FAULT DIAGNOSIS OF INDUSTRIAL MACHINES USING SENSOR SIGNALS AND CASE-BASED REASONING

Erik Olsson

2009

School of Innovation, Design and Engineering
FAULT DIAGNOSIS OF INDUSTRIAL MACHINES USING SENSOR SIGNALS AND CASE-BASED REASONING

Erik Olsson

Akademisk avhandling

som för avläggande av Teknologie doktorsexamen i Datavetenskap vid Akademin för innovation, design och teknik kommer att offentligen försvaras fredagen 18 september, 2009, 13.15 i Paros, Mälardalens högskola, Västerås.

Fakultetsopponent: Prof. Ashwin Ram, Georgia Institute of Technology, USA

School of Innovation, Design and Engineering
Abstract

Industrial machines sometimes fail to operate as intended. Such failures can be more or less severe depending on the kind of machine and the circumstances of the failure. E.g. the failure of an industrial robot can cause a hold-up of an entire assembly line costing the affected company large amounts of money each minute on hold. Research is rapidly moving forward in the area of artificial intelligence providing methods for efficient fault diagnosis of industrial machines. The nature of fault diagnosis of industrial machines lends itself naturally to case-based reasoning. Case-based reasoning is a method in the discipline of artificial intelligence based on the idea of assembling experience from problems and their solutions as “cases” for reuse in solving future problems. Cases are stored in a case library, available for retrieval and reuse at any time. By collecting sensor data such as acoustic emission and current measurements from a machine and representing this data as the problem part of a case and consequently representing the diagnosed fault as the solution to this problem, a complete series of the events of a machine failure and its diagnosed fault can be stored in a case for future use.
Abstract

Industrial machines sometimes fail to operate as intended. Such failures can be more or less severe depending on the kind of machine and the circumstances of the failure. E.g. the failure of an industrial robot can cause a hold-up of an entire assembly line costing the affected company large amounts of money each minute on hold. Research is rapidly moving forward in the area of artificial intelligence providing methods for efficient fault diagnosis of industrial machines. The nature of fault diagnosis of industrial machines lends itself naturally to case-based reasoning. Case-based reasoning is a method in the discipline of artificial intelligence based on the idea of assembling experience from problems and their solutions as "cases" for reuse in solving future problems. Cases are stored in a case library, available for retrieval and reuse at any time. By collecting sensor data such as acoustic emission and current measurements from a machine and representing this data as the problem part of a case and consequently representing the diagnosed fault as the solution to this problem, a complete series of the events of a machine failure and its diagnosed fault can be stored in a case for future use.

To my family
Preface

I would like to thank all the people who helped me making this thesis a fact. First of all I would like to thank my main and assistant supervisors Peter Funk and Ning Xiong at Mälardalen University, Västerås, Mats Jackson and Marcus Bengtsson at Mälardalen University, Eskilstuna for their support and dedication in my work. They have contributed a great deal to this thesis with lots of ideas and valuable discussions. Without them this thesis work would have been impossible. Secondly, I would like to thank my room colleagues, PhD students and friends Mobyen Ahmed and Shahina Begum. I would also like to thank Rostyslav Stolyarchuk at the State Scientific and Research Institute of Information Infrastructure, Lviv, Ukraine for his cooperation and valuable ideas concerning included paper C, Patrick Wehbi at ABB Robotics for his invaluable help concerning robot programming and my previously sponsoring company ABB Robotics, foremost Mats Åhgren which made the first part of my research possible.

Finally I would like to thank my family and my friends for making my life and work bearable!

Erik Olsson
Västerås, Mars 23, 2009
Preface

I would like to thank all the people who helped me making this thesis a fact. First of all I would like to thank my main and assistant supervisors Peter Funk and Ning Xiong at Mälardalen University, Västerås, Mats Jackson and Marcus Bengtsson at Mälardalen University, Eskilstuna for their support and dedication in my work. They have contributed a great deal to this thesis with lots of ideas and valuable discussions. Without them this thesis work would have been impossible. Secondly, I would like to thank my room colleagues, PhD students and friends Mobyen Ahmed and Shahina Begum. I would also like to thank Rostyslav Stolyarchuk at the State Scientific and Research Institute of Information Infrastructure, Lviv, Ukraine for his cooperation and valuable ideas concerning included paper C, Patrick Wehbi at ABB Robotics for his invaluable help concerning robot programming and my previously sponsoring company ABB Robotics, foremost Mats Åhgren which made the first part of my research possible.

Finally I would like to thank my family and my friends for making my life and work bearable!

Erik Olsson
Västerås, Mars 23, 2009
Publications included in the thesis


E. Olsson and P. Funk. Agent-Based Monitoring using Case-Based Reasoning for Experience Reuse and Improved Quality. Journal of Quality
Publications

Publications included in the thesis


E. Olsson and P. Funk. Agent-Based Monitoring using Case-Based Reasoning for Experience Reuse and Improved Quality. Journal of Quality

Publications not included in the thesis


E. Olsson, Mobyen U. Ahmed, P. Funk and N. Xiong, A Case Study of Communication in a Distributed Multi-Agent System in a Factory Production Environment, The 20th International Congress and Exhibition on Condition Monitoring and Diagnostics Engineering Management, Co-


Contents

1 Thesis

1.1 Research Questions

1.2 Research Contributions

1.3 Outline of Thesis

2 Theoretical Framework

2.1 Background

2.2 Introduction

2.3 Sensor Signals

2.3.1 Acoustic Emission

2.3.2 Discovered Fault Symptoms

2.4 Signal Pre-Processing

2.4.1 Bandwidth Filtering

2.4.2 The Discrete and Fast Fourier Transform

2.4.3 The Discrete Wavelet Transform

2.5 Signal Feature Extraction

2.5.1 Basic Signal Features

2.5.2 Wavelet Transform

2.5.3 Fourier Transform

2.5.4 Signal Thresholding

2.5.5 Standard Deviation

2.5.6 Feature Discrimination

2.5.7 Assembly of a Feature Vector

2.6 Classification
# Contents

## I  Thesis

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
</tr>
</tbody>
</table>

### 1 Introduction

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>5</td>
</tr>
<tr>
<td>1.2</td>
<td>5</td>
</tr>
<tr>
<td>1.3</td>
<td>8</td>
</tr>
</tbody>
</table>

### 2 Theoretical Framework

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>13</td>
</tr>
<tr>
<td>2.3.1</td>
<td>13</td>
</tr>
<tr>
<td>2.3.2</td>
<td>17</td>
</tr>
<tr>
<td>2.3.3</td>
<td>19</td>
</tr>
<tr>
<td>2.3.4</td>
<td>19</td>
</tr>
<tr>
<td>2.4</td>
<td>21</td>
</tr>
<tr>
<td>2.4.1</td>
<td>21</td>
</tr>
<tr>
<td>2.4.2</td>
<td>22</td>
</tr>
<tr>
<td>2.4.3</td>
<td>24</td>
</tr>
<tr>
<td>2.5</td>
<td>25</td>
</tr>
<tr>
<td>2.5.1</td>
<td>25</td>
</tr>
<tr>
<td>2.5.2</td>
<td>26</td>
</tr>
<tr>
<td>2.5.3</td>
<td>26</td>
</tr>
<tr>
<td>2.5.4</td>
<td>26</td>
</tr>
<tr>
<td>2.5.5</td>
<td>27</td>
</tr>
<tr>
<td>2.5.6</td>
<td>28</td>
</tr>
<tr>
<td>2.5.7</td>
<td>28</td>
</tr>
</tbody>
</table>

### 2.6 Classification

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.6</td>
<td>29</td>
</tr>
</tbody>
</table>
2.6.1 Case-Based Classification ........................................ 29
2.6.2 Neural Network Classification ................................. 33
2.6.3 A Neural Network Approach to CBR Classification ...... 36

3 A Comparison Between Five Case-Based Fault Diagnosis Systems for Industrial Machines 37
3.1 Introduction ......................................................... 37
3.2 The Systems .......................................................... 38
  3.2.1 ICARUS A Diagnostic System for Locomotives ...... 38
  3.2.2 Diagnosis of Electronic Circuits ............................ 39
  3.2.3 Satellite Diagnosis ............................................ 41
  3.2.4 Induction Motor Fault Diagnosis ........................... 42
  3.2.5 Diagnosis of Industrial Robots ............................... 43
3.3 Discussion ........................................................... 44
3.4 Conclusions .......................................................... 46

4 Conclusions and Future Work 47
4.1 Conclusions .......................................................... 47
4.2 Future Work .......................................................... 48
  4.2.1 Intelligent Maintenance Agents ............................. 48
  4.2.2 Localized and Distributed Case-Based Experience Sharing ................................................. 49

5 Paper Contributions 51
  5.1 Paper A ............................................................. 52
  5.2 Paper B ............................................................. 52
  5.3 Paper C ............................................................. 53
  5.4 Paper D ............................................................. 54
  5.5 Paper E ............................................................. 54
  5.6 Paper F ............................................................. 54

II Included Papers 60

6 Paper A:
  Fault Diagnosis in Industry Using Sensor Readings and Case-Based Reasoning 63
  6.1 Introduction ......................................................... 65
  6.2 Fault Diagnosis Based on Sensor Signals ....................... 67
  6.3 Case-Based Classification using Extracted Features ........ 68
CONTENTS

6.4 Application to Fault Diagnosis for Industrial Robots . . . 71
   6.4.1 Pre-processing and Feature Extraction ............ 72
6.5 Sound Classification and Results .......................... 73
6.6 Conclusions ........................................ 74
6.7 Acknowledgement ...................................... 75

7 Paper B:
Fault Diagnosis of Industrial Robots using Acoustic Signals and Case-Based Reasoning 79
7.1 Introduction ......................................... 81
7.2 Classifying Sound Recordings .......................... 82
   7.2.1 Filtering and Pre-processing .................... 82
   7.2.2 Features and Feature Vector .................... 83
   7.2.3 Classification Process .......................... 83
7.3 Classifying Sound Recordings .......................... 84
   7.3.1 Comparison to the OSA-CBM Architecture ......... 85
7.4 Pre-Processing ....................................... 85
   7.4.1 Time splitting .................................. 86
   7.4.2 The Discrete Wavelet Transform ................. 87
7.5 Feature Extraction Process .................... 90
   7.5.1 Feature Identification ......................... 90
   7.5.2 Assembly of a Feature Vector ................... 91
7.6 Fault Classification .................................. 92
7.7 Evaluation ........................................... 93
7.8 Example of Case Retrieval ............................ 96
7.9 How about FFT in This Context ......................... 98
7.10 Conclusions ........................................ 99

