The system shall perform its specified function using a method which can easily be explained to a qualified Signalling and Telecommunications maintenance technician.

8. The system shall be able to perform its specified function on multiple instances of the same actuator type, which may be installed in diverse situations, with varying loads and environmental conditions.

9. The system shall incorporate an interface which shall allow the user to interact effectively with it.

10. The system shall be suitably interfaced to its environment and the machine under test. If a practical implementation is pursued, consideration shall be given to railway regulations regarding the acceptance of equipment for use on or near the line. For both practical and lab-based implementations, consideration shall be given to local health and safety requirements.

These top-level requirements can then be decomposed further to result in a set of single, testable statements.

1. *A system shall be developed to detect and correctly diagnose present and/or emerging faults with an adequate level of accuracy. The scope of this function shall be limited to faults which can be reliably simulated on a working actuator.*

   1.1 A system shall be developed to detect and diagnose emerging faults in STME actuators.

   1.2 The system shall detect faults with an adequate level of accuracy to avoid false alarms.

   1.3 Faults shall only be considered for diagnosis if they can be reliably simulated on a working actuator.

   1.4 The list of faults to be diagnosed for each actuator type shall be agreed after consultation with maintainers.

2. *A doctoral thesis shall be submitted to the University of Birmingham, containing an account of the design process, the method developed, and the results of implementation. The submission shall be made no later than 25th September 2009 (in order to comply with regulations from the funding source, the EPSRC).*

   2.1 A doctoral thesis shall be submitted to the University of Birmingham.

   2.2 The thesis shall be submitted no later than 25th September 2008.

   2.3 The content of the thesis shall include an account of the design process, a description of the method developed, and the results of implementation.

3. *The method developed shall be demonstrated to solve the design problem by constructing a system in software, feeding it with simulated inputs (from measurements made elsewhere) or measurements from a lab-based actuator, and recording the results. The simulated inputs shall have diverse sources, in order to show that the system implements a generic algorithm. If a practical implementation is pursued, this requirement shall still be fulfilled as a simulation step.*

   3.1 The method developed to solve the problem shall be delivered as a theoretical, software-based implementation.

   3.2 The software implementation shall operate on data sets from several different actuator types.

4. *The method developed shall be demonstrated to solve the design problem by constructing a system in physical form, deploying it on live actuators and recording the results. The actuators used shall be diverse in nature, in order to show that the system implements a generic algorithm.*

   4.1 *(Optional)* - The method developed shall be delivered as a practical implementation.

4.2 *(Optional)* - The practical implementation shall be suitable for deployment on an operational railway.

4.3 *(Optional)* - The practical implementation shall operate on diverse types of actuator.

Items 5-10: no decomposition required.

These requirements are now sufficiently decomposed that they may be used for testing the system which is developed.

2.3 Impact on the system design

By defining the requirements fully at the start of the design process, many possible options for implementing the system can be excluded with a minimum of effort if they do not fulfil the requirements. This speeds up the process of developing a method for solving the design problem.

2.4 Test specifications

The test specifications for the system are based on the requirements in section 2. In order for testing to be successful, each of the performance requirements must be shown to have been fulfilled.

Table 2.1 shows the test specifications referenced to the performance requirements.

Req. No.	**Spec. No.**	**Test specification**
1.1	**1**	The trend gradient for fault f_i must be positive when f_i is simulated (*Detection*)
1.1/1.2	**2**	The trend gradient for f_i must be greater than all other gradients when f_i is simulated (*Diagnosis*)

Table 2.1: Requirements and test specifications

The term "trend gradient" refers to the gradient of a first-order linear trend constructed by least-squares approximation from the fault membership values paired with the corresponding fault severity figure. This severity figure shall be quantified separately for each fault; it may be a measurement which indicates the severity, or simply the number or magnitude of adjustments made to simulate the fault.

Chapter 3

Method research and evaluation

The set of requirements formed in section 2 exerts a substantial amount of constraint over the type of system which is to be developed. However, fault diagnosis in general is a huge field with large numbers of different possible approaches, and many versions thereof.

In this chapter, published literature on fault diagnosis shall first be reviewed. Then, a selection of approaches shall be examined in greater detail and evaluated against the requirements for this thesis.

3.1 Literature review

3.1.1 Background

Automated fault diagnosis has been an area of high research activity for several decades. Most of the effort in this field has been focussed on industrial processes, such as may be found in the chemical and manufacturing industries. Industrial processes usually have dozens or hundreds of variables and a large, expensive plant which is built as a one-off exercise. This means that investment in fault diagnosis has been higher, because it is a proportionally smaller cost to the operators and yet yields a high return in increased reliability.

The diagnosis of actuators, on the other hand, has not been the subject of as much research effort. Within a large process environment, the failure of a single actuator is unlikely to bring the process to a halt, because redundancy is built in to such a system. The plant diagnostic system identifies an actuator as causing a problem but does not seem to go into any further detail.

The nature of railways is such that failures of actuators can very often lead to a system halt whilst remedial action is carried out. Train door failures cause delays because of the safety procedures that are followed. For example, certain types of rail vehicle cannot be used if a single door is out of action, because they are passenger emergency exits and must always be available. Failed switch actuators cannot be used to throw points, therefore they limit the control a signaller has over the routing of trains.

3.1.2 Layout of the literature review

Section 3.1.3 is a review of diverse diagnosis methods which have been implemented across many different fields of engineering. A review will then be included of articles which are concerned with actuator diagnosis in particular. This consists of two main parts: an overview of the previous research carried out into railway actuator fault diagnosis (section 3.1.5), and then a review of methods developed for the DAMADICS[1] actuator problem (section 3.1.4), where several groups developed solutions for a valve actuator in a sugar factory.

[1] Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems

3.1.3 Diverse methods for fault diagnosis

Figure 3.1 shows a categorisation of methods for fault diagnosis. The diagram has been reproduced from a review paper [8], with the addition of automata to the class of qualitative causal models. There are three main categories of fault diagnosis methods: qualitative model-based, quantitative model-based, and process history-based.

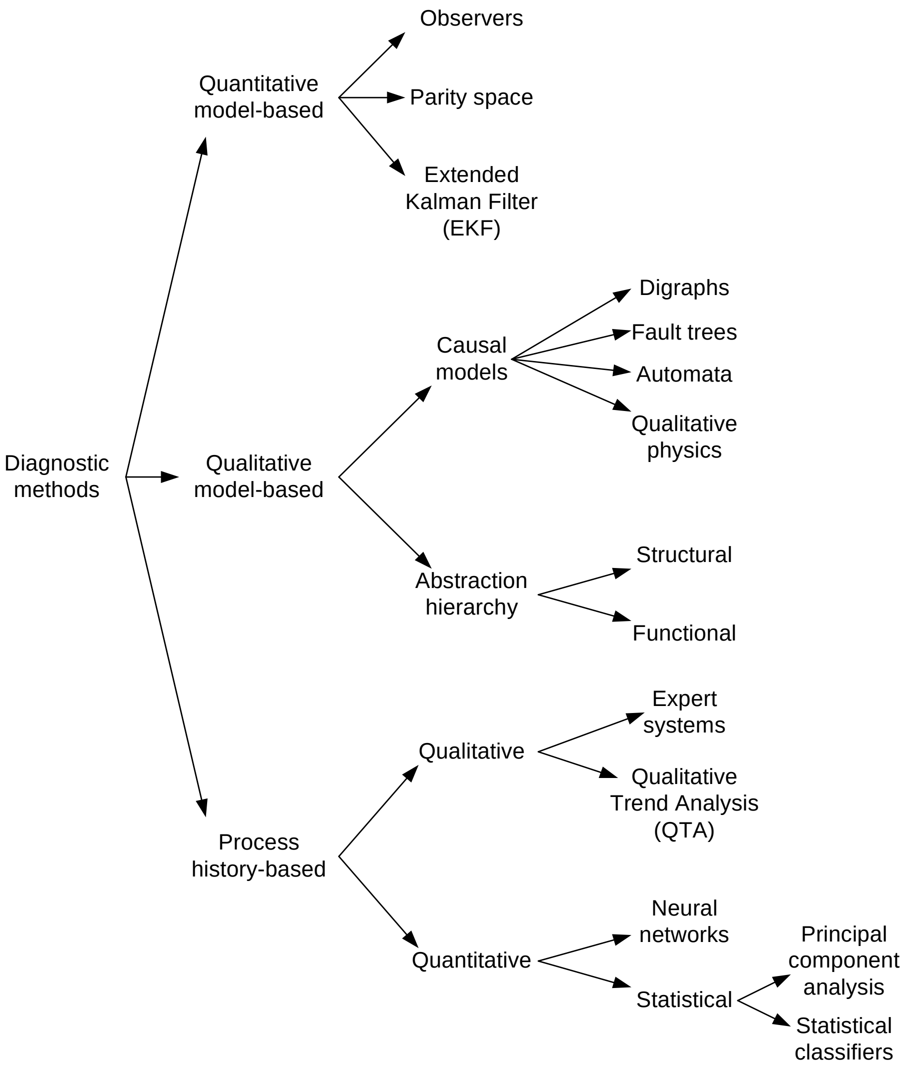

Figure 3.1: Categorisation of methods for fault detection and diagnosis [8]

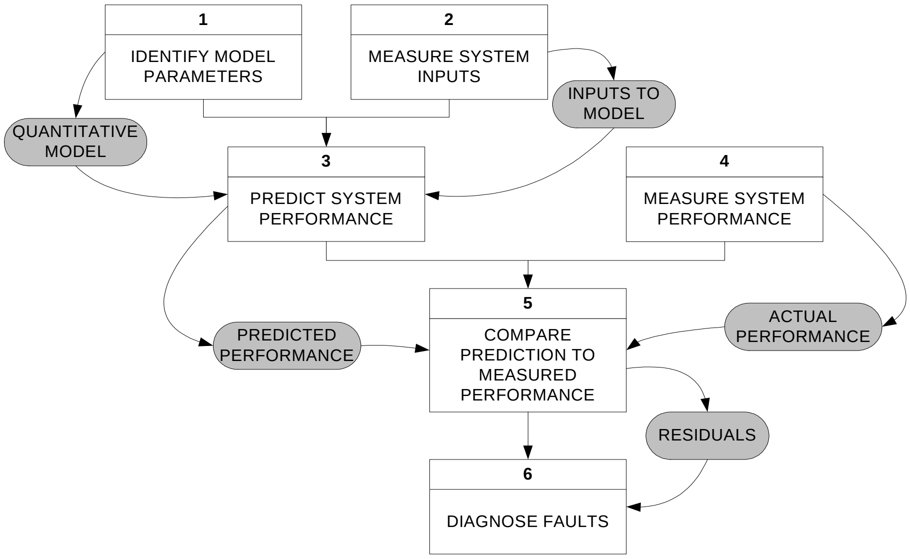

Figure 3.2: General structure of a quantitative model-based method for fault diagnosis

Quantitative model-based methods

The basic concept of these methods for fault diagnosis is to establish a mathematical model of the plant to be diagnosed, using a functional structure such as that shown in figure 3.2. This model then predicts the values of the plant's measurable variables. Signals known as residuals are generated by comparing the predicted variables to those measured from the plant itself. These residuals can then be used to determine if any faults are present, and what they might be. The most common functions for residual generation are subtraction of the model signal from the measured signal (to detect additive faults) and division of the measured signal by the model signal (to detect multiplicative faults).

Many methods exist for modelling the performance of dynamic systems, but the dominant form of model in the field of fault diagnosis is the state-space input-output model [8], usually in discrete form because variables are measured on a sampled basis.

One advantage of such a model is that it quite often exists already in the controller of a large piece of plant. State-space models are used to estimate the state variables of a system when they cannot be readily measured directly. This model is known as an *observer* and is used to provide more insight into a system's operation and therefore enable it to be better controlled. Mathematical models of this type have been used for some time for various applications including servo motors [9] and jet engines [10, 11].

Parity equations (or relations) are another widely-used method of obtaining residuals. These are essentially a rearrangement of the state-space input-output model which result in a parity vector whose values are ideally zero when no fault is present. The parity vector is then monitored for change which would indicate the presence of a fault. It is generated in such a way as to make it dependent only on faults introduced to the system and not on the system's state itself [12].

Höfling and Isermann [13] used parity equations to form a fault detection system which tracked the parameters of the monitored system. Parity equations can be used for parameter estimation. By tracking changes in parameters, faults can be detected and diagnosed.

Gertler and Monajemy [14] used dynamic parity relations to generate directional residuals, meaning that the introduction of a particular fault moves the residuals in a straight line through the parity space (the vector space made up of the residual signals). A desirable property of some such systems is that the movement of the residuals is orthogonal for each different fault, and therefore diagnosis is very simple. This method is similar in outcome to the use of observers or extended Kalman Filters tuned to each fault [15], but Gertler and Monajemy claim that their method makes the design process simpler.

Ballé and Füssel [16] proposed a solution to the problem of modelling non-linear systems by viewing them as piecewise linear and constructing a "LOcal LInear MOdel Tree" (LOLIMOT). This is a structure of linear models which are each valid for a certain part of the system's operation. This method was tested on a control valve actuator of the type used in chemical processes.

Another non-linear modelling solution was proposed by Demetriou and Polycarpou [17]. This used online approximators to model the performance of non-linear systems from a precise mathematical model. When the output of an online approximator is non-zero then a fault is detected. This system can also diagnose the characteristics of faults and can be designed to be robust to modelling errors.

Qualitative model-based methods

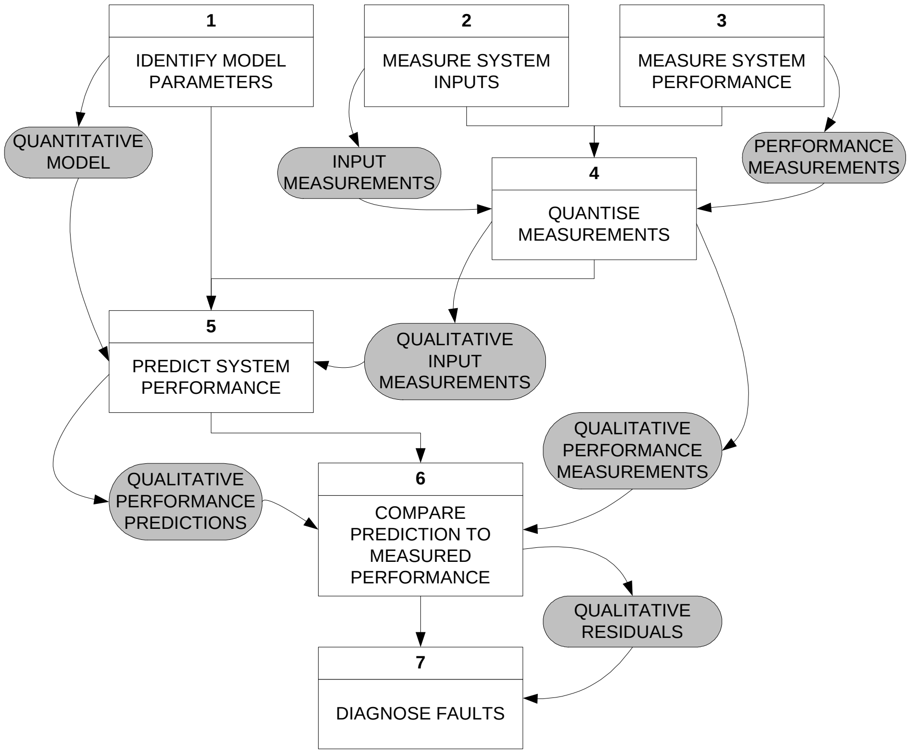
1
IDENTIFY MODEL PARAMETERS
2
MEASURE SYSTEM INPUTS
3
MEASURE SYSTEM PERFORMANCE
QUANTITATIVE MODEL
INPUT MEASUREMENTS
PERFORMANCE MEASUREMENTS
4
QUANTISE MEASUREMENTS
5
PREDICT SYSTEM PERFORMANCE
QUALITATIVE INPUT MEASUREMENTS
QUALITATIVE PERFORMANCE MEASUREMENTS
QUALITATIVE PERFORMANCE PREDICTIONS
6
COMPARE PREDICTION TO MEASURED PERFORMANCE
QUALITATIVE RESIDUALS
7
DIAGNOSE FAULTS

Figure 3.3: General structure of a qualitative model-based method for fault diagnosis

There are three main reasons cited in published literature for pursuing the use of a qualitative model. The first is that modelling inaccuracies are not a problem because qualitative methods do not rely on exact numerical modelling. The second is that qualitative methods can be used where it is not possible to make quantitative observations. The third is that qualitative models make it possible to incorporate empirical knowledge about a system in the operational model [18]. Figure 3.3 shows the functional flow of a generic qualitative model-based diagnosis system: it will be seen that there are some similarities with the structure of a quantitative model-based system.

A number of papers concerned with qualitative modelling and its application in fault diagnosis have been published by Professor Jan Lunze of the Ruhr-Universität Bochum, Germany, and various associates. There has been particular focus on the use of non-deterministic (or stochastic) automata as a modelling medium for physical processes [19]. Automata are also known as state machines. Stochastic automata were applied to the observation of qualitative states [20] and the diagnosis of transient faults [21]. A study was also carried out into the conditions which must exist for a deterministic automaton model to be valid [22], and the use of a semi-Markov process model, as a timed description of quantised event sequences, was evaluated [23]. State machines are also used as a qualitative model by Ramkumar *et al* [24].

These papers all deal with situations where variables cannot be directly measured and so a qualitative model must be used to process the qualitative variables which can be measured. The methods involve a partition or quantisation of the measurement space so that a discrete-event representation can be constructed.

Various probabilistic methods then model the transition between states and produce a diagnosis output. A probability level is calculated for each possible fault, which is a useful output because it allows a human observer to see the increasing probability that a certain fault is present over a period of time, and also can show how one fault initially appears more likely, but then diminishes as time goes on.

These methods are effective but there is always a compromise when quantised data is used as an input to a model. The probabilistic diagnosis algorithms are in place to deal with the ambiguity introduced by the use of these data. Quantisation also means that small changes in measured variables are not always detected straight away. This means that incipient faults may not be detected quickly enough.

Digraphs, also known as cause-effect graphs or signed directed graphs (SDG), were proposed by Iri *et al* [25] as a means of qualitatively simulating the performance of a physical system. The differential equations which mathematically represent a system were transformed to qualitative graphs by considering only the sign of each value (positive, negative or zero).

Digraphs can be used to represent a qualitative causal model, as part of a system to predict faults in a gas turbine [18]. Users of this system were able to view raw data, model behaviour using standard quantitative equations, and add further qualitative information from knowledge by editing the resulting digraphs. Tarifa and Scenna [26] applied digraphs to chemical processes, using a propane evaporator as an example. A fuzzy logic rule base was then used to diagnose faults according to a knowledge base established during the simulation of faults. Fuzzy logic is "concerned with the formal principles of approximate reasoning, with precise reasoning viewed as a limiting case" [27]. In fault diagnosis, this allows us to determine the *degree of presence* of a fault condition: that is, the level of confidence we have that the condition is currently present.

Qualitative physics is intended as a means for artificial intelligence systems to make reasoned decisions about the real world, based not only on standard mathematical behaviour models but also human-style deductions. A basic qualitative physics was proposed by de Kleer and Brown for perturbation analysis [28]. The limitations of this approach then led to the development by de Kleer and Brown of a more complete physics based on confluences [29] and also to the development of qualitative process theory by Forbus [28].

Fault trees are knowledge-based models which relate symptoms to particular fault conditions. They are used to determine the likely behaviour of equipment under failure modes. Heuristic knowledge and human reasoning are used to link observable

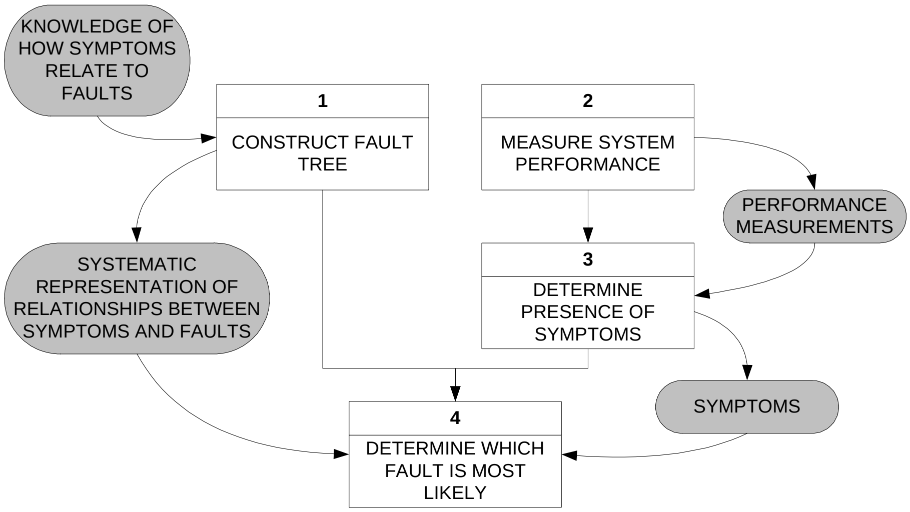
KNOWLEDGE OF HOW SYMPTOMS RELATE TO FAULTS
1
CONSTRUCT FAULT TREE
2
MEASURE SYSTEM PERFORMANCE
PERFORMANCE MEASUREMENTS
SYSTEMATIC REPRESENTATION OF RELATIONSHIPS BETWEEN SYMPTOMS AND FAULTS
3
DETERMINE PRESENCE OF SYMPTOMS
SYMPTOMS
4
DETERMINE WHICH FAULT IS MOST LIKELY

Figure 3.4: Fault tree diagnosis

symptoms to fault conditions. Providing there are enough symptoms to observe and that each fault has a unique set of symptoms, it is possible to diagnose faults accurately. Fault trees can be derived from digraph models [30] to form a diagnosis system for chemical processes. This approach was designed to include operator actions and disturbances from outside the plant. Figure 3.4 shows the functional flow of a generic fault tree diagnosis system.

Abstraction hierarchies are schemes for fault diagnosis where the end result is an isolation of the component or function which is faulty [31]. The decomposition of a system, according to the functions performed within it, has been proposed [32]. This allows the system state to be deduced qualitatively, leading to the isolation of the faulty component or subsystem.

Hierarchical schemes are suitable for large monitored plants such as the chemical processes which were being considered in these papers, however this thesis is concerned with the diagnosis of faults in a single actuator, and so the scale of these methods is not appropriate.

Process history-based methods

History-based methods are fundamentally different from model-based diagnosis methods because no prior knowledge about the monitored system is required. Previously measured data from the system is analysed to produce some knowledge from it which can be used to diagnose faults. This function is known as *feature extraction* [33]. Methods for the *application* of the knowledge are much the same as those for model-based methods.

There are several categories of feature extraction methods. These are shown in the process history branch of figure 3.1. Quantitative feature extraction can be statistical or non-statistical. Statistical methods are concerned with finding patterns in abstract mathematical representations of data.

Principal Component Analysis (PCA) is one of the most recent of these representations and has been applied to face recognition [34, 35] and non-destructive testing of metal objects for sub-surface cracks [36]. PCA is a mathematical transformation of vector or matrix data which reduces the dimensionality of the data.

Wavelet transforms are mathematical representations of signals with components in both the frequency and time domains. They have been used for feature extraction in many different fields including fault prediction for ball bearings [37], classification of geographical data [38, 39], classification of NMR[2] spectra [40], wear estimation for industrial turning processes [41] and vibration monitoring [42].

Wavelet transforms are a function of a signal, two scaling variables a and b, and a "mother wavelet" which is a mathematical function with zero mean and finite length. The scaling variables allow the wavelet transform to be carried out at various scales, which can produce a multi-resolution analysis of the signal.

Frequency-domain analysis has been used for some years to extract the features

[2] Nuclear Magnetic Resonance

of signals for diagnosis. Fourier transforms in particular can be used to track components of certain frequencies.

All these statistical methods rely on complex mathematical functions and produce fairly abstract data which requires further interpretation before it can be of use. The complexity of the functions limits their use on this thesis, where it has been specified that the methods used must be transparent and understandable to the technical staff who would be expected to use the system.

Some initial work was carried out into the possible use of wavelet transforms as a means of representing the waveforms from different actuators in some common domain, but it was found that the performance of each actuator produced results which could not be reliably linked in any qualitative way.

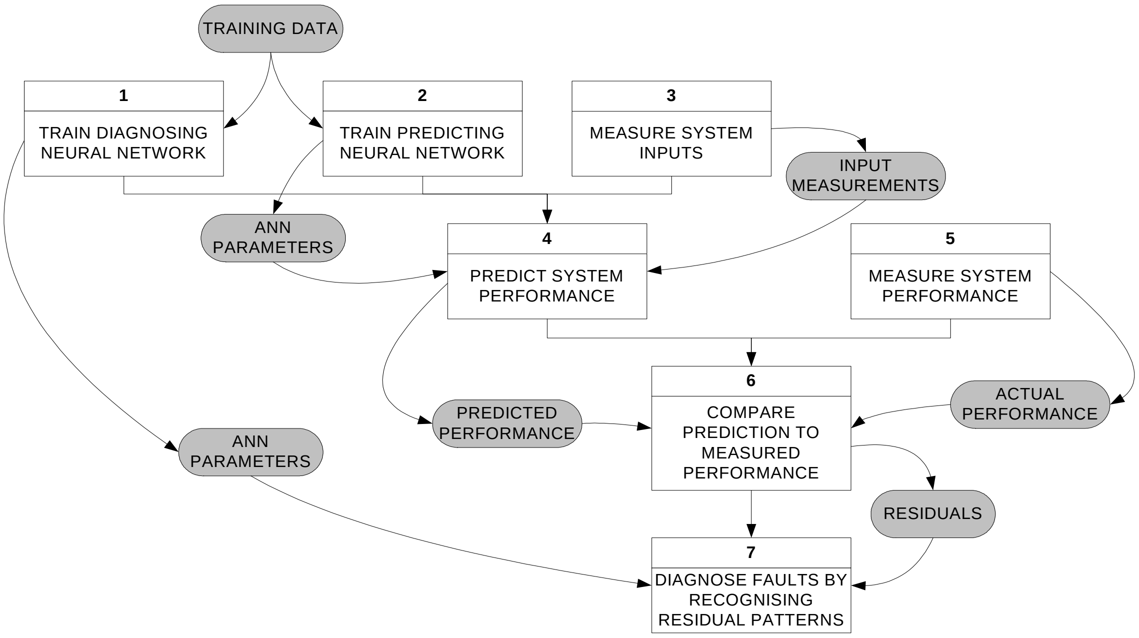
TRAINING DATA
1
TRAIN DIAGNOSING NEURAL NETWORK
2
TRAIN PREDICTING NEURAL NETWORK
3
MEASURE SYSTEM INPUTS
INPUT MEASUREMENTS
ANN PARAMETERS
4
PREDICT SYSTEM PERFORMANCE
5
MEASURE SYSTEM PERFORMANCE
6
COMPARE PREDICTION TO MEASURED PERFORMANCE
ACTUAL PERFORMANCE
PREDICTED PERFORMANCE
ANN PARAMETERS
RESIDUALS
7
DIAGNOSE FAULTS BY RECOGNISING RESIDUAL PATTERNS

Figure 3.5: An example of a diagnosis system using ANNs to model the monitored equipment, and to recognise faults through patterns in the residuals

Artificial neural networks (ANNs) are very popular computing methods which have applications throughout engineering. Their basic operation is modelled on that of neurons in the human body, which have input and output components and a vast amount of interconnection [43]. Figure 3.5 shows the functional flow of a neural network-based diagnosis system, where ANNs are used both for the diagnosis of fault from symptoms, and for the prediction of the outputs of the monitored asset.

They are capable of learning from input data and performing functions such as modelling of data profiles (which can be used to replace conventional mathematical models) and the classification of patterns (which can be the diagnosis of faults from an initial set of observed residuals generated by a model of any kind) [43].

Because of their versatility, ANNs of many different types have been employed in fault diagnosis research in recent times. A set of training data is all that is needed to program them to model systems and diagnose faults with superior accuracy to most other methods. Some examples of previous applications are catalytic cracking processes [44], power transformers [45] and robotic arm manipulators [46].

ANNs have the advantage, from the view of this thesis, of requiring no detailed analysis of the monitored systems, but rather a set of data which can be used to "teach" the network (training data). Training data can sometimes be easier or less costly to obtain than an accurate and detailed mathematical and physical analysis of an asset. However, the very nature of ANNs is "black-box", in that it is impossible to observe how a neural network arrives at a particular decision. It is very important in this thesis that the decisions can be traced right back to their origins so that technician users can interact fully with the system and draw their own conclusions from its output. Therefore, ANNs are not suitable for this thesis.

Qualitative process history-based methods attempt to solve the problem in non-numerical ways. Two diverse types of qualitative method based on process history are Qualitative Trend Analysis (QTA) and expert systems.

Qualitative Trend Analysis describes the process of qualitatively extracting relevant information from observations and using it to draw conclusions about the state of a monitored system. Cheung, Bakshi and Stephanopoulos [47, 48, 49, 50] developed a comprehensive scheme for the representation of trends in sensor data. The basic

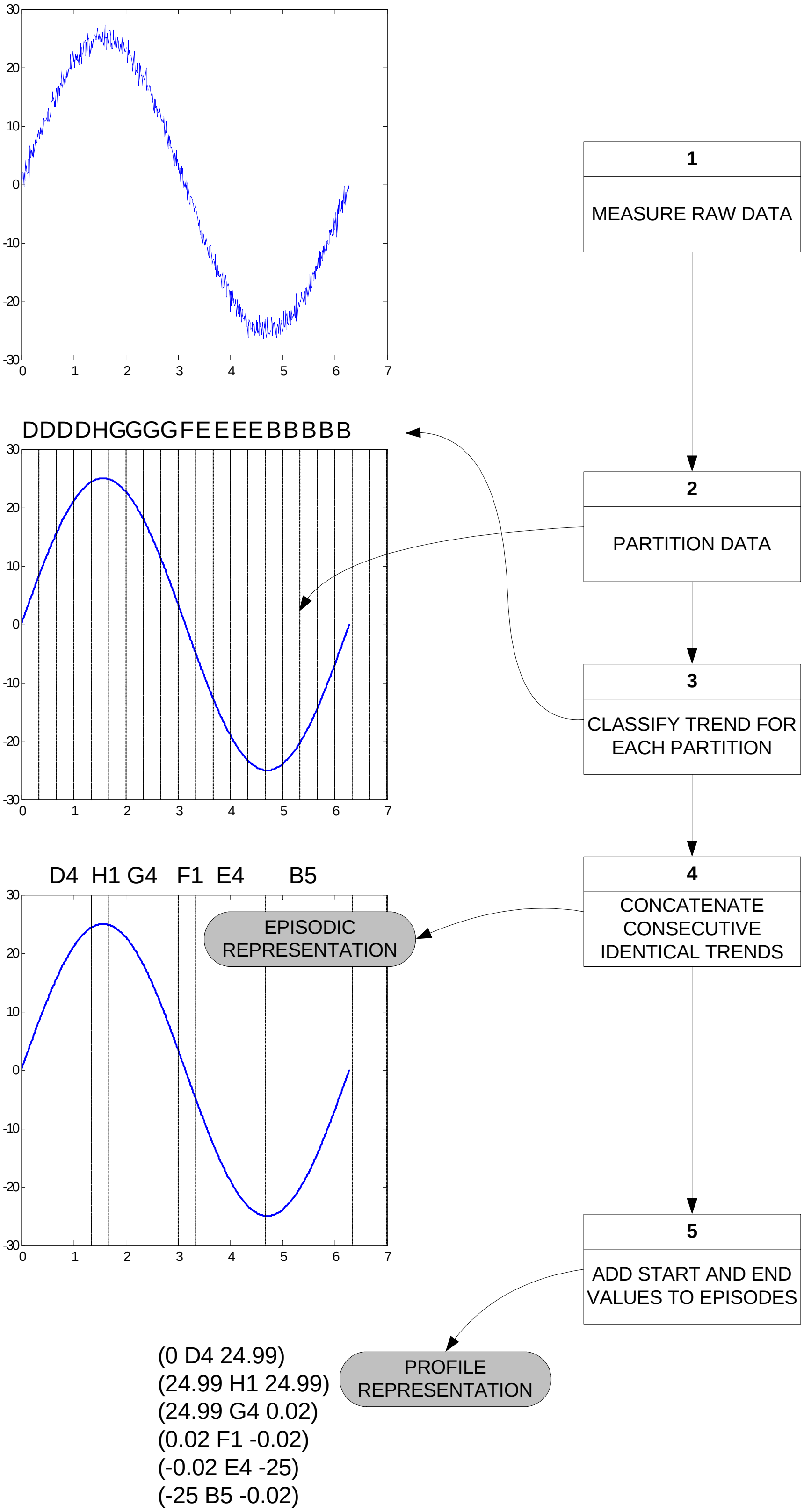

Figure 3.6: Qualitative Trend Analysis of measured data

method was to filter the waveform to the scale required for analysis, which removed some of the higher frequency information, and to then represent the waveform as a series of standardised curves.

This enabled sensor data to be partitioned into a sequence of *episodes*, where each episode is a trend of a particular shape. The episodes are represented by an “alphabet” of curve shapes. The problem with the scheme as presented in the sequence of papers [47, 48, 49, 50] was that the representation of the data constituted a classification problem, for which the authors had no ready solution but referred to the use of neural networks and other complex methods of classification.

This appeared to make the process very complex, but the idea of using an alphabet of basic shapes to represent waveform data is one which appeals greatly. Some quantitative information can be stored along with the sequence of shape characters. The starting and ending values in each episode are one example of this. Therefore, these representations contain both qualitative and quantitative information, making the approach ideal for this thesis. Shortly after the start of this thesis, the author proposed that if data were collected from many actuators of the same type and in the same condition, it would be possible to represent the data in such a way that the qualitative performance would be identical in all cases [51]. This can be achieved by using QTA.

Further work on QTA suggested a partitioning approach where the measured data were partitioned into intervals of the same length, and the qualitative state of each partition was determined separately [52]. Figure 3.6 illustrates the basic stages of this process.

A variation on QTA was proposed, where fuzzy rules were established for particular episodes of behaviour identified in measured data [53]. This approach was shown to be very successful in detecting and diagnosing faults in chemical processes. Given that it is possible to find a set of episodes in a qualitative representation which are common to all fault conditions for a given actuator, fuzzy rules established for particular episodes could be used to great effect on this thesis. The fuzzy sets developed in this paper were of uniform size, translating to heuristic categories such as “average”, “large”, or “very small”.

It is proposed that a useful development of this would be to establish episodic fuzzy rules based on the performance of an actuator under fault conditions. The faulty performance would be measured at the point where the fault is on the verge of preventing the actuator from moving. The fuzzy sets established would be constructed so as to provide a gradual increase in membership between the point of fault free operation and the point where the fault performance is measured. Then, as operational performance moves away from fault free and towards the fault, the membership increases, showing increased confidence that the fault is present.

Expert systems are methodical representations of human knowledge which diagnose faults by reasoning with IF-THEN rules. This can be done manually or using one of several expert system building tools. Their chief advantages are the ease with which they can be developed and the transparency of their reasoning [33]. Their chief drawbacks are that they are very specific to particular systems and are difficult to update [33].

The power distribution industry has focussed heavily on the use of expert systems for the diagnosis of faults in components such as transformers and insulators [54]. Dissolved gas analysis from transformer oil has been used to establish a set of diagnosis rules for the detection of faults in transformers [55]. This was found to be very successful, with detection rates for the tested faults in excess of 90%.

Butler [54] combined an expert system engine, built using the EXSYS automated tool, with a neural network model for the ageing of power components to form a diagnosis system for large networks of power distribution equipment. This was then used to find the location, within the network, of equipment which was behaving abnormally due to aging. Neural networks and expert systems were also combined for the diagnosis of steady-state faults in chemical processes [56], and for the diagnosis of power transformers [57].

3.1.4 The DAMADICS benchmark actuator project

Background

DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems) was a four-year European Commission-funded project to establish a benchmark for the assessment of diverse fault detection and isolation (FDI) methods [58].

The project was presented in a special edition of Control Engineering Practice. There was an introduction paper [58], several papers presenting methods for FDI based on the benchmark actuator, and several further papers on general fault diagnosis and soft computing. The benchmark FDI methods will be reviewed in this section.

The benchmark is an electro-pneumatic valve actuator used in the sugar production process at a factory in Poland. These actuators are used throughout industry in many different types of process. They are spring-and-diaphragm pneumatic control valves which regulate the flow of liquids through the factory. A diagram of the actuator is shown in figure 3.7. These actuators are mechanically and operationally quite different from the single-throw mechanical equipment used on railway actuators. One key difference is that the DAMADICS actuator, being a flow regulator, can have many different positions in normal operation, between its two extremes, whereas STME moves only from one extreme position to the other in normal operation. A similarity, however, is that the dynamics of the important variables are, like STME, of a similar scale in time to the time taken for movement.

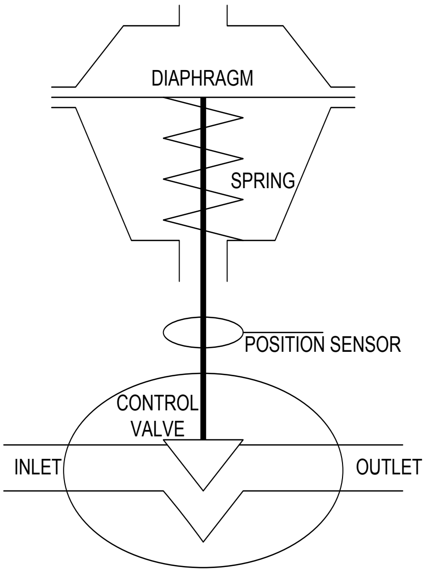

Figure 3.7: The pneumatic control valve actuator used as a benchmark in the DAMADICS project [58]

Qualitative reasoning and neural network diagnosis

Calado *et al* [59] proposed a qualitative reasoning approach with fuzzy neural network diagnosis. The inputs to the FDI system were the rates of change of the measured variables, normalised over the range $[-1, 1]$. The authors cited the inaccuracy of numerical models as the main reason for using a qualitative model. The model in this method simulates the process and predicts the rates of change of the variables, in the same format as the inputs. The real measured variables are converted to the same qualitative format and compared with the model's predictions in a discrepancy generator. This detects whether or not a fault is present.

The discrepancies are then fed to a hierarchical fuzzy neural network which diagnoses the fault. A fuzzy neural network is one where the neurons each have a particular fuzzy membership function, and the output of the neuron is the value of the membership function at the input value. The use of the qualitative model appears to have limited the usefulness of this method, because it was not possible for the system to detect all the faults. A major reason for this was that the small effects of incipient faults were, in some cases, indistinguishable from the noise in the variables.

This method was interesting because the qualitative processing of the input and measured variables meant that it might be possible to process data from several different actuators in such a way that the same rules could be used for all instances. This is a major requirement of the thesis. However, the qualitative model appears not to have enough functionality. It is fair to conclude that a purely qualitative approach to diagnosis is not sufficient to detect the small effects of incipient faults, which are the main target of this thesis.

State machines

Timed automata (state machines) were used in the method devised by Supavatanakul *et al* [60]. The actuator was modelled as a discrete event system, and the occurrence of faults was considered as a discrete phenomenon. The model was able to distinguish between fast and slow state transitions, so it could detect faults which affected the time taken for the actuator to perform its function. The discrete-event model was proposed as an alternative to standard quantitative models which was more robust to imprecision and uncertainty, and less complex.

The first function of this system was concerned with identifying the flow of states from measurements taken during simulations. A separate automaton was constructed for each fault. The input and output model values were then obtained. Sequences of extended states are determined, and then the possible active states were found. The automata which did not fit the possible state sequences were excluded from the candidate set. The remaining automata represented the faults which were possible.

The method was tested against the DAMADICS benchmark by applying it to data measured over about 100 seconds, where a particular fault occurred. The method was able to distinguish the fault which had occurred from three other actuator conditions: two faults and the fault free condition. However, it was not made clear whether the method was capable of distinguishing the other faults in the DAMADICS data set.

Interval observer model

A novel model-based method was used by Puig *et al* [61]. It aimed to tackle model uncertainties, parameter differences and disturbances by making the diagnosis stage robust to such unwanted inputs. The model consists of observers which predict the outputs of the plant to be within certain intervals. These intervals allow uncertainty to “propagate into the residuals”, so that the input to the fault detection function takes account of the uncertainty in the system. As with all model-based methods, there is the disadvantage of high mathematical complexity.

The modelling of uncertainty as an interval means that faults with effects smaller than the interval are likely to be dismissed as uncertainty. This is, to a certain extent, mitigated by the use of non-linear observers, which allow the use of less conservative intervals than would be possible with linear observers. This system was aimed at fault detection; the paper mentions that fault isolation (diagnosis) is a subject for further work. Therefore this method has only limited relevance to this thesis.

Neural networks for both modelling and diagnosis

Witczak *et al* [62] used a new type of neural network to both model the system under observation (by performing system identification) and diagnose faults. This neural network is the Group Method of Data Handling network (GMDH). The bounded-error approach was used in the design of the neural network to try and minimise modelling uncertainty. The fault diagnosis network monitored the residuals generated from comparison of the model to the real system, and maintained an adaptive threshold around the residuals to increase the sensitivity to faults. However, the authors conceded that some faults could not be detected and diagnosed. Significantly this included incipient faults. It seems that incipient faults are often lost when modelling uncertainty is taken into account, especially if a qualitative model is used.

Neural networks take a large amount of complex calculations to establish. This was addressed to some extent by ensuring that the most time-consuming calculations were carried out off line before monitoring started [62], however with a large set of identical actuators, some system identification would need to take place every time the system is installed on a new instance of the actuator. Programming an entire neural network with the individual behaviour of each actuator would be prohibitive.

Frequency analysis

Model-free fault detection and diagnosis methods are relatively rare, but Previdi and Parisini [63] proposed such a method based on a spectral estimation tool called the Squared Coherency Function (SCF). This function produces a frequency-based transform of the waveform. An envelope is established around the SCF of the fault-free waveform. Different envelopes are established for each type of fault.

Thus, the fault can be detected and part diagnosed by checking which envelopes are violated by a faulty waveform. This method is not suitable for use in the current thesis because it is only capable of picking up abrupt faults. Incipient faults would not be detected until they are severe enough to breach the threshold.

Fuzzy classifiers

Bocaniala and Sá da Costa [64] also proposed a model-free method which uses fuzzy classifiers to distinguish between faults by identifying regions within a multidimensional space (where each dimension corresponds to one measured variable) where there is correlation between variables measured during a particular fault simulation. Thus, if operational measurements fall within a particular region, a fuzzy membership function for that region becomes high in value, denoting that the actuator is in that particular state.

This method is very relevant to the current thesis because it provides a direct link between the symptoms of a fault and the diagnosis. It was also designed with a view to diagnosing faults at different levels of severity, although it does not take account of the slow development of incipient faults. The methods used to tune the fuzzy classifier are computationally expensive and would not lend themselves to use on a large number of actuators.

Conclusions from the DAMADICS project

The DAMADICS project demonstrates the application of several very diverse methods to a common benchmark actuator. This makes it possible to judge the comparative success of each method in a fair way.

The results from the application of a purely qualitative method [59] show that, although qualitative analysis can be useful in dealing with variations, it is not sufficient on its own to accurately detect the onset of incipient faults. In fact, most of the faults in the benchmark set for DAMADICS were abrupt faults. Despite this, it is clear that many of the methods were not suitable, for one reason or another, for this thesis. The main disadvantages were that they would either be impractical to adapt so that rules would apply to more than one actuator of a certain type, or that they were incapable of detecting incipient faults.

3.1.5 Fault diagnosis on the railways

Traditionally, the diagnosis of faults in railway equipment has been carried out, once a failure has occurred, by technicians called to the equipment's location. According to failure statistics from the Southern region of the British network [65], between 17 and 22% of failures on point actuators could not be traced to a particular fault when the technicians examined them, resulting in a record stating "Tested OK on arrival", despite the fact that the actuator had failed in service and therefore a fault was clearly present.

Clearly a human examination is not sufficient to detect all the possible faults in these actuators. More insight is required into the operation of the actuator, and so automated condition monitoring has emerged as a potential solution. Commercial solutions focus, in the main, on diagnosing abrupt faults using large arrays of sensors to detect faults in the positioning and locking mechanisms of the actuator being monitored [2]. There are currently several pilot schemes operating around the UK.

Little academic research has been carried out to advance this technology, however a number of papers have been published which aim to gain a better insight into the dynamics of an actuator and thereby to try and detect incipient faults which would not be seen on standard condition monitoring equipment until a failure was absolutely imminent.