8 Paper C:
Dynamic Modeling and Sound (Noise) Diagnostics of Robot Gearboxes for Fault Assessments 103
8.1 Introduction ......................................... 105
8.2 Sources of Gear Noise ................................ 106
8.3 Simulation of a Drive Model in Dymola / Modelica ... 107
8.4 Noise Experimental Setup ............................ 109
8.5 Recording of Noise .................................. 110
8.6 Crest Factor and Results Comparison .................... 113
8.7 Conclusions ........................................ 115
### 9 Paper D:
Identifying Discriminating Features in Time Series Data for Diagnosis of Industrial Machines

9.1 Introduction ................................................. 121
9.2 Background and Related Work ............................ 121
  9.2.1 Feature Discrimination .................................. 121
9.3 Computing Feature Vectors for Time-Series Data ........... 122
  9.3.1 Extracting Discriminating Features for Case Indexing .......................... 122
9.4 Case Indexing .................................................. 125
9.5 Example Implementation and Evaluation ..................... 125
  9.5.1 Measuring Current Time-Series .......................... 126
  9.5.2 Classification of Time-Series ............................ 126
  9.5.3 Computing Feature Vectors .............................. 127
9.6 Conclusions and Future Work ................................ 128

### 10 Paper E:
Using Cased-Based Reasoning Domain Knowledge to Train a Back Propagation Neural Network in order to Classify Gear Faults in an Industrial Robot

10.1 Introduction .................................................. 135
10.2 The CBR System .............................................. 136
10.3 Extracting Domain Knowledge ............................... 137
10.4 Training a Neural Network Classifier ...................... 138
10.5 Evaluation ...................................................... 140
10.6 Conclusions ...................................................... 140

### 11 Paper F:
Agent-Based Monitoring using Case-Based Reasoning for Experience Reuse and Improved Quality

11.1 Practical implications ........................................ 147
11.2 Introduction .................................................. 147
11.3 Intelligent Agents .......................................... 149
  11.3.1 The Maintenance Agent ................................ 150
11.4 Factors Affecting Decisions by Agents ..................... 150
11.5 Designing and Building Agent-Based Systems using Artificial Intelligence .......................... 152
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.5.1 Prototype Agent-Based Fault Diagnosis System Based on Sensor Signals and Case-Based Reasoning A Case Study</td>
<td>153</td>
</tr>
<tr>
<td>11.5.2 Signal Pre-Processing and Feature Extraction</td>
<td>156</td>
</tr>
<tr>
<td>11.5.3 Case-Based Classification Using Extracted Features</td>
<td>160</td>
</tr>
<tr>
<td>11.5.4 Sound Classification and Results</td>
<td>162</td>
</tr>
<tr>
<td>11.6 Conclusions</td>
<td>163</td>
</tr>
<tr>
<td>11.7 Acknowledgement</td>
<td>164</td>
</tr>
</tbody>
</table>
I

Thesis
Production companies often have large investments in modern production machines as well as high maintenance costs of such units [1]. Fast and precise identification of faults and problems in machines makes a crucial contribution to reduce maintenance costs and to enhance the reliability in manufacturing.

Fault diagnosis systems able to learn from experience, resulting in a more reliable performance of analysis of sensor readings can provide a number of advantages. Even though the benefits of this kind of systems are well known, they are still not widely accepted within industry. One reason might be the fear of investing too much in the implementation of such a system without knowing exactly what the results will be [2]. Another reason might be the bad reputation arising from unreliable systems repeatedly giving false alarms causing expensive loss of production capacity and resulting in technicians losing trust in the systems [3]. If systems could learn from previous experience for both correct and false alarms, the reliability and trust in them would increase.

Recent advances in research in the area of Artificial Intelligence (AI) have provided means to increase the reliability of this type of systems. For fault diagnosis purposes of industrial machines, streams of data can be gathered by various sensors. Sensor recordings can be regarded as evidence of origin for recognizing the working conditions of machines and can be used for construction of automatic fault diagnosis systems based on AI methods and techniques.

Case-Based Reasoning (CBR) is an attractive AI method for building
Chapter 1

Introduction

Production companies often have large investments in modern production machines as well as high maintenance costs of such units [1]. Fast and precise identification of faults and problems in machines makes a crucial contribution to reduce maintenance costs and to enhance the reliability in manufacturing.

Fault diagnosis systems able to learn from experience, resulting in a more reliable performance of analysis of sensor readings can provide a number of advantages. Even though the benefits of this kind of systems are well known, they are still not widely accepted within industry. One reason might be the fear of investing too much in the implementation of such a system without knowing exactly what the results will be [2]. Another reason might be the bad reputation arising from unreliable systems repeatedly giving false alarms causing expensive loss of production capacity and resulting in technicians losing trust in the systems [3]. If systems could learn from previous experience for both correct and false alarms, the reliability and trust in them would increase.

Recent advances in research in the area of Artificial Intelligence (AI) have provided means to increase the reliability of this type of systems. For fault diagnosis purposes of industrial machines, streams of data can be gathered by various sensors. Sensor recordings can be regarded as evidence of origin for recognizing the working conditions of machines and can be used for construction of automatic fault diagnosis systems based on AI methods and techniques.

Case-Based Reasoning (CBR) is an attractive AI method for building
reliable fault-diagnosis systems. A CBR system is centered around a case library containing retained cases describing problems and their respective solutions. The case library is continuously updated making the system increasing its experience in fault diagnosis. A CBR system contains several appealing properties [4]:

- A separation between its knowledge base and its reasoning function
- The advantage of a dynamic and revisable knowledge base
- The ability to explicitly show examples of solutions
- Increased user acceptance

The methodology of CBR lends itself naturally to fault diagnosis of industrial machines by representing sensor data as the problem and the repair action as the solution. CBR uses a database containing previously experienced problems and their solutions and use it to solve new problems of a similar nature [5]. The solutions can be collected from human experts or they can reflect previous search results in the case library. An example of an area in which CBR has been widely used is in medicine [6][7][8] where the symptom (the problem) and its diagnosis and treatment (the solution) are used as a case. Fault diagnosis of industrial machines and medical diagnosis of humans are analogous. When a machine fails to operate as intended it often shows unusual symptoms e.g. abnormal noises or shifting trends in driving current etc. In industry, Case-based fault diagnosis systems began to evolve after 1994 and they were until recent mainly installed in helpdesks, one example being Case Advisor [9], the first commercial helpdesk application that utilized CBR. Case-based systems for fault diagnosis of industrial machines still remain a new area and most systems existing today are prototypes on a research level. CheckMate [10] is one example of a case-based fault diagnosis system implemented for use in an industrial environment. It was implemented in order to aid technicians in repairing industrial printers. Further information about case-based fault diagnosis systems for industrial machines is given in chapter 3.

The aim of this thesis is to explore an approach to fault diagnosis of industrial machines using sensor signals along with methods and algorithms from signal processing and artificial intelligence. The approach is mainly based on the CBR methodology because of its appealing properties in this domain of applications.
1.1 Research Questions

Based on the previous section, the following research questions have been proposed:

1. **Is it possible to build automatic fault diagnosis able to improve its performance using methods and algorithms from artificial intelligence?**
   Recent advances in research in the area of artificial intelligence have provided methods and algorithms able to learn from experience and hence increase their performance. How to utilize these advances in order to improve the performance in fault diagnosis is an intriguing research challenge.

2. **How can we promote experience reuse in automatic fault diagnosis and how does such a scheme fit in industrial settings?**
   Artificial intelligence methods such as the CBR methodology contain several appealing properties for this domain of applications. CBR has the ability to explicitly show examples of solutions through past cases and its dynamic and revisable storage base enables system performance to continuously be enhanced by adding new and revising old cases. Also, it fosters experience reuse and sharing in the sense that classified cases from different sources can be easily added to a common library.

3. **How can automatic fault diagnosis with limited experience (sparsely populated case library) be reliable enough in an engineering context?**
   A key factor for user acceptance of a new system is its reliability, or in a CBR context, it must be able to display adequate performance even with a sparsely populated case library. Case retrieval must rely on robust case indexing algorithms in order to achieve adequate ranking of nearest neighbouring cases.

1.2 Research Contributions

Based on the research questions and the previous section; the main contributions of this thesis are:
1. Development of sensor-based methods and models for collection, use and reuse of experience for fault diagnosis and fault classification

This thesis explores fault diagnosis of industrial machines using sensor signals along with methods and algorithms from signal processing and artificial intelligence. The proposed methods and algorithms are presented along with a fault diagnosis framework and a prototype fault diagnosis system has been used for evaluation. Several methods and algorithms from signal processing and artificial intelligence have been used in this thesis work but the approach is mainly based on the CBR methodology where sensor signals such as acoustic emission and current readings are classified according to previously classified sensor signals stored as cases in a case library. Evaluations have shown that the proposed approach has been proven successful and reliable in diagnosing faults in gearboxes of industrial robots using acoustic emission and current readings using only a sparsely populated case library. Also, performance has been shown to improve as additional cases are added to the case library.

Sensor signals such as acoustic emission [paper A,B,C,E,F] and induction motor drive current [paper D] were used as fault diagnosis parameters and various signal filtering methods such as wavelet analysis [paper A, B, F], bandwidth filtering [paper C, E], time-domain averaging [paper A], time-splitting [paper A] and FFT analysis [paper A, D] have been applied. For feature extraction methods such as wavelet analysis [paper A, B, F], wavelet coefficient thresholding [paper A], standard deviation [paper D], crest factor and RMS calculation [paper C] and FFT analysis [paper A, D] along with approaches to classification such a neural network classification [paper F] and basic case-based classification involving Euclidean distance calculations [paper A, B, D, F].

2. An approach to automated decision support based on experience reuse for fault diagnosis in industrial settings

The methodology of CBR lends itself naturally to fault diagnosis of industrial machines by representing sensor data as the problem and the repair action as the solution [paper A, B, F]. When a new case occurs for the first time, an experienced technician may identify
and repair the fault and when the new case has been classified, it is added to the case library. The objective is to collect experience through cases and to achieve a more competent classification as additional cases are added to the case library. This approach aids technicians in making a correct objective diagnosis of industrial machines based on earlier classifications of similar sensor signals. The case retrieval can provide results that are user-friendly and offer a sort of automated decision support for technicians in diagnosis tasks and a CBR system has the ability to foster experience reuse and sharing in the sense that classified cases from different sources can be easily added to a common library. Intelligent agents deploying CBR enable the agents to gain experience by collecting past solved cases, adapt them to current problem and context e.g. the experience level of the technician [paper F]. By identifying similar situations, transfer relevant information and experience, and adapt these cases to the current situation will both transfer knowledge and help this decision process. Some decisions can be made autonomously by the agent in critical situations if no technician is close by. Using intelligent agents for monitoring is an important path to the next generation of monitoring systems and an approach to automated decision support based on experience reuse for fault diagnosis in industrial settings.