A neuro-fuzzy diagnosis approach has been proposed for the diagnosis of pneumatic point machines, where the variables measured were first partitioned into a small number of regions [66]. This allowed a piecewise linear input-output model to be used. Partitioning the operational variables in the same way, it was possible to generate residuals which were sensitive to faults. Local nodes on each actuator were capable of distinguishing sensor faults from actuator faults. Measured data from actuator faults was forwarded over a network to a central processor which ran an Adaptive Neuro-Fuzzy Inference System to diagnose the fault.

This system is interesting because it uses a simplified model in order to reduce complexity. However, it does not fully address the issue of tuning the rule base so that the rules apply equally to each instance of an actuator type. In practice, this system would have required the faults to be simulated on each actuator in order to correctly tune the model, and that is not practical.

Neural networks have been used to model the performance of pneumatic train doors and to diagnose faults [67]. An RBF neural network modelled the displacement profile of the door to a level of accuracy which could be predefined. Complexity of the network was reduced by filtering and removing corrupt values from the data set. Diagnosis was carried out with a Self-Organising Feature Map (SOFM) neural network. The results published in this paper indicated that the accuracy of the system was roughly 80%.

This system used the latest neural network technology but it is unclear how easy it would be to implement it on a practical scale.

3.2 Conclusions from literature

Both qualitative and quantitative fault diagnosis methods, be they model-based or not, have valuable properties. However, the problem in this thesis is to diagnose a large set of actuators of the same type, where the qualitative behaviour of each actuator is similar, but the quantitative behaviour is not. A new approach is required because neither a purely qualitative nor a purely quantitative approach is likely to succeed.

There are many problems with the use of quantitative models. The most obvious is that real world equipment can never be modelled with 100% accuracy. Model inaccuracies and uncertainties mean that residuals are unlikely to be solely dependent on faults, and this can lead to spurious detection of faults.

In previous years, the complexity of mathematical modelling has limited its use due to the physical limitations of computer equipment in carrying out the necessary calculations. Whilst modern computing power all but negates this disadvantage, the complexity of such methods makes them difficult for the layman to understand. A key requirement of this thesis is that the method used must be understandable to technical staff. This is so that they can engage more with the process and thereby gain more from its outputs.

Qualitative modelling is of particular interest in this thesis because it has the potential to solve the problem of quantitative variation between instances of similar actuators. It is fair to assume that all instances of a particular actuator will share similar characteristics of performance when viewed qualitatively.

However, the results of the research carried out into qualitative modelling suggest that the ambiguity introduced may impair the ability of a diagnosis system to detect incipient faults at an early stage of detection. In many cases, qualitative models are no simpler or easier to understand than quantitative ones.

Qualitative methods alone cannot detect incipient faults with an acceptable level of accuracy, but quantitative methods will be difficult to implement across a set of actuators, each with slightly different characteristics. A method is required which uses a hybrid analysis to transform real-world measurements into a form where a single set of rules can be applied to data from multiple actuators.

The use of process history to establish rules for diagnosis is appealing for this thesis, because the reasoning behind it is totally transparent to a technical user. Data from simulations of faults at maximum severity could be used to automatically establish rules which apply to episodes in the measurements from monitored equipment. These episodes could be established using qualitative trend analysis, which examines measured waveforms in a similar way to humans, looking for shapes and changes in those shapes. A combination of some of the methods discussed in this section, modified to the application in hand, seems like the best solution for fulfilling all the requirements of this thesis.

A model-based approach does not tackle the problem in a similar way to a human, whereas several process-history based methods are very intuitive. It is proposed that the best solution to the design problem in this thesis will be to use a history-based method with a hybrid qualitative-quantitative analysis of the prior data. This will be used to establish a set of rules which apply equally to the performance of any actuator of the correct type, providing that data measured from that actuator is also represented with the same hybrid analysis.

The resulting system will therefore analyse the shape and trends in measured waveforms in a similar way to a human, and decide, based on knowledge of the shapes and trends in faulty waveforms measured in the past, what faults appear to be present in the monitored data.

3.3 Method evaluation

Several methods described in the preceding section were evaluated in detail by representing them functionally and using this understanding to determine whether they would fulfil the requirements of the thesis. The result of these evaluations was compared against that of the author's own solution, which will be described in chapter 4.

The methods were chosen as representatives of different approaches to fault diagnosis. They were evaluated against the requirements by this simple scoring system:

- One point awarded for each completely fulfilled requirement
- Half a point for each partially fulfilled requirement
- No points for an unfulfilled requirement

Since the requirements are somewhat subjective, this scoring system is subjective by extension, and is therefore only indicative.

The following methods were chosen for this evaluation:

(a) The method developed in this thesis, described in section 4

(b) Qualitative model-based distributed FDD [66]

(c) Nonlinear model-based fault diagnosis with automatic rule generation [16]

(d) FDD using a fuzzy qualitative model [59]

(e) FDD using timed automaton models [60]

Certain other methods were initially examined for suitability but obvious deficiencies were found early on and a full evaluation was not necessary. One example of this was a time-fuzzy Petri-net scheme [68], which failed to describe adequately the diagnosis process.

3.3.1 Modelling of the system

The diagnosis problem was initially modelled as a single black box which produces the required outputs given certain inputs. These inputs impose requirements on the system, and their nature is determined by the method chosen. The black box model is shown in figure 3.8.

3.3.2 Evaluation procedure

For the high level currently being discussed, it was considered adequate to perform a simple set of checks to determine whether a proposed method was suitable for implementation. The steps taken in evaluation were as follows:

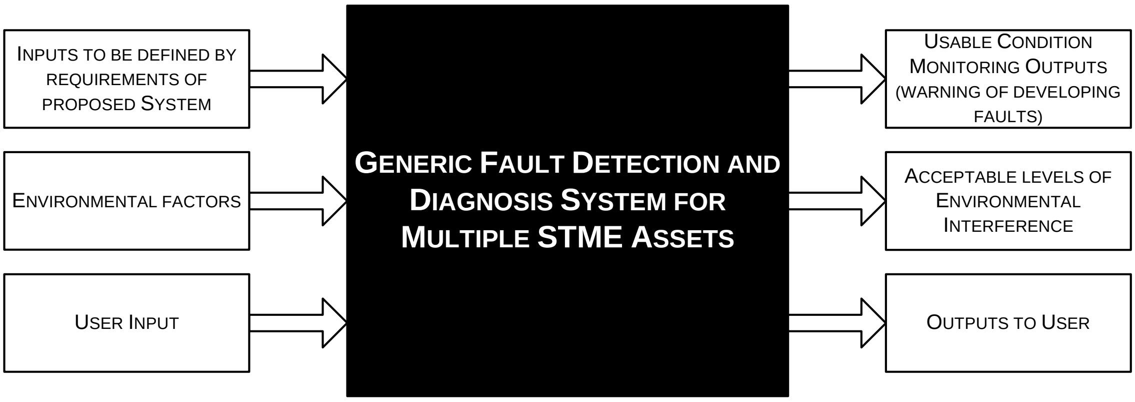
INPUTS TO BE DEFINED BY REQUIREMENTS OF PROPOSED SYSTEM
ENVIRONMENTAL FACTORS
USER INPUT
GENERIC FAULT DETECTION AND DIAGNOSIS SYSTEM FOR MULTIPLE STME ASSETS
USABLE CONDITION MONITORING OUTPUTS (WARNING OF DEVELOPING FAULTS)
ACCEPTABLE LEVELS OF ENVIRONMENTAL INTERFERENCE
OUTPUTS TO USER

Figure 3.8: Black-box representation of the system

- Gain a basic understanding of the method and express it as a functional flow diagram
- Determine to what extent the proposed method fulfils the system requirements
- Determine what requirements the proposed method imposes on the system
- Evaluate the balance between the proposed method's fulfilment and imposition of requirements

3.3.3 Semi-qualitative distributed fault detection and diagnosis using a piecewise linear model

Description

The proposed method [66] takes account of the quantitative differences between actuators of the same type by using a mix of qualitative and quantitative information to predict the actuator's performance. The measured waveforms are considered to be piecewise linear and can thus be partitioned and approximated to linear functions (function 1 in figure 3.9). The partitioning instants are pre-determined.

The qualitative and quantitative data from this approximation is then used to model the actuator's behaviour (function 4). The modelled output is compared with the measured output during operation, to generate residuals (function 5). These are then used to detect and diagnose faults (functions 6 and 7).

Function 6 considers two types of residual, one of which is unaffected by sensor faults. This allows it to distinguish sensor faults from actuator faults. Residuals which indicate actuator faults are sent over a network connection to a central processing point where a detailed diagnosis is carried out using an Adaptive Neuro-Fuzzy Inference System (ANFIS).

Figure 3.9 shows the functional flow for the proposed method.

Evaluation

This method is capable of detecting and diagnosing incipient faults, although indication has not been given as to how a fault trend is detected over a period of time. It should be suitable for any type of STME, and the functions are fairly easy to understand because the approach uses a qualitative analysis rather than complicated mathematics. The mix of qualitative and quantitative modelling overcomes the differences between actuators of the same type, making it suitable for use on large asset base.

However, this system imposes some important requirements itself, in terms of the data needed to make it function. The partitioning times need to be determined before it can operate - this is not included in the method and would need to be added, either with automatic analysis or human examination. A rule base is also required for fault diagnosis - this would have to be manually extracted from a FMEA or built automatically. In terms of measurement data, this system requires measurements during run-in and operation of each asset, which is perfectly achievable.

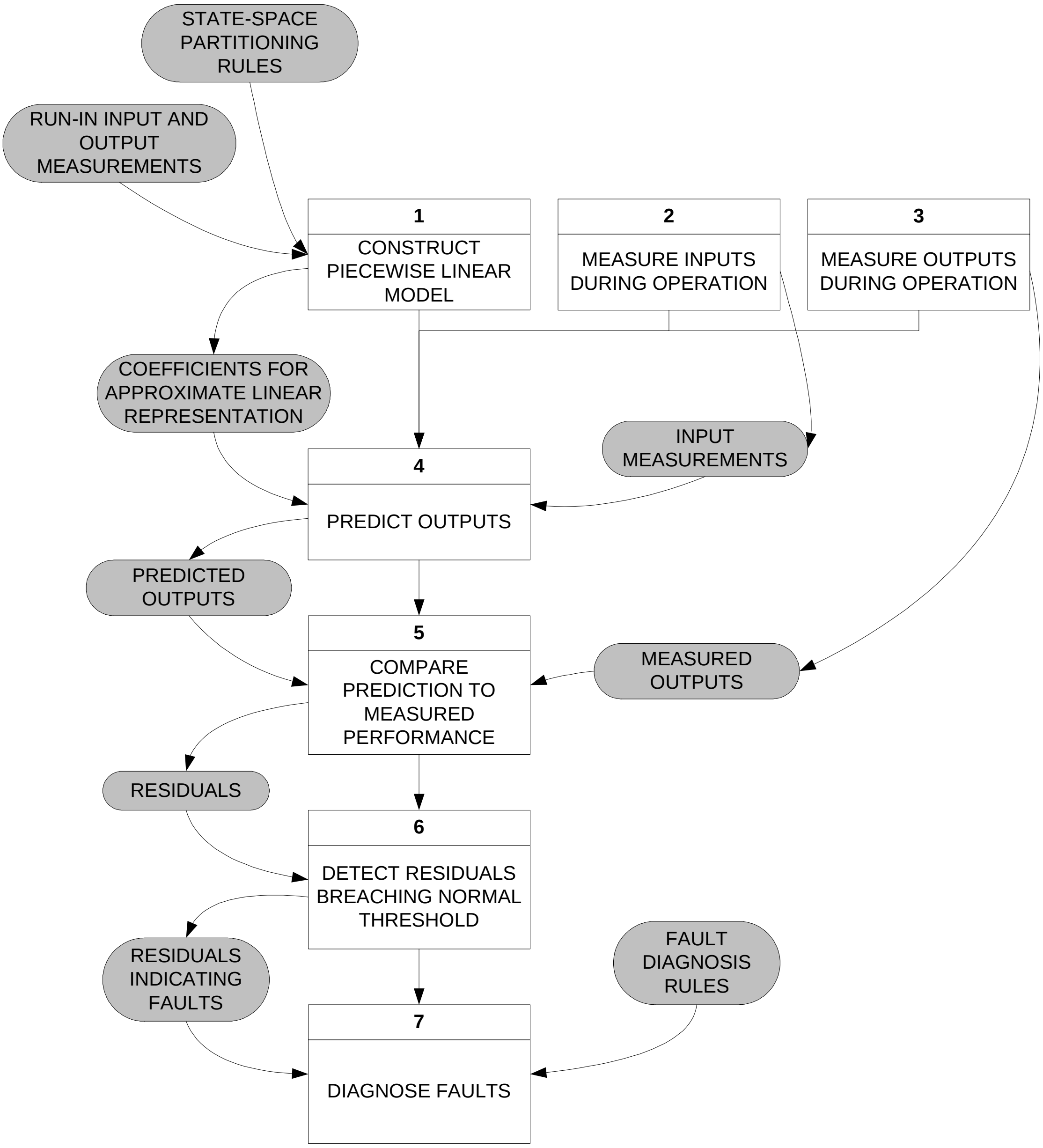
STATE-SPACE PARTITIONING RULES
RUN-IN INPUT AND OUTPUT MEASUREMENTS
1
CONSTRUCT PIECEWISE LINEAR MODEL
2
MEASURE INPUTS DURING OPERATION
3
MEASURE OUTPUTS DURING OPERATION
COEFFICIENTS FOR APPROXIMATE LINEAR REPRESENTATION
INPUT MEASUREMENTS
4
PREDICT OUTPUTS
PREDICTED OUTPUTS
5
COMPARE PREDICTION TO MEASURED PERFORMANCE
MEASURED OUTPUTS
RESIDUALS
6
DETECT RESIDUALS BREACHING NORMAL THRESHOLD
RESIDUALS INDICATING FAULTS
FAULT DIAGNOSIS RULES
7
DIAGNOSE FAULTS

Figure 3.9: Fault detection and diagnosis using a piecewise linear model

Conclusion

This proposed method is the only published scheme found so far by the author which takes account of the differences in performance between instances of actuators of the same type. The methods used within each function are generally easy to understand and fulfil the requirements. However, the proposed method requires considerable analysis to be carried out beforehand, in order to establish rule bases for fault diagnosis and for the partitioning of measured signals.

These operations would need to be carried out for each type of STME to be monitored. This makes the proposed method slightly less portable than one that establishes its own rule bases. This does not fulfil the requirements of this thesis, because it is implied that these tasks should be automated.

The general principles and architecture of this proposed method are the basis for the author's own proposal, because the general approach is the right one to overcome the differences between actuators of the same type.

3.3.4 Nonlinear model-based fault diagnosis with automatic rule generation

Proposed method

The proposed method diagnoses faults using a nonlinear model and automatically-generated fuzzy rules. It is proposed by Ballé and Füssel [16] and is based around a nonlinear plant, namely an electro-pneumatic valve actuator similar to that used for the DAMADICS benchmarking problem [58]. The functional flow of the proposed method is shown in figure 3.10.

Rules for fault diagnosis are automatically generated using a Self-Learning Classification Tree (SELECT) method (function 1 in figure 3.10), from data collected during an initialisation phase i.e. before introduction to service. This means that the system would not require any prior knowledge of fault performance and could function using measured data only. The SELECT method was presented in an earlier paper [69], the text of which is not readily available, although a summary was found [16].

The problem of varying performances between different actuators is overcome by a *system identification* function which analyses measured data from a run-in phase to determine the parameters of the model (function 2). This function provides sets of parameters to a linear-parameter-varying (LPV) model (function 3). LPV models have several sets of parameters, each of which is locally valid for a subregion of the total input space. This means, for example, that the model parameters change for different time regions.

The model uses the measurements of inputs to the plant, and the previous values of measured output and modelled output. This requires that the outputs of both the model and the plant are fed through a time delay (function 6). The length of time over which the previous outputs are considered depends on the current parameters of the model.

Modelled outputs and measured outputs are used to generate symptoms of several types (function 4). These symptoms are then used, along with the automatically generated rules, to diagnose faults (function 5).

Evaluation

This method is capable of detecting and diagnosing incipient faults but it does not explicitly state how trends would be detected over time. It establishes its own rule base, so it should be capable of working on any type of actuator. The process by which it achieves this is structured in a similar fashion to human thought, which makes this part of the method easy to understand. However, complex mathematical modelling is also used, so it does not completely fulfil the requirement to be easy to understand.

Implementing the method on a large asset base would be difficult because the rule base would need to be subtly different for each actuator, if it is generated using measured data only. This would mean that it would require fault simulations to be carried out on each monitored actuator, which is not practical. It is also not clear whether the piecewise linear model requires some pre-determination of the limits for

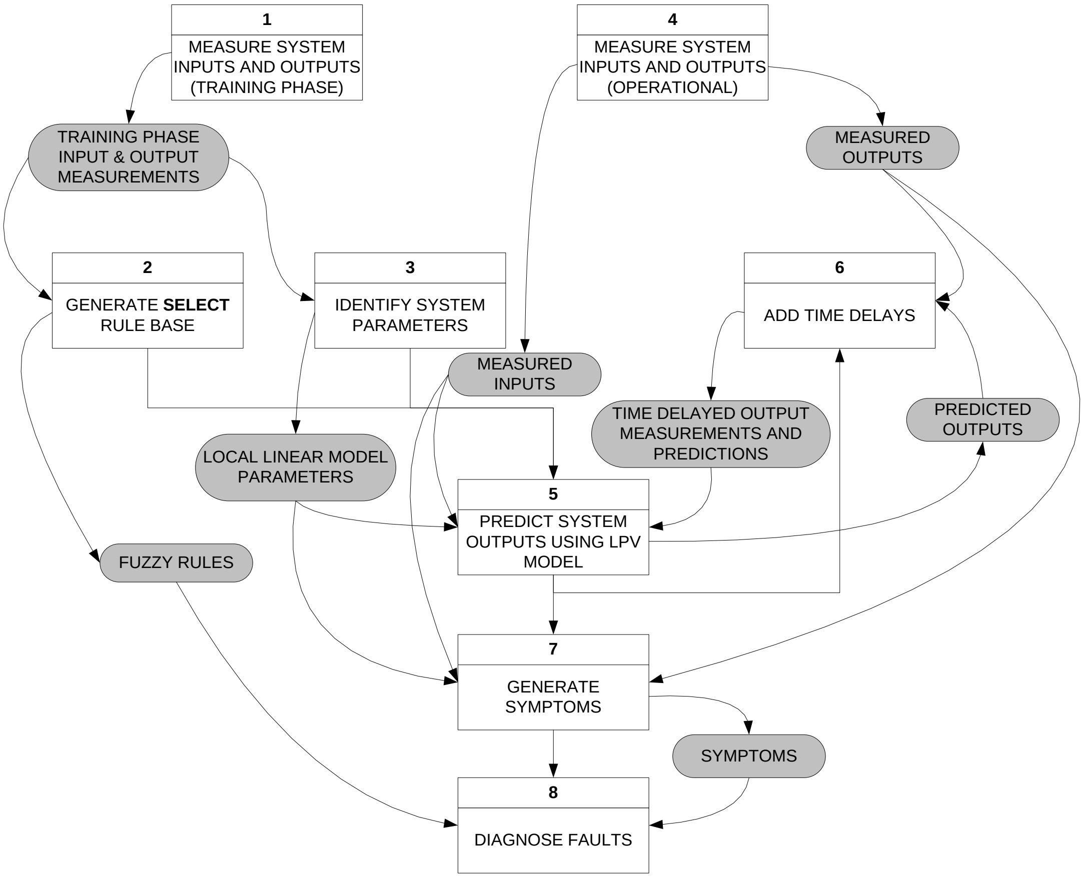
1
MEASURE SYSTEM INPUTS AND OUTPUTS (TRAINING PHASE)
4
MEASURE SYSTEM INPUTS AND OUTPUTS (OPERATIONAL)
TRAINING PHASE INPUT & OUTPUT MEASUREMENTS
MEASURED OUTPUTS
2
GENERATE SELECT RULE BASE
3
IDENTIFY SYSTEM PARAMETERS
6
ADD TIME DELAYS
MEASURED INPUTS
TIME DELAYED OUTPUT MEASUREMENTS AND PREDICTIONS
PREDICTED OUTPUTS
LOCAL LINEAR MODEL PARAMETERS
5
PREDICT SYSTEM OUTPUTS USING LPV MODEL
FUZZY RULES
7
GENERATE SYMPTOMS
SYMPTOMS
8
DIAGNOSE FAULTS

Figure 3.10: Fault diagnosis using a local linear model tree and self-learning classification

each piecewise segment - if it does, then this would also need to be carried out for each asset instance individually.

Conclusion

The model-based nature of this proposed method means it can deal well with variations in the inputs. However, it also increases system complexity.

The fact that the rule base is generated during a testing phase which requires all faults to be simulated means that it would be very difficult to implement such a method on a large number of similar actuators, unless the rules were general enough to apply to all instances of the actuator type. This is unlikely since the variations between actuators in fault-free performance would probably be enough to generate large residuals.

3.3.5 Fault detection and diagnosis using a fuzzy qualitative model

Proposed method

This method was developed as a means for solving the DAMADICS benchmark problem for actuator fault detection and diagnosis [59]. The DAMADICS problem is fully described in [58]. It is based on the diagnosis of faults in an electro-pneumatic flow control valve in a sugar factory.

The proposed method has a simple functional structure which is shown in figure 3.11. The classical physical model of the plant, in the form of mathematical equations, is used to generate (function 1 in figure 3.11) the parameters for a qualitative model based on confluences [29].

The subsequent functions operate on *linguistic variables*, which are qualitative representations of the properties of a quantitative signal. The plant input and output measurements are converted to linguistic variables by function 2, ready for use.

The qualitative model (function 3) is set up using the parameters generated by function 1 and takes as its inputs the linguistic variables which represent the measured inputs to the plant. The model predicts the nature of the linguistic variables which represent the plant outputs.

A discrepancy generator (function 4) then compares the linguistic variables predicted by the model with those representing the *measured* plant outputs.

The discrepancies, along with the linguistic representations of the measured variables, are fed to a diagnosis block (function 5) where a neural network processes them and gives diagnosed faults as the output. The neural network is designed to be trained using only a limited set of abrupt fault data, but to be able to diagnose incipient faults by recognising their symptoms when they are only at a small magnitude.

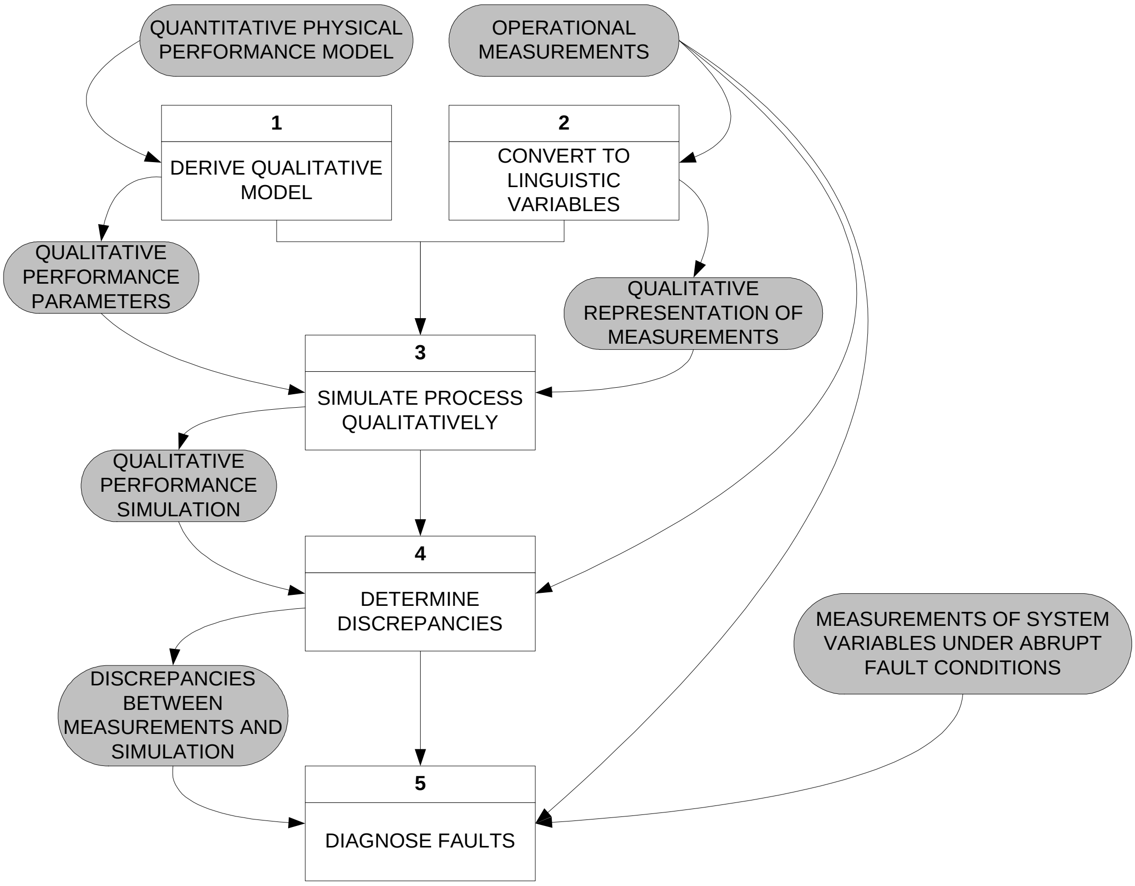
QUANTITATIVE PHYSICAL PERFORMANCE MODEL
OPERATIONAL MEASUREMENTS
1
DERIVE QUALITATIVE MODEL
2
CONVERT TO LINGUISTIC VARIABLES
QUALITATIVE PERFORMANCE PARAMETERS
QUALITATIVE REPRESENTATION OF MEASUREMENTS
3
SIMULATE PROCESS QUALITATIVELY
QUALITATIVE PERFORMANCE SIMULATION
4
DETERMINE DISCREPANCIES
MEASUREMENTS OF SYSTEM VARIABLES UNDER ABRUPT FAULT CONDITIONS
DISCREPANCIES BETWEEN MEASUREMENTS AND SIMULATION
5
DIAGNOSE FAULTS

Figure 3.11: Fault diagnosis with a fuzzy qualitative model

Evaluation

The method was designed to deal with both abrupt and incipient faults, but it was found to be poor at dealing with incipient faults because the development of symptoms over a long period of time was almost indistinguishable from signal noise [59] (section 5). Since incipient faults are the focus of this thesis, this method does not fulfil the requirement for detection and diagnosis. Additionally, it requires a detailed mathematical model for the actuator, something which may not be available. The black-box nature of neural networks will make it difficult for a user to understand why a particular output has been generated. The qualitative modelling approach taken, however, would be capable of taking account of operational differences between actuators of the same type.

Conclusion

The proposal has a straightforward functional flow and should be insensitive to small variations in operation between different instances of the same type of actuator, through the use of the qualitative model. However, a potentially complex mathematical model is required in order to generate the qualitative model.

This method has been shown in [59] to be incapable of dealing with incipient faults, which is a key requirement on the system to be produced in this thesis. The purely qualitative approach of this method appears to have contributed to its downfall, and suggests that a successful system will employ a mix of qualitative and quantitative processing.

3.3.6 Fault detection and diagnosis using timed automaton models

Proposed method

As part of the DAMADICS actuator benchmark diagnosis project [58], Supavatanakul et al [60] proposed a model-based diagnosis method which simulates actuator behaviour using timed automata. The functional flow of this method is shown in figure 3.12.

During an initial training phase, faults are simulated and measurements are taken. From these measurements, parameters for a series of automaton models are identified (function 1 in figure 3.12). For every fault which is simulated, an automaton is generated which models the actuator's behaviour under that fault condition.

The automaton models operate on qualitative representations of actuator outputs, measured during operation. These qualitative representations are obtained by discretisation (function 2). The value of the qualitative variable represents an interval on the quantitative variable being represented.

The automaton models (function 3) generate discrete outputs. In order for comparison to take place between the modelled and measured actuator outputs, the measured outputs are also discretised by function 2.

The diagnosis stage compares the I/O sequences of each automaton against the I/O sequence from the actuator. Where the sequences do not match, the automaton generating the sequence is rejected as a potential fault candidate. At the beginning of a time sequence, all faults are considered possible. The comparisons between I/O sequences are made every time instant. Over a period of time, fault conditions which are not possible are eliminated, leaving only the *set of possible faults*. This is the output of the diagnosis function.

Clearly, if the set of possible faults contains only one fault, then the fault has been diagnosed with certainty.

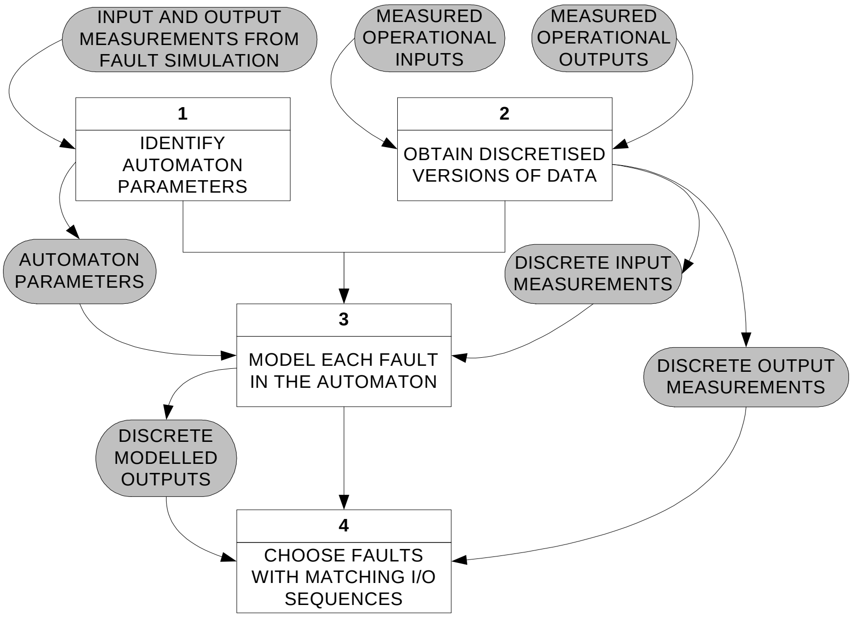
INPUT AND OUTPUT MEASUREMENTS FROM FAULT SIMULATION
MEASURED OPERATIONAL INPUTS
MEASURED OPERATIONAL OUTPUTS
1
IDENTIFY AUTOMATON PARAMETERS
2
OBTAIN DISCRETISED VERSIONS OF DATA
AUTOMATON PARAMETERS
DISCRETE INPUT MEASUREMENTS
3
MODEL EACH FAULT IN THE AUTOMATON
DISCRETE OUTPUT MEASUREMENTS
DISCRETE MODELLED OUTPUTS
4
CHOOSE FAULTS WITH MATCHING I/O SEQUENCES

Figure 3.12: Fault diagnosis using automaton models

Evaluation

The way this method works means that incipient faults would have to reach a significant level of severity before the system could eliminate enough fault candidates to give a useful diagnosis. It therefore does not fulfil the requirement to detect and diagnose incipient faults. The complex modelling approach, which uses multiple models, makes it difficult to understand, as does the use of set theory for the construction of the automata. The method could be used on more than one type of actuator, and the use of qualitative variables would mean that small differences in performance between instances could be accommodated by changes in the rules for discretisation of measured variables, although this would require an extra function which is not included.

Conclusion

This scheme is not suitable for the application being considered by the thesis, because it makes no provision for adapting the parameters to multiple actuators of the same type, and because it does not give as an output any indication of the severity of a fault.

The uncertainty introduced by discretising the variables is countered by making the set of possible faults a crisp set. However, since all members of this set have the same value of membership (1), there is no way to completely diagnose a fault unless the symptoms are so severe that none of the other fault candidates is valid. This is impractical for an actuator which may appear to work perfectly for many operations whilst an incipient fault is developing, because a diagnosis at a stage where operation has been affected, such that “no fault” is excluded from the set of possible faults, would be too late to be of any use for the purposes of condition-based maintenance.

3.3.7 Evaluation results

Fulfilment of system requirements

Table 3.1 shows the points scored by each method against each requirement considered to be key to the success of the system. This table indicates that the new method proposed in this thesis is the one which fulfils most of the requirements. As standalone schemes, the other methods do not fulfil all the thesis requirements, but certain aspects of their approach, as discussed in section 3.2, have given inspiration to the new method.

ID	**Method**
a	Solution proposed in this thesis
b	Semi-qualitative distributed fault detection and diagnosis using a piecewise linear model [66]
c	Nonlinear model-based fault diagnosis with automatic rule generation [16]
d	Fault detection and diagnosis using a fuzzy qualitative model [59]
e	Fault detection and diagnosis using timed automaton models [60]

Requirement	**Method**				
	(a)	**(b)**	**(c)**	**(d)**	**(e)**
Method must either be novel or be used in a novel way	1	0	0.5	0.5	0.5
Method must successfully predict and diagnose a reasonable number of faults	1	1	0.5	0	0.5
Method must be applicable to at least 3 different types of actuator	1	1	1	1	1
Method must be intuitive to human technicians	1	0.5	0.5	0.5	0
Method must be applicable to a large set of similar actuators	1	1	0	1	1
Total score (out of possible 5)	5	3.5	3	3	2.5

Table 3.1: Table of points scored by each method against the requirements

The new method also imposes the least requirements on the system. This is intuitively true, since it is the only one based completely on history rather than models.

It is therefore fair to conclude that the new combined approach of qualitative trend analysis and automatic fuzzy rule generation is more suitable for fulfilling the requirements of this thesis than methods which have previously been suggested.

Chapter 4

Functional design

Functional design is the process of refining the method of solving the design problem until a solution is produced which fulfils all the requirements. It is independent of the eventual implementation and therefore a functional design can be transferred between implementation platforms. Functional design is considered to be a fundamental stage in the design process, allowing the method to be determined without the distraction of concerns related to implementation (unless they are so important that they have had an effect on the requirements)[6]. This was especially important in this thesis because it was initially proposed that there might be an opportunity to construct an implementation suitable for field trials on the UK rail network. Therefore the eventual method would need to be suitable for implementation both in the lab and out on the track.

Section 4.1 describes the process of decomposition, where the function of the system is broken up into smaller functions until each function is simple enough to be implemented easily. The design is graphically represented as a set of Enhanced Functional Flow Block Diagrams, which show the functions and also the items passed between them as inputs and outputs. Each level of decomposition generates a new, more detailed set of diagrams.

Section 4.2 represents the eventual functional design in terms of the mathematical functions used.

4.1 Functional decomposition

4.1.1 Initial functional representation of the design solution

Figure 3.8 shows the initial view of the system, modelling it as a black box with inputs to be defined by the requirements of the solution developed, and outputs which correspond to the requirements.

The solution constructed in this thesis is designed to use process history data which is easy to obtain. This means that any fault simulation data must be collected from an actuator which is not operational, because it would be inconvenient and potentially dangerous to simulate faults on railway actuators which are then required for service.

From the outset, therefore, it was considered impossible to obtain accurate history data for all fault conditions *for every operational actuator of a certain type*. However, it was considered acceptable for each operational actuator to provide some initial training data which can be assumed to be fault-free.

These two data sets were considered to be the only available inputs for the purposes of training the system, because it would be impractical to attempt to obtain any more data from each individual actuator, as typically there might be thousands of instances of any one actuator type in service. For the purposes of monitoring it was assumed that the same variables as contained in the training data would be available for measurement when the actuators are in operation.

Thus there are three data sets, each of which pertain to a different situation. Three functions were developed to work on these data to produce a confidence level for each known fault: function 1 processes the fault simulation data and produces a rule set; function 2 uses the fault-free data from a particular actuator to adapt the rule set to that actuator's individual performance; function 3 processes the online operational data and evaluates the possibility of each fault being present. A fourth function was added to this to track the confidence levels and raise alarms at an appropriate point.

This design is the first-level functional decomposition of the system. It is shown in figure 4.1 in block diagram form.

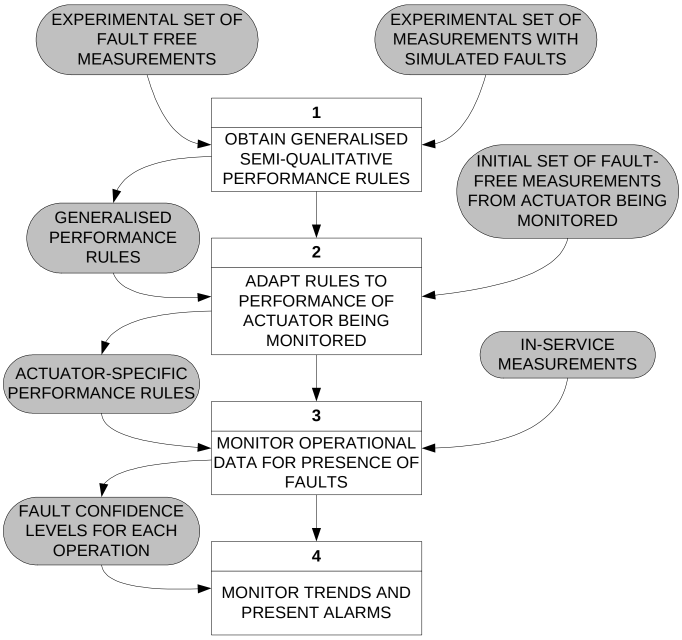
EXPERIMENTAL SET OF FAULT FREE MEASUREMENTS
EXPERIMENTAL SET OF MEASUREMENTS WITH SIMULATED FAULTS
1
OBTAIN GENERALISED SEMI-QUALITATIVE PERFORMANCE RULES
INITIAL SET OF FAULT-FREE MEASUREMENTS FROM ACTUATOR BEING MONITORED
GENERALISED PERFORMANCE RULES
2
ADAPT RULES TO PERFORMANCE OF ACTUATOR BEING MONITORED
IN-SERVICE MEASUREMENTS
ACTUATOR-SPECIFIC PERFORMANCE RULES
3
MONITOR OPERATIONAL DATA FOR PRESENCE OF FAULTS
FAULT CONFIDENCE LEVELS FOR EACH OPERATION
4
MONITOR TRENDS AND PRESENT ALARMS

Figure 4.1: First-level functional decomposition of the solution

4.1.2 Further decomposition

Each of the four first-level functions is complex and therefore each was decomposed into further functional diagrams.

In order that each of the three data sets can be operated on throughout the system, it is clear that a common format is required. Given that neither wholly quantitative nor wholly qualitative representation methods seem to be comprehensively infallible for representing the way mechanical equipment behaves, it was decided that a common representation for measured data would be based on qualitative trend analysis. A function to construct this representation therefore appears in the second-level diagrams as functions 1.1, 2.1 and 3.1.

Function 1 establishes a rule base and therefore the result of the representation made in function 1.1 is used to establish rules which will show whether a certain portion of the representation looks faulty or normal. Function 1.2 then constructs the rule base. This arrangement is shown in figure 4.2.

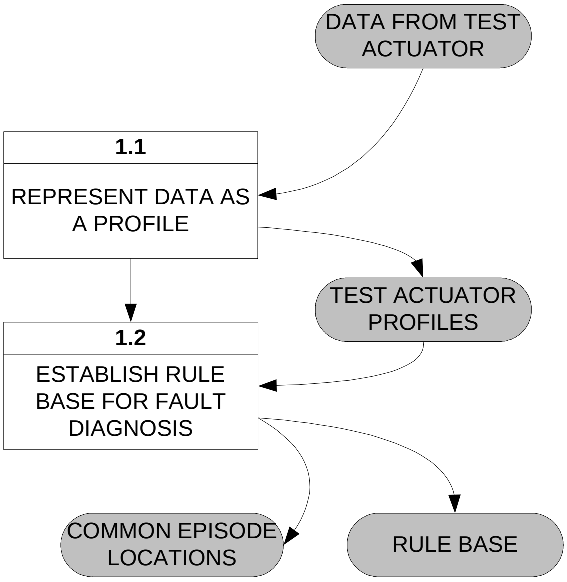

Figure 4.2: Second-level decomposition for function 1

The representation functions are based on the previous work done in the series of papers by Cheung, Stephanopoulos and Bakshi [47, 48, 49, 50]. The following definitions, corresponding with the QTA process illustrated in figure 3.6, are made for the representation of data in a combined qualitative and quantitative way:

Partition - A fixed-length portion of a waveform

Episode - A number of consecutive partitions which all have the same qualitative state, representing a trend of a particular shape. The letters $\mathcal{A}$-$\mathcal{I}$ are used to denote each different shape.

Profile - A sequence of episodes which represent a waveform of measured data.

Partitions and episodes contain start and end values, start and end times and a qualitative state.

The quantitative portion of the representation is obtained by storing the start and end values of the episode, and the start and end times. The resulting profile will now be called an *absolute* profile because the quantities in it are absolute values.

In order to eliminate information which is not relevant to the scale at which analysis of the waveforms is taking place, the raw measurements are first filtered to smooth them. This ensures that the profile is not unmanageably large, and also that the random variations between waveforms do not make the qualitative part of the profile change from waveform to waveform. In theory, providing the monitored equipment is in a constant state of repair, the qualitative portion of the profile will not change at all, and there will only be small, random variations in the quantitative portion.

Figure 4.3 shows how the profile is constructed from the original data. The output of function 1.1 must be refined before using it to form the rule base. Episodes are included in the formation of the rule base only if there is a similar episode in all profiles. This must be the case in order that variation between fault conditions can be correctly examined. Therefore, the first action of function 1.2 is to extract the common episodes from the profile set created by function 1.1.

A common episode is one which occurs in each profile and under every fault condition. The algorithm for detecting common episodes is described in detail in section 4.2. Once the common episodes have been extracted, it is possible to analyse the variations in profile values, both within a single fault condition and also between fault conditions. In order to make the rule base applicable to all actuators of the type in question,

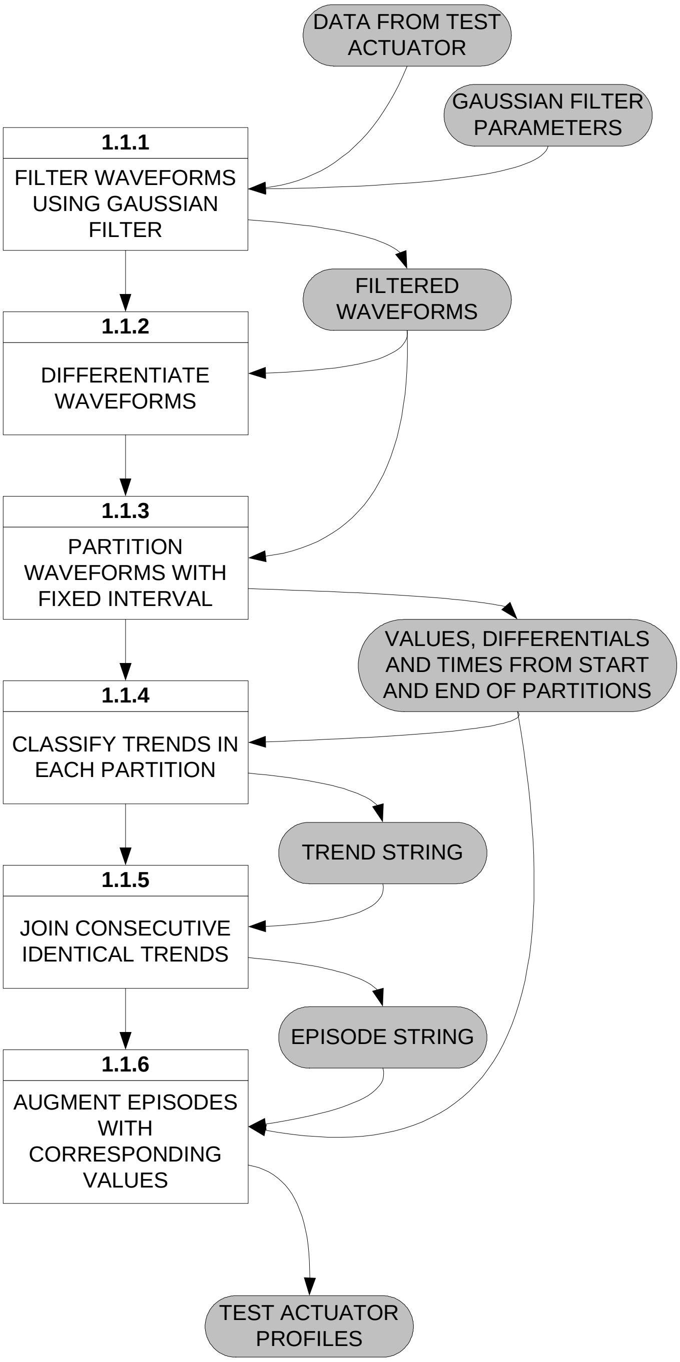
DATA FROM TEST ACTUATOR
GAUSSIAN FILTER PARAMETERS
1.1.1
FILTER WAVEFORMS USING GAUSSIAN FILTER
FILTERED WAVEFORMS
1.1.2
DIFFERENTIATE WAVEFORMS
1.1.3
PARTITION WAVEFORMS WITH FIXED INTERVAL
VALUES, DIFFERENTIALS AND TIMES FROM START AND END OF PARTITIONS
1.1.4
CLASSIFY TRENDS IN EACH PARTITION
TREND STRING
1.1.5
JOIN CONSECUTIVE IDENTICAL TRENDS
EPISODE STRING
1.1.6
AUGMENT EPISODES WITH CORRESPONDING VALUES
TEST ACTUATOR PROFILES

Figure 4.3: Third-level decomposition for function 1.1

the values in the common episodes are expressed as an additive variation from the fault free values (function 1.2.2).

Functions 1.2.3 and 1.2.4 eliminate episode values which are not useful for forming rules. There are two criteria for elimination: if a relative value is too close to the fault free value (function 1.2.3 eliminates values which fulfil this criterion, by checking the magnitude of relative values against a threshold) or if a relative value is too close to another fault's relative value (function 1.2.4 eliminates values fulfilling this criterion, by checking relative value differences between all fault conditions against a threshold). These functions ensure that the only values which are used to form rules are those where performance varies distinctly from the fault-free case, AND where performance variations for each fault are easily distinguished.

The parameters for the membership functions are then calculated. The side of the membership function closest to the fault-free case is established to be a gradual increase between fault-free and the average value for that fault. The further side of the function is established to decrease to 0 at a rate defined by a constant. Function 1.2.5 carries this out. The third-level decomposition of function 1.2 is shown in figure 4.4.

The level 2 decomposition of function 2 is shown in figure 4.5. Function 2 decomposes to two subfunctions. As previously mentioned, function 2.1 is a representation function almost identical to function 1.1. This is illustrated in figure 4.6. Function 2.2 uses the common episode locations generated in function 1 to reference the profiles generated in function 2.1. This is shown in figure 4.7.

The output of function 2 is a set of common episode values for the actuator currently being monitored. This allows operational measured data to be normalised to the fault-free values for that particular actuator. The resulting relative values can then be used to evaluate fault confidence in function 3.

Function 3 decomposes to two functions, as shown in figure 4.8. Once again, the

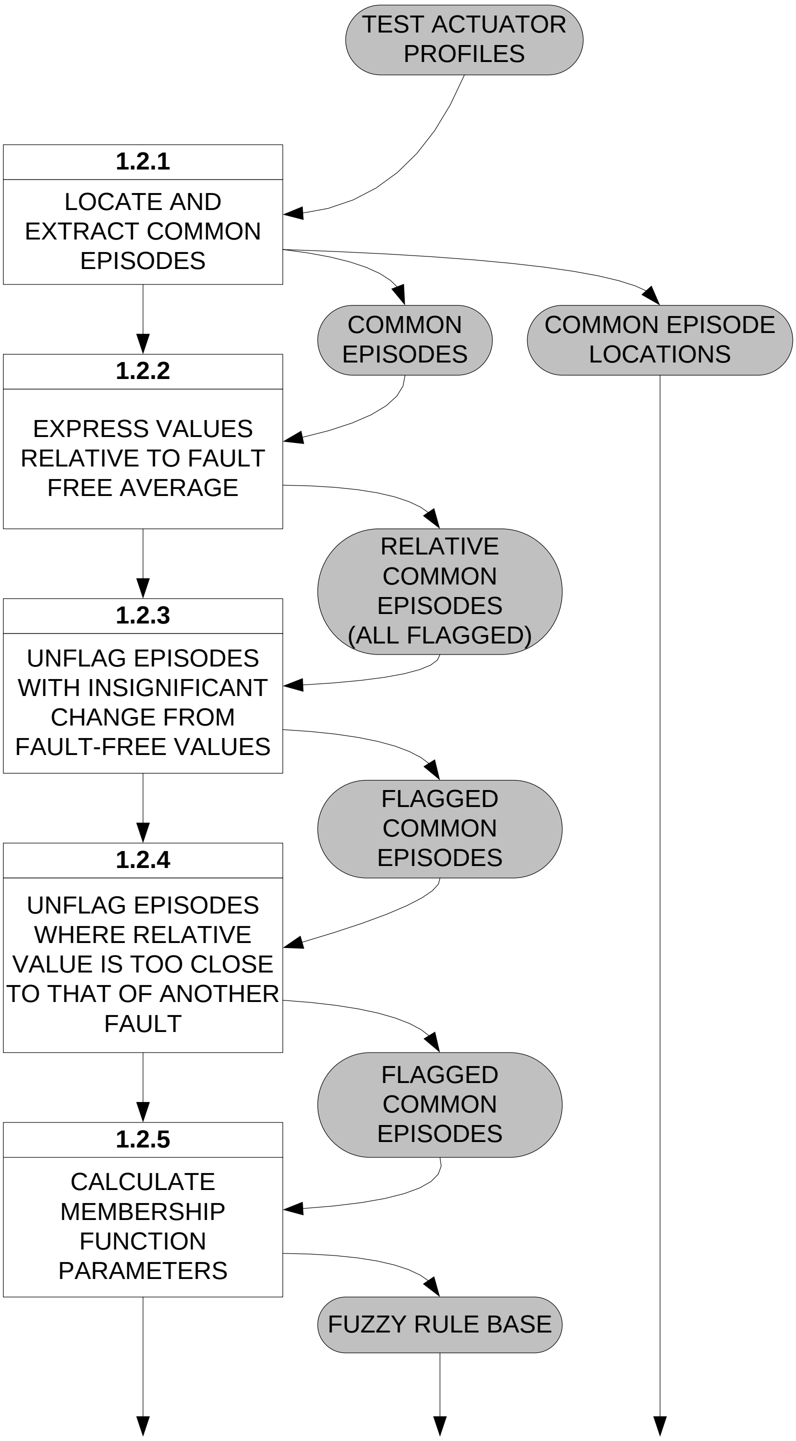
TEST ACTUATOR PROFILES
1.2.1
LOCATE AND EXTRACT COMMON EPISODES
COMMON EPISODES
COMMON EPISODE LOCATIONS
1.2.2
EXPRESS VALUES RELATIVE TO FAULT FREE AVERAGE
RELATIVE COMMON EPISODES (ALL FLAGGED)
1.2.3
UNFLAG EPISODES WITH INSIGNIFICANT CHANGE FROM FAULT-FREE VALUES
FLAGGED COMMON EPISODES
1.2.4
UNFLAG EPISODES WHERE RELATIVE VALUE IS TOO CLOSE TO THAT OF ANOTHER FAULT
FLAGGED COMMON EPISODES
1.2.5
CALCULATE MEMBERSHIP FUNCTION PARAMETERS
FUZZY RULE BASE

Figure 4.4: Third-level decomposition for function 1.2

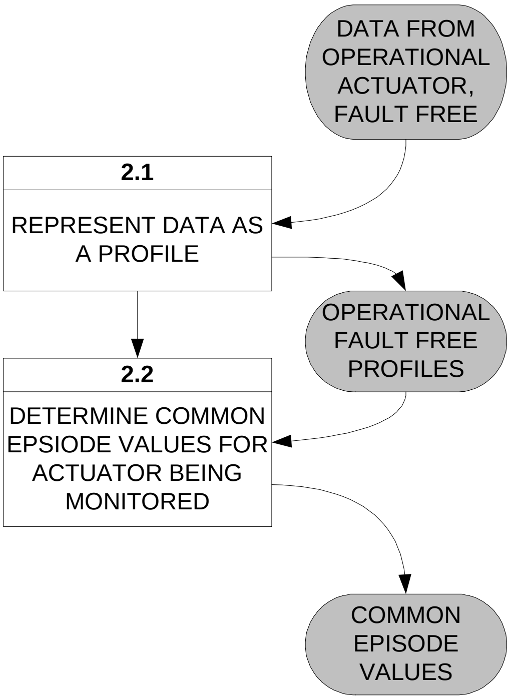
DATA FROM OPERATIONAL ACTUATOR, FAULT FREE
2.1
REPRESENT DATA AS A PROFILE
OPERATIONAL FAULT FREE PROFILES
2.2
DETERMINE COMMON EPSIODE VALUES FOR ACTUATOR BEING MONITORED
COMMON EPISODE VALUES

Figure 4.5: Second-level decomposition for function 2

first of these is concerned with the semi-qualitative representation of the measured data, but function 3.1 also uses the fault-free common episode values from function 2.2 to relativise the values from operational profiles. This is shown in figure 4.9. The relative quantities are the correct input to function 3.2, which evaluates them against the rule base to determine the presence of all possible faults by calculating the average membership score for all quantities in all common episodes. This is illustrated in figure 4.10. Function 4 processes the outputs of function 3 in order to make the results more accessible to users and thereby aid decision making. Since the system is not yet destined to be used by technicians, the interface requirements have been interpreted to apply chiefly to the testing of the system. Therefore, function 4 is designed to be of use in testing and understanding the system. The decomposition of this function is shown in figure 4.11.

The data which are used to test the system's operational performance are long runs of fault simulations, where the fault is gradually introduced. Function 4.1 collates the outputs of function 3 for each operation, and calculates a linear trend

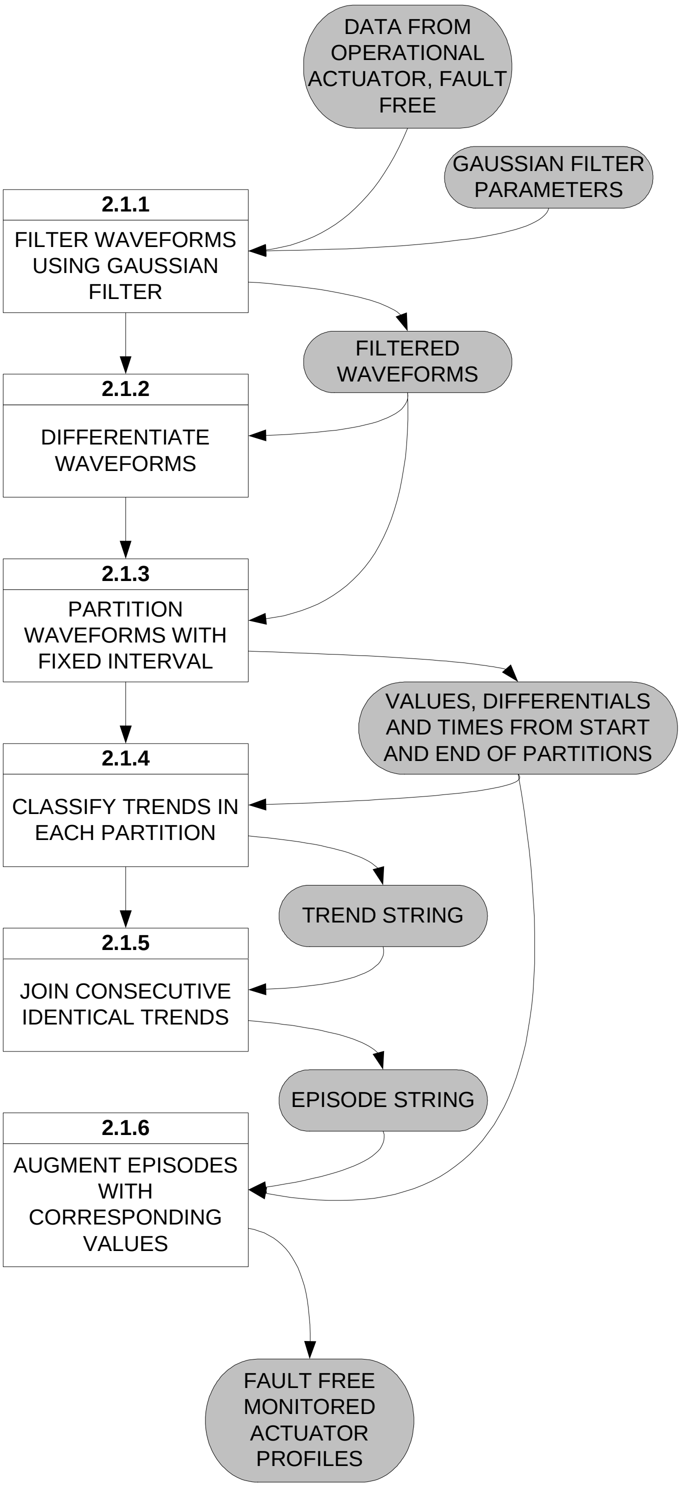
DATA FROM OPERATIONAL ACTUATOR, FAULT FREE
GAUSSIAN FILTER PARAMETERS
2.1.1
FILTER WAVEFORMS USING GAUSSIAN FILTER
FILTERED WAVEFORMS
2.1.2
DIFFERENTIATE WAVEFORMS
2.1.3
PARTITION WAVEFORMS WITH FIXED INTERVAL
VALUES, DIFFERENTIALS AND TIMES FROM START AND END OF PARTITIONS
2.1.4
CLASSIFY TRENDS IN EACH PARTITION
TREND STRING
2.1.5
JOIN CONSECUTIVE IDENTICAL TRENDS
EPISODE STRING
2.1.6
AUGMENT EPISODES WITH CORRESPONDING VALUES
FAULT FREE MONITORED ACTUATOR PROFILES

Figure 4.6: Third-level decomposition for function 2.1

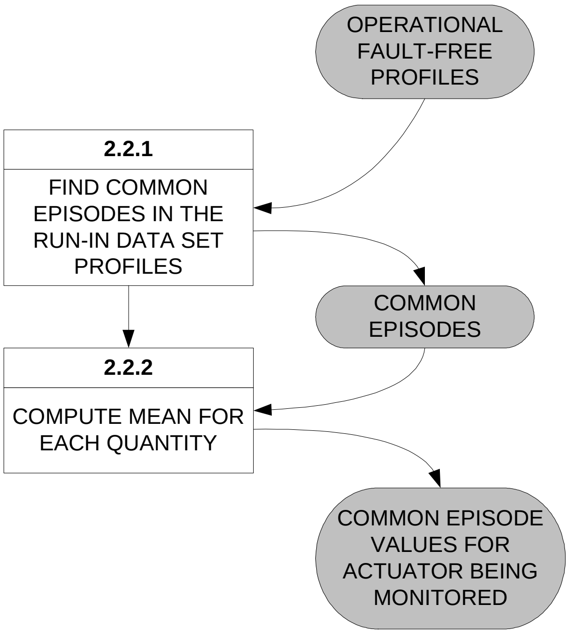
OPERATIONAL FAULT-FREE PROFILES
2.2.1
FIND COMMON EPISODES IN THE RUN-IN DATA SET PROFILES
COMMON EPISODES
2.2.2
COMPUTE MEAN FOR EACH QUANTITY
COMMON EPISODE VALUES FOR ACTUATOR BEING MONITORED

Figure 4.7: Third-level decomposition for function 2.2

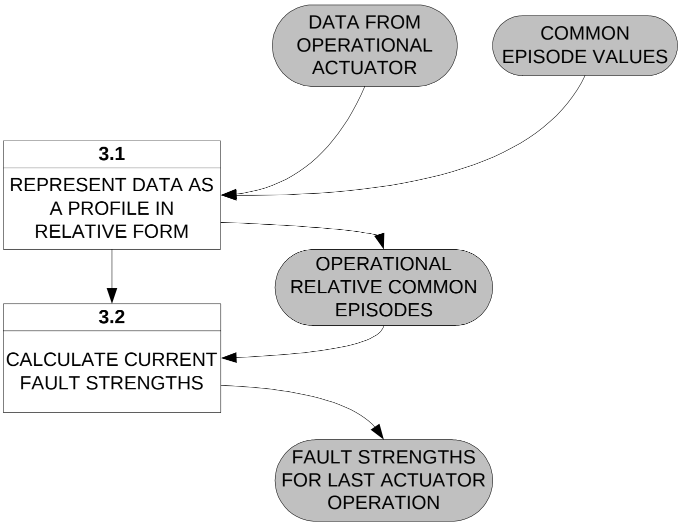
DATA FROM OPERATIONAL ACTUATOR
COMMON EPISODE VALUES
3.1
REPRESENT DATA AS A PROFILE IN RELATIVE FORM
OPERATIONAL RELATIVE COMMON EPISODES
3.2
CALCULATE CURRENT FAULT STRENGTHS
FAULT STRENGTHS FOR LAST ACTUATOR OPERATION

Figure 4.8: Second-level decomposition for function 3

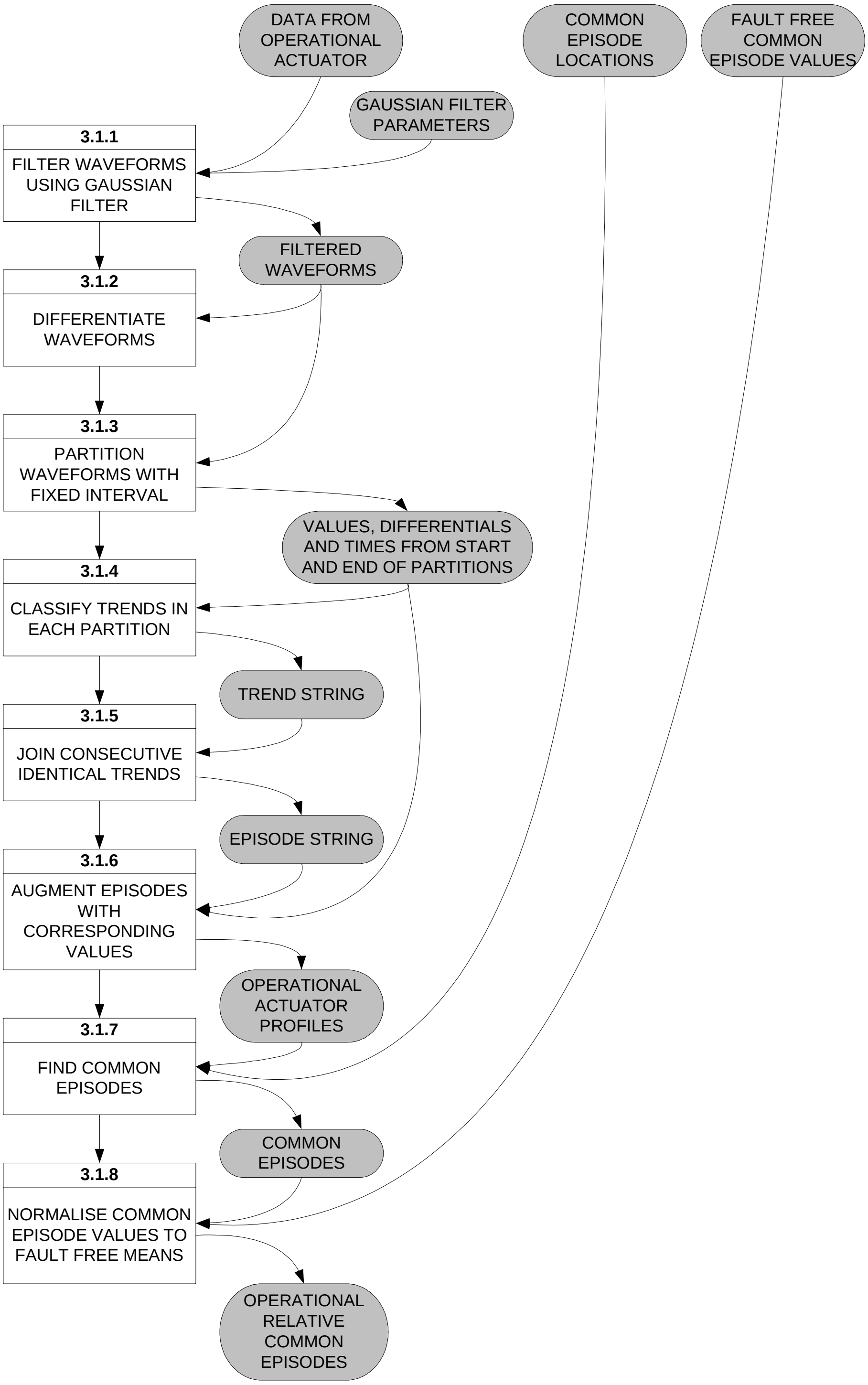
DATA FROM OPERATIONAL ACTUATOR
COMMON EPISODE LOCATIONS
FAULT FREE COMMON EPISODE VALUES
GAUSSIAN FILTER PARAMETERS
3.1.1
FILTER WAVEFORMS USING GAUSSIAN FILTER
FILTERED WAVEFORMS
3.1.2
DIFFERENTIATE WAVEFORMS
3.1.3
PARTITION WAVEFORMS WITH FIXED INTERVAL
VALUES, DIFFERENTIALS AND TIMES FROM START AND END OF PARTITIONS
3.1.4
CLASSIFY TRENDS IN EACH PARTITION
TREND STRING
3.1.5
JOIN CONSECUTIVE IDENTICAL TRENDS
EPISODE STRING
3.1.6
AUGMENT EPISODES WITH CORRESPONDING VALUES
OPERATIONAL ACTUATOR PROFILES
3.1.7
FIND COMMON EPISODES
COMMON EPISODES
3.1.8
NORMALISE COMMON EPISODE VALUES TO FAULT FREE MEANS
OPERATIONAL RELATIVE COMMON EPISODES

Figure 4.9: Third-level decomposition for function 3.1

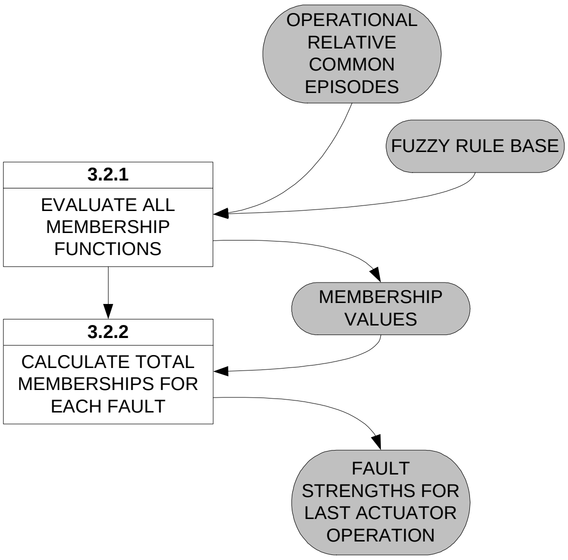
OPERATIONAL RELATIVE COMMON EPISODES
FUZZY RULE BASE
3.2.1
EVALUATE ALL MEMBERSHIP FUNCTIONS
MEMBERSHIP VALUES
3.2.2
CALCULATE TOTAL MEMBERSHIPS FOR EACH FAULT
FAULT STRENGTHS FOR LAST ACTUATOR OPERATION

Figure 4.10: Third-level decomposition for function 3.2

for the strength of each fault over the operations measured. The trend's gradient should be strongly positive for the fault strength corresponding to the fault condition being simulated. All other gradients should be less than that for the current fault. Function 4.2 plots the fault strengths for each possible fault, and each fault simulated. The trend is then superimposed on the graph.

In a practical system, a human factors assessment would be carried out to determine exactly how the data could be presented to maintenance staff in the best way to help them make decisions.

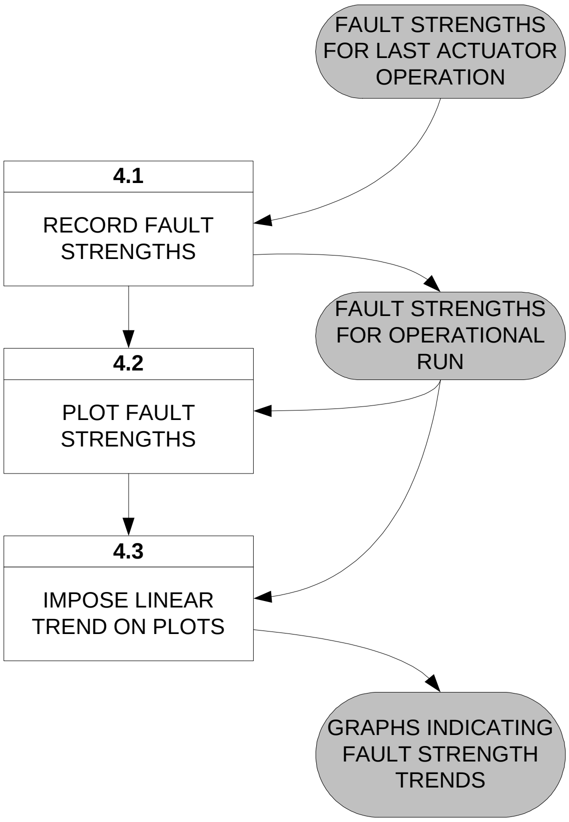
FAULT STRENGTHS FOR LAST ACTUATOR OPERATION
4.1
RECORD FAULT STRENGTHS
FAULT STRENGTHS FOR OPERATIONAL RUN
4.2
PLOT FAULT STRENGTHS
4.3
IMPOSE LINEAR TREND ON PLOTS
GRAPHS INDICATING FAULT STRENGTH TRENDS

Figure 4.11: Second-level decomposition for function 4

4.2 Mathematical representation of the functional design

Some notation conventions have been used throughout this chapter. Details of these conventions can be found in appendix B.

4.2.1 Input data processing

Let A be the set of actuators of an identical type. A's members may be installed in differing situations but the actuators themselves are all of the same design. Then, in order to diagnose faults, the diagnosis rules must first be established qualitatively by simulating faults on a test actuator $a_t \in A$. The system is then trained to understand the individual performances of the actuators in service, $a_1 \ldots a_{n_A}$ where n_A is the number of *monitored actuators* in set A, and thereby establish quantitative diagnosis rules for each individual actuator. Thus

$$A = \{a_t,\ a_1,\ \ldots,\ a_{n_A}\} \tag{4.1}$$

Actuators of type A may display faulty behaviour. Let the set E be the set of all possible behaviours which can be seen in actuators in set A. This includes normal or fault-free behaviour. Not all of these behaviours can be reliably diagnosed using available technology. In order to teach the system to diagnose a fault, it must first be simulated using a_t. Therefore, by prior analysis, a subset of behaviours G is chosen, whose members fulfil the following criteria:

- Can be simulated on a_t
- Resulting behaviour differs measurably[1] from fault free behaviour
- Would cause the actuator to fail if unchecked

[1] Given the sensors available

A new set F is now constructed, which shall contain behaviours which can be seen in an operational actuator *and* matched to a behaviour seen on a_t. Fault-free behaviour is labelled f_0 and the faulty behaviours $f_1 \dots f_{n_F-1}$, where n_F is the number of members of F i.e. the number of diagnosable behaviours. It follows that F shall only contain f_0 and G.

$$\begin{aligned} F &\subset E && (4.2) \\ F &= \{f_0,\, G\} && (4.3) \\ G &= \{f_1,\, \dots,\, f_{n_F-1}\} && (4.4) \end{aligned}$$

Using the test actuator a_t, each of the behaviours which are members of F is induced. For the faulty behaviours, the behaviour shall be induced at several levels of intensity, the highest of which shall result in the actuator being *on the point of failure*. The highest intensity for behaviour i shall be labelled $f_i\,(100)$ and the lower intensities $f_i\,(b)$ where b is the percentage intensity of the fault compared to $f_i\,(100)$. For example, if faulty behaviour f_2 can be simulated by the loosening of a certain bolt, and loosening it by 6 turns causes a_t to be on the point of failure, then the resulting behaviour is $f_2\,(100)$, and the faulty behaviour induced by loosening the bolt only 3 turns is $f_2\,(50)$.

Each fault simulation is carried out for R repetitions. R is set for each actuator type before testing starts and shall be suitably large. During each repetition, L samples are taken, at an interval of T seconds. L may vary between repetitions, depending on the measurement system used. It cannot be relied upon that L will be constant throughout. The variables to be measured shall be labelled $m_l\,(t)$. The number of variables to be measured shall be written as n_V. So, the measurement $\underline{\mathbf{M}}$ for one repetition shall *nominally* have the form:

$$\underline{\mathbf{M}} = \begin{bmatrix} T & m_1(T) & \dots & m_{n_V}(T) \\ \vdots & \vdots & \ddots & \vdots \\ LT & m_1(LT) & \dots & m_{n_V}(LT) \end{bmatrix} \tag{4.5}$$

This assumes that the time of each sample is measured exactly. Whilst this is not strictly true in all cases, any inaccuracies introduced by this assumption will be negated by the much greater effects of the filtering process, and will in any case be too small to affect calculations based on the dynamic scale of the actuator, since at a sampling frequency of 1 kHz, variations on time values are at least three orders of magnitude smaller than the fastest STME dynamics.

4.2.2 Function 1 - Rule base generation

Function 1.1 - Profile generation

Function 1.1.1 - Filtering

In order to make the process of qualitative representation easier, the waveforms in $\underline{\mathbf{M}}$ are filtered to remove high-frequency content. A Gaussian filter was used because previous work [48] suggested that this was the best filter for preserving the time position of turning points in the data. A Gaussian filter can be approximated satisfactorily by using repeated iterations of a windowing function (a 'boxcar' filter). Three iterations were found to be sufficient to filter the data adequately. The filter width was adjusted, for each variable, to the value which gave the best performance. Figure 4.12 shows the effects of Gaussian filtering on the current waveform from an AC switch actuator.

The effects of filtering seem quite drastic, but the important thing to note is that the effects of fault conditions can be seen both before and after filtering; the difference

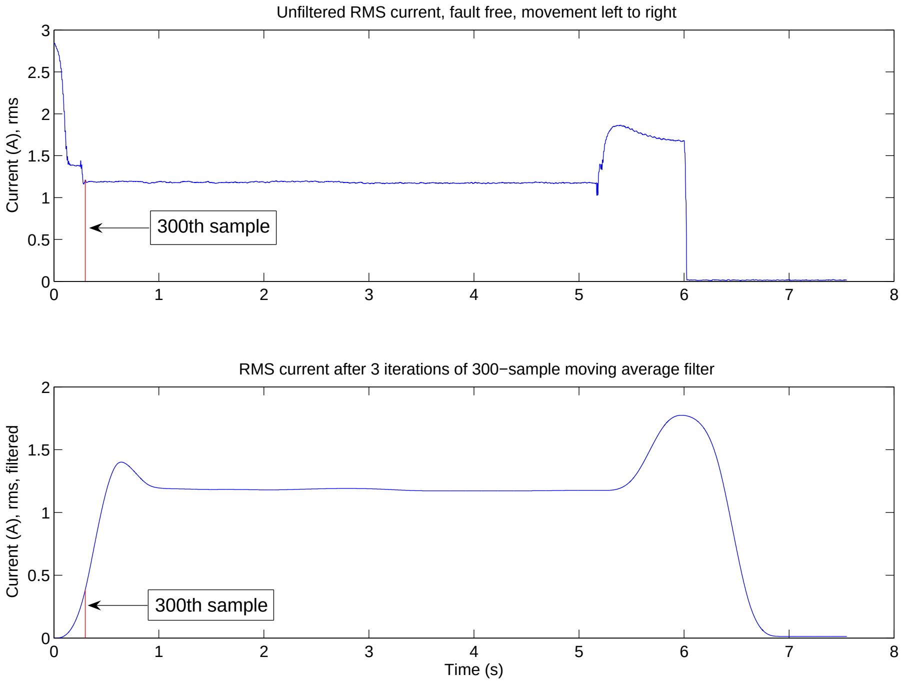

Figure 4.12: A current waveform, sampled from an AC switch actuator at 1 kHz, before and after filtering

is now there are only a few curve shapes for the system to try to spot and match, making the process more efficient.

The corresponding filter parameters are fed as arguments to a filtering function which gives as its output a vector of the same length as the input vector, allowing direct substitution of the unfiltered signal with the filtered one without modification to the time series.

The filter can be expressed as the following z-domain input-output relation:

$$Y(z) = \frac{1 + z^{-1} + \ldots + z^{-(w-1)}}{w} X(z) \tag{4.6}$$

where Y is the output signal, X is the input signal and w is the the number of samples over which the average is calculated (the width of the filter). This operation is carried out three times successively in order to approximate the Gaussian filter. In the implementation platform (MATLAB®), the filter is a "direct form II implementation of the standard difference equation" [70].

The output of the filtering process is a matrix $\underline{\mathbf{D}}$ which has the same layout as $\underline{\mathbf{M}}$ but represents the filtered values:

$$\underline{\mathbf{D}} = \begin{bmatrix} \underline{\mathbf{t}} & \underline{\mathbf{d}}_1 & \dots & \underline{\mathbf{d}}_{n_V} \end{bmatrix} \tag{4.7}$$

Function 1.1.2 - Differentiation

The next step is differentiation with respect to time. If a uniform time step length is assumed, differentiation could be simulated by simply taking the differences between values for each $\underline{\mathbf{d}}$ vector (the time series for each variable), but in practice the recorded time values must be used. The variable columns of each data matrix $\underline{\mathbf{D}}$ are differentiated with respect to the time column, resulting in a differential matrix $\underline{\dot{\mathbf{D}}}$. Differentiation is carried out twice, with individual differential matrices $\underline{\dot{\mathbf{D}}}_{\langle i,j\rangle}$ and $\underline{\ddot{\mathbf{D}}}_{\langle i,j\rangle}$, defined as follows:

$$\underline{\dot{\mathbf{D}}} = \begin{bmatrix} \underline{\mathbf{t}} & \underline{\dot{\mathbf{d}}}_1 & \dots & \underline{\dot{\mathbf{d}}}_{n_V} \end{bmatrix} \tag{4.8}$$

$$\text{where} \quad \dot{d}_{\langle k\rangle} = \begin{cases} 0 & k = 1 \\ \frac{d_{\langle k\rangle} - d_{\langle k-1\rangle}}{t_{\langle k\rangle} - t_{\langle k-1\rangle}} & k > 1 \end{cases} \tag{4.9}$$

$$\underline{\ddot{\mathbf{D}}} = \begin{bmatrix} \underline{\mathbf{t}} & \underline{\ddot{\mathbf{d}}}_1 & \dots & \underline{\ddot{\mathbf{d}}}_{n_V} \end{bmatrix} \tag{4.10}$$

$$\text{where} \quad \ddot{d}_{\langle k\rangle} = \begin{cases} 0 & k = 1 \\ \frac{\dot{d}_{\langle k\rangle} - \dot{d}_{\langle k-1\rangle}}{t_{\langle k\rangle} - t_{\langle k-1\rangle}} & k > 1 \end{cases} \tag{4.11}$$

Function 1.1.3 - Partitioning

The two data sets (value and differential) are now partitioned into the intervals best suited for extracting episodes. The partitioning interval is the interval, in samples, at which partition boundaries occur. It is written I_p and the corresponding number of partitions in the waveform is n_p. The partitioning interval will be the same for all variables to decrease computer expense. Since STME is partly defined by the timing of its dynamics in relation to the throw time, a suitable value of I_p for one variable is likely to be suitable for other variables, because the time characteristics of STME dynamics across all variables are very similar.

$$n_p = \left\lfloor \frac{L}{I_p} \right\rfloor \tag{4.12}$$

There will be a few samples left over at the end of the waveform unless $L - 1$ is a multiple of I_p. In practice, this should only be a small proportion of the waveform because I_p will be designed to be small compared to L.

The waveform is partitioned by storing the start and end values of each partition in new matrices. The end value of partition k becomes the start value of partition $k + 1$ and the same is true for time values. The following assignments are made for values of k from 1 to n_p and values of l from 1 to n_v:

$$x_{\langle k \rangle} = d_{\langle (k-1)I_p+1 \rangle} \tag{4.13}$$

$$y_{\langle k \rangle} = d_{\langle kI_p \rangle} \tag{4.14}$$

$$\dot{x}_{\langle k \rangle} = \dot{d}_{\langle (k-1)I_p+1 \rangle} \tag{4.15}$$

$$\dot{y}_{\langle k \rangle} = \dot{d}_{\langle kI_p \rangle} \tag{4.16}$$

$$\bar{\ddot{d}} = \frac{1}{I_p} \sum_{\kappa=(k-1)I_p}^{kI_p} \ddot{d}_{\langle \kappa \rangle} \tag{4.17}$$

$$t_{x\langle k \rangle} = t_{\langle (k-1)I_p+1 \rangle} \tag{4.18}$$

$$t_{y\langle k \rangle} = t_{\langle kI_p \rangle} \tag{4.19}$$

The value $\bar{\ddot{d}}$ represents the average of the second differential over the partition. This gives a generalised view of the shape of this part of the waveform and is more useful than the second differential value at a particular point.

Figure 4.13 shows graphically how values are chosen and stored in the partitioning process.

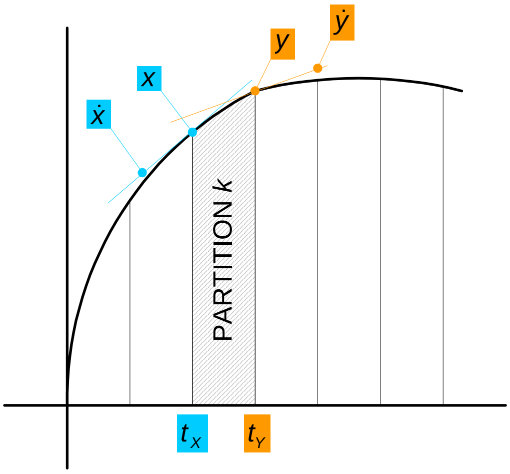

Figure 4.13: Assignment of values in the partitioning process

Function 1.1.4 - Qualitative classification

The values of the first and second differentials are used to classify the behaviour of each partition as a certain qualitative state. The possible qualitative states have distinct graphical shapes which are shown in figure 4.14. For each partition and each variable, the qualitative state value for the partition, s, is determined based on the *qualitative state* of the quantities x, $\dot{x}$, y, $\dot{y}$ and $\bar{\ddot{d}}$. The qualitative state of a variable is defined here as the sign of the variable. A qualitative state is written with square brackets around the original variable, e.g. $[x]$, and can take the values $+$, - or 0. Table 4.1 shows the rules for determining the partition's qualitative state. Each s element takes a letter value between $\mathcal{A}$ and $\mathcal{I}$.

The comparison of gradients is subject to a similarity threshold, to avoid rapid changes of qualitative state between partitions with very slightly different gradients but which are parts of the same overall trend.

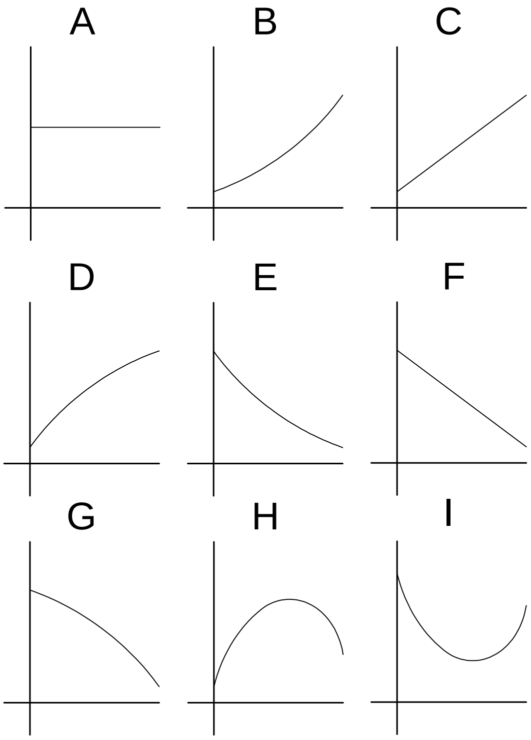

Figure 4.14: List of defined qualitative states

State	$[y-x]$	$[\dot{x}]$	$[\dot{y}]$	$\left[\ddot{d}\right]$
$\mathcal{A}$	any	$[0]$	$[0]$	$[0]$
$\mathcal{B}$	$[+]$	$[+]$	$[+]$	$[+]$
$\mathcal{C}$	$[+]$	$[+]$	$[+]$	$[0]$
$\mathcal{D}$	$[+]$	$[+]$	$[+]$	$[-]$
$\mathcal{E}$	$[-]$	$[-]$	$[-]$	$[+]$
$\mathcal{F}$	$[-]$	$[-]$	$[-]$	$[0]$
$\mathcal{G}$	$[-]$	$[-]$	$[-]$	$[-]$
$\mathcal{H}$	All other conditions where			$[-]$
$\mathcal{I}$	All other conditions where			$[+]$

Table 4.1: Table of criteria for deducing the qualitative state of a partition

```
for each variable do
    Establish episode 1 as the first interval;
    /* $k_p = 1$, $j = 1$, $k_e = 1$                                                */
    /* $\sigma_{\langle 1,l \rangle} = s_{\langle 1,l \rangle}$, $x_{e\langle 1,l \rangle} = x_{\langle 1,l \rangle}$, $y_{e\langle 1,l \rangle} = y_{\langle 1,l \rangle}$, $\tau_{x\langle 1,l \rangle} = t_{x\langle 1,l \rangle}$,
       $\tau_{y\langle 1,l \rangle} = t_{y\langle 1,l \rangle}$                                                    */
    while $s_{\langle k_p,l \rangle} = s_{\langle (k_p+1),l \rangle}$ do
        increment $k_p$;
    end
    /* The current interval is now the last interval of this
       episode                                                              */
    Replace final values in this episode with the final values of the current
    interval;
    /* $y_{e\langle 1,l \rangle} = y_{\langle k_p,l \rangle}$, $\tau_{y\langle 1,l \rangle} = t_{y\langle k_p,l \rangle}$                                  */
    Increment $k_p$ and $k_e$ once;
    while $k_p < n_p$ do
        Establish the $k_e$th episode as containing all values from the $k_p$th
        partition;
        while $s_{\langle k_p,l \rangle} = s_{\langle (k_p+1),l \rangle}$ do
            increment $k_p$;
        end
        Replace final values in this episode with the final values of the current
        interval;
        /* $y_{e\langle k_e,l \rangle} = y_{\langle k_p,l \rangle}$, $\tau_{y\langle k_e,l \rangle} = t_{y\langle k_p,l \rangle}$                            */
        Increment $k_p$ and $k_e$ once;
    end
end
```

Algorithm 1: Concatenation of partitions to episodes

Functions 1.1.5 and 1.1.6 - Episode formation Let k_p be the index of the partition, l the variable number, and k_e the number of the episode formed. Algorithm 1 shows the algorithm for the concatenation of partitions into episodes.

For each variable, those partitions which have the same qualitative state can be concatenated to form episodes. An episode's start value is the start value (x) from the first partition in the episode, and the end value of the episode is the end value (y) of the last interval.

The episodes are stored in a different manner to the partition data. The matrix of episodes, known as the *profile*, for each variable is labelled $\underline{\mathbf{P}}_l$ and has the following format:

$$\underline{\mathbf{P}}_l = \begin{bmatrix} \sigma_{\langle 1,l \rangle} & \tau_{x\langle 1,l \rangle} & \tau_{y\langle 1,l \rangle} & x_{e\langle 1,l \rangle} & y_{e\langle 1,l \rangle} \\ \vdots & & & & \vdots \\ \sigma_{\langle n_e,l \rangle} & \tau_{x\langle n_e,l \rangle} & \tau_{y\langle n_e,l \rangle} & x_{e\langle n_e,l \rangle} & y_{e\langle n_e,l \rangle} \end{bmatrix} \tag{4.20}$$

Function 1.2 - Rule base generation

The output of function 1.1 is a set of episode strings of varying lengths and with different qualitative representations. Each episode has associated quantities, a serial number (i.e. the kth episode in the profile), a start time and an end time.

These properties can be used to create links between profiles generated from different fault conditions. It is fair to assume that even if the qualitative representations for two faults, say f_1 and f_2, are quite different, there will still be common episodes which can be seen in both profiles, because fault simulations are to be carried out such that the actuator is *on the point* of failure but that the throw still reaches completion. This means that all stages of the actuator's throw will be present in all fault profiles, but that the dynamics will probably be different. The common episodes will, however, have diverse associated quantities, and these can be used to diagnose the fault condition.

Function 1.2.1 - Finding common episodes

Profiles cannot be compared directly together because it is not certain that all episodes in the sequence will be the same, or that there will be a fixed number of episodes for all possible faults. Therefore, those episodes which are not common to *all fault conditions* must first be discarded. These unwanted episodes can be unusual behaviours or they can be spurious small sequences induced by variations in the variables. The spurious sequences are not generally repeatable within a simulation and therefore must be discounted. An example of a spurious sequence is shown in figure 4.15.