3. Development of methods and algorithms for classifying cases using a sparsely populated case library

A CBR system has the ability to display adequate performance even with a sparsely populated case library as it does not require a complete case library for functioning properly from the beginning [paper A, B]. The case retrieval can provide intermediate results and it improves its classification performance as long as newly classified cases are injected into the case library. Case retrieval must rely on robust feature extraction and case indexing algorithms in order to achieve adequate ranking of nearest neighbouring cases [paper A, B, C, D, F]; especially when the system has a sparsely populated case library. Reducing the inherent high dimensionality in time series data is a desirable goal as algorithms used for CBR classification easily can be misguided if presented with data of to high dimension due to unwanted computation of similarities between irrelevant features. Selecting adequate features for clas-
sification of time series data can be a time-consuming task that requires good domain knowledge and a tedious manual inspection of the data. Individual weighting of important features [paper D] can be used in order to adjust and suppress unwanted features in the matching process but it often requires expert knowledge about the relevance of each feature and its impact in the matching process. Unsupervised feature discrimination where feature vectors for time series measurements are selected with respect to their discriminating power requires no expert knowledge and may also be used for individual weighting of features. A sparsely populated case library may also be extended by incorporating model based reasoning using adequately specified models and pre-classified sensor signals from the model simulation [paper C]. In order to succeed, it is important to find suitable diagnostic parameters that can be projected from model simulation results onto real measurements of sensor signals.

1.3 Outline of Thesis

The thesis is organized as follows. This chapter presents an introduction and the main research questions and research contributions to this domain of applications. Chapter 2 provides an introduction and theoretical background to methods and techniques applied in this research. Chapter 3 presents a comparison between five case-based fault diagnosis systems for industrial machines including the system described in this thesis. Chapter 4 concludes the first part of the thesis, revisit its research contributions and proposes future work. Chapter 5 summarizes the papers which form the second part of the thesis and the last six chapters contain the complete versions of the included papers.
Chapter 2

Theoretical Framework

This chapter mainly presents a theoretical background to the work this thesis is based on. Section 2.1 gives a short background to fault diagnosis of industrial machines. Section 2.2 introduces a fault diagnosis framework based on methods from artificial intelligence and modules from the OSA-CBM [11] standard. The last four sections of this chapter considers sensor signals, methods and algorithms that have been explored in this thesis work.

2.1 Background

Manual diagnosis of industrial machines has been performed as long as such machines have existed. Automatic diagnosis began to appear first when suitable computers became available in the 1970’s. Computer-aided diagnosis of industrial machines has many advantages and can be an effective and cost-saving investment for companies [2].

Most machinery failures give a warning in advance before they occur. This warning is usually a physical condition which indicates that a failure is about to occur [12] e.g. mechanical faults in induction motor driven gearboxes often show their presence through abnormal acoustic signals or abnormalities in motor drive current compared with normal ones. Using sensor technology it is possible to detect and measure the values of these conditions and their profiles.

Table 2.1 lists some common monitoring and fault diagnosis parameters and their associated sensors.
Table 2.1: Monitoring and fault Diagnosis Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
<td>Temperature detector</td>
</tr>
<tr>
<td>Vibration</td>
<td>Accelerometer</td>
</tr>
<tr>
<td>Acoustic Emission</td>
<td>Microphone</td>
</tr>
<tr>
<td>Electrical current</td>
<td>Ammeter, voltmeter</td>
</tr>
</tbody>
</table>

A typical monitoring and fault diagnosis system consists of one or several of the sensors listed in table 2.1 which output are fed to an analysis system. Figure 2.1 depicts a schematic figure of a selection of modules of the OSA-CBM [11] standard that form a typical monitoring and fault diagnosis system [13].

Figure 2.1: Four of the OSA-CBM standard modules for machine monitoring and fault diagnosis.

The modules in figure 2.1 (from left to right) are:

- **Sensor Module**: The sensor module provides the system with monitoring data (see table 2.1)
- **Signal Processing Module**: The Signal Processing Module receives sensor data and processes the data with e.g digital filters such as FFT, wavelet transform etc.
- **Condition Monitor Module**: The primary purpose of the Condition Monitor is to generate alerts based on preset operational limits
- **Decision Support Module**: The primary purpose of the decision support module is to generate recommended actions with respect to the condition of the system

This section introduces a fault diagnosis framework based on methods from artificial intelligence and the modules depicted in figure 2.1. The framework is illustrated in Figure 2.2. It includes signal filtering, feature extraction and a classifier as its main components. The classifier is used for decision support and it presents a diagnosis about the condition of the monitored object. A prototype system based upon this framework was implemented and tested on gearboxes on industrial robots. The system can, based on the symptoms, reason about the class of fault associated with the machine.
support module is to generate recommended actions with respect to the condition of the system.

## 2.2 Introduction

This section introduces a fault diagnosis framework based on methods from artificial intelligence and the modules depicted in figure 2.1. The framework is illustrated in Figure 2.2. It includes signal filtering, feature extraction and a classifier as its main components. The classifier is used for decision support and it presents a diagnosis about the condition of the monitored object. A prototype system based upon this framework was implemented and tested on gearboxes on industrial robots. The system can, based on the symptoms, reason about the class of fault associated with the machine.

![Figure 2.2: fault diagnosis framework based upon sensor signals](image)

Two common machine monitoring parameters have been used; acoustic emission [paper A,B,C,E,F] and electrical current [paper D]. These monitoring parameters were chosen because of:

- Their future ability to provide a physical distance between sensors
and the monitored object (as opposed to e.g. vibration monitoring that involves the attachment of accelerometers on the object).

- Sensorless monitoring; electrical current are usually readily available from within the machine and no extra sensors are needed.

- Acoustic emission can successfully be recorded using a simple electret condenser microphone connected to a computer with installed sampling equipment.

- Measuring acoustic emission in human audible frequencies provides an excellent ability to receive feedback from experienced technicians.

Signal pre-processing is used to purify the original sensor readings by removing unwanted components such as noise and/or to enhance components related to the condition of the object such that more reliable diagnosis results will be warranted. Noise can be caused internally by various parts in the diagnosed object or externally by disturbance from surroundings which is added to the received sensor data. Signal pre-processing has been dealt with by applying signal processing methods like wavelet analysis, bandwidth filtering, time domain averaging and fast Fourier transform and are further described in section 2.4.

Feature extraction is purported to identify characteristics of the sensor signals as useful symptoms for further analysis. This stage is critical for fault diagnosis in many industrial applications. In order to supply the diagnosis module (see Figure 2.2) with a moderate number of inputs for effective analysis and reasoning, representative features from the sensor signals have to be extracted. Time-based features are extracted from the profile of signal values with respect to time. Typical features of this kind can be peak value, start time, mean value, standard deviation, etc. Frequency-based features characterize sensor signals according to their amplitudes under significant frequencies and are mainly adopted as descriptors of condition parts of cases in this research. More information about time- and frequency-based signal features can be found in section 2.5 and fundamental signal analysis methods to yield frequency spectra are described in section 2.4.

Regarding fault classification a number of different methodologies can be considered. For complex diagnosis situations with nonlinear boundaries and many relevant features a classifier based on artificial neural
networks might be a good choice. Nevertheless the success of neural network functioning is conditioned upon the prior training of the network with sufficient examples, which unfortunately are not guaranteed in quite a few industrial environments. In section 2.6 an introduction to neural network classification is given. Algorithms used for classification can easily be misguided if presented with data of a too high dimension. E.g. the k-nearest neighbor algorithm which is often used for case-based classification performs best on smaller dimensions with less than 20 attributes. The inherent high dimensionality of extracted features can be reduced using methods such as feature discrimination described in section 2.5 in which we can transform the original signal into a reduced representation set of features where relevant information from the input data is retained and irrelevant information is lost. Case-based reasoning has the advantages of entailing no training beforehand but still exhibiting the ability of incremental learning if new useful cases are properly injected into the case library. This is the motivation to develop a case-based classifier of fault patterns which is introduced in this chapter and in the attached papers forming the second part of the thesis. In addition, an introduction to case-based reasoning and classification is given in section 2.6. I believe that applying CBR techniques for diagnosis is a strong candidate to deal with certain industrial problems with a high feature dimension but few known samples as support.

2.3 Sensor Signals

2.3.1 Acoustic Emission

Operating gears generate acoustic emission (AE) by the meshing of gear teeth. AE is transmitted to the shafting, bearings and transmission housing. The transmission housing then acts as a loudspeaker and radiates the AE to the surrounding environment.

AE is characterized by the generic properties of waves:

- Frequency
- Wavelength
- Period
- Amplitude
The frequency is given by:

\[ f = \frac{1}{T}, T = \text{time of 1 period} \]  

(2.1)

A more accurate description is given by:

\[ f = \frac{v}{\lambda}, v = \text{speed}, \lambda = \text{wavelength} \]  

(2.2)

Wavelength \( \lambda \) is inverse proportional with the frequency.

\[ \lambda = \frac{v}{f} \]  

(2.3)

AE is in most cases mainly caused by a imperfect engagement of the gear teeth. This imperfect action results in non-constant angular velocities caused by the dynamic forces at the gear teeth which in turn excite vibrations in the gear blanks and shafting. The gear housing walls normally prevent AE from the gear blanks reaching the human ear. The most significant transmission path of the AE is through the transmission housing. Figure 2.3 depicts the first part of a drive train of an axis in an industrial robot. It consists of a driving and a driven shaft.

![Figure 2.3: A part of a simple drive train.](image)

The gear ratio \( i \) of Figure 2.3 can be calculated as:

\[ i = \frac{Z_2}{Z_1} \]  

(2.4)
Where:
\(Z_1\) = number of teeth of the driving gear (pinion)
\(Z_2\) = number of teeth of the driven gear

The primary shaft rotational frequencies can be calculated using the following formulas [14] [12]:

\[
\begin{align*}
    f_{s1} &= \frac{N_1}{60} \quad (2.5) \\
    f_{s2} &= \frac{N_2}{60} = f_{s1} \frac{Z_1}{Z_2} \quad (2.6) \\
    f_m &= f_{s1}Z_1 \quad (2.7)
\end{align*}
\]

Where:
\(f_{s1}\) = driving shaft frequency, Hz
\(f_{s2}\) = driven shaft frequency, Hz
\(f_m\) = gear mesh frequency, Hz
\(N_1\) = driving shaft speed, rpm
\(N_2\) = driven shaft speed, rpm

The shaft and meshing frequencies can also be seen in the bands and sidebands of a Fast Fourier Transform spectrum (see Figure 2.4). The sidebands can be calculated from the gear mesh and shaft frequencies with the following formula:

\[
f_{sb} = f_m \pm n f_{s1}, f_m \pm n f_{s2} \quad (2.8)
\]

Figure 2.4 depicts a Fast Fourier Transform (FFT) [15] of a sound recording of the gear train of which the gear wheels described above form a part. From this FFT, it is possible to obtain information about the gearbox status by analyzing the peaks in the frequency spectrum.