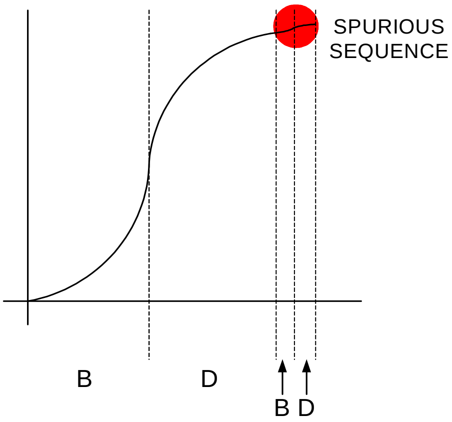

Figure 4.15: Illustration of spurious episode sequences

An episode shall be deemed to be spurious and ignored if its length or height are less than preset limits. This ensures that only episodes which represent major trends in the waveform are actually considered for matching. The limits are set separately for each possible episode value, each fault condition and each measured variable. This allows adjustments to take account of varying dynamic performances for different conditions.

It should be noticed that this method will also label as spurious all the turning point episodes ($\mathcal{I}$ and $\mathcal{J}$) in the profiles, because they are usually small in comparison to the length threshold and therefore little variation can be seen in any of the values.

The $\underline{\mathbf{P}}$ matrices will be shortened following this process, which will be carried out for each variable, each fault condition and each repetition.

For each fault, each repetition's qualitative data should, in theory, look identical. In practice, however, it is unavoidable that there will be some variations, especially if a fault significantly changes the qualitative performance of the actuator.

Algorithm 2 is used to isolate a set of matching episodes which can be found in all waveforms for all fault conditions, yet still be used to determine which condition is most likely. The scaling factor g is used to normalise the scales of units of time and the measured variable, for the magnitude calculation to be meaningful.

```
Set global counters to $\infty$ for each qualitative state;
for each profile do
    Separate the episodes according to qualitative state;
    for each possible qualitative state do
        Order the episodes of this qualitative state according to their
        magnitude, defined as $\sqrt{g^2 \left(\tau_y - \tau_x\right)^2 + \left(y - x\right)^2}$;
        Count the number of episodes of this qualitative state in this profile;
        if the count is less than the global counter for this qualitative state
        then
            Set the global counter to the value of the local count;
        end
    end
end
/* Minimum numbers of episodes for each qualitative state have
   now been found                                              */
for each profile do
    for each qualitative state do
        if local count = minimum count then
            Assume all episodes match directly as common episodes;
        end
        else if local count > minimum count then
            Find matches using least-squares approximation;
        end
        /* The number of matches will be equal to the minimum
           count for this QS                                   */
    end
end
```

Algorithm 2: Episode matching

Let the number of common episodes obtained be represented by n_c.

Resulting $\underline{\mathbf{P}}$ matrices will have the form

$$\underline{\mathbf{P}}_l = \begin{bmatrix} \sigma_{\langle 1,l \rangle} & \tau_{x\langle 1,l \rangle} & \tau_{y\langle 1,l \rangle} & x_{e\langle 1,l \rangle} & y_{e\langle 1,l \rangle} \\ \vdots & & & & \vdots \\ \sigma_{\langle n_c,l \rangle} & \tau_{x\langle n_c,l \rangle} & \tau_{y\langle n_c,l \rangle} & x_{e\langle n_c,l \rangle} & y_{e\langle n_c,l \rangle} \end{bmatrix} \tag{4.21}$$

Function 1.2.2 - Relative quantity calculation

In order to make the rules fully transferable from one actuator to another (assuming that the changes in behaviour are proportional from one actuator to another), the episode values are expressed in terms of the difference between the current measured value and the average value for fault-free behaviour. Thus, for each episode, each repetition (with R repetitions for each fault) and each fault condition, the following calculations are made:

For $k = 1$ to n_c, $l = 1$ to n_v

$$\overline{\underline{\mathbf{x}}}_{f0\langle k,l\rangle} = \frac{1}{R}\sum_{j=1}^{R} \underline{\mathbf{x}}_{f0\langle j,k,l\rangle} \tag{4.22}$$

$$\overline{\underline{\mathbf{y}}}_{f0\langle k,l\rangle} = \frac{1}{R}\sum_{j=1}^{R} \underline{\mathbf{y}}_{f0\langle j,k,l\rangle} \tag{4.23}$$

$$\overline{\underline{\mathcal{T}}}_{x\ f0\langle k,l\rangle} = \frac{1}{R}\sum_{j=1}^{R} \underline{\mathcal{T}}_{x\ f0\langle j,k,l\rangle} \tag{4.24}$$

$$\overline{\underline{\mathcal{T}}}_{y\ f0\langle k,l\rangle} = \frac{1}{R}\sum_{j=1}^{R} \underline{\mathcal{T}}_{y\ f0\langle j,k,l\rangle} \tag{4.25}$$

For $i = 2$ to $n_F - 1$, $j = 1$ to R, $k = 1$ to n_C, $l = 1$ to n_V

$$\underline{\mathbf{x}}_{rel\ (f_i)} = \underline{\mathbf{x}}_{f_i} - \overline{\underline{\mathbf{x}}}_{f0\langle k,l\rangle} \tag{4.26}$$

$$\underline{\mathbf{y}}_{rel\ (f_i)} = \underline{\mathbf{y}}_{f_i} - \overline{\underline{\mathbf{y}}}_{f0\langle k,l\rangle} \tag{4.27}$$

$$\underline{\mathcal{T}}_{x\ rel\ (f_i)} = \underline{\mathcal{T}}_{x\ f_i} - \overline{\underline{\mathcal{T}}}_{x\ f0\langle k,l\rangle} \tag{4.28}$$

$$\underline{\mathcal{T}}_{y\ rel\ (f_i)} = \underline{\mathcal{T}}_{y\ f_i} - \overline{\underline{\mathcal{T}}}_{y\ f0\langle k,l\rangle} \tag{4.29}$$

Each common relative episode value is given a usage flag which is initially set to 1, which indicates that it shall be used to form rules.

Functions 1.2.3 and 1.2.4 - Optimisation

The two optimisation functions allow the system to avoid making inefficient rules by checking common episode values against criteria and clearing the usage flag if the value fails the test. Function 1.2.3 uses algorithm 3, which eliminates episode values which do not noticeably deviate from the fault-free case.

```
for each common episode value do
    Compare the magnitude of the relative value with the deviation threshold;
    if relative value does not exceed the threshold then
        clear the usage flag;
    end
end
```

Algorithm 3: Exclusion of values which do not vary appreciably from the fault-free case

```
for each common episode value with usage flag set do
    for each fault condition $f_i$ do
        for each fault condition $f_{i2}$ except $f_i$ do
            Find $v_{f_i} - v_{f_{i2}}$ where $v$ is the current episode value;
            If it is less than the fault distinction threshold then clear the usage flag;
        end
    end
end
```

Algorithm 4: Exclusion of values which are too similar to those from other fault conditions

Function 1.2.4 uses algorithm 4. This function eliminates the values which are too close to those of other faults, thereby increasing the efficiency of the rules which are left to be formed. The threshold is defined separately for each variable and each value type.

Function 1.2.5 - Membership functions

The peak of a fault's membership function should be set to occur where the difference value corresponds to the measurements made at maximum fault severity. Once a $\underline{\mathbf{P}}$ matrix of the form in equation 4.21 has been obtained for each measured variable and each fault, the rules can be generated. The rules are expressed as parameters for fuzzy membership functions.

Variation *away* from normal behaviour will result in a rapid decrease in membership of the fault free set. Variation between normal and the fault will result in a gradual change in membership. All faults will be simulated up to the the very point of failure. Therefore, variation of any one variable beyond that measured during simulation of fault $f_1(100)$ is unlikely to be a result of f_1, because the actuator would have failed. It is possible that another fault f_2 could cause a further increase in a certain value, beyond that seen when the actuator is failing with fault f_1. Figure 4.16 illustrates this concept, showing how membership functions for different faults can overlap for different sizes of variation.

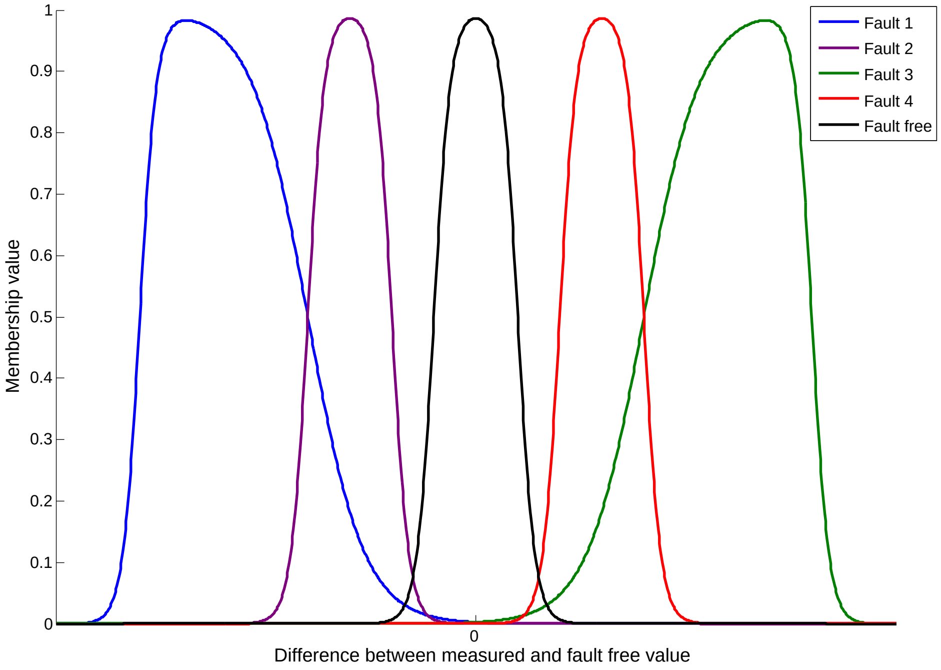

Figure 4.16: Membership functions as rules on one variable in an episode

$$f\left(x; h_1, c_1, h_2, c_2\right) = e^{\frac{-(x-c_1)^2}{2h_1^2}} - e^{\frac{-(x-c_2)^2}{2h_2^2}} \tag{4.30}$$

Gaussian membership functions, as expressed in equation 4.30, were chosen for rule formation because of their smooth transitions and their logical set of parameters which map well to the variable space. The parameters h_1 and h_2 govern the gradient of the left- and right-hand transitions respectively. Smaller values of h result in quicker transitions. The c_1 and c_2 parameters govern the left- and right-hand points, respectively, at which the function is at its maximum value. Setting $c_1 > c_2$ results in the maximum value of the membership function being less than 1. Conversely, setting $c_1 < c_2$ results in an interval in the variable, of size $c_2 - c_1$, over which the membership is continuously 1. The system is programmed to set $c_1 = c_2$, so that the membership is only ever briefly 1 before it begins the decay transition. This is important because it recognises the possibility that an increase or decrease in a measured value may be a result of more than one possible condition.

The rising transition from the fault free point to the faulty difference is referred to as the attack transition; the falling transition which occurs if the difference moves further in the same direction is referred to as the decay transition. A standard value of h was chosen by experimentation for the decay transitions. The attack transitions' h values were set proportional to the difference, so that gradual change between fault free and faulty results in a transition from 0 to 1. The proportional constant was chosen by experimentation. Figure 4.17 illustrates some typical membership functions.

The system is designed to deal correctly with all possible fault difference values, positive and negative. It must therefore assign the values of h_1, h_2, c_1 and c_2 differently, depending on the sign of the difference.

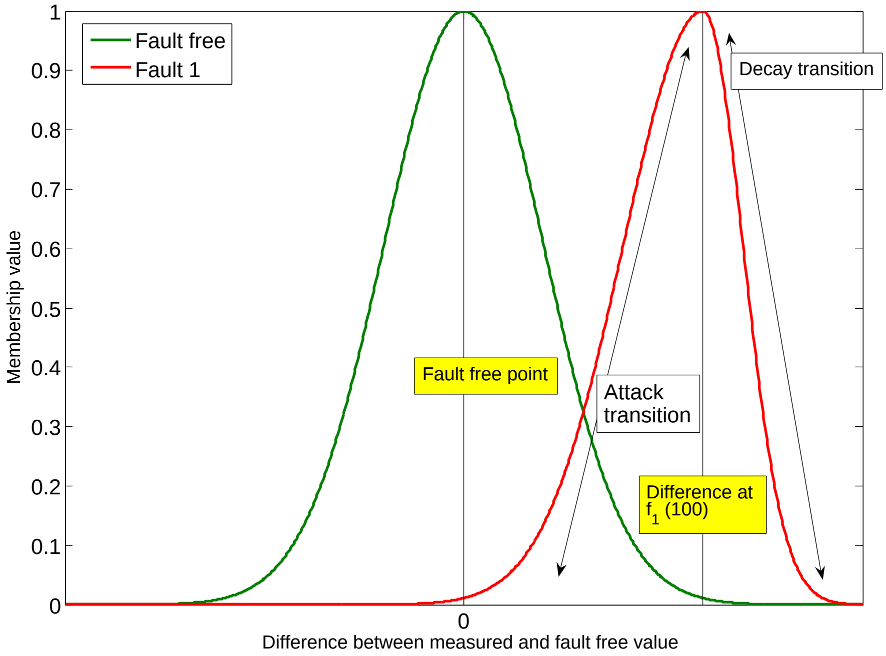

Figure 4.17: Establishing membership functions for faults

4.2.3 Function 2 - Finding base quantities for fault free operation

Function 2.1 - Representation of measured data

The same process is carried out on measured data during the running in period as was carried out on the simulated fault data in function 1. The running in data should be completely free of faults. Profiles are generated for each running-in operation. The common episodes are found using the process in function 1.2 and the values associated with them are stored for use in calculating percentage changes from normal behaviour.

The STME on which the system is to be installed will be checked to ensure no faults are present, and then R throws shall be carried out in each direction, to measure fault free data. For each repetition, a profile $\underline{\mathbf{P}}$ is obtained in exactly the same way as in function 1.1.

The $\underline{\mathbf{P}}$ matrix is then processed so that it only contains the *common episodes*, resulting in a matrix of the form in equation 4.21.

The means of episode values in each $\underline{\mathbf{P}}$ matrix are collected as a single set of values $\overline{\underline{\mathbf{P}}}$:

$$\overline{\underline{\mathbf{P}}}_l = \begin{bmatrix} \sigma_{\langle 1,l \rangle} & \frac{\sum_{j=1}^{R} \tau_{x\langle 1,l \rangle}}{R} & \frac{\sum_{j=1}^{R} \tau_{y\langle 1,l \rangle}}{R} & \frac{\sum_{j=1}^{R} x_{e\langle 1,l \rangle}}{R} & \frac{\sum_{j=1}^{R} y_{e\langle 1,l \rangle}}{R} \\ \vdots & & & & \vdots \\ \sigma_{\langle n_c,l \rangle} & \frac{\sum_{j=1}^{R} \tau_{x\langle n_c,l \rangle}}{R} & \frac{\sum_{j=1}^{R} \tau_{y\langle n_c,l \rangle}}{R} & \frac{\sum_{j=1}^{R} x_{e\langle n_c,l \rangle}}{R} & \frac{\sum_{j=1}^{R} y_{e\langle n_c,l \rangle}}{R} \end{bmatrix} \tag{4.31}$$

A matrix such as this shall be generated for each variable measured. These matrices are then the benchmark against which operationally measured values will be compared.

4.2.4 Function 3 - On-line diagnosis of faults

Function 3.1 - Profile representation

The process for representing operationally measured waveforms is almost identical to that in function 1.1, except that the quantities are expressed as the difference between the absolute, measured value and the fault free values obtained in function 2.2.

The input for each operation is a single matrix of measured values $\underline{\mathbf{M}}$ which is represented, as with previous functions, as a set of profiles $\underline{\mathbf{P}}_l$, one for each variable measured.

The profiles are then processed, using the common episode locations from function 1.2 to leave an edited matrix containing only the common episodes, as seen in equation 4.21. The episode values in $\underline{\mathbf{P}}$ are converted to differences by subtracting the average fault free quantities, just as in function 1.2.2.

Function 3.2 - Calculation of fault scores for the current operation

The next step is to calculate fault memberships for each possible fault, according to algorithm 5.

```
for each possible fault do
    for each variable do
        for each common episode value do
            if usage flag is set then
                Evaluate the membership function and add to the total for this
                fault;
                Increment the number of rules evaluated for this fault;
            end
        end
        Divide total of memberships by number of rules evaluated;
    end
end
```

Algorithm 5: Diagnosis of faults through rule evaluation

This routine occurs every time the actuator operates. Over time, the average membership for a fault will increase, if that fault is gradually becoming more severe.

4.2.5 Function 4 - Trend analysis and alarms

The objective of the system is to attract attention to the detected *development* of fault conditions. The system represents this development as an increase in the strength of the fault from operation to operation.

Previous fault diagnosis systems have suffered from poor reliability because the alarm levels have been set arbitrarily. It is more appropriate to set these levels based on experience from trial installations. Therefore function 4 focuses on detecting positive trends in the fault strengths and presenting them for human examination.

For each fault simulated in a case study, a row of graphs is produced, showing the fault score against the severity of the fault (each data point being a single throw where the fault is incrementally increased).

A linear trend is imposed on the graph. This trend is calculated using a least-squares method. The gradient of this trend gives an indication of how fast or strongly the fault is developing. Therefore, the fault with the highest trend gradient is the one whose effects are appearing most strongly in the measured data.

The fault strength is the total of the membership functions for each variable and each episode in a profile constructed from the measurements of one operation.

Figure 4.18 shows a sample of ideal output from function 4. The rows represent the different conditions imposed on the actuator, and the columns represent the fault score for each possible fault. When the actuator is fault-free, the fault score is high for the fault-free condition and low for all the other possible faults, and none of the scores changes appreciably. The x-axis for this condition represents the number of operations in the fault-free data set. For the imposed faults, it represents the incremental adjustments which simulate the introduction of a fault. Here, a gradual increase can be seen in the score corresponding to the simulated fault, a decrease in the fault-free score (showing that the actuator's condition is moving outside normal boundaries), and no change in the score for other possible faults.

This ideal situation is not expected to be seen in the testing phase, because the rule base is not optimised in this system. Optimisation of the rule base means deleting some rules, but this has not been included as part of the system because it was considered more important for the rules to consider all available parts of the waveform at all times. Therefore, it is likely that if two faults exhibit some similar characteristics, their scores may both increase when one of the faults is gradually introduced. However, the score for the fault being simulated should have the highest gradient on the graph.

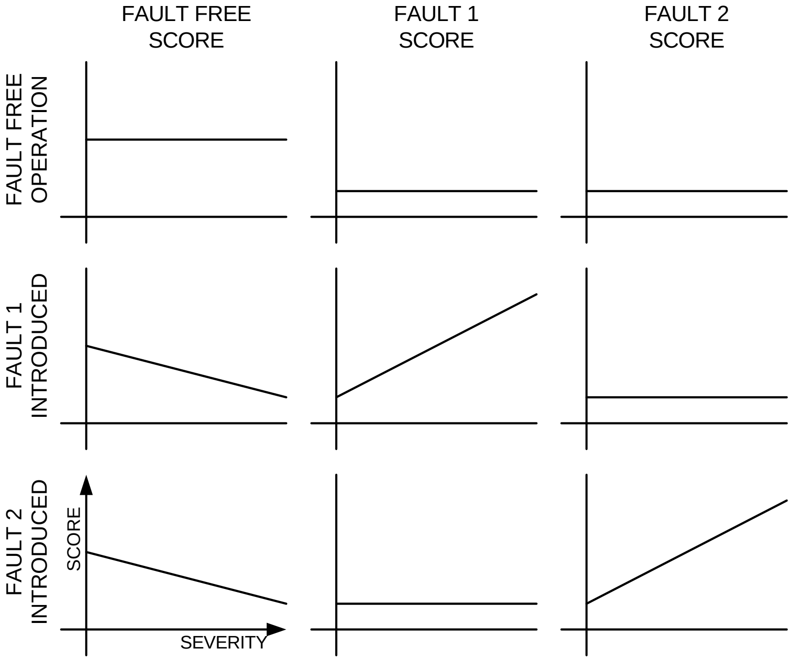

Figure 4.18: Ideal sample output of function 4

Chapter 5

Case study I: External doors on class 158 DMU

5.1 Background

Classes 158 and 159 are inter-regional/inter-city DMUs with a single saloon and vestibules at the ends of each vehicle. The external doors are double-leaf swing-plug doors driven by a single Tebel Pneumatik air actuator and a mechanical linkage with over-centre locking bars for each leaf. Figure 5.1 shows a plan view of the door mechanism when locked, unlocked but still closed, and open. The actuator (1, the dashed area) opens and closes the door using a rotary action about the pivot (2). This moves the locking rods (4), which are attached to pillars (6) that then rotate. Drive arms link the pillars to the door leaves; thus when the pillar rotates, the door leaves swing outwards.

The principle of the locking mechanism is that by driving the mechanism a few degrees past the point where the doors are shut, it becomes impossible for the door to be opened by force against the leaves (since pushing outwards on the door leaves now just increases the strength of the lock). The lengths of the over-centre locking

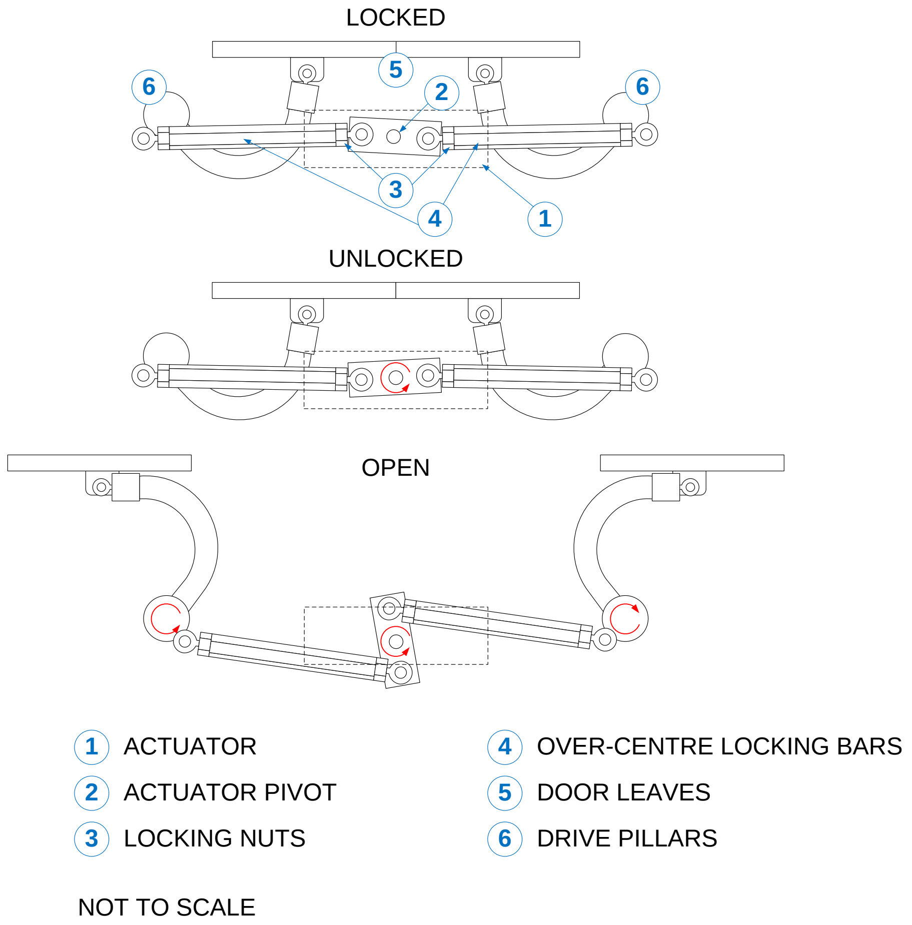

Figure 5.1: Class 158/159 door mechanism, plan view, not all parts shown

bars are adjustable by maintainers, to account for the variations between each door's performance. Lengthening the bars increases the force required to lock the door; shortening them decreases the force required. In either case, if the bar length is outside acceptable boundaries, the door may fail to lock. The length of the bar is set according to the air pressure in the actuator at the instant of locking. This is measured using a dial gauge during a closing throw where the air has been bled from the actuator and then reapplied. The requirement is that the locking pressure must be between 4.0 and 4.5 bar. In practice, the doors are set towards the high end of this range, since the pressure tends to drop over time.

Two failure modes were simulated by adjusting the locking bar to be either too long or too short. By starting from a point where the doors are correctly adjusted, the progression of an incipient fault can be simulated.

Detailed failure mode listings were not available for this door, but an informal discussion with the specialist engineer who facilitated the data collection revealed that the over-centre locking bars shorten naturally during operation, eventually bringing the locking pressure below the threshold. This failure mode therefore reflects directly in the simulation where the bar was shortened; conversely, the bar lengthening was not considered to be a natural failure mode but rather a fault which would cause effects that the diagnosis system should be able to spot.

5.1.1 Setup of experiments and controls

The experiments were carried out on vehicle 52744, in unit 158883, stabled on an outdoor road at Salisbury DMUD[1]. The engine was running so that the on-board air supply could be used. It should be noted that the main reservoir air pressure fluctuated slightly throughout testing because of the way the air compressor is designed to switch in and out, so the starting values for the pressure waveforms are of limited significance: the actuator operates satisfactorily for all values of incoming air pressure experienced during this data collection session.

The general layout of the experiment is shown in figure 5.2. A PC and data acquisition unit were used to record the data. An air pressure transducer was installed on each side of the actuator, and a displacement transducer was added to measure the rotational displacement of one drive pillar.

Triggering measurement using the control buttons

Using opto-isolators, the push buttons which control the door were interfaced to the data acquisition system, allowing measurement to be triggered when the buttons were pressed. This meant that all data files were synchronised to the start of the door operation, allowing for better comparison between captured files.

[1] Diesel Multiple Unit Depot

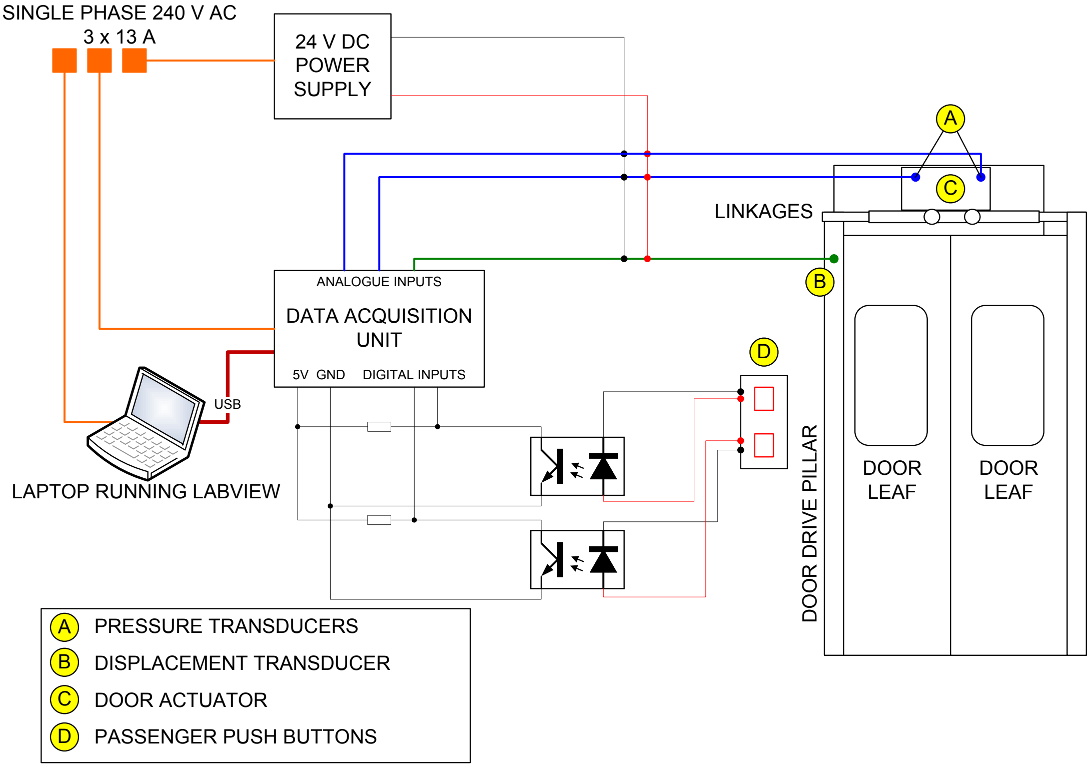

Figure 5.2: Setup of data measurement equipment on the door unit

The open and close push buttons are both normally-open and provide a pulse of +24 V when pressed. The measurement system was configured to wait for a high pulse on the digital output of the opto-isolator before starting measurement.

Displacement transducer

A drawstring displacement transducer was used to measure the door's displacement between the two stable positions. This sensor has a maximum drawstring length of 25 cm and therefore could not be attached to any part of the door leaf, because it moves through a greater distance than 25 cm. The transducer was positioned so that some of the string could be wrapped around one of the vertical drive pillars. This measured the rotational displacement of the drive system, which can be expressed as a percentage of the total measurable displacement.

Air pressure transducers

An air pressure transducer was fitted in line with the incoming air pipes on either side of the actuator using brass pipe fittings and spare hose. The hoses were kept as short as possible to minimise the trapped air in the system, as this would affect the accuracy of the measurements.

5.1.2 General format of experiments

Each door was first adjusted so that the over-centre locking pressure was around 4.5 bar. Control measurements were then taken. Each door was operated ten times in each direction.

5.1.3 Lengthening of the over-centre locking rod

Starting from settings which gave an over-centre locking pressure of around 4.5 bar, each door was gradually adjusted to make the over-centre locking rod longer. The severity of the fault was measured by taking the over-centre locking pressure after each adjustment. Pressures of up to 7.5 bar (the full pressure of the incoming air supply) were measured at this point.

The effect of this fault is to increase the force required to lock the door. The graphs in figure 5.3 show the results of the experiment on one of the doors. The top row shows graphs of the three variables during opening, and the bottom row shows them during closing. The green waveforms are from the starting condition, becoming more red as the length of the bar was increased.

In the opening waveforms, the instant of unlocking becomes later as the length of the bar is increased. This can be seen around point 1, as the unlocking characteristics occur later in each successive waveform. Note that the incoming air pressure varies over a small range; this is not related to the fault simulations but rather a characteristic of the air system.

The symptoms of this fault are much less visible in the closing graphs, because the locking dynamics are smaller in the variables measured. Point 2 on the closing graphs is where the lock engages. The dynamics of locking become later as the length of the bar is increased, just as with opening.

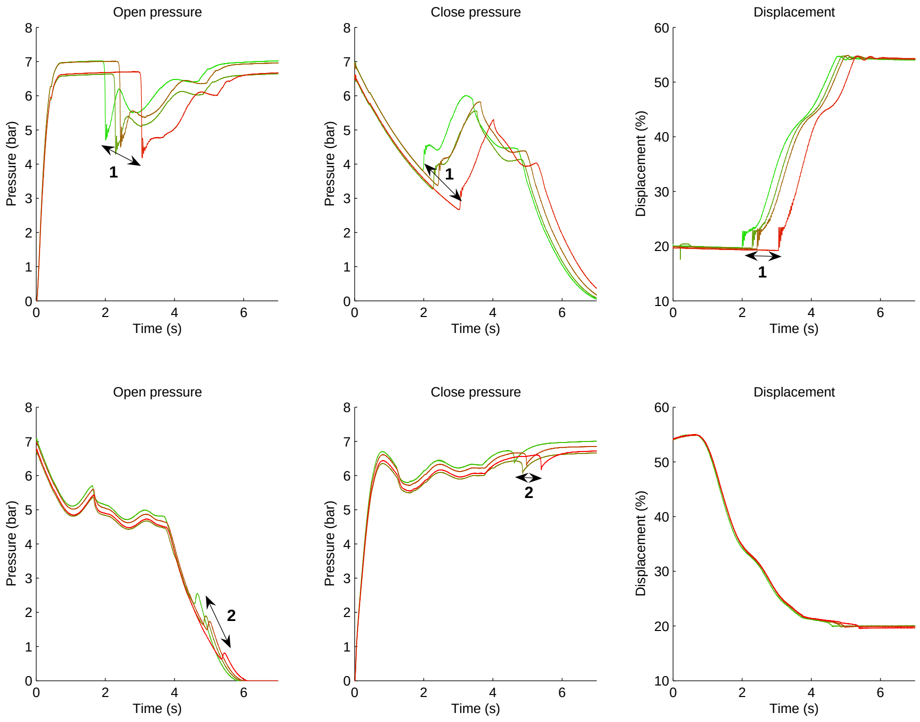

Figure 5.3: Lengthening the over-centre locking rod on ‘A’ door

5.1.4 Shortening of the over-centre locking rod

Starting from settings which gave an over-centre locking pressure of around 4.5 bar, each door was gradually adjusted to make the over-centre locking rod shorter. The severity of the fault was measured by taking the over-centre locking pressure after each adjustment. Pressures of as low as 1.6 bar were recorded.

Figure 5.4 shows the graphs for this experiment, again with the opening graphs on the top row and the closing graphs on the bottom. As the bar is progressively shortened, the instant of unlocking becomes earlier, as can be seen at point 1 in the opening graphs. The dynamics of unlocking occur earlier and become less pronounced as the bar is shortened, which is expected given the way the door operates. Because of the earlier unlocking, there is a general trend of earlier movement throughout the opening operation: the door reaches its open position earlier, as can be seen in all three measurements, but most pointedly on the displacement waveform at point 2.

The locking dynamics during closing are again quite small, but there is still a general trend at point 3, for the locking to occur earlier and with less pronounced dynamics as the bar is shortened. Note also from the displacement profile that the doors seem to settle in a slightly less closed position when the bar is shortened - presumably because the lock is looser.

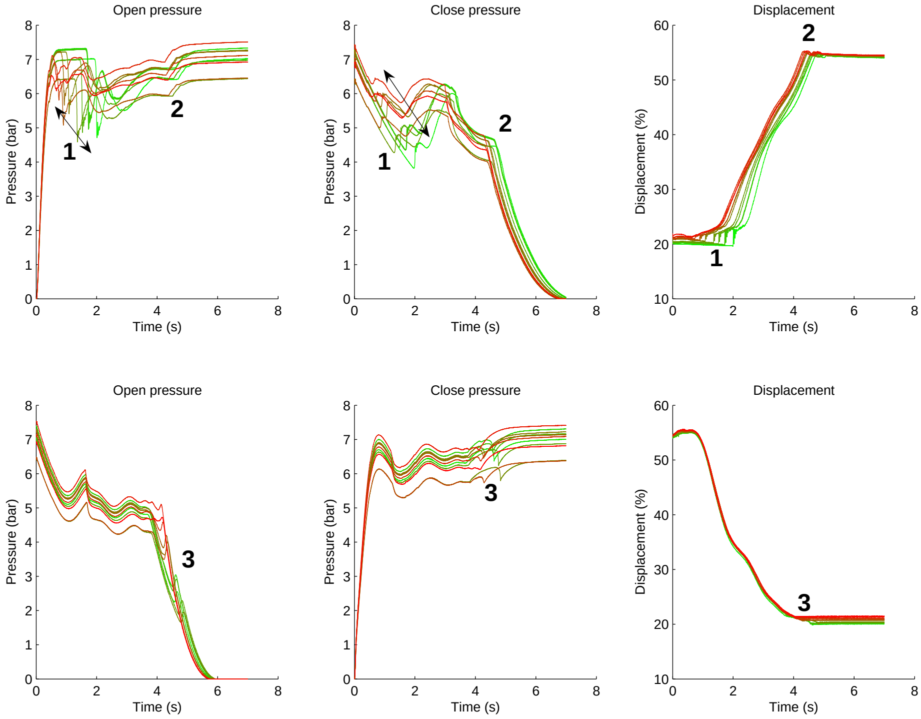

Figure 5.4: Shortening the over-centre locking rod on ‘A’ door

Each of the four doors underwent the same experiments, and, qualitatively speaking, the effects of the faults were the same in each case. This means that the data collected are the right kind to test the system developed in this thesis.

5.1.5 Variations in performance

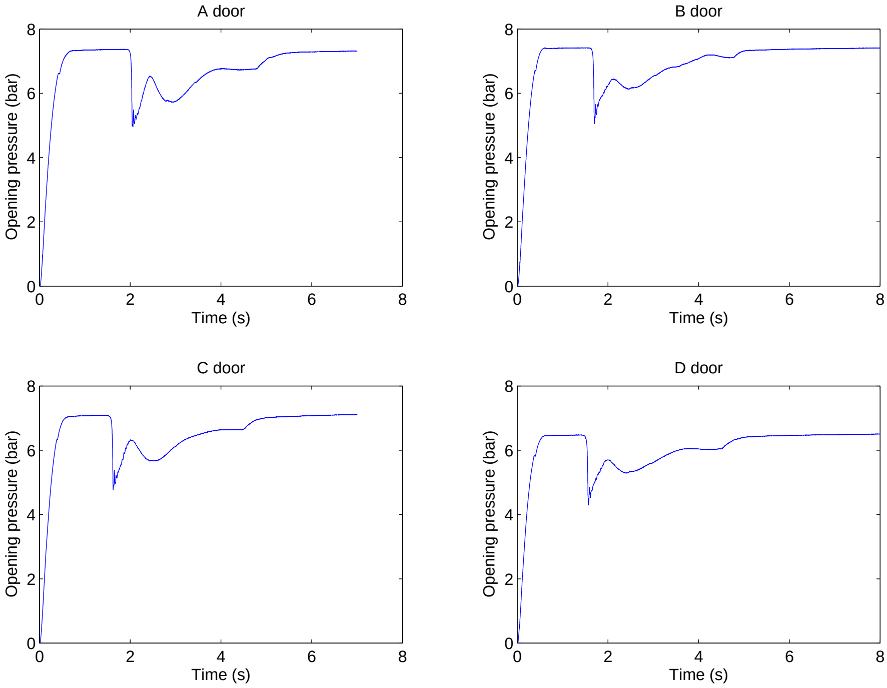

Figure 5.5: Comparison of fault-free operation for the four actuators under test

Figure 5.5 shows graphs of the pressure on the opening side of the actuator, during opening operations, from each of the four doors tested. The figures show that, broadly speaking, the performance was qualitatively the same in each case, although B door showed a slightly different shape around the 3-4 second area from the others. The variations in peak air pressure are due to the natural range of pressures available from the air supply on board the train; the air compressor did not operate continuously, but rather when pressure reached a lower limit.

Quantitatively, there was little variation between the four doors; perhaps most significant was the dip around the 1.8-2 second mark, which corresponds to the unlocking point: the time at which this occurred seemed to vary from door to door. It was to be expected that there would be few quantitative differences, since the actuator was moving more or less the same load in each case; the only differences would have been in the adjustable parts of the mechanism - some of which were not altered during the course of testing, such as the door height relative to the vehicle body.

5.2 Test results

The data set from case study I comprises complete fault simulations for four doors. There are 16 possible combinations of Test Actuator and Monitored Actuator. Using the MATLAB implementation of the fault diagnosis system, testing was carried out on all 16 possible actuator combinations. Disregarding those combinations where the TA and MA are the same, there are 12 combinations which test whether the rules can be transferred from one actuator to another.

5.2.1 Diagnosis system in action

Figure 5.6 shows waveforms from the fault free condition and the shortened over-centre locking rod simulation, before and after filtering and the identification of the profile. The stem plots on the filtered waveforms show start (circular markers) and end (diamond markers) values associated with the episodes in the profile. Episodes which were shorter than the minimum length threshold were rejected, which is why the whole of each waveform is not covered by an episode.

Comparing the episodes identified for the two waveforms, we can see that, in this case, there is poor matching between the two profiles. The faulty profile has only five episodes, yet the fault-free one has seven. Of these, only three can be said to be reasonably comparable: the $\mathcal{B}$ and $\mathcal{D}$ episodes at the start of the waveform, and

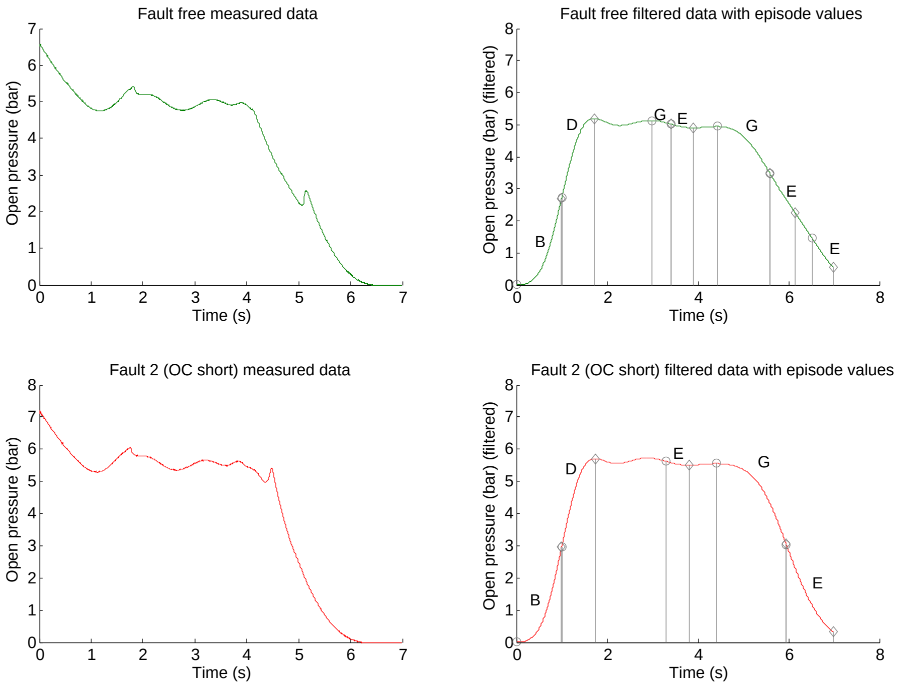

Figure 5.6: Episode identification in open pressure waveforms from closing operations, C door, with fault-free and OC long conditions

the $\mathcal{G}$ episode at the end.

This graphic demonstrates that there can be difficulties in identifying common episodes for every waveform, but there is usually a core of qualitative information which stays the same and can be used for rule formation. However, in this fault condition, the only significant change to the waveform occurs in an area where the system is not consistent in classifying episodes, making it difficult for the system to diagnose the fault from this data. These waveforms were from a closing operation; those from the opening operation show much more pronounced effects which were easier for the system to spot, so the fault was detected in the opening data.

5.2.2 System outputs

Tables 5.1 and 5.2 show the number of combinations which fulfilled each test specification, for each condition simulated. Since the fault free condition was not changing under adjustment, gradients of membership trends were irrelevant, so instead the average membership values were compared in the same way.

Spec	Condition	OK combinations	% effectiveness
1	Fault free	12	100
	OC Long	12	100
	OC Short	12	100
2	Fault free	12	100
	OC Long	11	91.7
	OC Short	11	91.7

Table 5.1: Test results for opening movements on the pneumatic door

Spec	Condition	OK combinations	% effectiveness
1	Fault free	12	100
	OC Long	9	75.0
	OC Short	8	66.7
2	Fault free	12	100
	OC Long	6	50.0
	OC Short	9	75.0

Table 5.2: Test results for closing movements on the pneumatic door

The performance of the system clearly shows that, in the majority of cases, it is capable of detecting the introduction of a fault, and of determining which fault is present. The performance of the system is markedly worse when working on data from closing movements. This can be explained by the relative magnitude of the fault effects. In figures 5.3 and 5.4, it can clearly be seen that the introduction of the faults causes much larger variations in the opening waveforms than in the closing waveforms. This means that the system has useful information to make rules with, especially when the effects of heavy filtering of the waveforms may be to eliminate evidence of the one small kink in the pressure waveforms whose movement betrays the presence of faults.

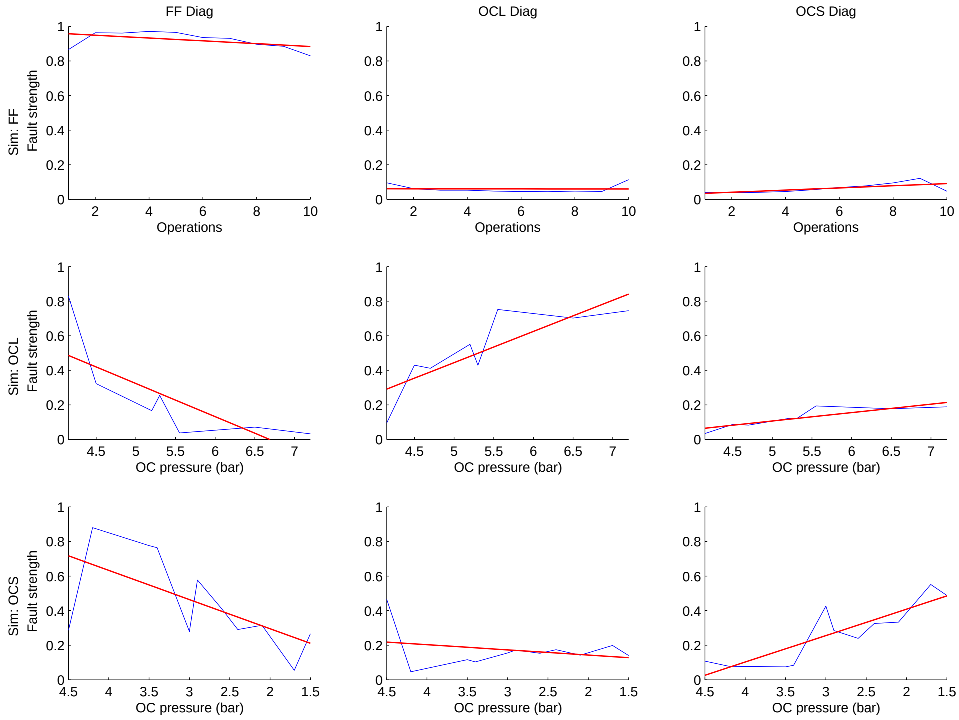

Figure 5.7: Results graphs for opening operations, A door as test actuator and C as monitored actuator

Figure 5.7 shows the results of running the system for a single actuator combination. The blue line is the fault strength at each point simulated; the red line is a first-order linear trend for the fault strength, calculated using the standard MATLAB `polyfit` function, which uses least-squares approximation. The rules were established using the test dataset from actuator A, and the monitoring dataset from actuator C. For the fault-free simulation, it can clearly be seen that the diagnosis of the fault free condition is high for each of the ten operations in the dataset.

When faults 1 and 2 are simulated, the diagnosis for the fault free condition gradually reduces in strength, showing that the actuator is drifting away from a healthy position. The diagnosis for the corresponding fault also increases, with a firm positive trend as indicated by the red line. The diagnosis for the fault which is not being simulated can be seen to stay quite low; even though it rises, it does not

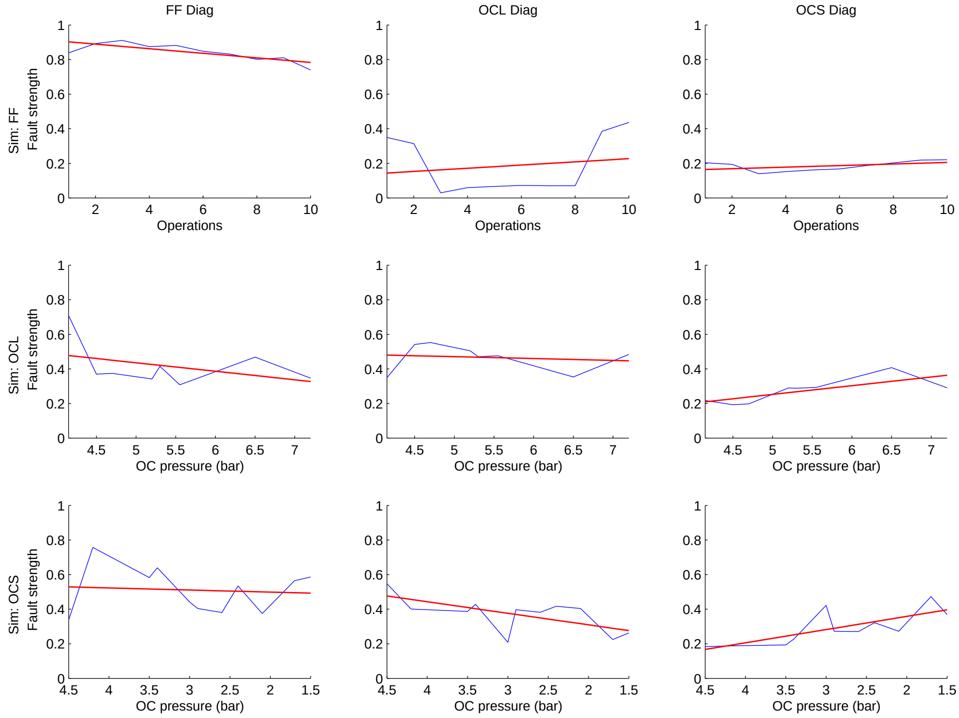

Figure 5.8: Results graphs for closing operations, A door as test actuator and C as monitored actuator

rise as much as the simulated fault, so it diagnoses the fault correctly.

Comparing this with the results from closing operations (seen in figure 5.8) for the same actuator combination, we see again the disparity between opening and closing operations. The system is much weaker, with weaker positive or negative trends for the fault strengths of the simulated fault. This is because the system was unable to construct a strong rule set for the faults, which is understandable, given that the variation in performance under faulty conditions is barely distinct for closing operations.

5.3 Conclusions

In this case study, two distinct and opposite faults were simulated on the door actuators of a rail vehicle. The fault diagnosis system was then run on the data and proved to be 100 % successful in detecting and 91.7 % successful in diagnosing incipient faults from opening operation measurements. Despite the symptoms of the faults being very small in the closing operations, the system was still able to detect the onset of faults in some cases. These successful results show that the concept of this diagnosis system is a valid one.

The difference in success rates between diagnoses from opening and closing operations suggests that the effects of a fault need to be quite noticeable in order for the system to be able to spot them after filtering. One area for future work might be to improve the algorithm for the classification of qualitative states, perhaps to the point where filtering is no longer required.

In particular cases where the system was not effective, there were both false alarms (significant fault trends under fault free conditions) and misdiagnoses. It seems that, under both fault simulations, there was an upward trend for both diagnoses, despite the fact that the faults have diametrically opposite effects on the measured variables (it would therefore be expected that when fault 1 is simulated, fault 2 shows no increase in strength at all).

Overall, this case study is pleasing because the system worked correctly for all combinations of test and monitored actuators. Despite the disappointing closing results, the system would have detected the faults from the opening data, which means that in practice, the fault would have been detected and diagnosed correctly.

Chapter 6

Case study II: HW Switch Actuator

6.1 Background

The HW switch actuator is widely used on railway junctions throughout the UK. It is driven by a 110 V DC permanent magnet motor connected through a reduction gearbox to the drive of the switch. A clutch (which can be either magnetic or mechanical, depending on the age of the machine) is used to disconnect the motor, should the force in the drive exceed a certain tolerance. A sketch of the HW actuator is shown in figure 6.1.

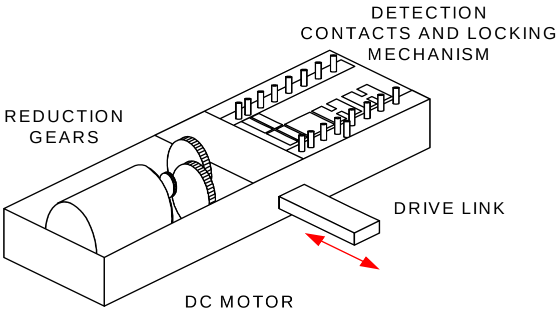

Figure 6.1: Sketch of a HW switch actuator showing main components

A mechanical locking mechanism is situated at the end of the machine, locking the points in place and making electrical contacts for detection. A backdrive is sometimes installed on long switches in order to spread the throw force further up the switch. A diagram of the transmission of forces in a backdrive is shown in figure 6.2.

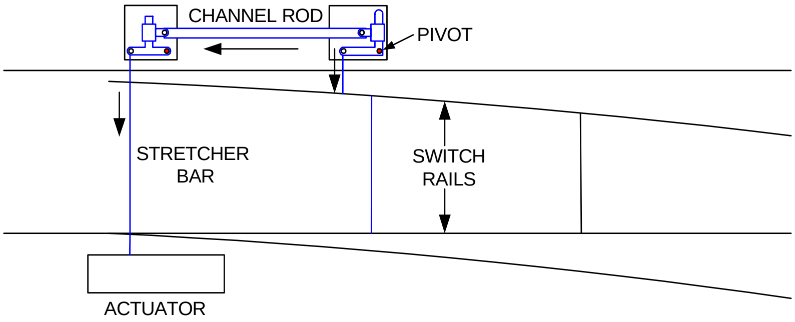

Figure 6.2: Transmission of forces in a backdrive

Although the motors in these machines are rated to 110 V DC, it is common to drive them at a higher voltage. Lineside DC voltages are typically around 120 V DC. This is produced by rectification from AC with smoothing by batteries. However, one installation, which forms part of the test group for this case study, was found to be driven from full-wave rectified mains power with no smoothing. If this mains voltage was not stepped down by a transformer then the rectified average voltage is 153 V DC, which means that the DC motor is being operated approximately 40% out of tolerance.

Despite this, motor faults are relatively uncommon in electric switch actuators. The most common cause of failure is incorrect adjustment of the many adjustable mechanical parts around a switch. If any one of these is misaligned, the forces required to throw the points increase markedly, until they overcome the clutch in the actuator, resulting in a failed throw.

6.2 Experimental setup

An automatic test drive unit was built to automatically operate the actuator and measure data using a data acquisition unit. The force in the drive was measured using a purpose-built load pin which replaced a standard bolt in the drive link. The displacement of the switch was measured using a drawstring sensor and the motor current was measured using a current clamp on the drive wire to the motor. A diagram of the experimental setup components is shown in figure 6.3.

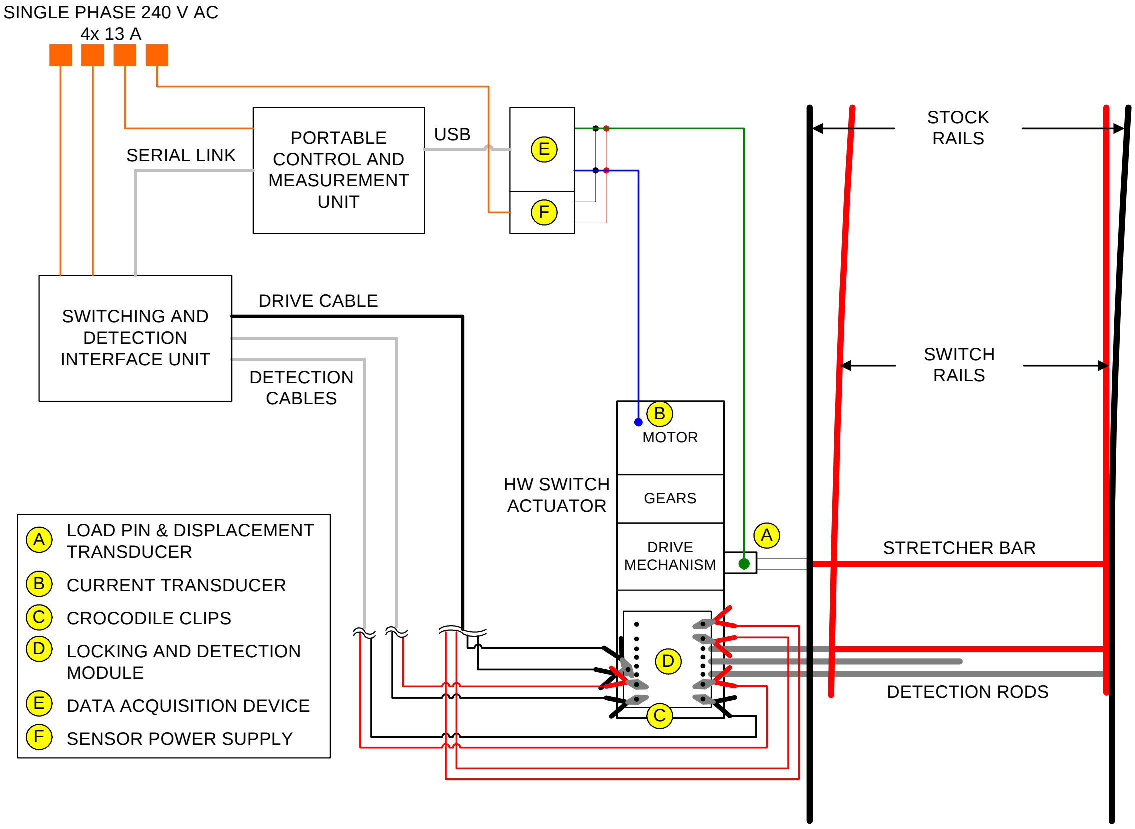

Figure 6.3: Experimental setup for data collection from the HW switch actuator

6.3 Faults simulated

A variety of faults was simulated on several instances of the HW actuator. The focus was on incipient faults, since abrupt faults can usually be detected using current methods. The following subsections describe each fault in detail, along with its simulation.

6.3.1 Misadjustment of the backdrive

The fulcrum points of the backdrive can be adjusted in order to transmit more or less force to the heel of the switch. By adjusting the mechanism, as shown in figure 6.4, it was possible to simulate the backdrive falling out of adjustment in both directions (i.e. overdriving and underdriving).

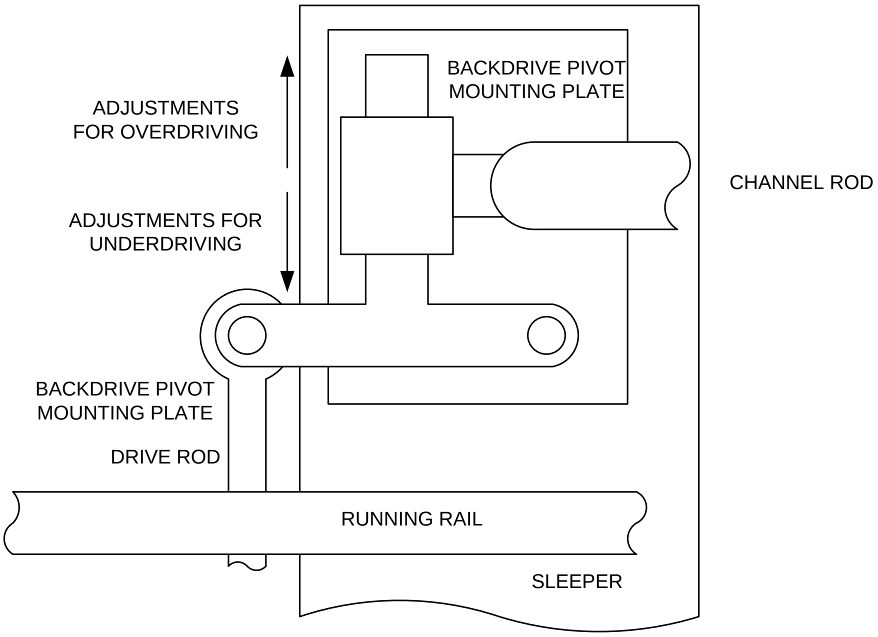
BACKDRIVE PIVOT MOUNTING PLATE
ADJUSTMENTS FOR OVERDRIVING
CHANNEL ROD
ADJUSTMENTS FOR UNDERDRIVING
BACKDRIVE PIVOT MOUNTING PLATE
DRIVE ROD
RUNNING RAIL
SLEEPER

Figure 6.4: Adjustment of force in backdrive

The effects of this fault were quite subtle, but nonetheless visible in some of the output waveforms. Figure 6.5 shows the effects when the backdrive is underdriven: the forces decrease in the drive and therefore less force is required to throw the switch over, resulting in lower end forces for locking. Point 1 shows the decrease in forces for the normal to reverse movement, and point 2 shows the same decrease in the opposite throw direction.

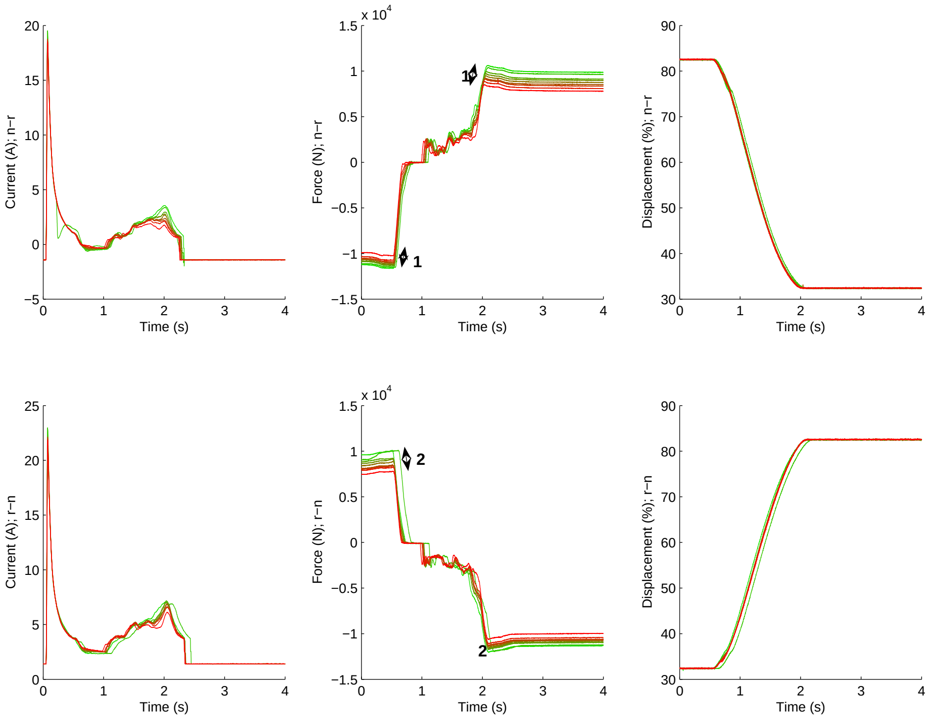

Figure 6.5: Underdriving the backdrive on the switch at Escrick

6.3.2 Overdriving to one side

Misadjustment of the drive link can cause the switch to push too hard to one side before locking takes place. On the HW drive link a pair of nuts can be adjusted so that the actuator pushes further to one side before engaging the lock. This is shown in figure 6.6.

Even small adjustments have a large effect on the forces, so adjustments of the nuts were made in increments of $\frac{1}{6}$ turn. Eventually, at a certain level of adjustment, the force becomes high enough to stall the actuator before locking takes place. This fault was simulated in both directions. Figure 6.7 shows the effects when the switch was overdriven towards reverse. At point 1, we can see evidence in both the force and current waveforms that the actuator is working harder to push the switch over; likewise, at point 2, we see that the force in the switch is higher before unlocking from the reverse position, as the fault worsens.

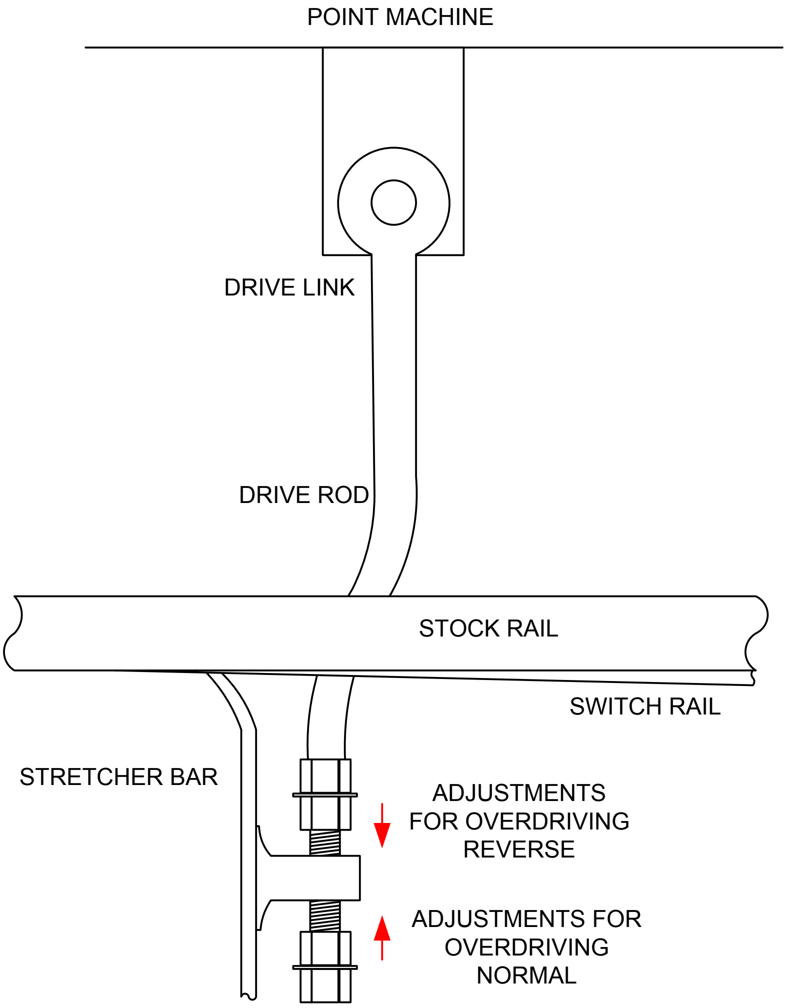
POINT MACHINE
DRIVE LINK
DRIVE ROD
STOCK RAIL
SWITCH RAIL
STRETCHER BAR
ADJUSTMENTS FOR OVERDRIVING REVERSE
ADJUSTMENTS FOR OVERDRIVING NORMAL

Figure 6.6: Adjustment of the HW drive link to overdrive the switch to one side

6.3.3 Relevance of fault simulations to real failure modes

Over a five-year period on the Southern Zone of the UK railway network, the following five faults were the most common recorded causes of failure on HW switch actuators [65]:

- No fault found on examination (17 %)
- Detector rod out of adjustment (9 %)
- Drive rod out of adjustment (9 %)
- Facing point lock out of adjustment (6 %)

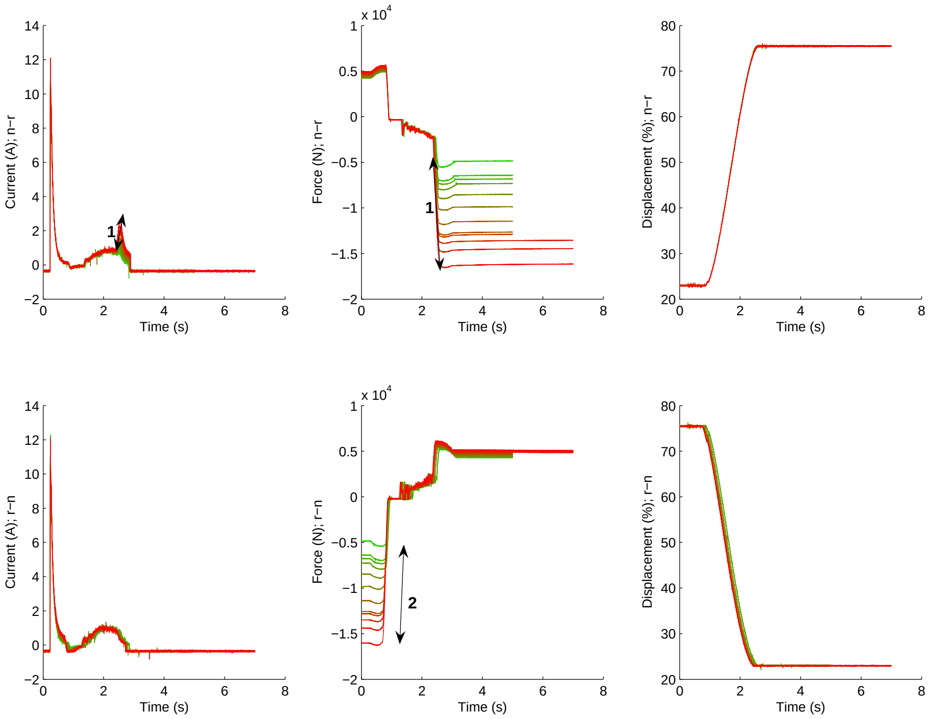

Figure 6.7: Effects of overdriving the switch to reverse, from switch A at Bristol

- Drive rod out of adjustment/gauge (5 %)

The high proportion of failures where diagnosis was not possible indicates that the actuator was suffering intermittent faults which did not present when the maintenance staff attended the actuator. However, of the remaining faults, the two which concern the drive rod adjustment are directly related to the simulations of overdriving towards one side or the other. Given that there was a high number of adjustment faults throughout the top 25 failure modes, it was considered to be useful to simulate poor adjustment of the backdrive, since it is a plausible failure mode and causes effects which are smaller than straight overdriving, adding a further diagnosis challenge for the purposes of testing the system.

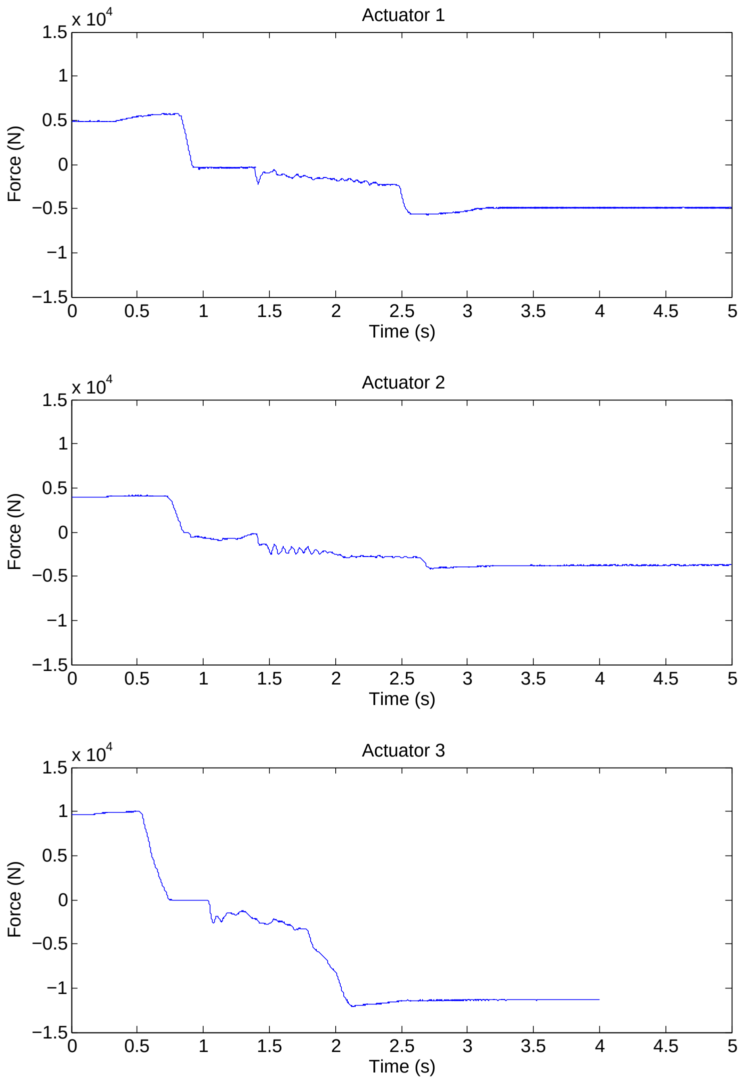

Figure 6.8: Comparison of fault-free normal to reverse operations for each actuator

6.3.4 Variations between actuators

Figure 6.8 shows force waveforms under the same conditions for normal to reverse operation, for each of the three actuators. It is clear that there were significant quantitative differences between the actuators, because each is throwing a different length of rail. The friction characteristics and initial states of the actuators are also likely to have been quite variable, since the actuators were not new. The locking forces were smaller for 1 and 2 than for 3. However, the general shape of the waveforms, and the rough locations of step changes, were similar in each case, allowing qualitative comparison to be performed by the diagnosis system.

6.4 Test results

The data set from case study II comprises simulations of two faults (overdriving towards normal and reverse) on three HW switch actuators.

Two of the three actuators have backdrives, and on these actuators, two additional faults were simulated (overdriving and underdriving of the backdrive). There are 9 possible combinations of Test Actuator and Monitored Actuator. Using the MATLAB implementation of the fault diagnosis system, testing was carried out on all 9 possible actuator combinations. Disregarding those combinations where the TA and MA are the same, there are 6 combinations which test whether the rules can be transferred from one actuator to another, for the overdrive faults, and two combinations which test the same for the backdrive faults.

6.4.1 Diagnosis system in action

Episode identification and matching

Figure 6.9 shows waveforms from the fault free condition and the overdriving towards reverse fault simulation, before and after filtering and the identification of the profile. The stem plots on the filtered waveforms show start (circular markers) and end (diamond markers) values associated with episodes in the profile. Episodes which were shorter than the minimum length were rejected, which explains why there is not complete coverage of the waveform. There is generally good agreement between the two profiles, with most of the episodes common to both.

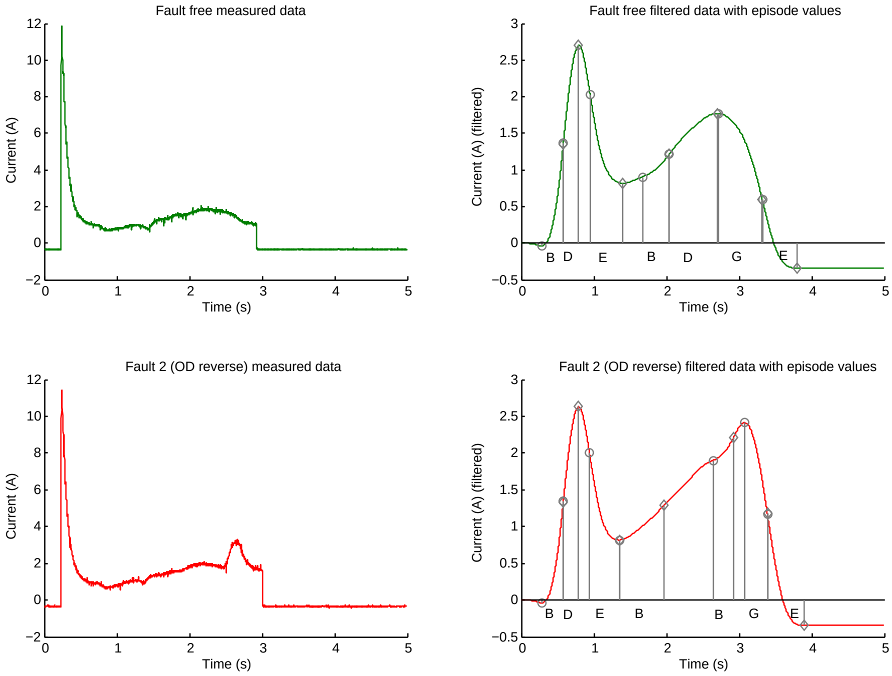

Figure 6.9: Identification of non-trivial episodes in HW current waveforms

σ	Fault free				OD Reverse			
	τ_x	τ_y	x	y	τ_x	τ_y	x	y
$\mathcal{B}$	0.290	0.579	-0.039	1.357	0.290	0.579	-0.039	1.334
$\mathcal{D}$	0.580	0.789	1.367	2.701	0.580	0.789	1.344	2.632
$\mathcal{E}$	0.950	1.399	2.022	0.818	0.940	1.349	2.001	0.817
$\mathcal{B}$	1.670	2.039	0.905	1.212	1.360	1.969	0.817	1.288
$\mathcal{G}$	2.710	3.319	1.773	0.597	3.080	3.380	2.412	1.177
$\mathcal{E}$	3.320	3.790	0.593	-0.341	3.390	3.889	1.171	-0.343

Table 6.1: Profiles for HW current under fault free and OD reverse conditions, common episodes only

The two $\mathcal{B}$ episodes in the middle of the faulty waveform compete for the match with the single episode in the fault-free waveform. The first of the two wins because its magnitude is greater than the second, and therefore is considered to match the magnitude of the second episode in the fault-free waveform.

Tracking of episode values

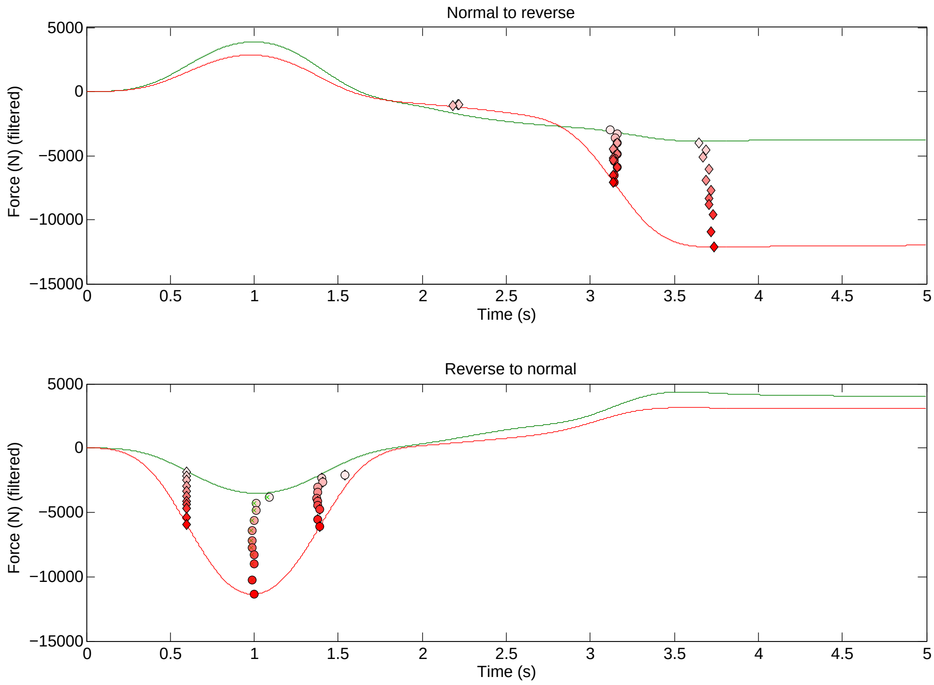

Figure 6.10: Fault free (green) and OD reverse (red) filtered waveforms, overlaid with episode values where rules were evaluated

Figure 6.10 shows force waveforms from the overdriving reverse fault simulation - the green plot is at the start of the simulation, i.e. fault free, and the red is at the end, i.e. maximum severity. The shaded markers overlaid on the plots show episode values (circles for start values, diamonds for end values) where these were used in the diagnosis evaluation. The system is, in this case, strong at identifying the right episode values to use for diagnosis: the ones where the fault causes the largest effects. Two episode mismatches can be seen in the normal to reverse waveform: the two isolated diamonds in the middle of the plot. This is where an episode has been recorded and matched to the one later in the waveform where the actual changes have taken place. The correct match for these values would be at the end of the waveform, where a column of diamonds descends towards the end of the faulty waveform.

6.4.2 System outputs

Tables 6.2 and 6.3 show the number of combinations which fulfilled each test specification, for each condition simulated. Since the fault free condition was not changing under adjustment, gradients of membership trends were irrelevant, so instead the average membership values were compared in the same way.

Spec	Condition	OK combinations	% effectiveness
1	Fault free	6/6	100
	OD Normal	6/6	100
	OD Reverse	6/6	100
	Backdrive OD	1/2	50
	Backdrive UD	1/2	50
2	Fault free	6/6	100
	OD Normal	6/6	100
	OD Reverse	5/6	83
	Backdrive OD	1/2	50
	Backdrive UD	0/2	0

Table 6.2: Test results for normal to reverse movements on the HW actuator

Spec	Condition	OK combinations	% effectiveness
1	Fault free	6/6	100
	OD Normal	6/6	100
	OD Reverse	5/6	83
	Backdrive OD	2/2	100
	Backdrive UD	2/2	100
2	Fault free	6/6	100
	OD Normal	6/6	100
	OD Reverse	4/6	67
	Backdrive OD	2/2	100
	Backdrive UD	0/2	0

Table 6.3: Test results for reverse to normal movements on the HW actuator

These results show a clear distinction in success between the overdriving faults, which were detected and diagnosed well, and the backdrive faults, which are not detected with the same level of success. This is probably a combination of two factors: one, that the magnitude of fault effects for the backdrive faults is much smaller than that for the overdriving faults, and two, that the order in which the faults are presented to the rule-establishing function (function 1) affects the way the rules are made, because the margins imposed between rule values are applied sequentially.

Figures 6.11 and 6.12 show successful fault diagnoses from the combination of actuators A and B, under the simulation of overdriving faults.

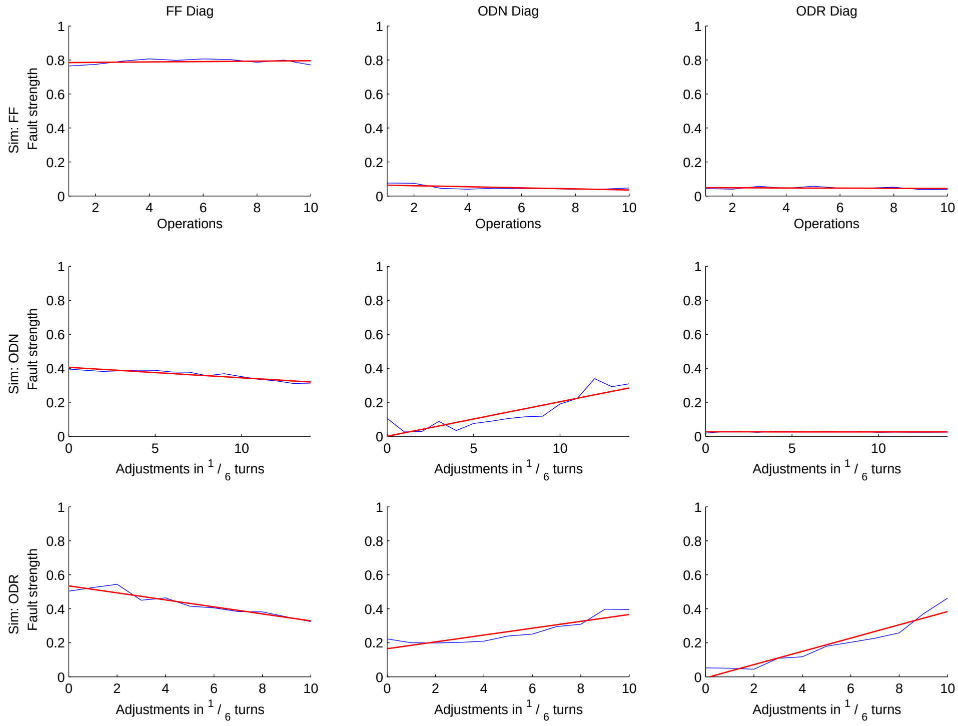

Figure 6.11: Fault diagnoses for normal to reverse movements, test actuator A, monitored actuator B

6.5 Conclusions

In this case study, several adjustment faults were induced on a set of three HW switch actuators. The fault diagnosis system was then run on the dataset. The results show that the system is weak at transferring rules between actuators which have very large differences in parameters, but that it is still sometimes capable

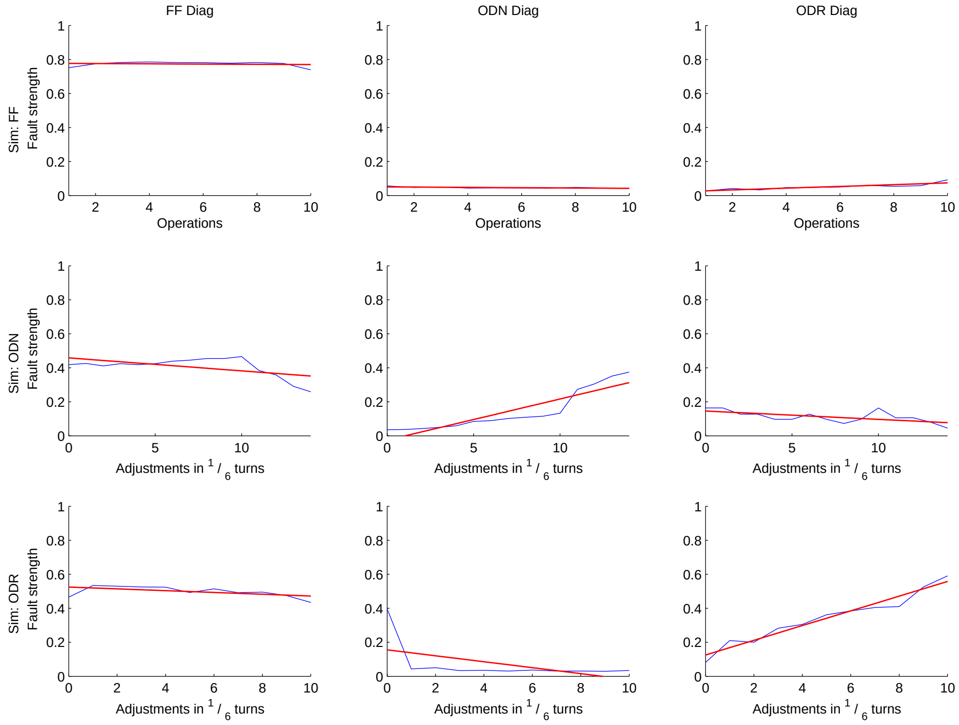

Figure 6.12: Fault diagnoses for reverse to normal movements, test actuator A, monitored actuator B

of correctly diagnosing incipient faults. This weakness identifies opportunities for further improvements in the future.

However, it is pleasing to note a near-100% success rate for the detection and diagnosis of faults in the main actuator drive. Faults of this nature have been known to lead to accidents (such as the Potters Bar derailment) and so the detection of this fault is particularly relevant to industry.

In this case study, there were no false alarms, but there were some misdiagnoses. However, when the system worked well, the diagnosis for other faults tended to fall as the simulated fault diagnosis rose. This showed that the rules had established themselves well in these cases. The lack of false alarms is encouraging because it is very important that technical staff trust the system to only warn them when a fault is really present.

Chapter 7

Conclusions

7.1 Summary

In this thesis, a practical design problem has been analysed, decomposed and solved using a systems engineering approach to structure the traditional research activities. Methods applied in different industry sectors were researched and evaluated.

Qualitative trend analysis, a method previously employed in the diagnosis of chemical plants, inspired the solution developed here. The system uses QTA to obtain abstract qualitative and quantitative information from measured waveforms. The information is used to form fuzzy rules which can be applied to all STME actuators of a particular type, avoiding the need for lengthy training periods, which would be impractical on the railways.

Two case studies, each a practical industrial application of the system, were carried out with the generous cooperation of Network Rail and South West Trains. The results show that the system is capable of fulfilling its requirements, but that there are some weaknesses which need to be addressed.

7.2 Limitations

The key limitation that the system presents is that the additive relativisation method is not sufficiently accurate or consistent to give reliable results under most conditions. The approach is, however, much more detailed and sensitive than the analysis currently implemented in the industry. The output of the system would not be dependable in the context of a highly demanding infrastructure or vehicle system, although the incidence of false alarms (i.e. fault indications during fault free operation) was very low, and in fact in case study II, there were no false fault indications when the actuator was fault free.

This is a problem because the application of a condition monitoring system on the railways would be as an aid to condition-based maintenance of assets. Condition-based maintenance eliminates costly and inefficient periodic maintenance tasks, instead directing tasks to be carried out as and when the condition monitoring system deems it necessary. The monitoring system must therefore be very accurate and reliable. False alarms damage the credibility of the system, making it less effective because maintenance staff are wont to disregard its outputs if they have experience of them being inaccurate.

The system still relies on the measurement of several dynamic values, which requires monitoring and data acquisition equipment to be installed in situ on the actuator, something which might incur significant expense in retrofitting.

7.3 Proposals for future work

The system has demonstrated that the concept of using qualitative trend analysis for fault diagnosis is valid, and that it has the potential to solve the problem of diagnosing multiple, distributed assets without carrying out exhaustive testing on each instance.

To achieve a practical system, the weak points of the design must be addressed. One of these, as already mentioned, is use of differences to express relative change between faulty and fault-free performance. Whilst the QTA approach is effective and intuitive, it would be beneficial to investigate the implementation of different ways of expessing the relative changes.

Another weak point is the determination of the qualitative trends from measured data. The system as presented in this thesis made heavy use of low-pass filtering in order to be able to recognise shapes and trends on a scale which allowed easy comparisons between waveforms from different actuators or different conditions. The performance of the system might be improved if trends on the correct scale could be identified from unfiltered data. This was particularly evident in the data from the closing doors in case study I - there were clear fault effects in the unfiltered data, but these were difficult to retain in the filtered data, making the system perform quite badly on all the closing data. The authors of the papers [47, 48, 49, 50] which inspired the QTA approach for this problem suggested the use of neural networks or other AI approaches to determine accurately the shape of trends in the data. This is one possibility which was not explored in this thesis, because it seemed to be a very complex way of solving only one small part of the design problem, and therefore went against the requirement for a simple method that was easily understandable.

Further work should also assess the relative effectiveness of using particular measured parameters in the diagnosis system; it can be seen from the graphs of fault simulations that the displacement parameter, for example, seems to exhibit very few effects when faults are introduced. Few of the rules established by the system used values from the displacement waveforms. The effects were so small that they could not be distinguished from random variations, and were rightly ignored by the system so that it would not be confused by those variations.

The system developed here has been presented to the INNOTRACK EU project and may be taken forward by one of the industrial partners, a S&C[1] manufacturer with a condition monitoring product for their switches.