The peak at around 600 Hz corresponds to the meshing frequency of the driving gear. This frequency can be calculated using formula 2.7 by inserting the rotational frequency of the driving shaft which was 43 Hz and the number of teeth on \(Z_1\) which was 14:

\[
f_m = f_{s1}Z_1 = 43 \times 14 = 602 \text{ Hz}
\]
As depicted in 2.4, the shaft frequencies can often be read from the sidebands; $f_{s1}$ corresponds to the driving shaft rotational frequency and $f_{s2}$ corresponds to the driven shaft rotational frequency. Harmonics occur at integer multiples of the fundamental frequencies. The first harmonic can be seen at the right in the figure at 1200 Hz. The same sidebands occur in the harmonic(s).

**Recording Acoustic Emission**

AE can successfully be recorded using a simple electret condenser microphone connected to a computer with installed sampling machines. Three sampling parameters are important to consider when setting up the recording machines:

- Sampling frequency
- Bit depth
- Nyqvist theorem

Sampling frequency (sample rate) must be chosen accordingly to get the right amount of information. Computer sampling machines makes measurements of sound at fixed intervals or sampling frequencies e.g. 8,16,24,44.1,48,96,192kHz etc. Each measurement is saved as an integer number at a fixed bit depth e.g. 8,16,24 bit where $8bit = 2^8 = 256$
2.3 Sensor Signals

measurement representations. When measuring AE, one sampling channel is enough. Two channels (stereo) needs twice the sampling rate e.g. CD-quality=44.1kHz=2*22.5kHz channels but it is not required. The sampling theorem asserts that the uniformly spaced discrete samples are a complete representation of the signal if its bandwidth is less than half the sampling rate. This is called the Nyquist Theorem [16]. It implies that to fully capture a signal with limited bandwidth $B$ the sampling rate must be $2B$. By using sampling rate $B$ measurements of e.g. a sinusoidal signal of frequency $B$ may result in only a line whereas using a sample rate of $2B$, the full signal can be captured.

2.3.2 Discovered Fault Symptoms

Transmission Error

In most cases, the dominant source of AE is vibration due to transmission error (geometric inaccuracies) introduced during the manufacture of the gear. Transmission error is defined as [14]:

"the difference between the actual position of the output gear and the position it would occupy if the gears were perfectly conjugate"

Gear Tooth Impacts

Gear tooth impacts occur when there are tooth deflections or spacing errors in a gear. This will result in a premature contact at the tooth tip causing an impact between the gears. These impacts can cause large frequency AE levels and also shorten the life of a gear due to reductions in gear tooth fatigue life.

Figure 2.5 shows two recordings of the axes of an industrial robot; a recording of a normal axis at the left and a recording with an abnormality at the right. As can be seen in the figure, the normal recording is smooth and steady, containing no prominent peaks. The faulty recording at the right resembles the normal recording except for two very prominent peaks. These peaks are the results of impacts due to a notch in one of the gear wheels in the gearbox. In [paper A,B,F] these peaks were extracted as features and classified in a case-based approach. Impulses are not always detectable in an FFT spectrum [paper A,B]. Under these circumstances wavelet analysis (see section 2.4) might be more successful.
By measuring the time $t$ between two repeating impulses the shaft speed can be obtained (see 2.5 and 2.6) using the formula:

$$f = \frac{1}{t} = \frac{N}{60} \quad (2.9)$$

**Gear Play**

Excessive play between two mating gears can result in undefined rattling impulse noises. These noises can occur when an instant torque is applied to the output shaft of the gearbox or when the driving shaft changes its direction of rotation. Figure 2.6 depicts a filtered sound recording of a rattling gearbox of an industrial robot.

It can be difficult to determine which part of the gearbox causes such rattle. It is not always straightforward and in this case, the experience of experts is very valuable.

**Friction**

Increased friction between two mating gears is a potential source of increased vibration. The meshing action between two gears is characterized by a combination of rolling and sliding. The sliding forces between two gear teeth as they mesh will increase with increased friction resulting in increasing gear noise. Increased friction proved to be detectable through indirect current measurements [paper D].
2.3 Sensor Signals

2.3.3 Induction Motor Drive Current

Indirect Measurements of Motor Drive Current

The induction motor drive current from the motor driving the gearbox can be measured using the appropriate measuring equipment. Current measurements $M_c$ as discussed in this thesis are actually derived from measurements of motor torque $M_t$ using constant $c$ which were readily available from within the machine and no extra sensors were needed:

\[ M_c = M_t \times c \]  \hfill (2.10)

Figure 2.7 depicts the drive train of the measured gearbox. It consists of a pinion driving the first reduction gear which in turn drives a second reduction gear that is connected to the output shaft of the gearbox. Fig 2.8 depicts an indirect current measurement from the induction motor driving the above illustrated gearbox.

2.3.4 Discovered Fault Symptoms

Knocking due to Friction

Knocking Gearboxes have been shown to be detectable through indirect current measurements [paper D]. These impacts are likely the results of
spacing errors between gears caused by a too tightly adjusted gearbox. This will result in increased friction between two mating gears. It will probably shorten the life of a gear due to reductions in gear tooth fatigue life. The forces between two gear teeth as they mesh will increase with increased friction resulting in increasing current which can be detectable in a properly filtered current measurement. Filtering frequencies can be derived from gearbox properties using equations:

\[ f_{sn} = f_{sn-1} \frac{Z_n}{Z_{n+1}} \quad (2.11) \]
\[ f_m = f_{sm} Z_{m+1} \quad (2.12) \]

Figure 2.9 shows two filtered current measurements of gearboxes of in-
dustrial robots; the left is a measurement of a normal gearbox and the right measurement comes from a too tightly adjusted gearbox. Measurement of the faulty gearbox has an increase in current compared to the normal one.

Figure 2.9: Current measurements of a normal and a faulty gearbox

2.4 Signal Pre-Processing

Signals from a gearbox must (usually) be processed before any important information related to the gear wheels can be extracted from it. This chapter discusses five signal pre-processing methods:

- Bandwidth filtering
- Fast Fourier Transformation
- Wavelet Transformation

2.4.1 Bandwidth Filtering

Bandwidth filtering can be effective when frequencies of interest are known and unwanted noise easily can be filtered out. By applying various kinds of bandwidth filters as shown below, important signal characteristics such as gear mesh frequencies and band limited spectrum’s can be filtered out. Common bandwidth filters include:

- Band pass
- Band stop
A band pass filter allows for a part of a frequency spectrum or band to pass. It leaves out all frequencies above and below the selected frequency. It is also called a notch filter as it leaves only a notch of a frequency band to pass. A band stop filter is the inverse of a band pass filter. It stops a selected frequency band while letting the frequency spectrum on the sides of the band to pass.

The low pass filter is set to a frequency breakpoint where all frequencies below that point are able to pass and no above will. The high pass filter is the inverse letting only frequencies above the breakpoint to pass.

2.4.2 The Discrete and Fast Fourier Transform

Fourier series decomposes a periodic function into a sum of sines and cosines. Fourier series were introduced by Joseph Fourier (1768-1830) and led to a revolution in mathematics. G. Strang in 1993 said:

"The Fast Fourier transform - the most valuable numerical algorithms of our lifetime."

Fourier series have applications in many fields such as electrical engineering, vibration, acoustics and signal processing. A Fourier series consists of a sum of sines and cosines:

\[
\frac{a_0}{2} + \sum_{n \in N} a_n \cos nt + \sum_{n \in N} b_n \sin nt
\]  

(2.13)

This sum can successfully approximate integrable functions \( f \) on \([\pi, -\pi]\). The terms \( a_n \) and \( b_n \) are called the Fourier coefficients of \( f \). By using Euler’s formula:

\[
e^{int} = \cos (nt) + i \sin (nt)
\]  

(2.14)

we can represent \( f \) as a sum of Fourier coefficients:

\[
f (t) = \sum_{n=-\infty}^{\infty} c_n e^{int}
\]  

(2.15)
Now suppose \( f \) is defined for all real \( t \) without truncation to a finite interval e.g. \( -\pi \) to \( \pi \). Instead we integrate over \( \mathbb{R} \). This expression is called the Fourier transform of \( f \). For some functions it it impractical to evaluate the Fourier transform. Instead we can truncate the range of integration to a finite interval \([a, b]\) and then approximate the integral for \( \hat{f}(\omega) \) by a finite sum:

\[
\hat{f}(\omega) \approx \sum_{k=1}^{N-1} f(t_k) e^{i\omega t_k} \Delta t
\]

This sum is called the discrete Fourier transform \( Df \) of \( f \) and it is very useful as it can be computed as a matrix product. This implementation is called the Fast Fourier Transform (FFT) and is very commonly used in computers. The DFT in matrix form can be derived by first transforming 2.16 to (see [15] pp. 384-385):

\[
Df(n) = \sum_{k=0}^{N-1} f(k) w^{-nk}, \text{where } w = e^{2\pi i/N}
\]

And by viewing \( f \) and \( Df \) as vectors then \( Df = M_N f \) where:

\[
f = \begin{pmatrix} f(0) \\ \vdots \\ f(N-1) \end{pmatrix}
\]

And:

\[
Df = \begin{pmatrix} Df(0) \\ \vdots \\ Df(N-1) \end{pmatrix}
\]

And:

\[
M_N = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{-2\pi i/N} & e^{-2\pi i/N} & e^{-(N-1)\cdot 2\pi i/N} \\ 1 & e^{-2\pi i/N} & e^{-2\pi i/N} & e^{-(N-1)\cdot 2\pi i/N} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & e^{-(N-1)\cdot 2\pi i/N} & e^{-2(N-1)\cdot 2\pi i/N} & e^{-(N-1)^2\cdot 2\pi i/N} \end{pmatrix}
\]
The efficiency of this transform is prodigious. It can reduce a computation by a factor of a thousandth of the original number and it has led to one of the major technological breakthroughs of the twentieth century.