7.4 Effectiveness of the system

The system performs markedly worse in case study II, when compared to case study I. It highlights the weak point of the design, which is that a raw difference, obtained from a matching episode value, is used to calculate relative performance. This method works better when, as in case study I, the magnitude of fault effects is similar between actuators. For train doors, this seems intuitively true, since each door is of similar mass and each actuator is installed in similar conditions.

For switch actuators, however, the situation is very different: the load the actuator is moving may vary from switch to switch because it is proportional to the length of the switch, and also may be affected by such factors as cant.

This weak point is also demonstrated by both case studies, in that the membership of faults calculated by the system does not reach 1 when the rules are being applied from another actuator - the differences calculated simply do not synchronise well enough for this to be the case.

Overall, however, the test results show that the system is capable of correctly detecting and diagnosing rising trends in incipient faults, even as the fault is in an early stage of development. It therefore has the potential to be part of an integrated diagnostic system.

With some refinements to the algorithms for episode matching and qualitative trend detection, the system would become much more effective and reliable. One desirable improvement would be to develop a way to spot qualitative trends, at the correct

[1] Switches and crossings

dynamic scale, without performing any filtering on the waveform at all. This may require sophisticated pattern recognition systems such as neural networks, but the need to keep the system simple should be considered when such refinements are considered.

After improvement, this system would be capable of achieving very accurate fault predictions. Once experience has been gained of exactly how quickly the faults develop naturally, it will be a trivial matter to start predicting the time to failure in the presence of a particular fault, allowing true condition-based maintenance to be achieved. Maintainers could perform their tasks reactively instead of periodically - and with accurate condition monitoring, ensuring no faults are missed in the prediction system, this would save money, time and risk to staff. It is also possible that if maintainers are able to take ownership of the system's data and outputs, they may gain a deeper understanding of the behaviour of the assets they maintain, making them more likely to make good decisions in maintenance.

7.5 Final words

A systematic, requirements-driven process was used to form a design for this fault diagnosis system. By drawing on inspiration from several areas of the fault diagnosis research base, it was possible to design a system which used existing approaches in novel ways. This has resulted in a refreshing and new approach to the problem of condition monitoring, and promises much for the future.

Although there is work to be done if this system is to be used in practice, it is gratifying to consider that the concept is valid. If implementation is achieved, it may well save millions of pounds per year in maintenance costs, allowing the rail industry to reinvest that money in improvements for passengers and customers.

Bibliography

[1] Network Rail Infrastructure Ltd and Jarvis plc, "Statement regarding Potters Bar rail crash," 2004.
URL `http://www.networkrailmediacentre.co.uk/Content/Detail.asp?ReleaseID=207&NewsAreaID=2`

[2] F. B. Zhou, M. D. Duta and M. P. Henry, "Remote condition monitoring for railway point machine," in *ASME/IEEE Joint Rail Conference*, pp. 103–108 (2002).

[3] J. C. Hoskins, K. M. Kaliyur and D. M. Himmelblau, "Incipient fault detection and diagnosis using artificial neural networks," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 81–86 (1990).

[4] N. Lehrasab and S. Fararooy, "Formal definition of single throw mechanical equipment for fault diagnosis," *IEE Electronics Letters*, **vol. 34** (23), pp. 2231–2232, 1998.

[5] T. Doran, (ed.) *IEEE standard for the application and management of the systems engineering process* (IEEE, 2005).

[6] B. S. Blanchard and W. J. Fabrycky, *Systems Engineering and Analysis* (Prentice Hall, 1998), 3rd edn.

[7] C. Roberts, "Project concept specification," 2005.

[8] V. Venkatasubramanian, R. Rengaswamy, K. Yin and S. N. Kavuri, “A review of process fault detection and diagnosis part I: Quantitative model-based methods,” *Computers and Chemical Engineering*, **vol. 27**, pp. 293–311, 2003.

[9] M. S. Z. Abidin, R. Yusof, M. Khalid and S. M. Amin, “Application of a model-based fault detection and diagnosis using parameter estimation and fuzzy inference to a dc servomotor,” in *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, pp. 783–788 (2002).

[10] R. Patton and J. Chen, “A robustness study of model-based fault diagnosis for jet engine systems,” in *Proceedings of 1st IEEE Conference on Control Applications*, pp. 871–876 (1992).

[11] D. Gayme, S. Menon, C. Ball, D. Mukavetz and E. Nwadiogbu, “Fault detection and diagnosis in turbine engines using fuzzy logic,” in *NAFIPS 2003: 22nd International Conference of the North American Fuzzy Information Processing Society*, pp. 341–346 (2003).

[12] P. M. Frank, “Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results,” *Automatica*, **vol. 26** (3), pp. 459–474, 1990.

[13] T. Höfling and R. Isermann, “Fault detection based on adaptive parity equations and single parameter tracking,” *Control Engineering Practice*, **vol. 4** (10), pp. 1361–1369, 1996.

[14] J. J. Gertler and R. Monajemy, “Generating directional residuals with dynamic parity relations,” *Automatica*, **vol. 31** (4), pp. 627–635, 1995.

[15] R. N. Clark, “The dedicated observer approach to instrument failure detection,” in *Proceedings of the 18th IEEE Conference on Decision and Control inc. the Symposium on Adaptive Processes*, vol. 18, pp. 237–241 (1979).

[16] P. BALLÉ and D. FÜSSEL, "Closed-loop fault diagnosis based on a nonlinear process model and automatic fuzzy rule generation," *Engineering Applications of Artificial Intelligence*, **vol. 13**, pp. 695–704, 2000.

[17] M. A. DEMETRIOU and M. M. POLYCARPOU, "Incipient fault diagnosis of dynamical systems using online approximators," *IEEE Transactions on Automatic Control*, **vol. 43** (11), pp. 1612–1617, 1998.

[18] L. TRAVÉ-MASSUYÈS and R. MILNE, "Gas-turbine condition monitoring using qualitative model-based diagnosis," *IEEE Expert*, pp. 22–31, 1997.

[19] J. LUNZE, "Qualitative modelling of continuous-variable systems by means of non-deterministic automata," *Intelligent Systems Engineering*, **vol. 1** (1), pp. 22–30, 1992.

[20] G. LICHTENBERG and J. LUNZE, "Observation of qualitative states by means of a qualitative model," *International Journal of Control*, **vol. 66** (6), pp. 885–903, 1997.

[21] F. SCHILLER, J. SCHRÖDER and J. LUNZE, "Diagnosis of transient faults in quantised systems," *Engineering Applications of Artificial Intelligence*, **vol. 14**, pp. 519–536, 2001.

[22] J. LUNZE, B. NIXDORF and J. SCHRÖDER, "Deterministic discrete-event representations of linear continuous-variable systems," *Automatica*, **vol. 35**, pp. 395–406, 1999.

[23] J. LUNZE, "Diagnosis of quantized systems based on a timed discrete-event model," *IEEE Transactions on Systems, Man and Cybernetics*, **vol. 30** (3), 2000.

[24] K. B. RAMKUMAR, P. PHILIPS, H. A. PRESIG, W. K. HO and K. W. LIM, "Structured fault-detection and diagnosis using finite state-automaton," in *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society*, vol. 3, pp. 1667–1672 (1998).

[25] M. Iri, K. Aoki, E. O'Shima and H. Matsuyama, "An algorithm for diagnosis of system failures in the chemical process," *Computers and Chemical Engineering*, **vol. 3**, pp. 489–493, 1979.

[26] E. A. Tarifa and N. J. Scenna, "Fault diagnosis, direct graphs, and fuzzy logic," *Computers and Chemical Engineering*, **vol. 21** (Supplement), pp. S649–S654, 1997.

[27] L. A. Zadeh, "Fuzzy logic," *Computer*, **vol. 21**, 1988.

[28] K. D. Forbus, "Qualitative physics: past, present and future," in *Exploring artificial intelligence*, chap. 7, pp. 239–296 (Morgan Kaufmann Publishers Inc., 1988).

[29] J. de Kleer and J. S. Brown, "A qualitative physics based on confluences," *Artificial Intelligence*, **vol. 24**, pp. 7–83, 1984.

[30] N. H. Ulerich and G. J. Powers, "On-line hazard aversion and fault diagnosis in chemical processes: the digraph + fault tree method," *IEEE Transactions on Reliability*, **vol. 37** (2), pp. 171–177, 1988.

[31] V. Venkatasubramanian, R. Rengaswamy and S. N. Kavuri, "A review of process fault detection and diagnosis part II: Qualitative model-based methods," *Computers and Chemical Engineering*, **vol. 27**, pp. 313–326, 2003.

[32] F. E. Finch and M. A. Kramer, "Narrowing diagnostic focus using functional decomposition," *AIChE Journal*, **vol. 34** (1), pp. 25–36, 1988.

[33] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri and K. Yin, "A review of process fault detection and diagnosis part III: Process history based methods," *Computers and Chemical Engineering*, **vol. 27**, pp. 327–346, 2003.

[34] K. I. Kim, K. Jung and H. J. Kim, "Face recognition using kernel principal component analysis," *IEEE Signal Processing Letters*, **vol. 9** (2), pp. 40–42, 2002.

[35] J. Yang, D. Zhang, A. F. Frangi and J.-Y. Yang, "Two-dimensional PCA: A new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **vol. 26** (1), pp. 131–137, 2004.

[36] A. Sophian, G. Y. Tian, D. Taylor and J. Rudlin, "A feature extraction technique based on principal component analysis for pulsed eddy current NDT," *NDT&E International*, **vol. 36**, pp. 37–41, 2003.

[37] K. Mori, N. Kasashima, T. Yoshioka and Y. Ueno, "Prediction of spalling on a ball bearing by applying the discrete wavelet transform to vibration signals," *Wear*, **vol. 195**, pp. 162–168, 1996.

[38] M. O. U. Johannes R. Sveinsson and J. A. Benediktsson, "Cluster-based feature extraction and data fusion in the wavelet domain," in *IEEE International Geoscience and Remote Sensing Symposium*, vol. 2, pp. 867–869 (2001).

[39] N. C. Fitton and S. J. D. Cox, "Optimising the application of the Hough transform for automatic feature extraction from geoscientific images," *Computers & Geosciences*, **vol. 24** (10), pp. 933–951, 1998.

[40] D. Li, W. Pedrycz and N. J. Pizzi, "Fuzzy wavelet packet based feature extraction method and its application to biomedical signal classification," *IEEE Transactions on Biomedical Engineering*, **vol. 52** (6), pp. 1132–1139, 2005.

[41] S. Pittner and S. V. Kamarthi, "Feature extraction from wavelet coefficients for pattern recognition tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **vol. 21** (1), pp. 83–88, 1999.

[42] G. G. Yen and K.-C. Lin, "Wavelet packet feature extraction for vibration monitoring," *IEEE Transactions on Industrial Electronics*, **vol. 47** (3), pp. 650–667, 2000.

[43] I. A. Basheer and M. Hajmeer, "Artificial neural networks: Fundamentals, computing, design and application," *Journal of Microbiological Methods*, **vol. 43**, pp. 3–31, 2000.

[44] V. Venkatasubramanian and K. Chan, "A neural network methodology for process fault diagnosis," *AIChE Journal*, **vol. 35** (12), pp. 1993–2002, 1989.

[45] Y.-C. Huang, "Condition assessment of power transformers using genetic-based neural networks," *IEE Proceedings on Science, Measurement and Technology*, **vol. 150** (1), pp. 19–24, 2003.

[46] A. T. Vemuri, M. M. Polykarpou and S. A. Diakourtis, "Neural network based fault detection in robotic manipulators," *IEEE Transactions on Robotics and Automation*, **vol. 14** (2), 1998.

[47] J. T.-Y. Cheung and G. Stephanopoulos, "Representation of process trends - part I: A formal representation framework," *Computers and Chemical Engineering*, **vol. 14** (4/5), pp. 495–510, 1990.

[48] J. T.-Y. Cheung and G. Stephanopoulos, "Representation of process trends - part II: The problem of scale and qualitative scaling," *Computers and Chemical Engineering*, **vol. 14** (4/5), pp. 511–539, 1990.

[49] B. R. Bakshi and G. Stephanopoulos, "Representation of process trends - part III: Multiscale extraction of trends from process data," *Computers and Chemical Engineering*, **vol. 18** (4), pp. 267–302, 1994.

[50] B. R. Bakshi and G. Stephanopoulos, "Representation of process trends - part IV: Induction of real-time patterns from operating data for diagnosis

and supervisory control," *Computers and Chemical Engineering*, **vol. 18** (4), pp. 303–332, 1994.

[51] J. A. Silmon and C. Roberts, "A systems approach to fault detection and diagnosis for condition-based maintenance," in *1st IET International Conference on Railway Condition Monitoring* (2006).

[52] R. Rengaswamy and V. Venkatasubramanian, "A syntactic pattern-recognition approach for process monitoring and fault diagnosis," *Engineering Applications of Artificial Intelligence*, **vol. 8** (1), pp. 33–51, 1995.

[53] I. B. Özyurt, L. O. Hall and A. K. Sunol, "SQFDiag: Semiquantitative model-based fault monitoring and diagnosis via episodic fuzzy rules," *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, **vol. 29** (3), pp. 294–306, 1999.

[54] K. L. Butler, "An expert system-based framework for an incipient failure detection and predictive maintenance system," in *Proceedings of the International Conference on Intelligent Systems Applications to Power Systems*, pp. 321–326 (1996).

[55] C. E. Lin, J. M. Ling and C. L. Huang, "An expert system for transformer fault diagnosis using dissolved gas analysis," *IEEE Transactions on Power Delivery*, **vol. 8** (1), pp. 231–238, 2003.

[56] W. R. Becraft and P. L. Lee, "An integrated neural network/expert system approach for fault diagnosis," *Computers and Chemical Engineering*, **vol. 17** (10), pp. 1001–1014, 1993.

[57] Z. Wang, Y. Liu and P. J. Griffin, "A combined ANN and expert system tool for transformer fault diagnosis," *IEEE Transactions on Power Delivery*, **vol. 13** (4), pp. 1224–1229, 1998.

[58] M. Bartyś, R. Patton, M. Syfert, S. de las Heras and J. Quevedo, "Introduction to the DAMADICS actuator FDI benchmark study," *Control Engineering Practice*, **vol. 14**, pp. 577–596, 2006.

[59] J. M. F. Calado, J. M. G. Sá da Costa, M. Bartyś and J. Korbicz, "FDI approach to the DAMADICS benchmark problem based on qualitative reasoning coupled with fuzzy neural networks," *Control Engineering Practice*, **vol. 14**, pp. 684–698, 2006.

[60] P. Supavatanakul, J. Lunze, V. Puig and J. Quevedo, "Diagnosis of timed automata: Theory and application to the DAMADICS actuator benchmark problem," *Control Engineering Practice*, **vol. 14**, pp. 609–619, 2006.

[61] V. Puig, A. Stancu, T. Escobet, F. Nejjari, J. Quevedo and R. Patton, "Passive robust fault detection using interval observers: Application to the DAMADICS benchmark problem," *Control Engineering Practice*, **vol. 14**, pp. 621–633, 2006.

[62] M. Witczak, J. Korbicz, M. Mrugalski and R. J. Patton, "A GMDH neural network-based approach to robust fault diagnosis: Application to the DAMADICS benchmark problem," *Control Engineering Practice*, **vol. 14**, pp. 671–683, 2006.

[63] F. Previdi and T. Parisini, "Model-free actuator fault detection using a spectral estimation approach: the case of the DAMADICS benchmark problem," *Control Engineering Practice*, **vol. 14**, pp. 635–644, 2006.

[64] C. D. Bocaniala and J. Sá da Costa, "Application of a novel fuzzy classifier to fault detection and isolation of the DAMADICS benchmark problem," *Control Engineering Practice*, **vol. 14**, pp. 653–669, 2006.

[65] Advantage Technical Consulting, "Review of the reliability of point motors and track circuits," 2002.

[66] C. Roberts, H. P. Dassanayake, N. Lehrasab and C. J. Goodman, “Distributed quantitative and qualitative fault diagnosis: railway junction case study,” *Control Engineering Practice*, **vol. 10**, pp. 419–429, 2002.

[67] N. Lehrasab, H. P. Dassanayake, C. Roberts, S. Fararooy and C. J. Goodman, “Industral fault diagnosis: pneumatic train door case study,” *Proceedings of the IMechE Part F: Rail and Rapid Transit*, 2002.

[68] E. Rocha Loures and J.-C. Pascal, “Detection and diagnosis of hybrid dynamic systems based on time fuzzy petri nets,” in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1825–1831 (2004).

[69] D. Füssel, “Self learning classification tree (SELECT) - a human-like approach to fault diagnosis,” in *EUFIT ’97*, pp. 1870–1875 (1997).

[70] The Mathworks Inc., “Documentation for the MATLAB software suite,” HTML help files, 2004.

[71] A. Bredebusch, J. Lunze and H. Richter, “A Petri-net representation of the qualitative behaviour of a dynamical continuous-time system,” in *IEE Second International Conference on Intelligent Systems Engineering*, pp. 223–228 (1994).

[72] G. Lichtenberg and J. Lunze, “Identification of discrete event models for continuous-variable systems,” in *UKACC International Conference on Control*, vol. 1, pp. 711–715 (1996).

[73] Q. Shen and R. R. Leitch, “Qualitative model-based diagnosis of continuous dynamic systems,” in *IEE First International Conference on Intelligent Systems Engineering*, pp. 147–152 (1992).

[74] J. Lunze, “Qualitative modelling of dynamical systems for on-line diagnosis,” in *IEE First International Conference on Intelligent Systems Engineering*, pp. 153–158 (1992).

[75] S. Charbonnier, C. Garcia-Beltan, C. Cadet and S. Gentil, “Trends extraction and analysis for complex system monitoring and decision support,” *Engineering Applications of Artificial Intelligence*, **vol. 18** (1), pp. 21–36, 2005.

[76] L. J. de Miguel and L. F. Blázquez, “Fuzzy logic-based decision-making for fault diagnosis in a dc motor,” *Engineering Applications of Artificial Intelligence*, **vol. 18** (4), pp. 423–450, 2005.

[77] C. G. Looney, “Fuzzy petri nets for rule-based decision making,” *IEEE Transactions on Systems, Man and Cybernetics*, **vol. 18** (1), pp. 178–183, 1988.

[78] S.-M. Chen, “Weighted fuzzy reasoning using weighted fuzzy petri nets,” *IEEE Transactions on Knowledge and Data Engineering*, **vol. 14** (2), pp. 386–397, 2002.

[79] J. Lunze and J. Schröder, “Sensor and actuator fault diagnosis of systems with discrete inputs and outputs,” *IEEE Transactions on Systems, Man and Cybernetics*, **vol. 34** (2), 2004.

[80] P. Supavatanakul and J. Lunze, “Timed discrete-event method for diagnosis of industrial actuators,” in *IEEE International Conference on Industrial Technology*, pp. 1354–1359 (2002).

[81] D. Förstner and J. Lunze, “Discrete-event models of quantized systems for diagnosis,” *International Journal of Control*, **vol. 74** (7), pp. 690–700, 2001.

[82] Q. Shen and R. Leitch, “Fuzzy qualitative simulation,” *IEEE Transactions on Systems, Man and Cybernetics*, **vol. 23** (4), pp. 1038–1061, 1993.

[83] R. Du and K. Yeung, “Fuzzy transition probability: a new mothod for monitoring progressive faults. part 1: the theory,” *Engineering Applications of Artificial Intelligence*, **vol. 17**, pp. 457–467, 2004.

[84] Q. H. Wang and J. R. Li, "A rough set-based fault ranking prototype system for fault diagnosis," *Engineering Applications of Artificial Intelligence*, **vol. 17**, pp. 909–917, 2004.

[85] F. E. H. Tay and L. Shen, "Fault diagnosis based on Rough Set Theory," *Engineering Applications of Artificial Intelligence*, **vol. 16**, pp. 39–43, 2003.

[86] Q. Jing, W. Xisen, P. Zhihua and X. Yongcheng, "A research on fault diagnostic expert system based on fuzzy Petri nets for FMS machining cell," in *IEEE International Conference on Industrial Technology*, pp. 122–125 (1996).

[87] E. Németh, R. Lakner, K. M. Hangos and I. T. Cameron, "Hierarchical CPN model-based diagnosis using HAZOP knowledge," Scl-009/2003, Process Control Research Group, MTA SZTAKI, Hungary, 2003, `http://daedalus.scl.sztaki.hu/PCRG`.

[88] J. Jin and J. Shi, "Automatic feature extraction of waveform signals for in-process diagnostic performance improvement," *Journal of Intelligent Manufacturing*, **vol. 12**, pp. 257–268, 2001.

[89] M. Gao, M. Zhou, X. Huang and Z. Wu, "Fuzzy reasoning Petri nets," *IEEE Transactions on Systems, Man and Cybernetics*, **vol. 33** (3), pp. 314–324, 2003.

[90] R. Isermann, "On fuzzy logic applications for automatic control, supervision and fault diagnosis," *IEEE Transactions on Systems, Man and Cybernetics*, **vol. 28** (2), pp. 221–235, 1998.

[91] S. Fararooy and N. Lehrasab, "Generic test environment for single-throw mechanical equipment," *IEE Computing and Control Engineering Journal*, **vol. 9** (3), pp. 141–149, 1998.

[92] Railway Group Standards, *GM/RT 1310: Design requirements and acceptance of portable/transportable infrastructure plant and work equipment* (Railtrack PLC, 1998).

[93] Railway Group Standards, *GE/RT 8015: Electromagnetic compatibility between railway infrastructure and trains* (Railway Safety, 2002).

[94] British Standard, *BS EN 60529: Degrees of protection provided by enclosures (IP code)* (British Standards Institute, 2000).

[95] R. Isermann and M. Ulieru, "Integrated fault detection and diagnosis," in *IEEE International Conference on Systems, Man and Cybernetics*, pp. 743–748 (1993).

[96] R. Isermann, "Model based fault detection and diagnosis methods," in *American Control Conference*, pp. 1605–1609 (1995).

[97] J. M. Vinson and L. H. Ungar, "Dynamic process monitoring and fault diagnosis with qualitative models," *IEEE Transactions on Systems, Man and Cybernetics*, **vol. 25** (1), pp. 181–189, 1995.

[98] T. I. Salsbury, "A controller for HVAC systems with fault detection capabilities based on simulation models," in *Proceedings of Building Simulation '99*, vol. 1, pp. 147–154 (1999).

[99] "2005 quarter 3 network condition monitor," Tech. rep., HM Office of the Rail Regulator, 2005.

[100] M. Pilling and L. Wilkinson, "Reliability based maintenance and condition monitoring," Tech. rep., Asset Management Consulting Ltd and Network Rail.

[101] S. Dash, R. Rengaswamy and V. Venkatasubramanian, "Fuzzy logic based trend classification for fault diagnosis of chemical processes," *Computers and Chemical Engineering*, **vol. 27**, pp. 347–362, 2003.

[102] S. Dash, M. R. Maurya, V. Venkatasubramanian and R. Rengaswamy, "A novel interval-halving framework for automated identification of process trends," *AIChE Journal*, **vol. 50** (1), pp. 149–162, 2004.

[103] D. Düştegör, E. Frisk, V. Cocquempot, M. Krysander and M. Staroswiecki, "Structural analysis of fault isolability in the DAMADICS benchmark," *Control Engineering Practice*, **vol. 14**, pp. 597–608, 2006.

[104] J. M. Kościelny, M. Bartyś, P. Rzepiejewski and J. Sá da Costa, "Actuator fault distinguishability study for the DAMADICS benchmark problem," *Control Engineering Practice*, **vol. 14**, pp. 645–652, 2006.

[105] D. J. Pedregal, F. P. García and F. Schmid, "RCM^2 predictive maintenance of railway systems based on unobserved components models," *Reliability Engineering and System Safety*, **vol. 83**, pp. 103–110, 2004.

[106] M. H. Chen, D. Lee and T. Pavlidis, "Residual analysis for feature detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **vol. 13**, pp. 30–40, 1991.

[107] H. Guo and S. B. Gelfand, "Classification trees with neural network feature extraction," *IEEE Transactions on Neural Networks*, **vol. 3** (6), pp. 923–933, 1992.

[108] I. Mierswa and K. Morik, "Automatic feature extraction for classifying audio data," *Machine Learning*, **vol. 58**, pp. 127–149, 2005.

[109] Z. Wang, Z. He and J. D. Z. Chen, "Filter banks and neural network-based feature extraction and automatic classfication of electrogastrogram," *Annals of Biomedical Engineering*, **vol. 27**, pp. 88–95, 1999.

[110] T. Kohonen, "The self-organising map," *Proceedings of the IEEE*, **vol. 78** (9), pp. 1464–1480, 1990.

[111] S. K. Setarehdan and J. J. Soraghan, "Automatic left ventricular feature extraction and visualisation from echocardiographic images," in *Computers in Cardiology*, pp. 9–12 (1996).

[112] Y. Mallet, D. Coomans, J. Kautsky and O. de Vel, “Classification using adaptive wavelets for feature extraction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **vol. 19** (10), pp. 1058–1066, 1997.

[113] K. K. Simhadri, S. S. Iyengar, R. J. Holyer, M. Lybanon and J. M. Zachary, Jr, “Wavelet-based feature extraction from oceanographic images,” *IEEE Transactions on Geoscience and Remote Sensing*, **vol. 36** (3), pp. 767–778, 1998.

[114] J. Holmquist, E. Bengtsson, O. Eriksson, B. Nordin and B. Stenkvist, “Computer analysis of cervical cells: automatic feature extraction and classification,” *Journal of Histochemistry and Cytochemistry*, **vol. 26** (11), pp. 1000–1017, 1978.

[115] E. A. Ashton, K. J. Parker, M. J. Berg and C. W. Chen, “A novel volumetric feature extraction technique with applications to MR images,” *IEEE Transactions on Medical Imaging*, **vol. 16** (4), pp. 365–371, 1997.

[116] A. Eronen and A. Klapuri, “Musical instrument recognition using cepstral coefficients and temporal features,” in *IEEE International Conference on Acuoustics, Speech and Signal Processing*, vol. 2, pp. 753–756 (2000).

[117] I. Kapouleas, “Segmentation and feature extraction for magnetic resonance brain image analysis,” in *10th International Conference on Pattern Recognition*, vol. 1, pp. 583–590 (1990).

[118] I. Daubechies, “The wavelet transform, time-frequency localisation and signal analysis,” *IEEE Transactions on Information Theory*, **vol. 36** (5), pp. 961–1005, 1990.

[119] J. Mora, D. Llanos, J. Meléndez, J. Colomer, J. Sánchez and X. Corbella, “Classification of sags measured in a distribution substation based on qualitative and temporal descriptors,” in *17th International Conference on Electricity Distribution* (2003).

[120] J. Jin and J. Shi, “Diagnostic feature extraction from stamping tonnage signals based on design of experiments,” *Transactions of the ASME*, **vol. 122**, pp. 360–369, 2000.

[121] J.-J. J. Chen and R. Shiavi, “Temporal feature extraction and clustering analysis of electromyographic linear envelopes in gait studies,” *IEEE Transactions on Biomedical Engineering*, **vol. 37** (3), pp. 295–302, 1990.

[122] R. T. Olszewski, “Generalised feature extraction for structural pattern recognition in time-series data,” Ph.D. thesis, Carnegie Mellon University, 2001.

[123] E. J. Keogh and M. J. Pazzani, “An enhanced representation of tme series which alows fast and accurate classification, clustering and relevance feedback,” in *Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining*, pp. 239–241 (1998).

[124] L. Satish, “Short-time Fourier and wavelet transforms for fault detection in power transformers during impulse tests,” *IEE Proceedings in Science, Measurement and Technology*, **vol. 145** (2), pp. 77–84, 1998.

[125] M. Ayoubi, “Neuro-fuzzy structure for rule generation and application in the fault diagnosis of technical processes,” in *Proceedings of the American Control Conference*, pp. 2757–2761 (1995).

[126] W.-H. Chen, C.-W. Liu and M.-S. Tsai, “On-line fault diagnosis of distribution substations using hybrid cause-effect network and fuzzy rule-based method,” *IEEE Transactions on Power Delivery*, **vol. 15** (2), pp. 710–717, 2000.

[127] Y. Lu, T. Q. Chen and B. Hamilton, “A fuzzy system for autmotive fault diagnosis: fast rule generation and self tuning,” *IEEE Transactions on Vehicular Technology*, **vol. 49** (2), pp. 651–660, 2000.

[128] W. G. Fenton, T. M. McGinnity and L. P. Maguire, “Fault diagnosis of electronic systems using intelligent techniques: a review,” *IEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, **vol. 31** (3), pp. 269–281, 2001.

[129] J. E. Larsson, B. Hayes-Roth and D. M. Gaba, “Goals and functions of the human body: an MFM model for fault diagnosis,” *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, **vol. 27** (6), pp. 758–765, 1997.

[130] M. W. Kim, J. G. Lee and C. Min, “Efficient fuzzy rule generation based on fuzzy decision tree for data mining,” in *Proceedings of the IEEE International Fuzzy Systems Conference*, vol. 3, pp. 1223–1228 (1999).

[131] F. J. Uppal, R. J. Patton and V. Palade, “Neuro-fuzzy based fault diagnosis applied to an electro-pneumatic valve,” in *Proceedings of the 15th IFAC World Congress*, pp. 21–26 (2002).

[132] G. Dounias, A. Tsakonas, J. Jantzen, H. Axer, B. Bjerregaard and D. G. von Keyserlingk, “Genetic programming for the generation of crisp and fuzzy rule bases in classification and diagnosis of medical data,” in *Proceedings of the 1st International NAISO Congress on Neuro-Fuzzy Technologies* (2002).

[133] R. Rengaswamy, T. Hägglund and V. Venkatasubramanian, “A qualitative shape analysis formalism for monitoring conrol loop performance,” *Engineering applications of artificial intelligence*, **vol. 14**, pp. 23–33, 2001.

[134] R. Rengaswamy and V. Venkatasubramanian, “A fast training neural network and its updation for incipient fault detectionand diagnosis,” *Computers and Chemical Engineering*, **vol. 24**, pp. 431–437, 2000.

[135] Y. Kitamura and R. Mizoguchi, "An ontological analysis of fault process and category of faults," in *Proceedings of 10th International Workshop on Principles of Diagnosis*, pp. 118–128 (1999).

[136] W. Pedrycz and F. Gomide, "A generalised fuzzy petri net model," *IEEE Transactions on Fuzzy Systems*, **vol. 2** (4), pp. 295–301, 1994.

[137] M.-S. Chen and S.-W. Wang, "Fuzzy clustering analysis for optimising fuzzy membership functions," *Fuzzy Sets and Systems*, **vol. 103**, pp. 239–254, 1999.

[138] C. Z. Janikow, "Fuzzy decision trees: Issues and methods," *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, **vol. 28** (1), pp. 1–14, 1998.

[139] W. Wang and S. M. Bridges, "Genetic algorithm optimisation of membership functions for mining fuzzy association rules," in *Proceedings of the 7th International Joint Conference on Information Systems, Fuzzy Theory and Technology* (1999).

[140] S. Guillaume, "Designing fuzzy inference systems from data: an interpretability-oriented review," *IEEE Transactions on Fuzzy Systems*, **vol. 9** (3), pp. 426–443, 2001.

[141] K. sang Tang, K. fung Man, Z. feng Liu and S. Kwong, "Minimal fuzzy memberships and rules using hierarchical genetic algorithms," *IEEE Transactions on Industrial Electronics*, **vol. 45** (1), pp. 162–169, 1998.

[142] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Transactions on Systems, Man and Cybernetics*, **vol. 22** (6), pp. 1414–1427, 1992.

[143] International Technical Committee, "Operational availability of railway control systems," Tech. Rep. 2, Institution of Railway Signal Engineers, 2006.
URL `http://www.irse.org`

[144] V. Puig and J. Quevedo, “Passive robust fault detection using fuzzy parity relations,” *Mathematics and Computers in Simulation*, **vol. 60**, pp. 193–207, 2002.

[145] D. Dvorak and B. Kuipers, “Model-based monitoring of dynamic systems,” in *Proc. 11th Int. Joint Conf. on Artificial Intelligence (IJCAI-89)*, pp. 1238–1243 (Morgan Kaufmann, San Mateo, CA, 1989).
URL `citeseer.ist.psu.edu/dvorak89modelbased.html`

Appendix A

Details of the design process

This section describes in detail the process which was carried out in order to arrive at the final design.

Initially, a set of requirements was developed, covering all aspects of the design problem, including consideration of the application environment. This requirements set was derived from the initial brief of the thesis, as shown in figure A.1.

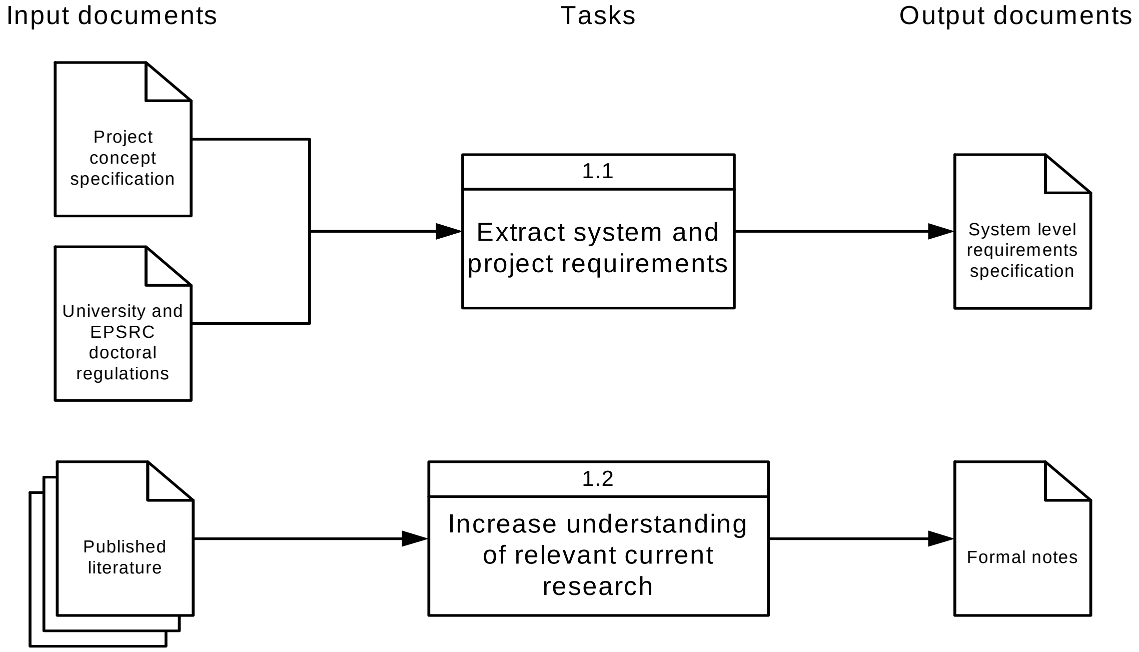
Input documents
Tasks
Output documents
Project concept specification
University and EPSRC doctoral regulations
1.1
Extract system and project requirements
System level requirements specification
Published literature
1.2
Increase understanding of relevant current research
Formal notes

Figure A.1: Phase I of the thesis - initial requirements definition

The function of the system was defined such that all the requirements are met. The requirements and functions were refined to progressively finer levels of detail by evaluating options, choosing a solution and recording the decision as a new set of requirements which apply to the next level of functionality. A system may be defined functionally in several ways; the optimum method is chosen by evaluating the predicted performance of the functions against the requirements. The best performing functional definition was chosen. Figure A.2 shows the flow of this design stage.

Once this process was complete, components were chosen to fulfil the functions of the system. Again, the components were evaluated against the requirements to ensure that the process arrived at the best possible solution. Specifications were made for the correct integration of components, to ensure that interfaces work correctly.

Requirements, functions and components were refined several times, resulting in the definition of progressively finer details of the system. When all components had been defined, implementation could take place. The components were constructed according to their requirements. Then they were connected together to form the physical system. Integration tests are carried out to ensure that all interfaces work correctly. Figure A.3 shows the process of physical design and implementation as it applies to this thesis, which is mostly implemented in software.

Phase IV was the final phase of this systems engineering process, because the theoretical implementation is the final deliverable of the thesis. During this phase, the system was tested against artificial data (allowing changes to be made if necessary) before using real data to formally test the system. The system was checked against its functional design to ensure it conformed to the functional flow specified. The performance of the system was then checked against the requirements to ensure that all requirements were fulfilled. It is then fair to conclude that the design problem is solved and that the system works as required. Figure A.4 shows this testing process.

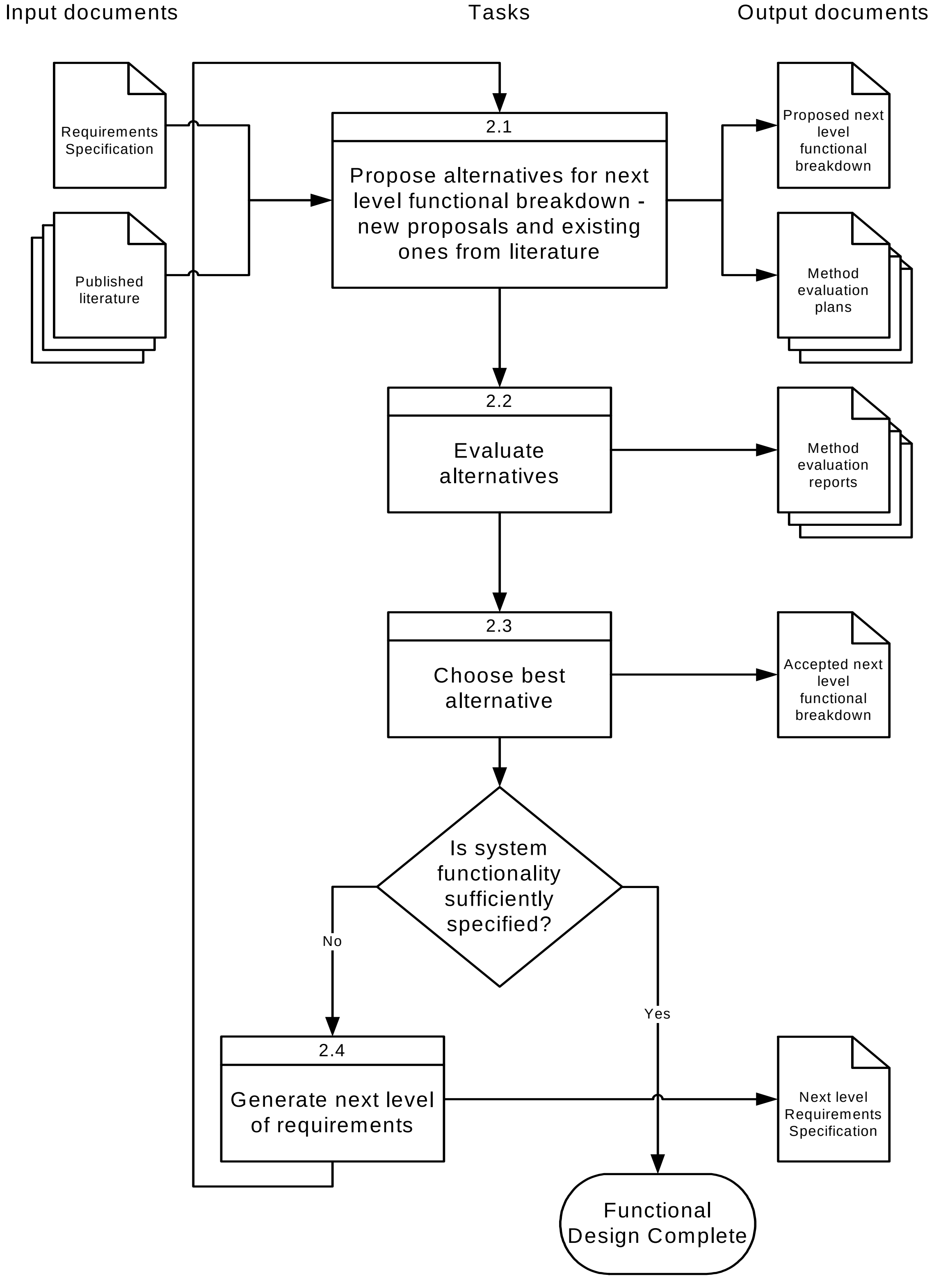
Input documents
Tasks
Output documents
Requirements Specification
Published literature
2.1
Propose alternatives for next level functional breakdown - new proposals and existing ones from literature
Proposed next level functional breakdown
Method evaluation plans
2.2
Evaluate alternatives
Method evaluation reports
2.3
Choose best alternative
Accepted next level functional breakdown
Is system functionality sufficiently specified?
No
Yes
2.4
Generate next level of requirements
Next level Requirements Specification
Functional Design Complete

Figure A.2: Phase II of the thesis - detailed system definition

Documentation was maintained throughout this process in order to keep track of design decisions. Changes to the system were controlled and recorded to ensure that

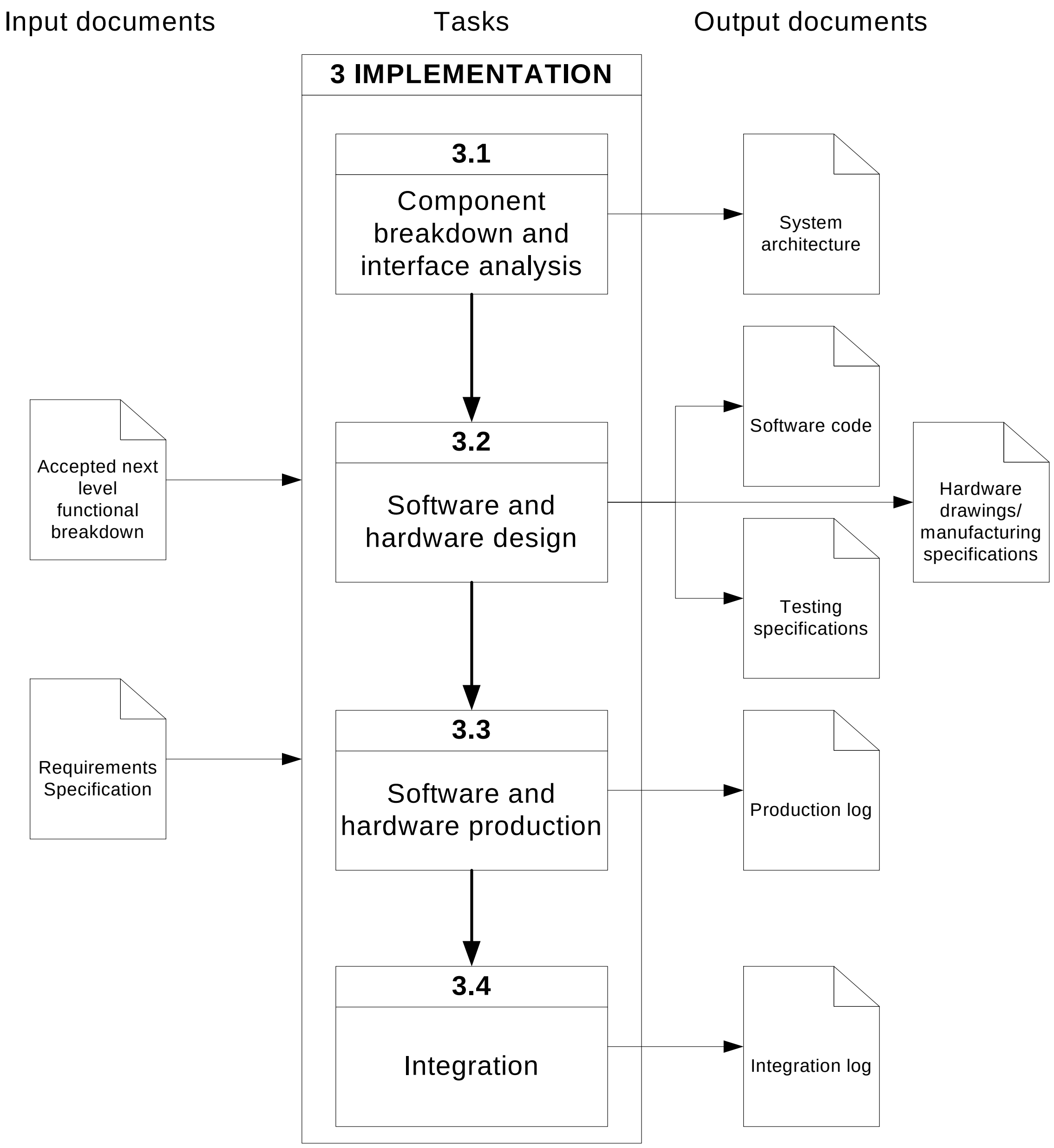
Input documents
Tasks
Output documents
3 IMPLEMENTATION
3.1
Component breakdown and interface analysis
System architecture
Accepted next level functional breakdown
3.2
Software and hardware design
Software code
Hardware drawings/ manufacturing specifications
Testing specifications
Requirements Specification
3.3
Software and hardware production
Production log
3.4
Integration
Integration log

Figure A.3: Phase III of the thesis - implementation

they did not interfere with any requirements.

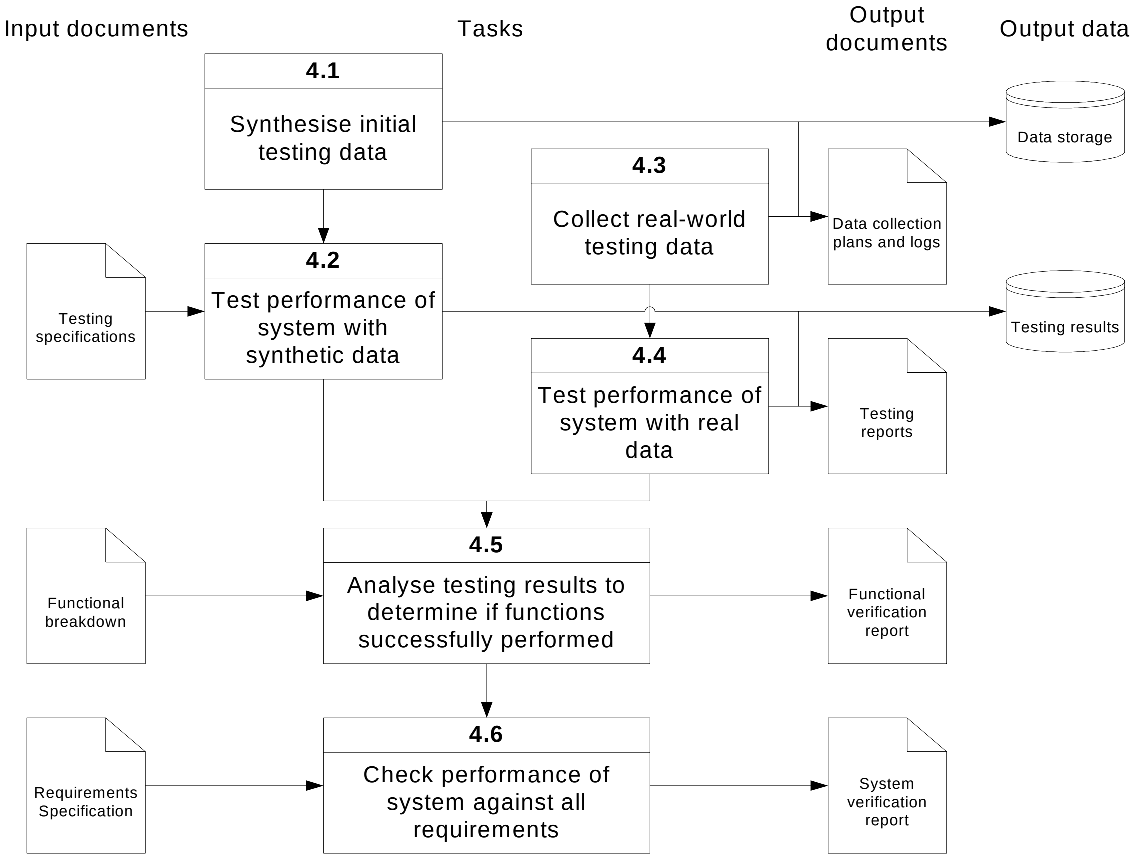
Input documents
Tasks
Output documents
Output data
4.1
Synthesise initial testing data
Data storage
4.3
Collect real-world testing data
Data collection plans and logs
Testing specifications
4.2
Test performance of system with synthetic data
Testing results
4.4
Test performance of system with real data
Testing reports
Functional breakdown
4.5
Analyse testing results to determine if functions successfully performed
Functional verification report
Requirements Specification
4.6
Check performance of system against all requirements
System verification report

Figure A.4: Phase IV of the thesis - testing and verification

Appendix B

Conventions for mathematical notation

B.1 Subscripts

Subscripts for symbols shall indicate either an index value within a vector or matrix, or some sort of qualification to the symbol. The convention shall be that qualification symbols shall be in unenclosed text, and indices shall be enclosed within brackets even if only one index is displayed where more than one might be displayed.

Where a symbol has a subscript indicating that it is a numbered element of a matrix, the coordinates shall be written in triangular brackets. Subscripts to matrix or vector symbols shall be enclosed in parentheses. An example of the indexing convention is shown below:

$$\underline{\mathbf{A}} = \begin{bmatrix} a_{\langle 1,1\rangle} & a_{\langle 1,2\rangle} \\ a_{\langle 2,1\rangle} & a_{\langle 2,2\rangle} \end{bmatrix} \tag{B.1}$$

$$\underline{\mathbf{A}}_{(1,1)} = a_{\langle 1,1\rangle} \tag{B.2}$$

B.2 Columns and rows of matrices

Let

$$\underline{\mathbf{A}} = \begin{bmatrix} a_{\langle 1,1\rangle} & a_{\langle 1,2\rangle} \\ a_{\langle 2,1\rangle} & a_{\langle 2,2\rangle} \end{bmatrix} \tag{B.3}$$

Sometimes it is necessary to isolate a column or row from $\underline{\mathbf{A}}$ for a separate operation. If an operation were to be carried out on the vector formed by the first column of $\underline{\mathbf{A}}$, this portion of the matrix shall be written thus:

$$\underline{\mathbf{A}}_{(:,1)} = \begin{bmatrix} a_{\langle 1,1\rangle} \\ a_{\langle 2,1\rangle} \end{bmatrix} \tag{B.4}$$

The colon represents all values along a particular dimension, between two limits. If no limits are displayed, as in the above example, this means that all elements along that dimension are included. For example, if there were a 10 × 10 matrix $\underline{\mathbf{B}}$, the notation $\underline{\mathbf{B}}_{(2:5,1)}$ would represent a column vector containing elements 2 to 5 of the first column of $\underline{\mathbf{B}}$.

This notation has been chosen because these functions are to be implemented in MATLAB, and this form of notation is analogous to the indexing notation used in MATLAB code.

B.3 Element-wise operation

Most of the functions in the following sections operate element-wise on vectors extracted, using the method in the previous section, from larger matrices. An element-wise operation shall be denoted by a circumflex accent ($\hat{}$) on top of the normal operator or, in function notation, the parentheses containing the operand.

For example:

$$\text{Element-wise multiplication } \underline{\mathbf{c}} = \underline{\mathbf{a}}\hat{.}\underline{\mathbf{b}} \tag{B.5}$$

$$\text{Element-wise function } \underline{\mathbf{c}} = \hat{f}\left(\underline{\mathbf{x}}\right) \tag{B.6}$$

B.4 Dimension notation

In order to avoid confusion, counter symbols are defined in table B.1 for consistent use throughout this document. This should establish firmly which dimension is being dealt with at any one time. When figures replace counters in subscripts, they will always be listed in the order shown, whether or not all dimensions are used.

Symbol	Dimension
i	Fault number
j	Measurement repetitions
k	Time (sample number, interval number or episode number)
l	Measured variable number
m	Element of membership function vector

Table B.1: Table of dimension notations

Appendix C

Result graphs for case study I (pneumatic train door actuator)

This section displays the results graphs for all 12 of the possible actuator combinations.

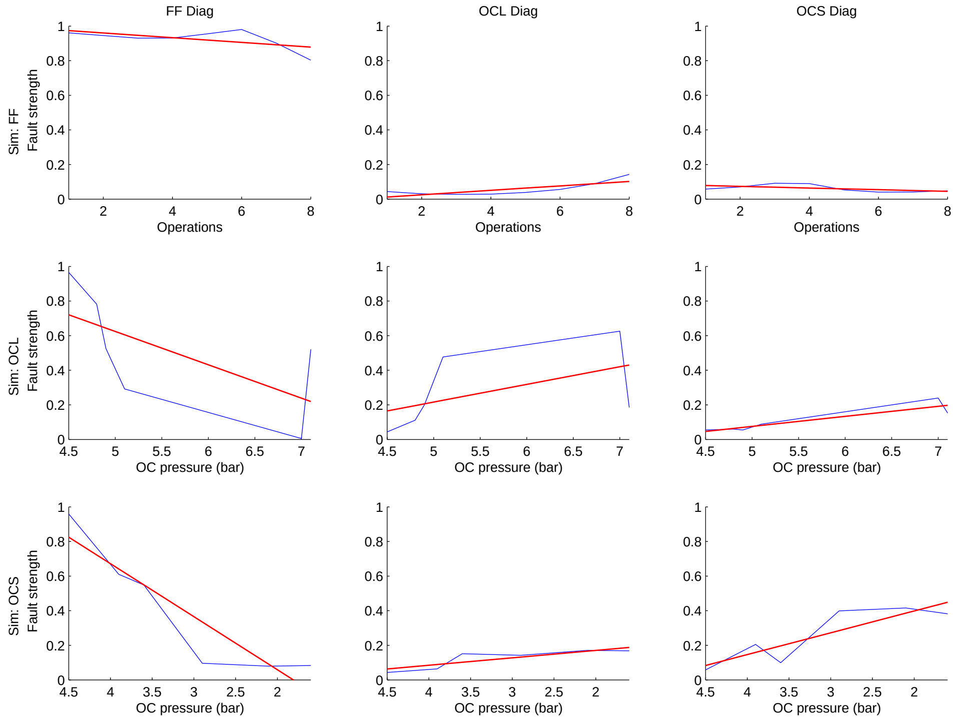

Figure C.1: Results graphs for opening operations, A door as test actuator and B as monitored actuator

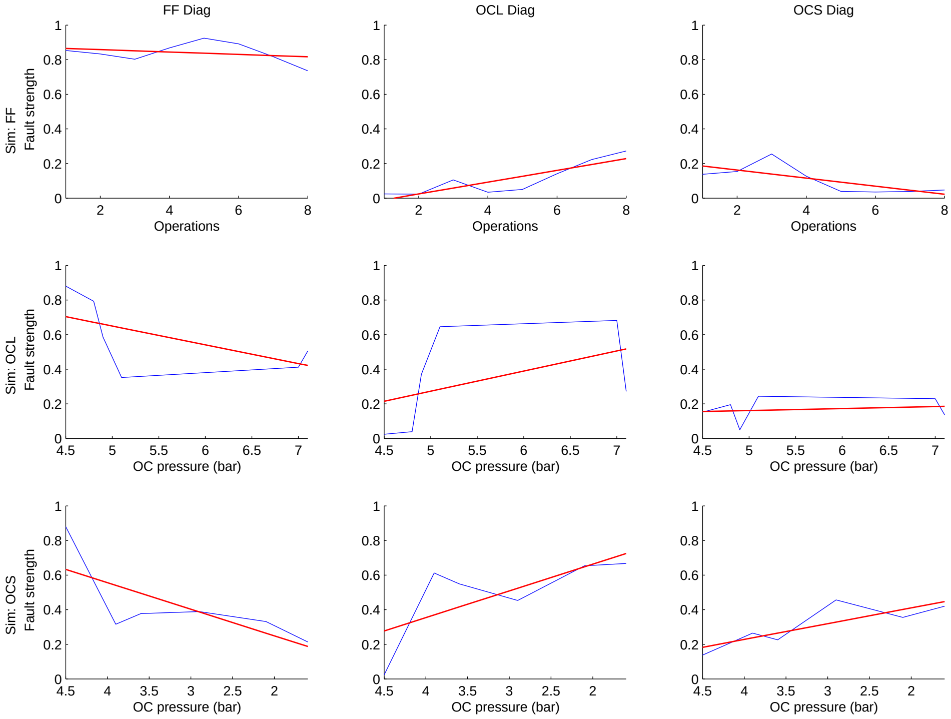

Figure C.2: Results graphs for closing operations, A door as test actuator and B as monitored actuator

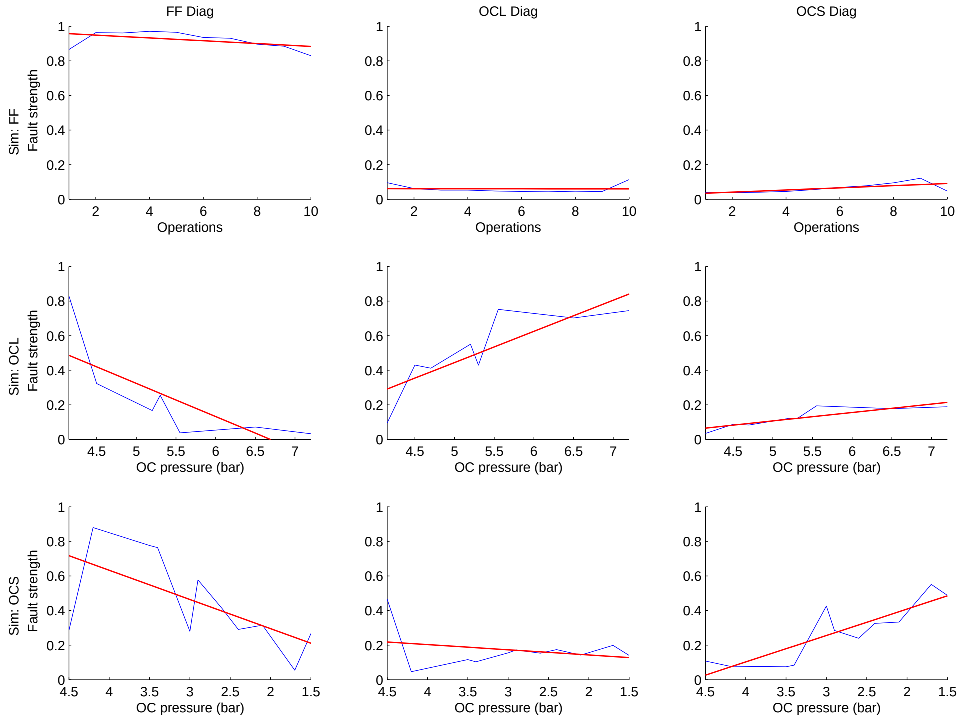

Figure C.3: Results graphs for opening operations, A door as test actuator and C as monitored actuator

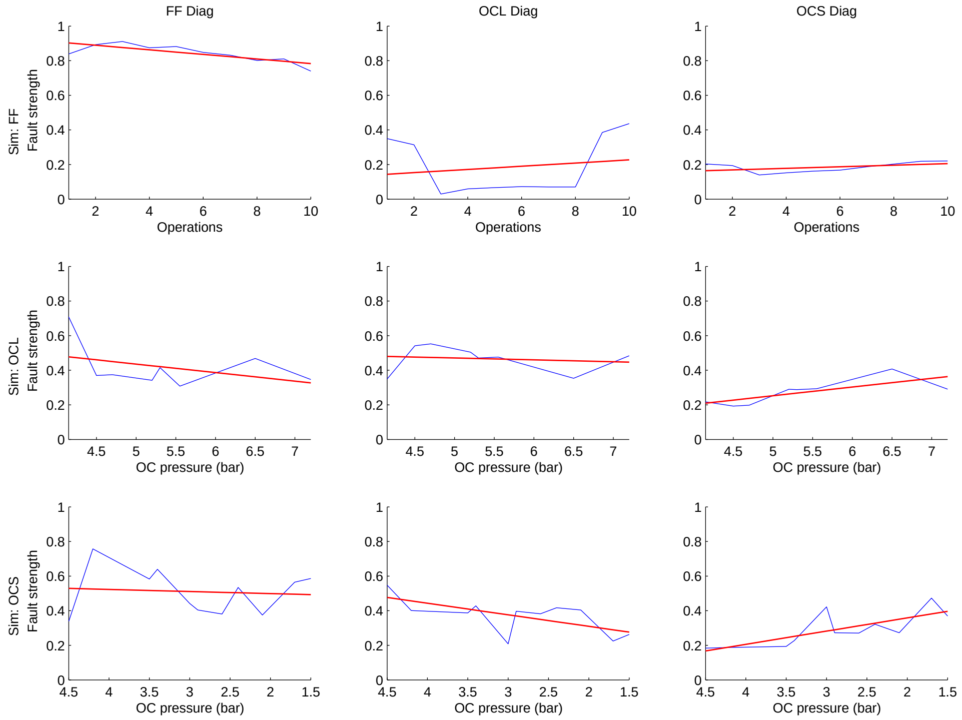

Figure C.4: Results graphs for closing operations, A door as test actuator and C as monitored actuator

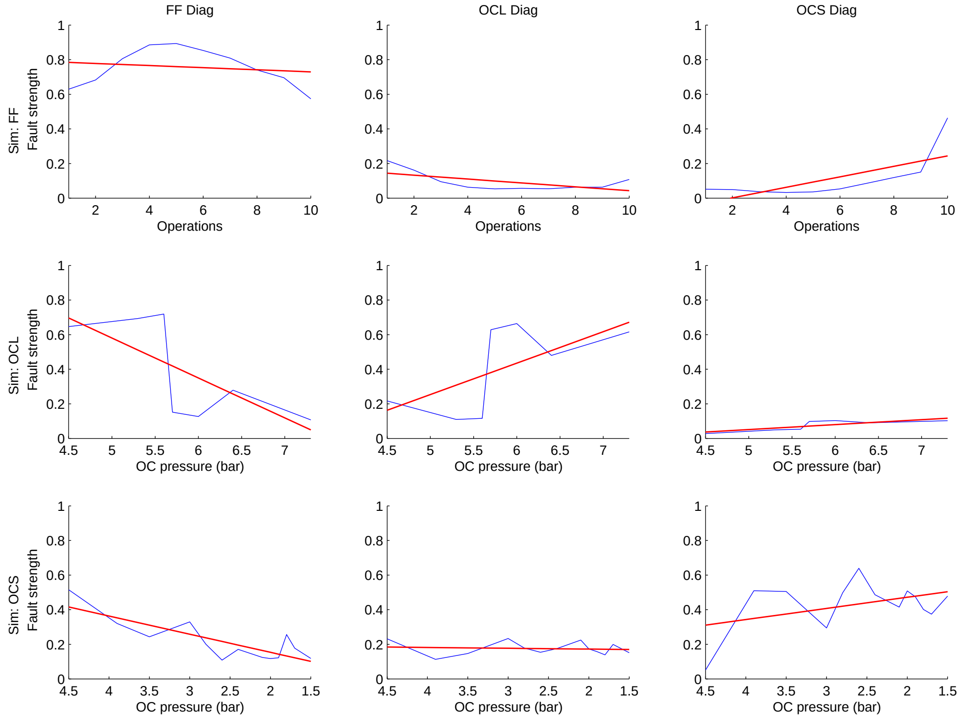

Figure C.5: Results graphs for opening operations, A door as test actuator and D as monitored actuator

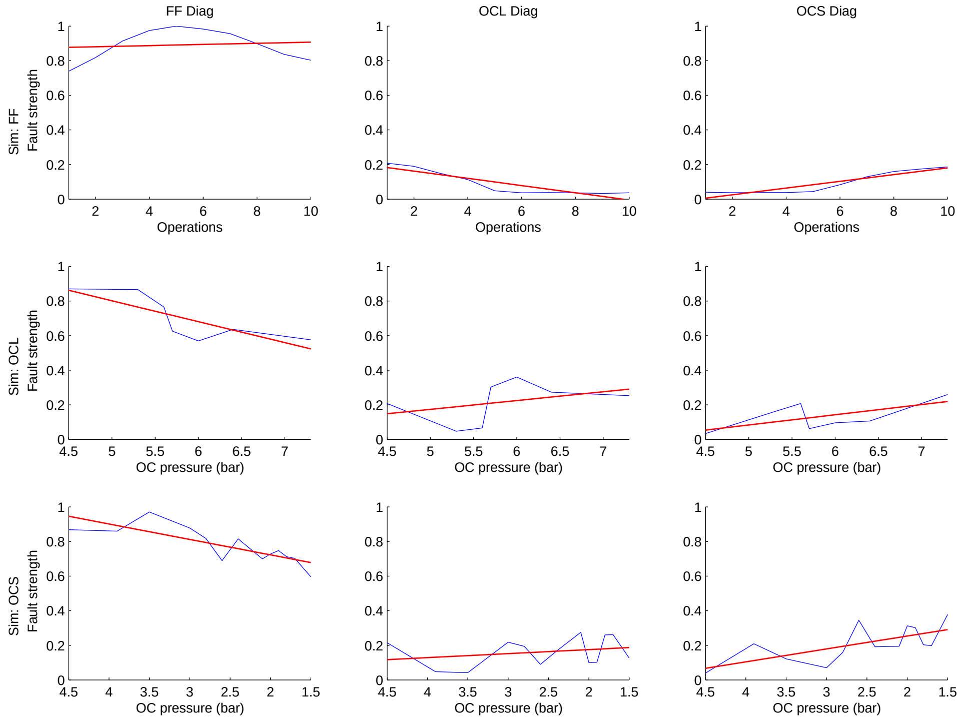

Figure C.6: Results graphs for closing operations, A door as test actuator and D as monitored actuator

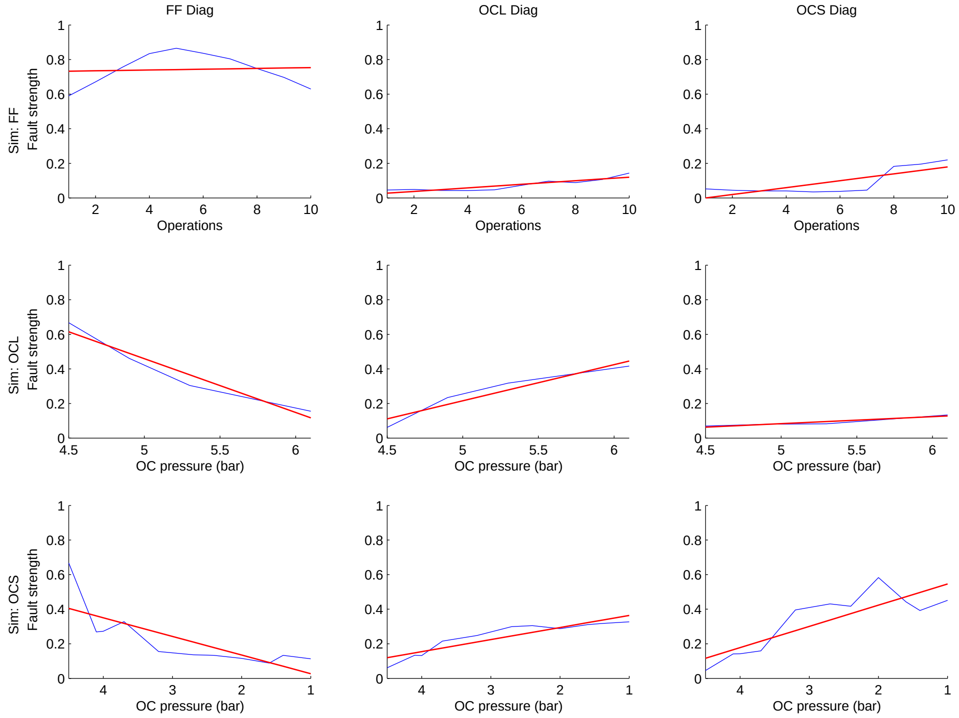

Figure C.7: Results graphs for opening operations, B door as test actuator and A as monitored actuator

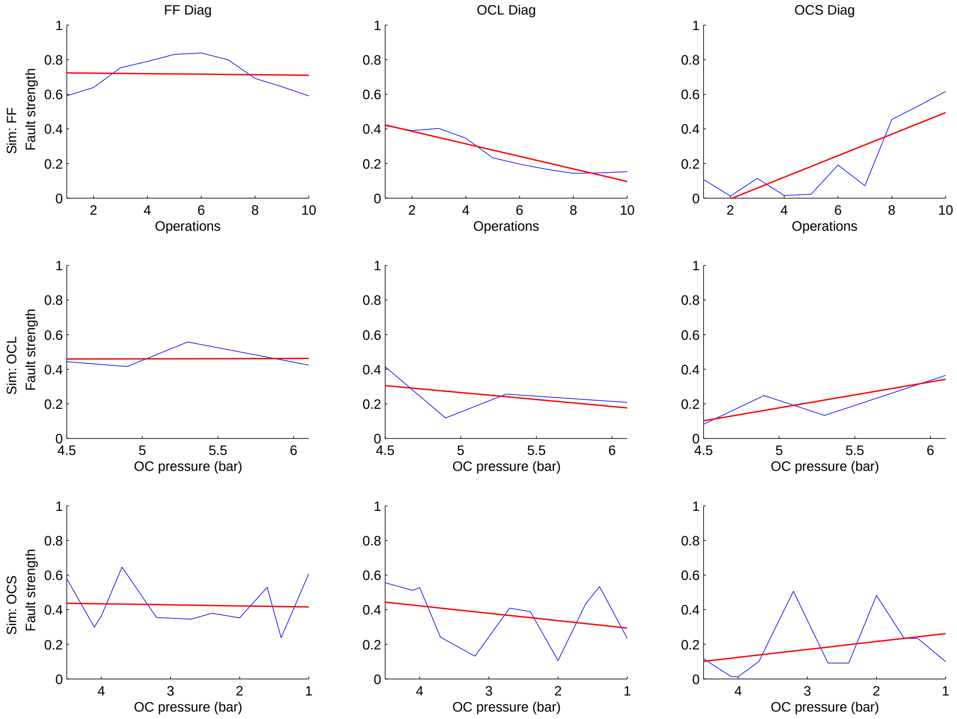

Figure C.8: Results graphs for closing operations, B door as test actuator and A as monitored actuator

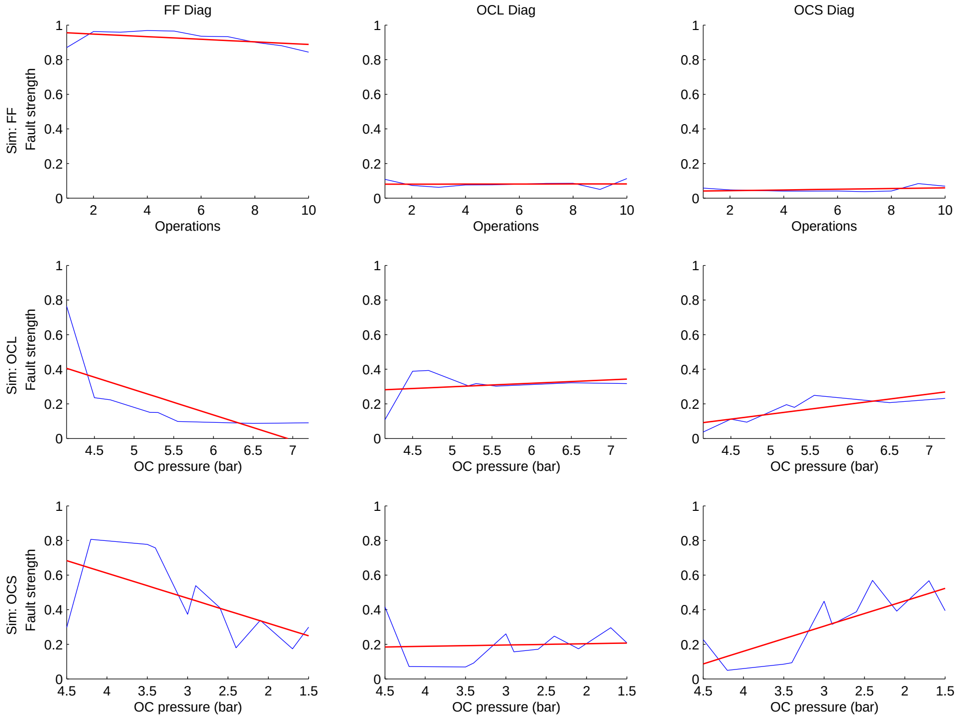

Figure C.9: Results graphs for opening operations, B door as test actuator and C as monitored actuator

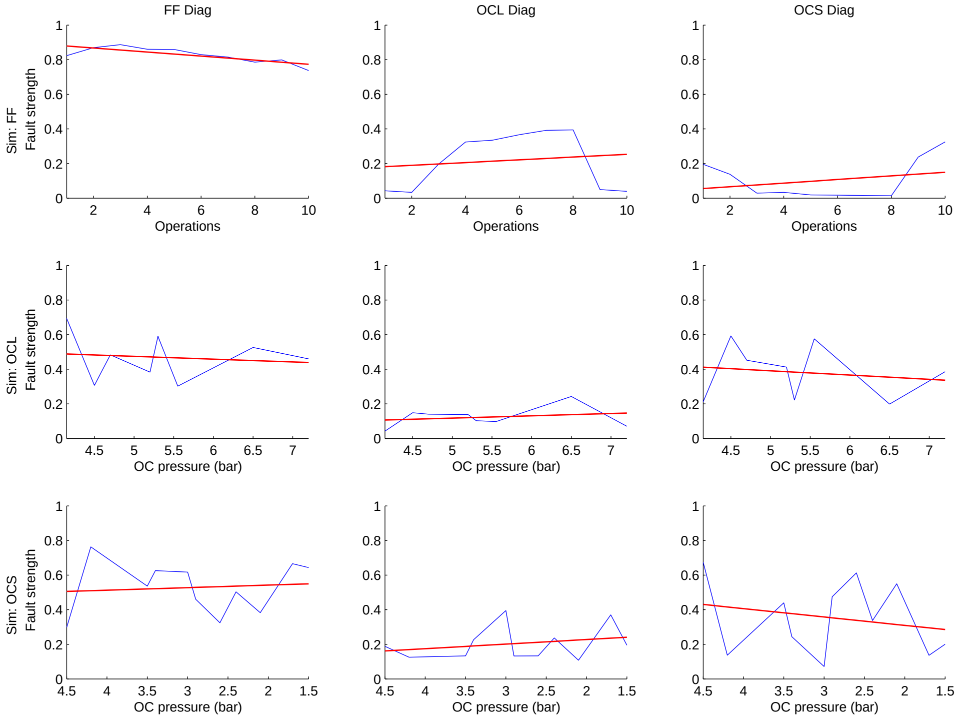

Figure C.10: Results graphs for closing operations, B door as test actuator and C as monitored actuator

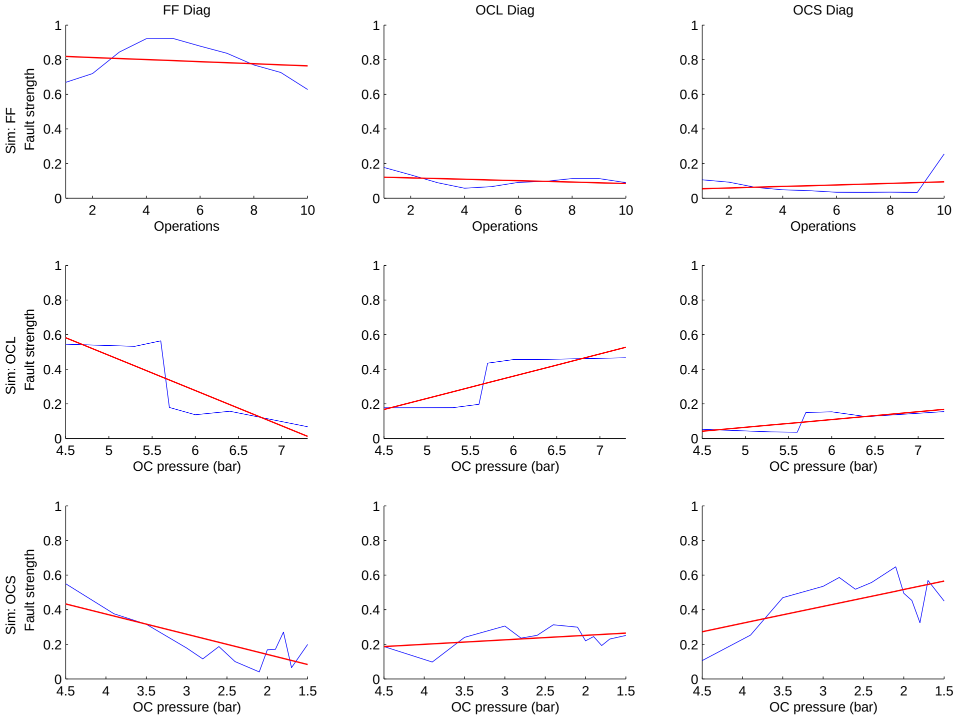

Figure C.11: Results graphs for opening operations, B door as test actuator and D as monitored actuator

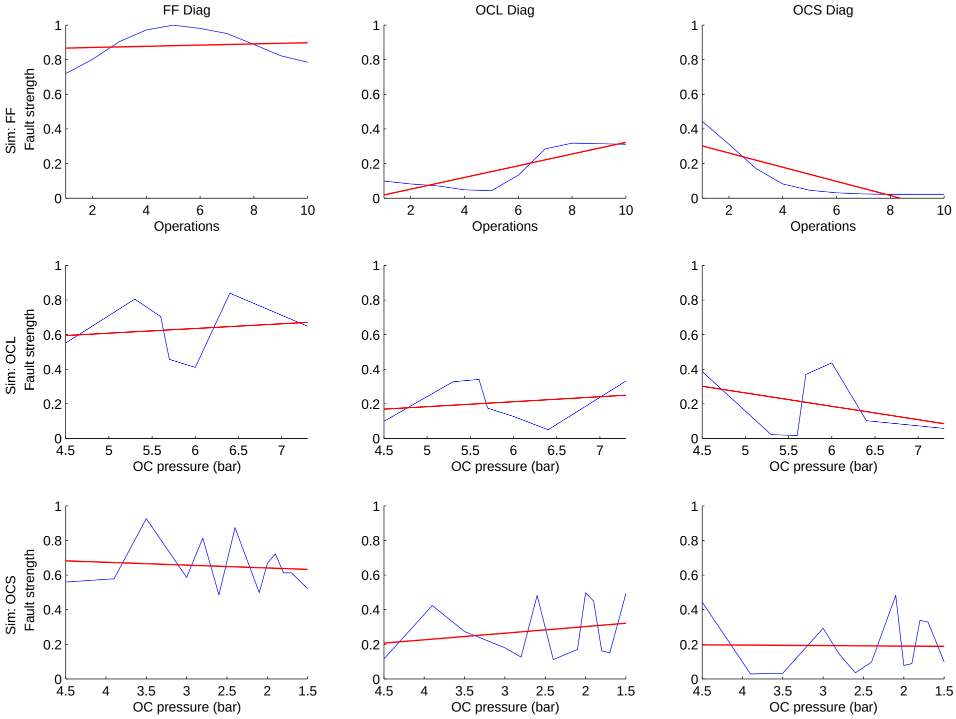

Figure C.12: Results graphs for closing operations, B door as test actuator and D as monitored actuator

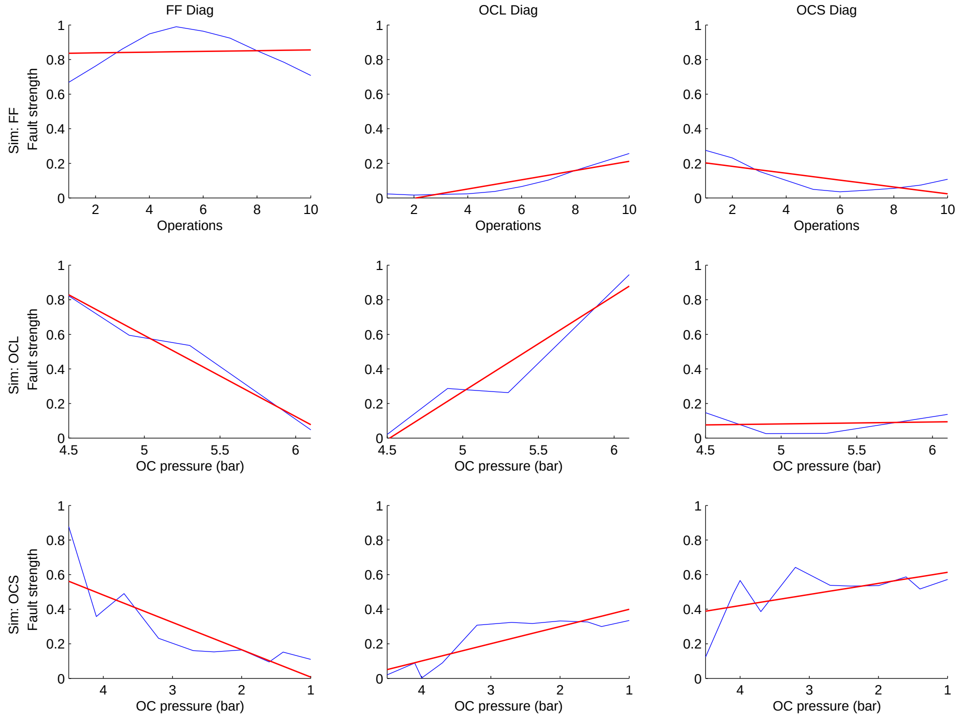

Figure C.13: Results graphs for opening operations, C door as test actuator and A as monitored actuator

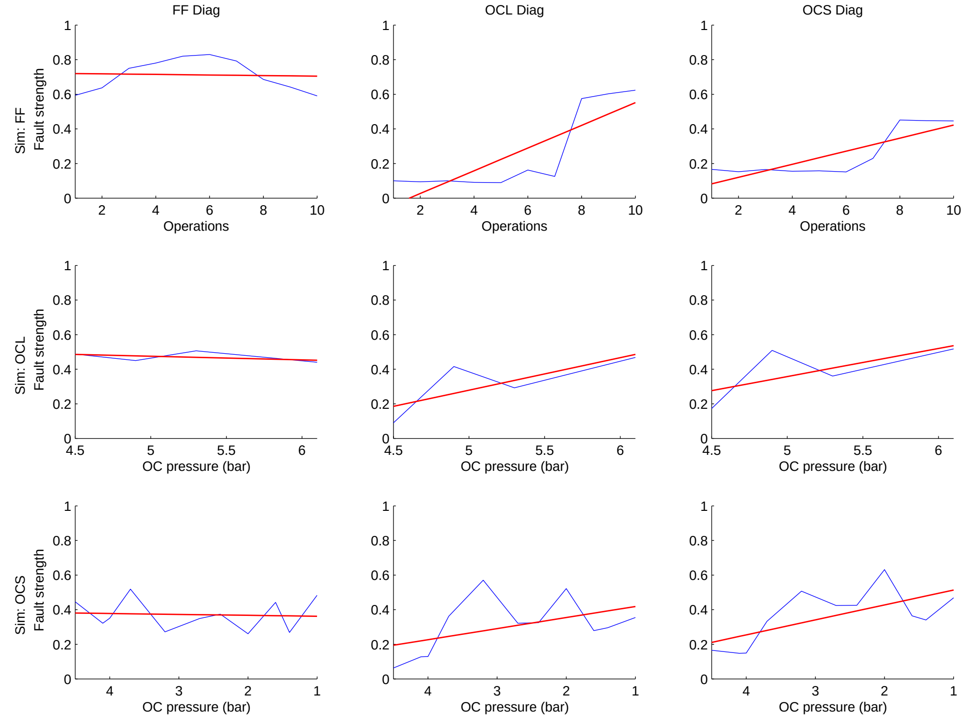

Figure C.14: Results graphs for closing operations, C door as test actuator and A as monitored actuator

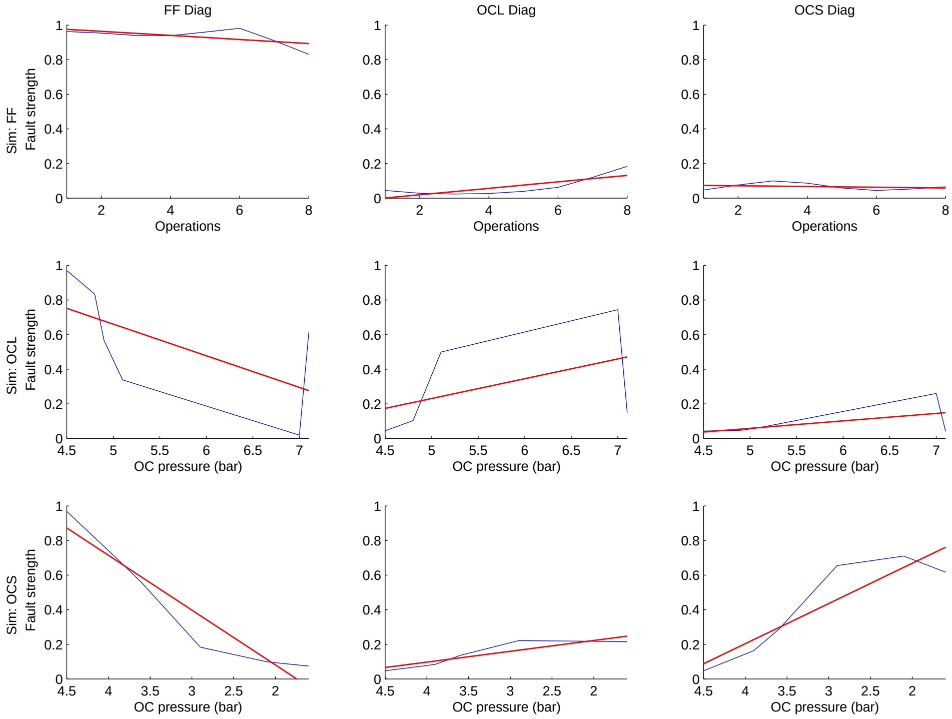

Figure C.15: Results graphs for opening operations, C door as test actuator and B as monitored actuator

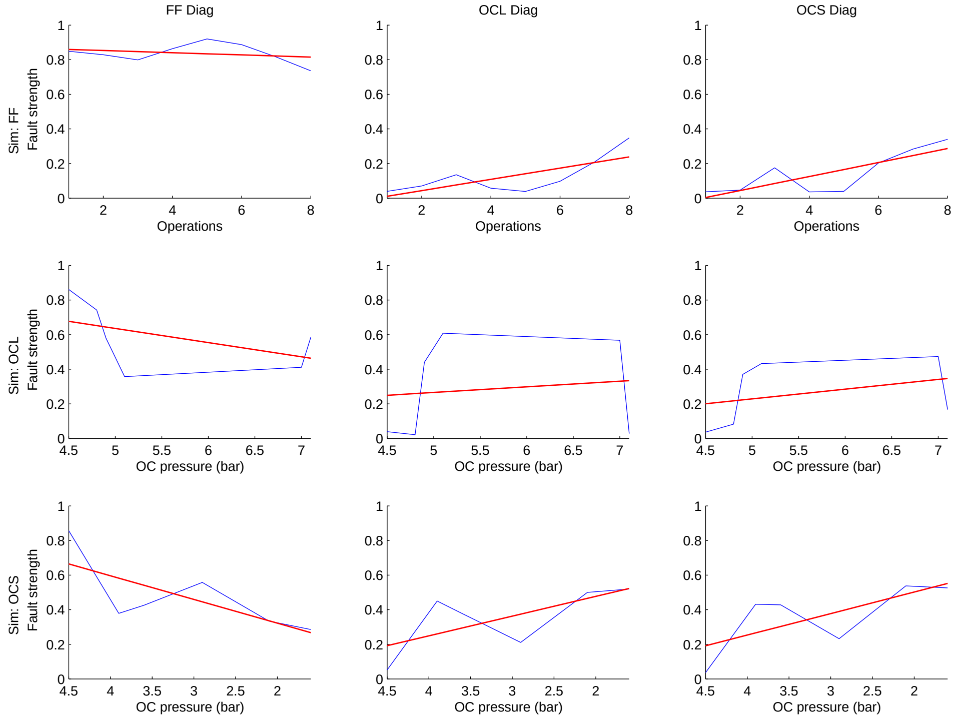

Figure C.16: Results graphs for closing operations, C door as test actuator and B as monitored actuator