### 2.4.3 The Discrete Wavelet Transform

Wavelet transforms are popular in many engineering and computing fields for solving real-life application problems. Wavelets can model irregular data patterns, such as impulse sound elements better than the Fourier transform [paper B]. The signal \( f(t) \) will be represented as a weighted sum of the wavelets \( \psi(t) \) and the scaling function \( \phi(t) \) by:

\[
f(t) = A_1 \phi(t) + A_2 \psi(t) + \sum_{n \in \mathbb{Z}, m \in \mathbb{Z}} A_{n,m} \psi(2^n t - m) \tag{2.21}
\]

Where \( \psi(t) \) is the mother wavelet and \( \phi(t) \) is the scaling function.

In principle a wavelet function can be any function with positive and negative areas canceling out. That means a wavelet function has to meet the following condition:

\[
\int_{-\infty}^{\infty} \psi(t) dt = 0 \tag{2.22}
\]

Dilation’s and translations of the mother wavelet function define an orthogonal basis of the wavelets as expressed by

\[
\psi_{(sl)}(t) = 2^{s/2} \psi \left(2^{-s} t - l \right) \tag{2.23}
\]

Where variables \( s \) and \( l \) are integers that scale and dilate the mother function \( \psi(t) \) to generate other wavelets belonging to the Daubechies wavelet family. The scale index \( s \) indicates the wavelet’s width, and the location index \( l \) gives its position. The mother function is rescaled, or ”dilated” by powers of two and translated by integers. To span the data domain at different resolutions, the analyzing wavelet is used in a scaling equation as following:

\[
\phi(t) = \sum_{k=-1}^{N-2} (-1)^k c_{k+1} \psi(2t + k) \tag{2.24}
\]

Where \( \phi(t) \) is the scaling function for the mother function \( \psi(t) \), and \( c_k \) are the wavelet data values.
The coefficients \( \{c_0, c_n\} \) can be seen as a filter. The filter or coefficients are placed in a transformation matrix, which is applied to a raw data vector. The coefficients are ordered using two dominant patterns, one works as a smoothing filter (like a moving average), and the other works to bring out the ”detail” information from the data.

The result of the wavelet transformation is a measurement of the likeness between the scaled wavelet basis function and the analysed signal. The result contains a number of coefficients that describes energy level of the input signal in the time and frequency domain. It can be represented as a scalogram.

### 2.5 Signal Feature Extraction

Signal feature extraction is a method to reduce the often high dimension of a sensor signal to a reduced dimension in order to supply e.g. a pattern classifier with a moderate number of inputs for effective analysis and reasoning. Feature extraction can be seen as a transformation of the original signal into a reduced representation set of signal features. The primary goal of feature extraction is to:

- represent signal characteristics
- reduce signal dimension
- preserve relevant information
- lose irrelevant information

#### 2.5.1 Basic Signal Features

According to the domain from which features are derived we can distinguish between two categories of features: time-based features and frequency-based features. Time-based features are extracted from the profile of signal values with respect to time. Time-based features are suitable to represent e.g. regular or stochastic events in time. Typical features of this kind can be peak value, mean value, RMS value, standard deviation, Peak-to-peak value, Crest Factor etc. Below are mathematical definitions of five common time-based features given:
Theoretical Framework

\[ Peak = |x|_{max} \]  \hspace{1cm} (2.25)
\[ Mean = \frac{1}{n} \sum_{i=1}^{n} x_i \]  \hspace{1cm} (2.26)
\[ RMS = \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \]  \hspace{1cm} (2.27)
\[ Peak - to - peak = |x|_{max} + |x|_{min} \]  \hspace{1cm} (2.28)
\[ CrestFactor(CF) = \frac{S_{max}}{RMS} \]  \hspace{1cm} (2.29)

Frequency-based features characterize sensor signals according to their amplitudes under significant frequencies. Many fundamental signal analysis methods are available to yield frequency spectra such as the wavelet transform and the fast Fourier transform.

2.5.2 Wavelet Transform

Wavelet analysis [15] is an effective tool of transforming analogue sensor signals to a frequency spectra. It has been shown to perform better than Fourier transform under circumstances with heavy background noise [17]. Technical details of wavelet analysis are given in 2.4 and details of wavelet-based feature extraction are discussed in [paper A,B,F].

A comparative study was also performed in [paper A] between wavelet analysis and Fourier transform demonstrating the superiority of the wavelet approach in producing high quality features for case-based classification.

2.5.3 Fourier Transform

FFT analysis is another common method for feature extraction from signals and it has been shown to be useful in some classification tasks. Technical details of the Fourier transform are discussed in 2.4 and details of Fourier-based feature extraction are discussed in [paper A,D].

2.5.4 Signal Thresholding

Thresholding is a simple method to extract features according to some pre-set threshold. The threshold can be based on signal features and
be set to elicit deviating parts of a signal e.g. high/low peak amplitude, RMS value, standard deviation etc. Signal thresholding is easy to implement and can be powerful when appropriate thresholds can be derived. On the other hand, it can be hard to derive correct threshold parameters and parameters can vary with time. Below is an illustration of signal thresholding according to peak amplitude.

![Thresholding impulse peaks](image)

**Figure 2.10: Thresholding impulse peaks**

Technical details about signal thresholding in combination with wavelet analysis for wavelet-based feature extraction are discussed in [paper A,B,F].

### 2.5.5 Standard Deviation

Standard Deviation is a measurement of the spread of values in signal $X$. It is defined as the square root of the variance. Variance is a measure of statistical dispersion according to:

$$\text{var}(X) = E((X - \mu)^2), \mu = E(X) \quad (2.30)$$

Where 2.30 calculates the average of the squared distance of its possible values in $X$ from the expected value $\mu$. The result of 2.30 is squared and standard deviation is variance converted to measurement units such as:

$$\text{std}(X) = \sqrt{\text{var}(X)} \quad (2.31)$$

A graphic representation of a standard deviation “bell” curve in combination with thresholds $\delta$ is depicted in fig 2.11

Technical details about standard deviation in combination with FFT analysis for FFT-based feature extraction are discussed in [paper D].
2.5.6 Feature Discrimination

Feature discrimination relies on the fact that certain measurement values in a signal have a stronger discriminating power than others. By letting the values with the strongest discriminating power represent signal features we have hopefully achieved a great reduction in dimension of the signal and a more qualitative knowledge representation of it. The basic idea of feature discrimination as discussed in this thesis can be summarized as [paper D]:

1. Collect classified signal measurements in a case
2. Represent difference between values in measurements with standard deviation
3. Keep only points with enough deviation with respect to signals in the case library
4. Let these points represent the signal

2.5.7 Assembly of a Feature Vector

Using a feature vector as the signature for sensor signals is a well adopted method to detect and identify faults in industrial machines. It is also commonly used in CBR systems. A vector of frequency-based features can be formulated as [paper B]:

\[
FV = [Amp(f_1), Amp(f_2), ..., Amp(f_n)] (2.32)
\]

where \(\text{Amp}(f_i)\) denotes the function of amplitude which depends on frequency \(f_i\) and \(n\) is the number of frequencies in consideration.

A vector of time-based features of signal \(X\) as defined in (2.26)-(2.29) can be formulated as:

\[
FV = [\text{Peak}(X), \text{Mean}(X), \text{RMS}(X), \text{CF}(X)] (2.33)
\]

where \(\text{Peak}(X)\) denotes the peak value of signal \(X\), \(\text{Mean}(X)\) denotes the mean value of signal \(X\), \(\text{RMS}(X)\) denotes the root mean square value of signal \(X\) and \(\text{CF}\) denotes the crest factor of signal \(X\).

More details about time-based features are discussed in [paper C].
2.6 Classification

\[ FV = \{Amp(f_1), Amp(f_2), ..., Amp(f_n)\} \]  \hspace{1cm} (2.32)

where \(Amp(f_1)\) denotes the function of amplitude which depends on frequency \(f_i\) and \(n\) is the number of frequencies in consideration.

A vector of time-based features of signal \(X\) as defined in (2.26)-(2.29) can be formulated as:

\[ FV = \{Peak(X), Mean(X), RMS(X), CF(X)\} \]  \hspace{1cm} (2.33)

where \(Peak(X)\) denotes the peak value of signal \(X\), \(Mean(X)\) denotes the mean value of signal \(X\), \(RMS(X)\) denotes the root mean square value of signal \(X\) and \(CF\) denotes the crest factor of signal \(X\).

More details about time-based features are discussed in [paper C].

2.6 Classification

Two main signal classification methods are discussed in this thesis: Case-based classification involving Euclidean distance calculations and Neural network classification.

2.6.1 Case-Based Classification

History of CBR

CBR is derived from instance-based learning which is a machine learning method [18] used in the artificial intelligence discipline. The technique of CBR had its theoretical origins in the mid 1970’s and originally came from research in cognitive science [19]. It a feasible model of the reasoning process performed by our brain e.g. when we are subjected to stereotypical situations such as going to a restaurant or visiting a hairdresser. If a similar situation is encountered a second time, memories of these situations are already recorded in our brains and stored as scripts that inform us what to expect and how to behave. The original work in CBR was performed by Schank and Abelson in 1977. In 1983 Janet Kolodner developed the first CBR system designated CYRUS [20]. Cyrus was an implementation of Schank’s dynamic memory model and contained knowledge, as cases, about the travels and meetings of a former U.S.
Secretary of state. CBR has been known outside the research community since about 1990 when Lockheed began to use a CBR system named CLAVIER [21] for the baking of composite parts in an industrial oven.

The Structure of Case-Based Reasoners

The designs of most CBR systems share some common features. The basic parts of the system are the case and the case library. The structure of cases can be very different, depending on the systems in which they are used but in general they all share some common parts:

- A problem description, generally a set of features
- A solution to the problem

The features are used to match the case against other cases. They can be generic text, symbols, numerical values etc. The problem description is the reason for the existence of the case. It describes the problem to be solved. The solution describes how the problem has been solved when encountered in the past. The solution may be altered and adapted if the problem differs in any way from that described in the case. Cases are stored in a case library, commonly stored in a database with routines for storing, retrieving and manipulating cases.

A Case-Based Reasoner operates with the case library as the central part of the system. When a new problem occurs the case-based reasoner:

1. Retrieves the appropriate case from the case library.
2. Reuses the retrieved case in the current situation.
3. Revises the retrieved case if needed.
4. Retains the revised case in the case library.

This cycle enables the Case-Based Reasoner to improve its ability to solve problems over time as more and more cases are stored in the case library.

A new problem is matched against cases previously stored in the case library and those most similar are retrieved from the library. A solution is suggested based on the retrieved case(s) that represents the closest match to the new case. If the proposed solution is inappropriate it will
probably need to be revised, resulting in a new case that can be retained in the case library. Figure 2.12 depicts the CBR cyclical process applied to the classification and diagnosis of sensor data.

![CBR process diagram](image)

Figure 2.12: The CBR process.