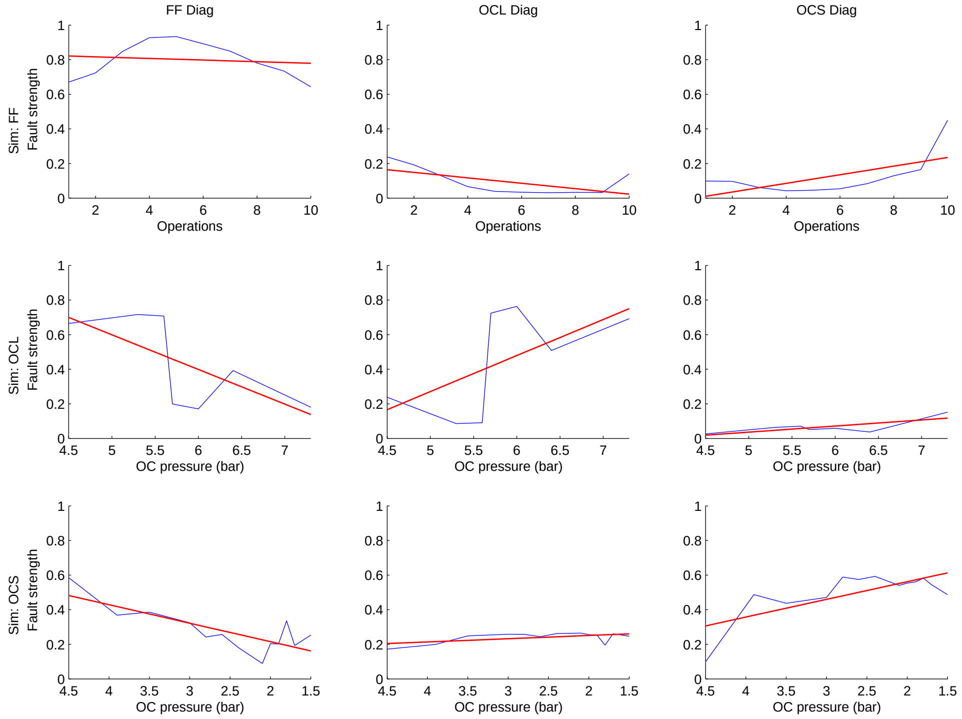

Figure C.17: Results graphs for opening operations, C door as test actuator and D as monitored actuator

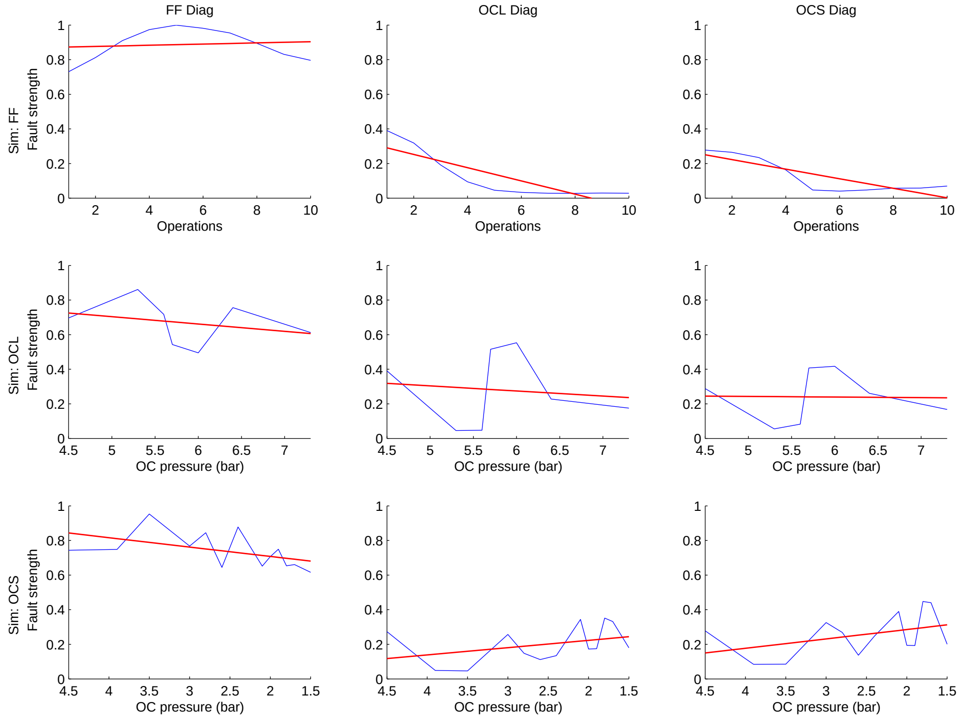

Figure C.18: Results graphs for closing operations, C door as test actuator and D as monitored actuator

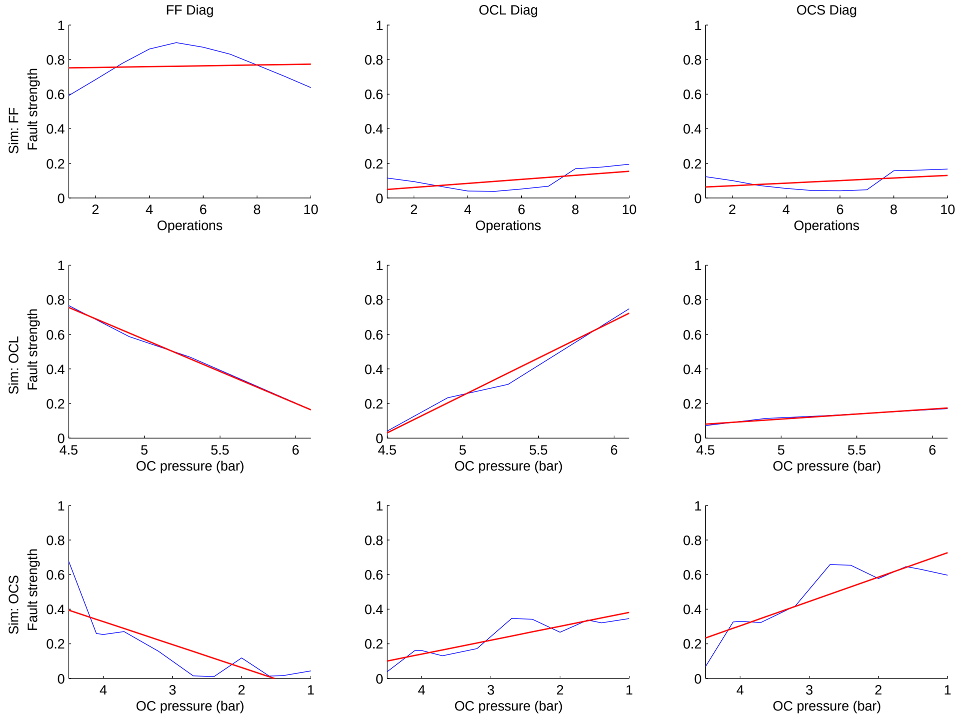

Figure C.19: Results graphs for opening operations, D door as test actuator and A as monitored actuator

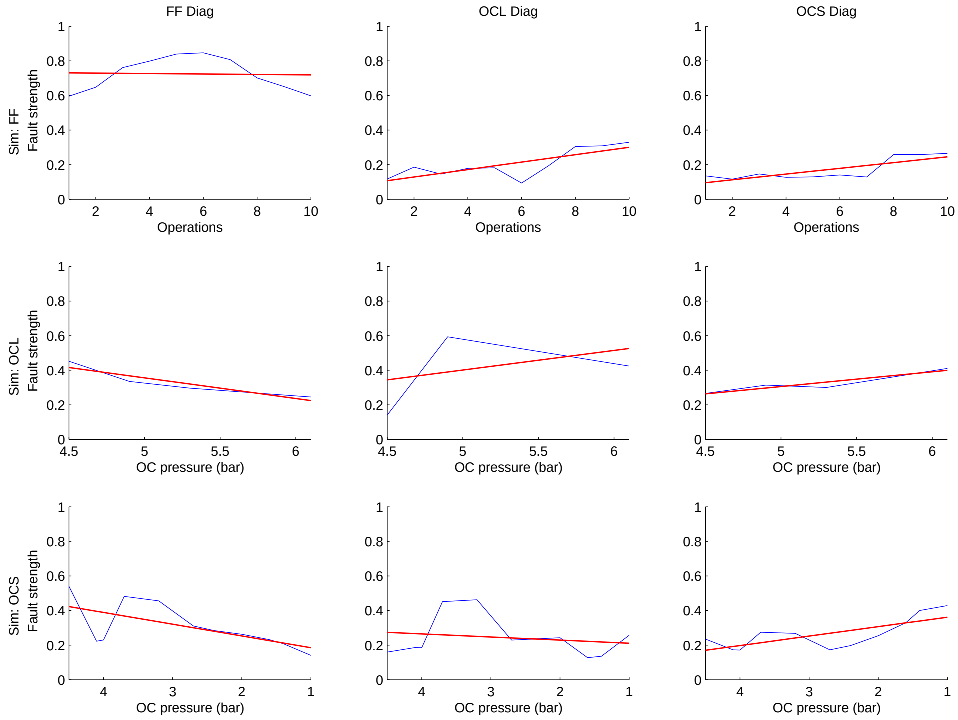

Figure C.20: Results graphs for closing operations, D door as test actuator and A as monitored actuator

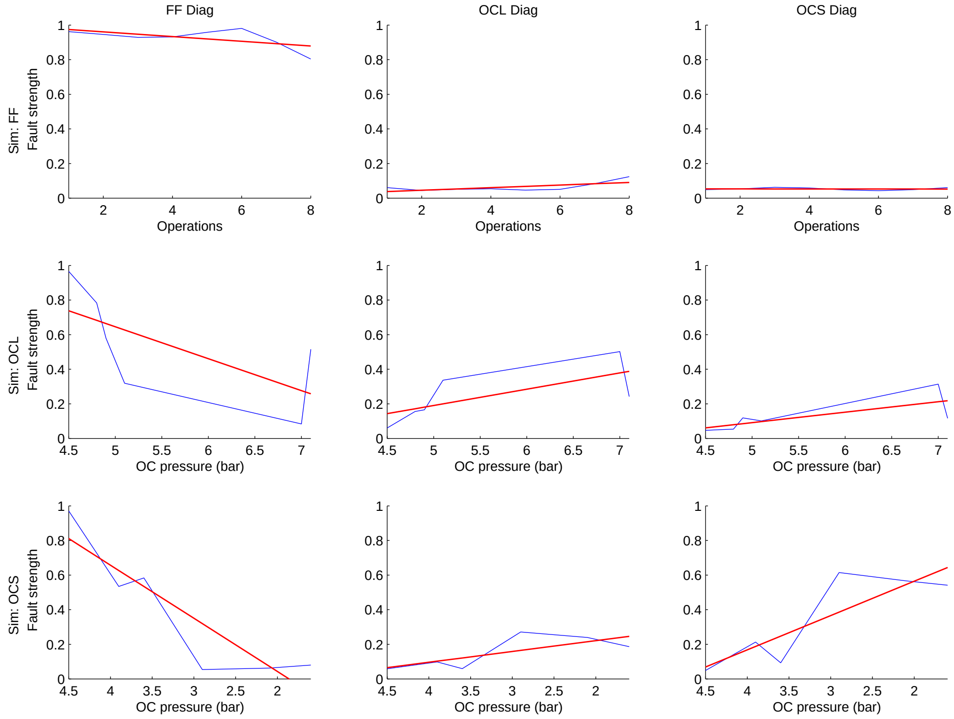

Figure C.21: Results graphs for opening operations, D door as test actuator and B as monitored actuator

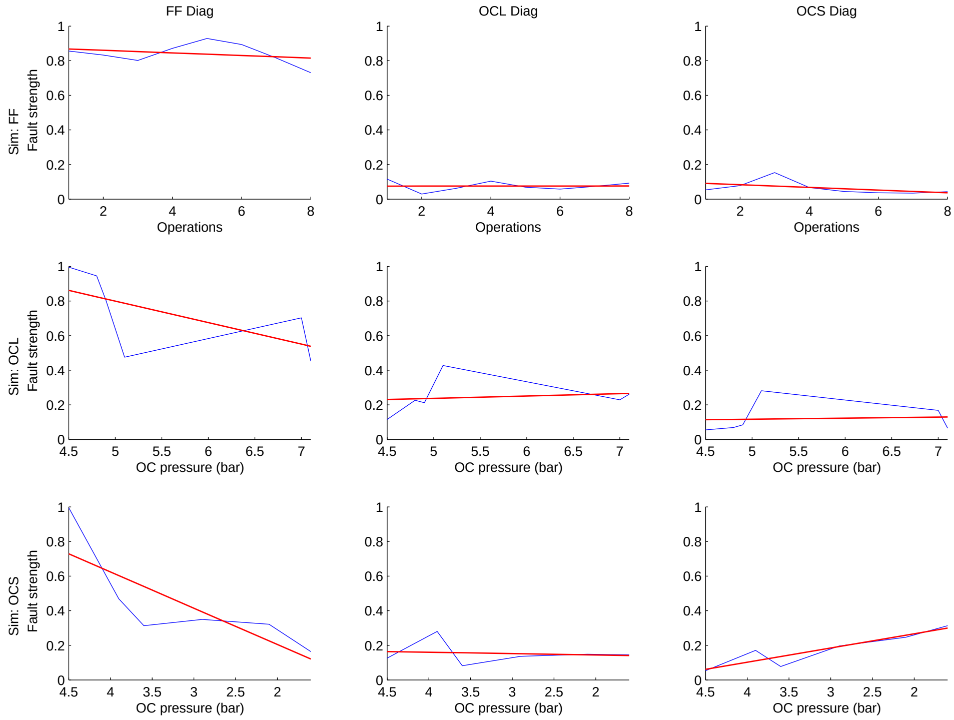

Figure C.22: Results graphs for closing operations, D door as test actuator and B as monitored actuator

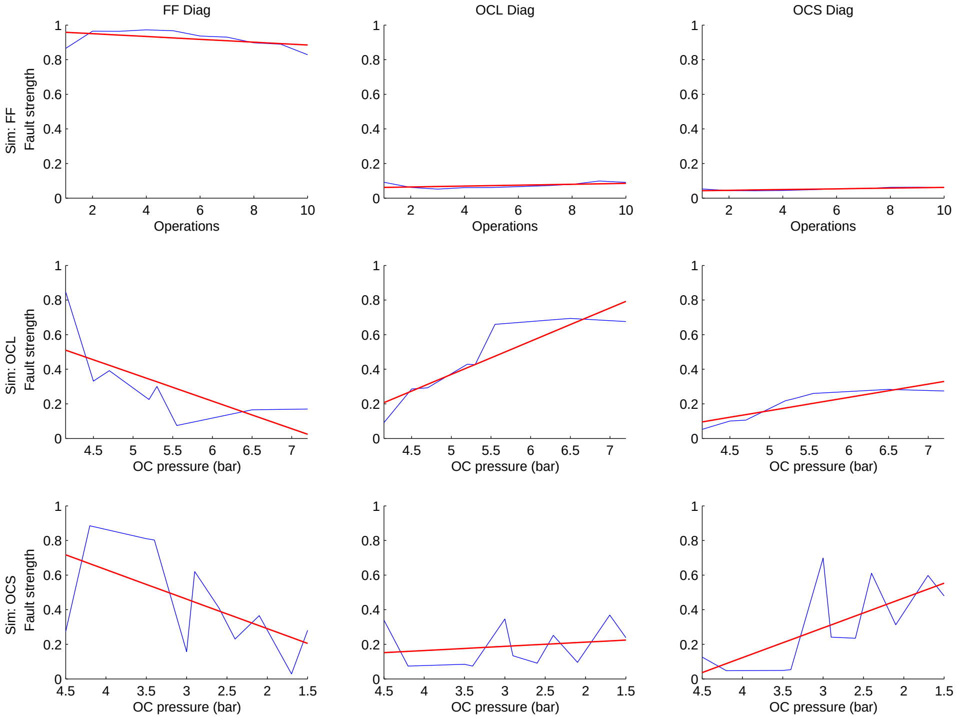

Figure C.23: Results graphs for opening operations, D door as test actuator and C as monitored actuator

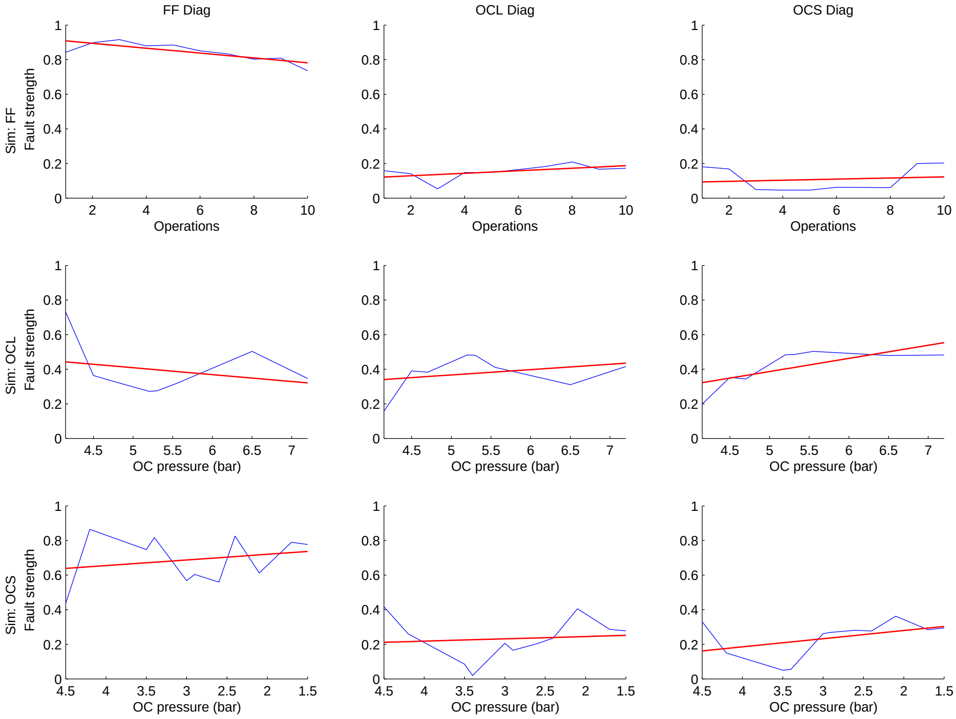

Figure C.24: Results graphs for closing operations, D door as test actuator and C as monitored actuator

Appendix D

Result graphs for case study II (DC electric switch actuator)

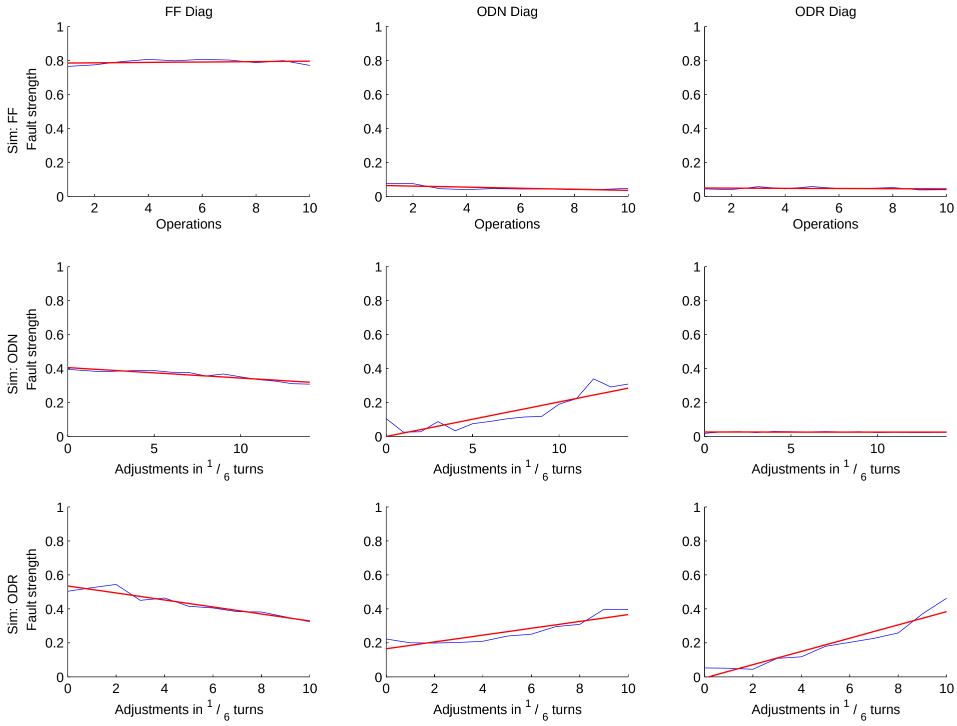

Figure D.1: Results graphs for normal to reverse operations, switch A as test actuator and switch B as monitored actuator

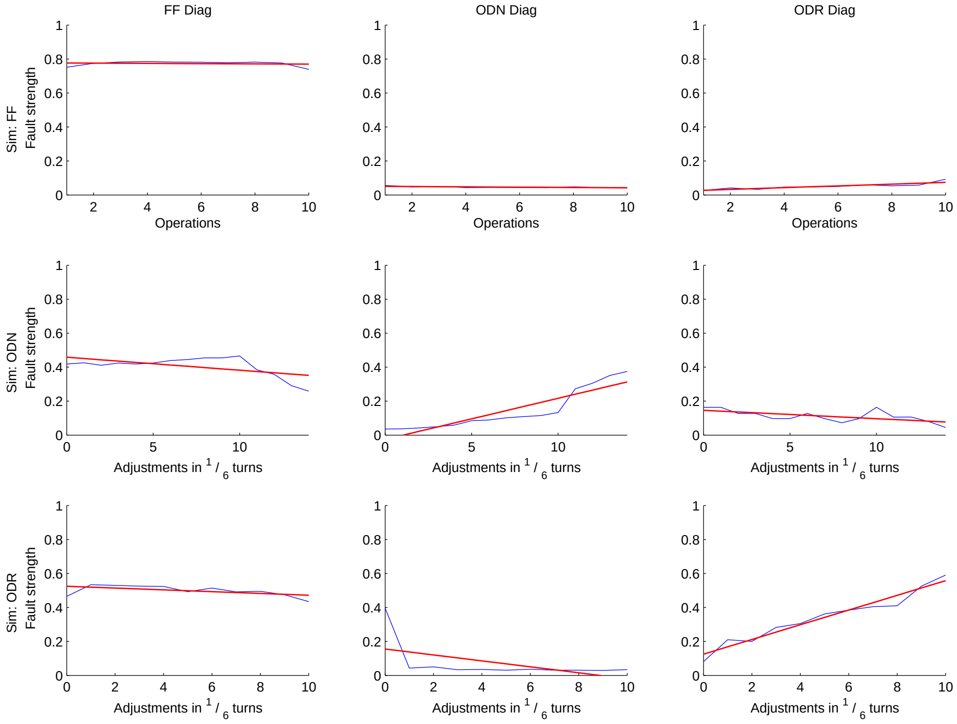

Figure D.2: Results graphs for reverse to normal operations, switch A as test actuator and switch B as monitored actuator

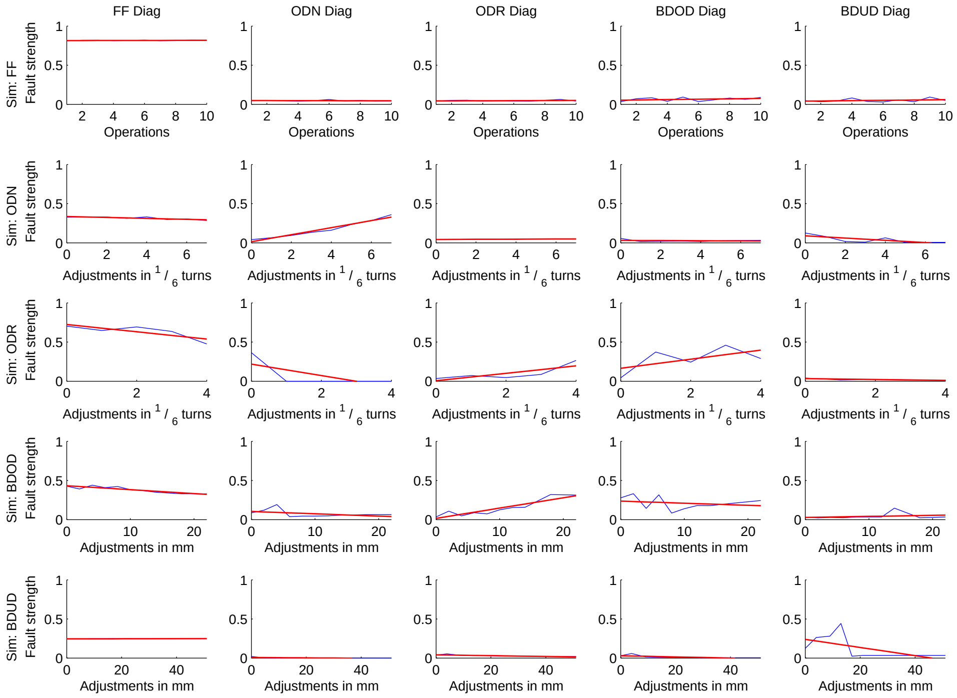

Figure D.3: Results graphs for normal to reverse operations, switch A as test actuator and switch C as monitored actuator

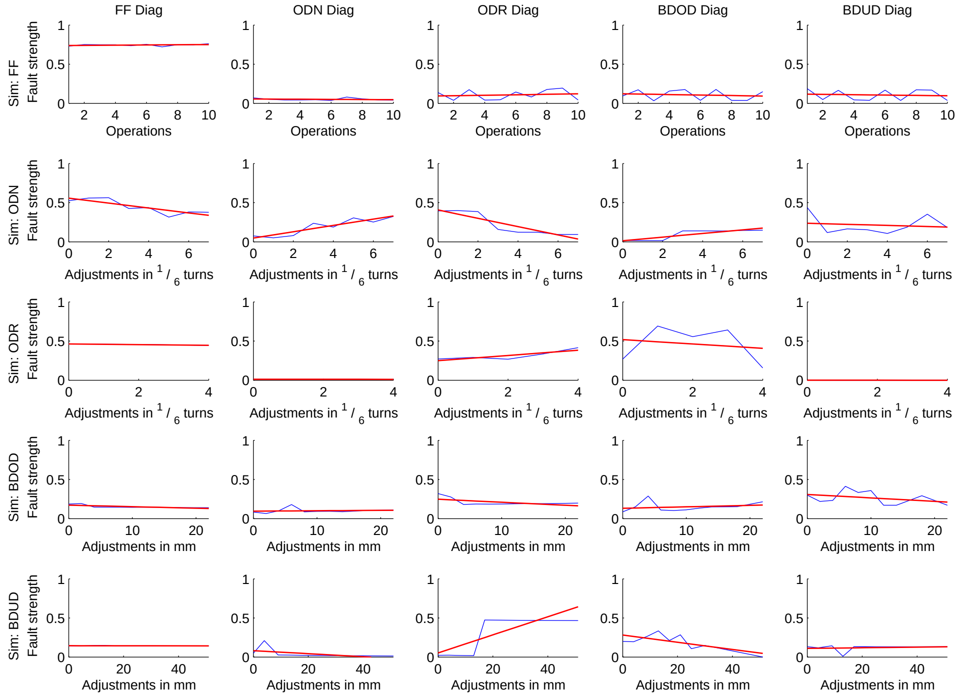

Figure D.4: Results graphs for reverse to normal operations, switch A as test actuator and switch C as monitored actuator

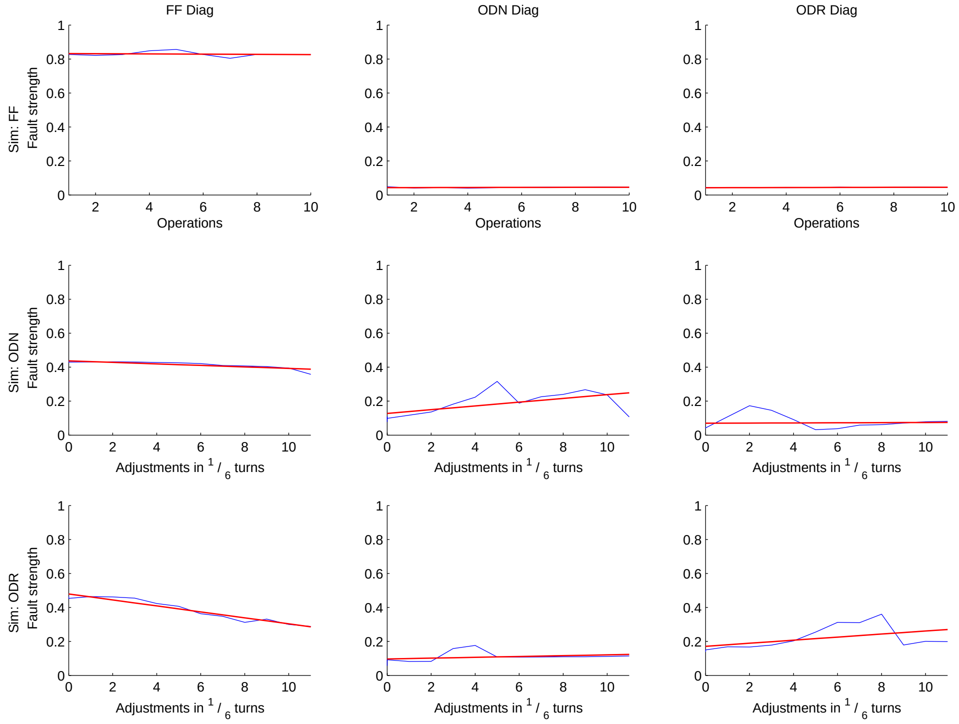

Figure D.5: Results graphs for normal to reverse operations, switch B as test actuator and switch A as monitored actuator

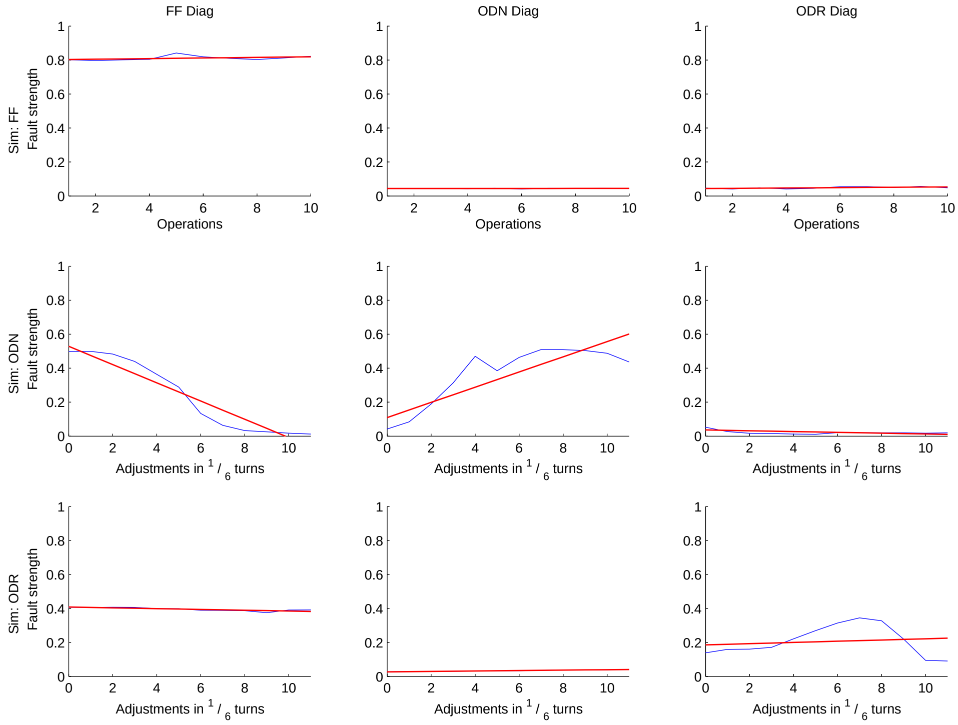

Figure D.6: Results graphs for reverse to normal operations, switch B as test actuator and switch A as monitored actuator

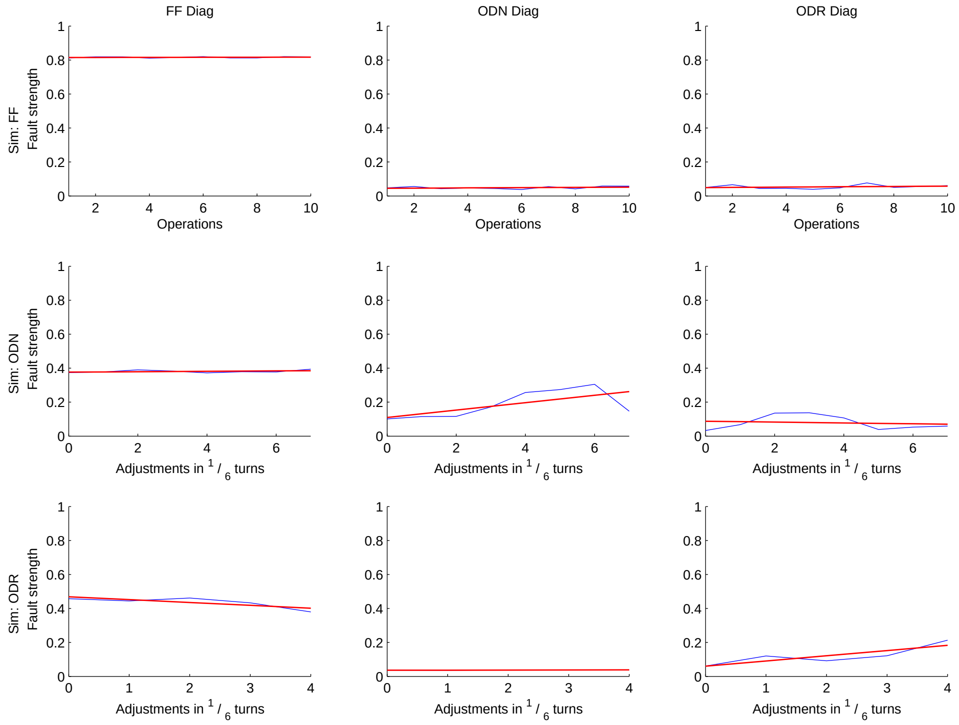

Figure D.7: Results graphs for normal to reverse operations, switch B as test actuator and switch C as monitored actuator

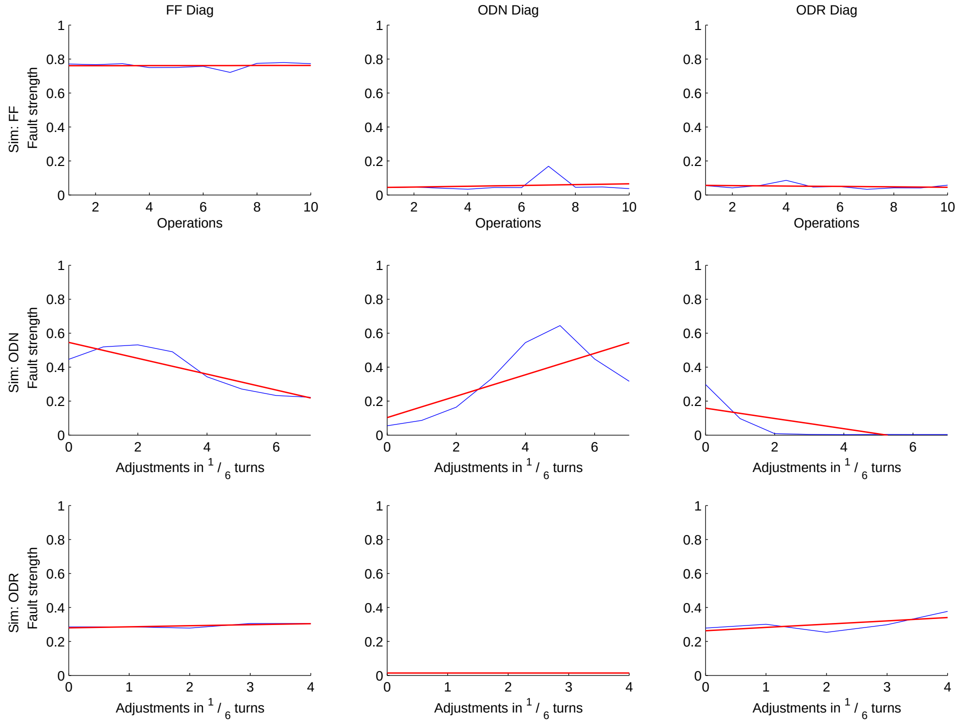

Figure D.8: Results graphs for reverse to normal operations, switch B as test actuator and switch C as monitored actuator

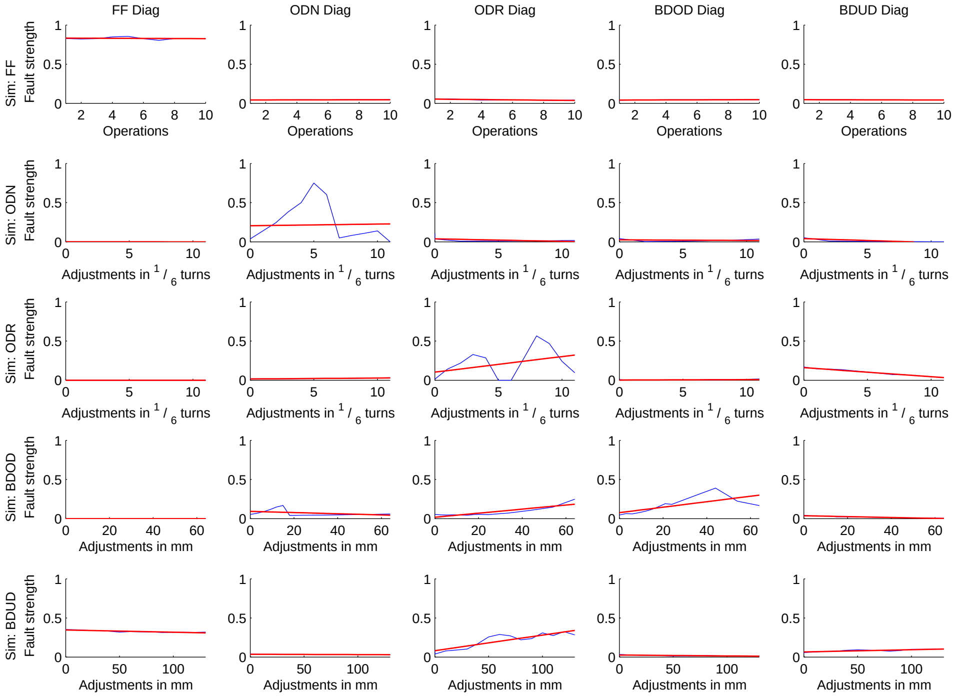

Figure D.9: Results graphs for normal to reverse operations, switch C as test actuator and switch A as monitored actuator

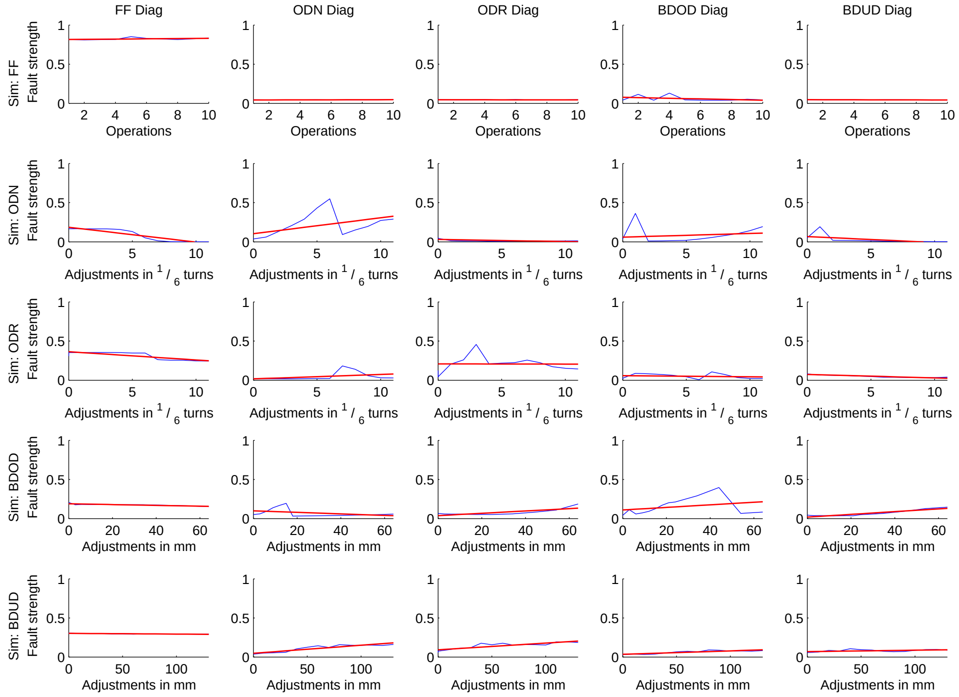

Figure D.10: Results graphs for reverse to normal operations, switch C as test actuator and switch A as monitored actuator

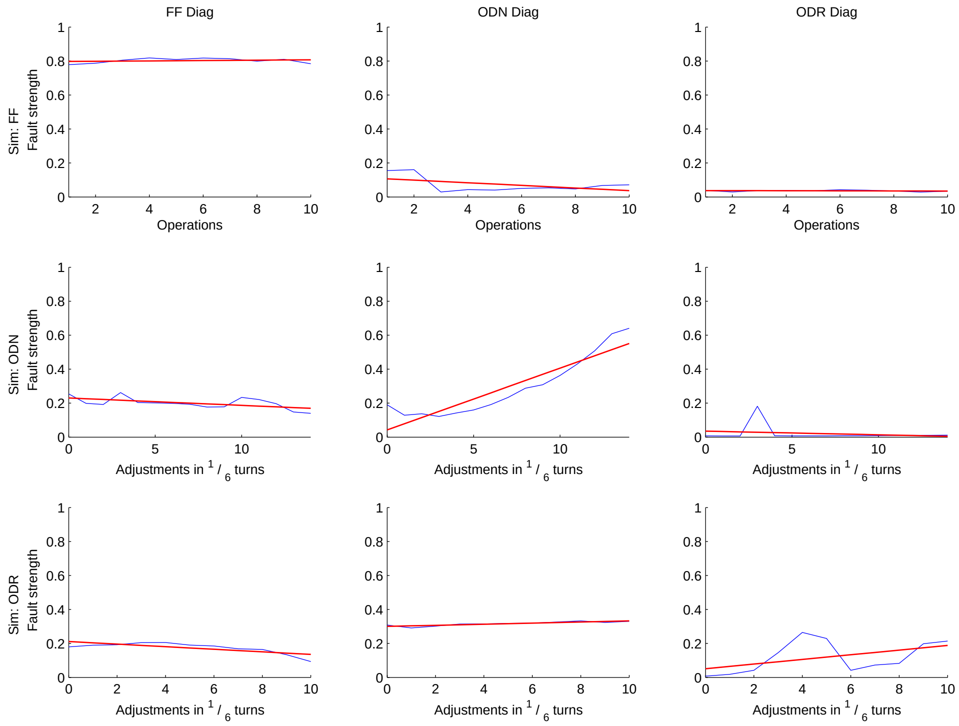

Figure D.11: Results graphs for normal to reverse operations, switch C as test actuator and switch B as monitored actuator

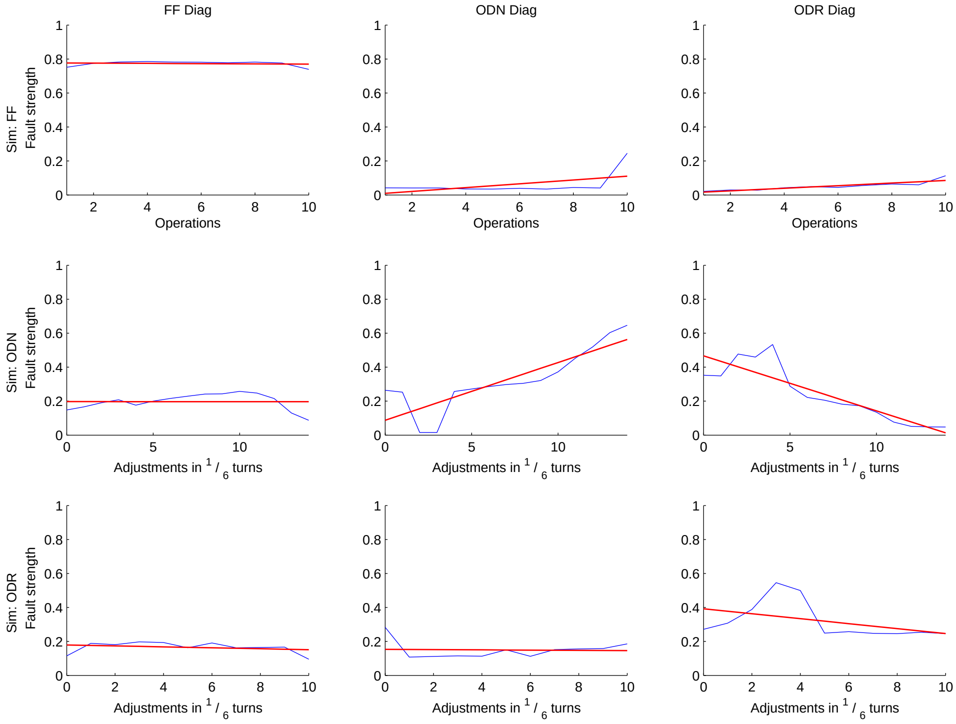

Figure D.12: Results graphs for reverse to normal operations, switch C as test actuator and switch B as monitored actuator