### Case Retrieval

To retrieve cases similar to a new problem the system needs a matching function able to identify such similar cases. Most often, cases are retrieved by some kind of similarity measurement. The similarity measurement is based on certain selected characteristics and enables the quick retrieval of appropriate cases from the case library. E.g. in a machine diagnosis system, these features might be the type of machine, specifications of the machine, various extracted sensor data from the machine etc.

The similarity measurement calculation usually results in the retrieval of cases not identical with the new case but separated by a certain “distance”. A common technique used when calculating the distance measurement is the nearest neighbor retrieval. The formula for the nearest neighbor distance calculation is shown in 2.34.
\[ \text{Similarity}(N, R) = \sum_{i=1}^{n} w_i \times f(N_i, R_i) \] (2.34)

Where:

- \( N \) is the new case
- \( R \) is the retrieved case
- \( n \) is the number of features in each case
- \( i \) is an individual feature from 1 to \( n \)
- \( f \) is a similarity function for attribute \( i \) in cases \( N \) and \( R \)
- \( w \) is a weight that controls the importance of attribute \( i \)

As shown in 2.34 weights can be used in the retrieval process to discern features that are more or less important in the retrieval process. By weighting certain attributes, the nearest neighbor calculation can be made more realistic.

**Adaptation**

When a case is retrieved, the CBR system will try to reuse the solution it contains. In many circumstances this solution may be appropriate. But if the proposed solution is inadequate, the CBR system might try to adapt the proposed solution. Adaptation means that the system tries to transform the proposed solution (if close enough) to a more appropriate solution suited for the new case. In general there are two kinds of adaptation procedure in CBR:

- **Structural adaptation**
- **Derivational adaptation**

Structural adaptation begins with the original solution and adapts this by the application of adaptation rules and formulas. Derivational adaptation derives a new solution from the rules or formulas that created the original solution. In this method, the rules that created the original solution must be saved in the case.

Today, most CBR systems do not use adaptation. They simply reuse the solution suggested by the closest matching case. If any adaptation is needed, this is performed manually.
Extending a Case Library using Model-Based Reasoning

A case library of pre-classified sensor signals can be assembled in order to automate fault diagnosis using the CBR methodology. A key factor for user acceptance of such a system is its reliability, or in a CBR context, it must facilitate a reliable case library. A sparsely populated case library may be extended by incorporating model-based reasoning using adequately specified models and pre-classified sensor signals from the model simulation. In order to succeed, it is important to find suitable diagnostic parameters that can be projected from model simulation results onto real measurements of sensor signals. An example of a diagnostic parameter known as the Crest Factor (CF) was successfully used in order to classify simulation results from a dynamic model of a gearbox [paper C]. Gear vibrations on the force level were extracted from the model and projected onto the sound recordings of a real gearbox stored in a CBR system.

2.6.2 Neural Network Classification

History of Neural Networks

Neural Networks are actually among the first work recognised as AI. McCulloch and Walter Pitts [22] proposed a model of artificial neurons in 1943 where each neuron could be characterised as being on or off according to its stimulation from other neurons. Donald Hebb [23] introduced a learning rule for neural networks in 1949 by modifying the connection strength between them. His rule is called the Hebbian learning rule and it is widely used. Frank Rosenblatt (among others) [24] continued to work on McCulloch and Pitts original neuron model in 1962 and developed the perceptron and proved that the perceptron convergence algorithm could adjust the connection strength of a perceptron to match any input data if such a match existed. In 1969 Minsky and Papert proved that a two-input perceptron could not be trained to identify when its inputs were different. This discovery put a nail in the coffin for neural network fundings until the late 1980s even though multilayer backpropagation networks were already invented and didn’t have that flaw. In the mid 1980s, several different groups re-invented the back-propagation learning algorithm and successfully applied it to many learning problems causing a new neural network era to begin.
The Structure of Neural Networks

This section will focus on multi layer feed-forward networks. A multi layer feed-forward network represents a function $f(x)$ of its input $x$. The network is composed of units that are called neurons or nodes. The nodes are connected to each other with directed links that serves to propagate the activation $x_i$ from one node to another. Each link also has a numeric weight $w_i$ associated with it. The weight determines the strength of the connected link between two nodes. The internal state of a feed-forward network is represented by its weights. A node is actually a threshold function $h(x)$ that gets activated when appropriate inputs are given to it:

$$h(x) = k(d(x))$$ (2.35)

Where $d(x)$ computes a weighted sum of its inputs $x$:

$$d(x) = \sum_{i=0}^{n} w_i x_i$$ (2.36)

$h(x)$ is a threshold activation function deriving its output from $d(x)$ according to its threshold function. A simple threshold function can be a function which outputs 1 when input is positive and 0 otherwise. A more commonly used threshold function is the sigmoid function $\frac{1}{1+e^{-x}}$ which have the advantage of being differentiable which is important for some weight learning algorithms.

Network Learning

A network of interconnected nodes can be trained to approximate a function $f(x)$. Fig 2.13 depicts a two layer feed-forward neural network with two input nodes, two hidden nodes, one output node and two untrained weights $w_1$ and $w_2$.

This example describes how weights are adjusted in a network when it learns to approximate a function $f'(x)$ from $f(x)$. The network output $f(x)$ is the linear combination of the activation of its nodes $h_1(x)$ and $h_2(x)$ according to output weights $w_1$ and $w_2$:

$$f(x) = w_1 \times h_1(x) + w_2 \times h_2(x)$$ (2.37)
Theoretical Framework

This section will focus on multi layer feed-forward networks. A multi layer feed-forward network represents a function \( f(x) \) of its input \( x \).

The network is composed of units that are called neurons or nodes. The nodes are connected to each other with directed links that serves to propagate the activation \( x_i \) from one node to another. Each link also has a numeric weight \( w_i \) associated with it. The weight determines the strength of the connected link between two nodes. The internal state of a feed-forward network is represented by its weights. A node is actually a threshold function \( h(x) \) that gets activated when appropriate inputs are given to it:

\[
  h(x) = k(d(x)) \tag{2.35}
\]

Where \( d(x) \) computes a weighted sum of its inputs:

\[
  d(x) = \sum_{i=0}^{n} w_i x_i \tag{2.36}
\]

\( h(x) \) is a threshold activation function deriving its output from \( d(x) \) according to its threshold function. A simple threshold function can be a function which outputs 1 when input is positive and 0 otherwise. A more commonly used threshold function is the sigmoid function \( 1/(1 + e^{-x}) \) which have the advantage of being differentiable which is important for some weight learning algorithms.

Network Learning

A network of interconnected nodes can be trained to approximate a function \( f(x) \). Fig 2.13 depicts a two layer feed-forward neural network with two input nodes, two hidden nodes, one output node and two untrained weights \( w_1 \) and \( w_2 \).

This example describes how weights are adjusted in a network when it learns to approximate a function \( f'(x) \) from \( f(x) \). The network output \( f(x) \) is the linear combination of the activation of its nodes \( h_1(x) \) and \( h_2(x) \) according to output weights \( w_1 \) and \( w_2 \):

\[
  f(x) = w_1 \times h_1(x) + w_2 \times h_2(x) \tag{2.37}
\]

Figure 2.13: A two-layer neural network

The learning procedure adjust the weights in the network to minimize the classification error. The network is trained to approximate function \( f(x) \) with \( f'(x) \) such that \( f'(x) \) classifies the training set inputs \( x \) of two variables:

\[
  x_1 \in \text{class}_1 \\
  x_2 \in \text{class}_0 \\
  x_3 \in \text{class}_0
\]

This implies that weights \( w_1 \) to \( w_2 \) in the network must be adjusted according to:

\[
  f'(x_1) = f(x_1) = w_1 \times h_1(x_1) + w_2 \times h_2(x_1) = 1 \\
  f'(x_2) = f(x_2) = w_1 \times h_1(x_2) + w_2 \times h_2(x_2) = 0 \\
  f'(x_3) = f(x_3) = w_1 \times h_1(x_3) + w_2 \times h_2(x_3) = 0
\]

and by minimizing the sum of classification errors \( f(x) - f'(x) \) the weights can be adjusted accordingly:

\[
  (f(x_1) - f'(x_1)) + \\
  (f(x_2) - f'(x_2)) + \\
  (f(x_3) - f'(x_3)) \\
  \rightarrow w_1, w_2
\]

This is a simple example of weight adjustment using only one weight
update but the principle is the same when training larger feed-forward networks. The idea is to adjust weights to minimize the measurement of error on the training set. The general gradient decent algorithm [25] for weight adjustment is given below:

\[ w_j \leftarrow w_j + \alpha \times \text{Err} \times g'(in) \times x_j \]  

(2.38)

where \( \text{Err} = y - hw(x) \) for true output \( y \) minus network output \( hw(x) \) and \( g' \) is the derivate of the activation function and \( \alpha \) is the learning rate. The derivate of the activation function gives the gradient decent and weights are adjusted accordingly to decrease for negative errors and increase for positive errors.

### 2.6.3 A Neural Network Approach to CBR Classification

A neural network can be used as an alternative approach to CBR classification [paper E]. This approach may be usable when only a small and simple classifier is wanted that may use only a part of the knowledge stored in a CBR system. Once successfully trained, a neural network classifier can be directly applied on noisy sensor data without the use of the usual sensor signal classification steps involving filtering and feature extraction. It can represent the part of the case-base used in its training process and it will respond accordingly e.g. it can act as decision support in response to its input. In this alternative approach, the domain knowledge stored in a CBR system is used in order to train a neural network to provide decision support in the area of fault diagnosis. The approach is to compile domain knowledge from the CBR system using attributes from previously stored cases. These attributes holds vital information usable in the training process. The approach may be usable when a light-weight classifier is wanted due to e.g. lack of computing power or when only a part of the knowledge stored in the case base of a CBR system is needed. Further, no use of the usual sensor signal classification steps such as filtering and feature extraction are needed once the neural network classifier is successfully trained.

More details about using a neural network learning and fault classification of unfiltered acoustic signals are given in [paper E].
Chapter 3

A Comparison Between Five Case-Based Fault Diagnosis Systems for Industrial Machines

3.1 Introduction

This chapter addresses case-based reasoning (CBR) [5] systems used for fault diagnosis of industrial machines. The chapter is intended to provide a comparison between the system described in this thesis and four additional CBR systems. The additional systems were chosen because of their well-documented CBR-part [26] and their application in the area of fault diagnosis. All systems in this survey were created or reported after about 1999 and are published in major Proceedings and Journals such as the ECCBR and ICCBR Proceedings and Journal of Intelligent and Fuzzy Systems. The chapter is structured as follows: Section 3.2 gives an overview of five CBR fault diagnosis systems of industrial machines. Section 3.3 discusses and compares features of the systems. Section 3.4 gives a brief conclusion of the systems.
3.2 The Systems

This section describes five CBR systems for fault diagnosis of industrial machines. The first system is a diagnostic system for locomotives. It collects fault codes from locomotives and uses them for off-board locomotive diagnosis. The second system diagnoses electric circuits. It uses measurement data from the circuit as features and matches them with similar cases. The proposed solution is then adapted to the new case. The third system monitors the health of satellites by looking for anomalies in the down linked data from the satellite. The fourth system uses a combination of a neural network and CBR to diagnose induction motors. The last system is described in this thesis and diagnoses industrial robots with the aid of e.g. acoustic signals.

3.2.1 ICARUS A Diagnostic System for Locomotives

Locomotives are large and complex machines that are very difficult and expensive to repair. Due to their complexity, they are often best served and repaired by their manufacturer. The manufacturer often have a long time service contract with their customers and it is important for the manufacturer to reduce the service costs as much as possible.

ICARUS [27] is a case-based reasoning tool for off-board locomotive diagnosis. Locomotives are equipped with many sensors that can monitor their state and generate fault messages. ICARUS is designed to handle the fault codes that are generated by the locomotives.

Each fault code is saved in a fault database. Connected to each fault is a repair log taken from a repair database. The fault log combined with the repair log is a case in ICARUS.

Most repair logs contains a fault cluster. This means that many small faults occur before a repair is performed. The cluster of faults is used as features for case matching. Each cluster is assigned a weight between 1 and 0. The value of the weight is set to represent a clusters ability to isolate a specific repair code. If a cluster is connected to only one repair code its weight will be 1. If a cluster is connected to evenly distributed repair codes in the case base its weight will be lower. Clusters below a certain weight threshold will be assigned zero weights.

The weights are used in the matching formula. The degree of likeness between a new case and a stored case is calculated as:
3.2 The Systems

This section describes five CBR systems for fault diagnosis of industrial machines. The first system is a diagnostic system for locomotives. It collects fault codes from locomotives and uses them for off-board locomotive diagnosis. The second system diagnoses electric circuits. It uses measurement data from the circuit as features and matches them with similar cases. The proposed solution is then adapted to the new case. The third system monitors the health of satellites by looking for anomalies in the downlinked data from the satellite. The fourth system uses a combination of a neural network and CBR to diagnose induction motors. The last system is described in this thesis and diagnoses industrial robots with the aid of, e.g., acoustic signals.

3.2.1 ICARUS Diagnostic System for Locomotives

Locomotives are large and complex machines that are very difficult and expensive to repair. Due to their complexity, they are often best served and repaired by their manufacturer. The manufacturer often have a long time service contract with their customers and it is important for the manufacturer to reduce the service costs as much as possible.

ICARUS [27] is a case-based reasoning tool for off-board locomotive diagnosis. Locomotives are equipped with many sensors that can monitor their state and generate fault messages. ICARUS is designed to handle the fault codes that are generated by the locomotives.

Each fault code is saved in a fault database. Connected to each fault is a repair log taken from a repair database. The fault log combined with the repair log is a case in ICARUS.

Most repair logs contain a fault cluster. This means that many small faults occur before a repair is performed. The cluster of faults is used as features for case matching. Each cluster is assigned a weight between 1 and 0. The value of the weight is set to represent a cluster’s ability to isolate a specific repair code. If a cluster is connected to only one repair code its weight will be 1. If a cluster is connected to evenly distributed repair codes in the case base its weight will be lower. Clusters below a certain weight threshold will be assigned zero weights.

The weights are used in the matching formula. The degree of likeness between a new case and a stored case is calculated as:

\[
\frac{[\sum w_c]^2}{[\sum w_s][\sum w_n]} \quad (3.1)
\]

Where:

\[ w_c = \text{weights in common clusters between stored and new case} \]
\[ w_s = \text{weights of clusters in stored case} \]
\[ w_n = \text{weights of clusters in new case} \]

The repair code associated with the case with the highest degree of likeness is the retrieved case.

The system was validated with a case library consisting of 50 repair codes. Each repair code was associated with 3-70 cases. Each case was removed from the case base and matched to all other cases in the case base. If the repair code of the case was in the top three nearest neighboring cases, the match was considered as a success. As a result the overall accuracy of the system was 80%.

3.2.2 Diagnosis of Electronic Circuits

Diagnosis of electronic circuits is based on the analysis of the circuit response to a certain input stimuli. Input signals are generated and measurements are acquired in certain nodes of the circuit. A traditional way of doing this is to use fault dictionaries. Fault dictionaries are based on selected measurements on faulty systems. The comparison is performed by a nearest neighbor calculation and the closest case is taken as a diagnosis. The problem with fault dictionaries occurs when a new fault is found that cannot be matched with the ones already stored in the dictionary. To deal with this a case-based approach is suitable to be able to automatically extend the dictionary with new faults as they occur [26].

The case consists of two parts. Part one is the numeric part that contains the case identification number and the measurements taken from the circuit. The second part contains information about the fault diagnosis.

The class corresponds to the class of component that is diagnosed. The components are divided into different classes if they have different accepted deviations from their normal value. E.g. +/-10% can be an
A Comparison Between Five Case-Based Fault Diagnosis Systems for Industrial Machines

Table 3.1: Case Structure. The Measurement Part.

<table>
<thead>
<tr>
<th>Case id</th>
<th>Measure1</th>
<th>Measure2</th>
<th>...</th>
<th>MeasureN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case i</td>
<td>M1</td>
<td>M2</td>
<td>...</td>
<td>MN</td>
</tr>
</tbody>
</table>

Table 3.2: Case Structure Fault Part

<table>
<thead>
<tr>
<th>Class</th>
<th>Comp.</th>
<th>Deviation</th>
<th>Hierarchy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Comp.</td>
<td>X%</td>
<td>M_i L_i</td>
</tr>
</tbody>
</table>

accepted deviation for a class of components. The component field contains the component location. The deviation field contains the measured deviation of the component. The hierarchy field contains a description of which level in the circuit hierarchy the components is.

A normalized Euclidean distance function is used to retrieve the cases from the case base and the k nearest neighbors where k=3 is retrieved. The solution is adapted to the new case by transformational reuse [5]. A learning algorithm is then applied to decide whether the case should be saved as a new case in the case base or not. E.g. if the diagnosis is correct there is no need to retain the new case in the library. But if the retrieved cases produce a misclassification of the new case, the case might be added to the case base according to the results of the learning algorithm.

The system has been tested with the DROP4 [28] and the All-KNN learning algorithms. All cases are also equipped with weights to improve the classification.

A measurement on a circuit is performed resulting in the k=3 nearest neighbors in Table 3.3.

Table 3.3: An Example of Case Retrieval.

<table>
<thead>
<tr>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>Comp</th>
<th>Devi</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Case</td>
<td>0.6</td>
<td>0.7</td>
<td>0.2</td>
<td>C_1</td>
</tr>
<tr>
<td>Neighbor1</td>
<td>0.6</td>
<td>0.7</td>
<td>1.1</td>
<td>C_1</td>
</tr>
<tr>
<td>Neighbor2</td>
<td>0.7</td>
<td>0.4</td>
<td>1.3</td>
<td>C_1</td>
</tr>
<tr>
<td>Neighbor3</td>
<td>0.7</td>
<td>0.4</td>
<td>1.3</td>
<td>C_2</td>
</tr>
</tbody>
</table>
3.2 The Systems

Neighbor 1 and 2 have the same component as the new case but the deviation is smaller in both cases. Neighbor 3 has a different component. The new case will be selected as a component C1 because of its similarity in the measurements. The deviation is far from normal so the case will be introduced in the case base.

The system has been tested on a filter circuit that is commonly used as a benchmark for electronic circuits. The filter consists of several capacitors and resistors. The average result with the All-KNN retain algorithm was 89% and the average result with the DROP4 retain algorithm was 88%.

### 3.2.3 Satellite Diagnosis

Satellites are monitored from the ground using down linked data (telemetry). The case-based diagnosis program can be resembled as an expert apprentice. The program remembers the human experts actions along with the context that is defined by the down linked data. It then attempts to make its own diagnosis when similar data appears in another occasion [29].

The features in the case are not state values taken at a certain point of time. Because of the telemetry’s streaming values the features are instead trends extracted from the streaming data flow. The length of the trend is different for different parameters. The table below shows a sample case with two parameters:

<table>
<thead>
<tr>
<th>Case ID</th>
<th>Length of Time Series</th>
<th>Sampling Rate</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>1000</td>
<td>45</td>
<td>-3</td>
<td>10</td>
</tr>
<tr>
<td>2345</td>
<td>2000</td>
<td>60</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

A case is constructed from the streaming data at a time called the case point. A case is constructed looking back from the case point a certain length of time. The attribute values are picked using a window of the same length as the sampling rate. For each window only one average value is saved as representing that window. The length of the time series corresponding to an attribute is l/s were l is the length specified in the case schema and s is the sampling rate.
The distance between two time series \( R, W \) is calculated by dividing
the time series into smaller sequences \( R_i, W_i \). An Euclidean distance
calculation between each \( R_i, W_i \) is performed and a global distance \( d_g \)
is calculated from all the obtained distances between the time series
sequences:

\[
d_g(R, W) = \frac{1}{k} \sum_{i=1}^{k} d_i(R_i, W_i)
\]  
(3.2)

The system notifies the user if a new case is considered interesting. The
new case is considered interesting in two ways:

1. A similarity threshold determines if the new case should be con-
sidered as an anomaly. If the similarity of all the retrieved cases is
below that threshold the case is considered to be an anomaly and
the user is automatically notified.

2. If some of the retrieved cases are above the first threshold. Another
threshold determines if the new case is similar enough to some other
case in the case base that is previously diagnosed as an anomaly.
If so, the system will notify the user of the type of anomaly. In
both situations the user is able to give feedback to the system.

3.2.4 Induction Motor Fault Diagnosis

Induction motors are very common within industry as prime movers in
machines. Induction motors have a simple construction and are very
reliable. But working in a tough environment driving heavy loads can
introduce various faults in the motors. A system for fault diagnosis of
induction motors is presented here. The system has interesting features
such as a neural network combined with a case-based reasoning system
[30].

A case consists of 6 categories of features and 20 variables. Among
the variables are measurement positions, rotating frequency components
and characteristic bearing frequencies. The case also includes the type
of machine to be measured, the symptom, the corrective action etc.

The system uses an ART-Kohonen neural network [31]) (ART-KNN)
to guide the search for similar cases in the case base.
CBR is used to select the most similar match for a given problem. The advantage with the ART-KNN compared to other neural networks such as the Kohonen Self Organizing Map [32] is that it can learn new knowledge without losing old knowledge. When a new case is presented to the system the ART-KNN learns the new case in one of two ways:

1. If the similarity of the new case compared to the cases already learned by the network is below a certain threshold; the similarity coefficient. The network learns the case by adding new nodes to its layers.

2. If the similarity of the case is above the threshold, the network learns the case by adjusting its old nodes to resemble the new case.

Cases are then indexed in the case base by clusters of features in the ART-KNN. The indexed cases are then matched against the new case with a standard similarity calculation.

The system has been tested with measurements from an AC motor in a plant. The motor had a rotor fault which resulted in high levels of noise and vibration. The system was trained with 60 cases containing different motor defects such as bearing faults, rotor damages and component looseness.

The system retrieved two previous cases from the case base together with results from a modified cosine matching function. The retrieved cases both indicated a bearing fault. The average result of a test of all cases in the case base was 96.88%.

### 3.2.5 Diagnosis of Industrial Robots

Mechanical fault in industrial robots often show their presence through abnormal acoustic signals.

At the factory end test of industrial robots a correct classification of the robot is very critical. An incorrect classification of a faulty robot may end up in the factory delivering a faulty robot to the customer.

The industrial robot diagnosis system uses case-based reasoning and acoustic signals as a proposed solution of recognizing audible deviations in the sound of an industrial robot [paper A,B].

The sound is recorded by a microphone and compared with previously made recordings; similar cases are retrieved and a diagnosis of the robot can be made.
A Comparison Between Five Case-Based Fault Diagnosis Systems for Industrial Machines

Features are extracted from the sound using wavelet analysis [15]. A feature in the case is a normalized peak value at a certain frequency. The case contains peak values from many frequencies. The case also contains fields for information of the robot model and type of fault (if any). There is also room to enter how the fault was repaired. Table 3.5 displays a part of the case structure.

Table 3.5: A part of the case structure for robot diagnosis.

<table>
<thead>
<tr>
<th>Serial Number</th>
<th>Type</th>
<th>Fault</th>
<th>Diagnosis and Repair</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>45634</td>
<td>4500</td>
<td>2</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Cases are retrieved using a nearest neighbor function that calculates the Euclidean distance between the new case and the cases stored in the case library. A list with the k nearest neighbors is retrieved based on the distance calculations. The system learns by adding new cases to the case base. A technician enters the diagnosis and repair action manually in each case.

The system has been evaluated on recordings from axis 4 on an industrial robot. Sounds from 24 healthy robots and 6 faulty robots were collected to enable case-based classification of the condition of the robots. The prototype system demonstrated quite good performance by making right judgments in 91% of all tests.

3.3 Discussion

When comparing different case-based reasoning systems with each other one must focus on the features that are shared by all case-based reasoners.

Below is a comparative discussion of five common problems that have to be faced when implementing a case-based reasoner and how they are solved in each system. The problems are as follows:

1. Feature extraction and case representation.
2. Case retrieval and indexing.
3. Case reuse.
4. Case revision and retain.

5. Case base maintenance.

1. ICARUS uses combinations of fault codes as features because that is the way a locomotive signals its faults. A repair action on a locomotive is also very expensive, thus several faults must be combined before a repair action can be executed. Often machines cannot provide such fault codes. Instead features such as filtered measurements from different kinds of sensors are used. This is the situation for the electronic circuit diagnosis system, the induction motor diagnosis system, the satellite diagnosis system and the industrial robot diagnosis system. They all collect single measurements or time series measurements, e.g. current, vibration, acoustic signals, streaming telemetry data etc. The data collecting sensors can be an integrated part of the object or an external portable measurement device.

   The basic case representation is similar for the systems in this survey. The three basic components of the case are the features, the problem description and the repair action. Sometimes the repair action is implicit in the fault description. As in the electronic circuit diagnosis system, the repair action is equal as to replacing the faulty component.

2. The case retrieval process most commonly uses some kind of distance calculation combined with weights to calculate a distance between the new and stored cases. The k nearest neighbours to the new case is then retrieved. This kind of retrieval is used in all systems except the induction motor diagnosis system and the satellite health diagnosis system. The satellite health diagnosis system uses two similarity thresholds; one for anomaly detection and one for event detection. The induction motor diagnosis system uses a neural network to first index relevant cases in the case base. After that a straightforward k nearest neighbour distance calculation is performed to calculate the distance between the indexed cases and the new case.

3. All systems in this survey implements the reuse phase by suggesting the diagnosis extracted from the retrieved k nearest neighboring cases. The satellite diagnosis system also has a threshold for sorting out irrelevant cases not to be considered for reuse. In addition to this form of reuse the circuit diagnosis system uses adaptation [5] by transforming the past solution of the k=3 nearest neighbors to an appropriate solution
for the new case. The new solution is then inserted into the new case as
the proposed solution.

4. The simplest form of retaining is to just add the new case in the case
base. The industrial robot diagnosis system uses this kind of retain-
ing (the robot diagnosis case base is then manually investigated by an
experienced technician in order to remove irrelevant cases and provide
relevant cases with more diagnostic information). To few removals of
cases can in time cause problems with an overfilled case base making the
system perform less well. Most system implements some kind of user
interaction before a case is retained. This is performed in the satellite
diagnosis system and in ICARUS by letting an experienced technician
decide whether the case is relevant or not. The retaining process can be
extended by calculating if the new case has any ability to improve the
future diagnosis of the system. The simplest form is to look if a similar
case already exists in the case base. If it does, there is no need to retain
the case. The circuit diagnostic system also incorporates a machine-
learning algorithm that calculates the ability of the case to improve the
performance of the system.

5. Most systems in this survey are only prototypes and have not yet im-
plemented any automatic maintenance process of the case memory. The
circuit diagnosis system implements a confidence factor [33] to prevent
bad cases from spoiling the performance of the system. The case base is
maintained by removing cases when the performance of the case drops
below a certain confidence index.

3.4 Conclusions

This chapter has briefly compared five fault diagnosis systems that uses
case-based reasoning as their primary approach to problem solving. Case-
based reasoning is still new in the area of fault diagnosis of industrial
machines and most systems mentioned in this chapter are still proto-
types. Some parts of the CBR process seem to be implemented to a
higher extent than others in the systems. E.g. feature extraction and
case retrieval seems to be fully implemented but adaptation is not widely
implemented. Also, automatic maintenance of the case memory seems
not to be implemented in the majority of the systems.
Chapter 4

Conclusions and Future Work

4.1 Conclusions

This thesis explores an approach to fault diagnosis of industrial machines using sensor signals along with methods and algorithms from signal processing and artificial intelligence. The approach is based on sensor readings and a relevant feature identification and extraction process based on those sensor signals. The approach is mainly based on the CBR methodology and it enables the collection of valuable sensor data from machines on a regular basis for use in fault diagnosis and for storage for future use. Evaluations have shown that the proposed approach has been proven successful and reliable in diagnosing faults in gearboxes of industrial robots using acoustic emission and current readings in combination of sparsely populated case library, also performance has been shown to improve as additional cases are added to the case library.

As previously mentioned, the main contributions of this thesis are:

1. Development of sensor-based methods and models for collection, use and reuse of experience for fault diagnosis and fault classification

2. An approach to automated decision support based on experience reuse for fault diagnosis in industrial settings
3. Development of methods and algorithms for classifying cases using a sparsely populated case-library

4.2 Future Work

Future work involves the integration of the proposed approach into an agent-based approach for use in condition monitoring of industrial applications. A future scenario is depicted where intelligent maintenance agents are able to autonomously perform necessary actions and/or aid a human in the decision making process. Agents may utilize the concept of localized and distributed case-based experience sharing.

4.2.1 Intelligent Maintenance Agents

An intelligent maintenance agent is specialized in interpreting data from the device it is connected to. The agent observes its environment through one or more sensors. Additional information about the environment may also be acquired through communication with other agents or systems. The agent may have some basic domain knowledge about when to bring the findings to the attention of a human and when to shut down a process. The agent also has social skills to communicate its findings. It may also ask for additional information to make a final decision and it has facilities to receive appropriate feedback [paper F]. Handling groups of sensors with a dependency between measurements enabling sensor agents to collaborate and learn from experience, resulting in more reliable performance. Figure 4.1 depicts an outline of a maintenance agent in its environment.

Industrial machines may be monitored by maintenance agents. A maintenance agent is able to report if anomalies occurs and has the ability to immediately shut down failing machines if necessary and report to a technician, e.g. if a robot is loosing its grip on an object during assembly or if some machine or robot breaks down. Figure 4.2 depicts a scenario of an agent reporting different failure codes according to the severeness of the failure in a manufacturing process it also depicts the process of distributed experience sharing.
4.2 Future Work

Future work involves the integration of the proposed approach into an agent-based approach for use in condition monitoring of industrial applications. A future scenario is depicted where intelligent maintenance agents are able to autonomously perform necessary actions and/or aid a human in the decision making process. Agents may utilize the concept of localized and distributed case-based experience sharing.

4.2.1 Intelligent Maintenance Agents

An intelligent maintenance agent is specialized in interpreting data from the device it is connected to. The agent observes its environment through one or more sensors. Additional information about the environment may also be acquired through communication with other agents or systems. The agent may have some basic domain knowledge about when to bring the findings to the attention of a human and when to shut down a process. The agent also has social skills to communicate its findings. It may also ask for additional information to make a final decision and it has facilities to receive appropriate feedback [paper F]. Handling groups of sensors with a dependency between measurements enabling sensor agents to collaborate and learn from experience, resulting in more reliable performance. Figure 4.1 depicts an outline of a maintenance agent in its environment.

Industrial machines may be monitored by maintenance agents. A maintenance agent is able to report if anomalies occur and has the ability to immediately shut down failing machines if necessary and report to a technician, e.g. if a robot is losing its grip on an object during assembly or if some machine or robot breaks down. Figure 4.2 depicts a scenario of an agent reporting different failure codes according to the severeness of the failure in a manufacturing process it also depicts the process of distributed experience sharing.

4.2.2 Localized and Distributed Case-Based Experience Sharing

Human experience is a valuable asset and could be even more valuable if artificially stored and reused in an efficient way. Technicians have experience which may have been collected during many years both from successful solutions as well as from very costly mistakes. It is possible to save a large amount of time and money if such experiences could be captured and stored in such a way that it can be reused in the future and shared between collaborative units. Such kind of human thinking,
intelligence and reasoning-models can be found in the CBR methodology [34].

Maintenance agents and technicians can take advantage of such experience sharing by having access to an appropriate experience sharing interface that has access to a local and/or distributed database containing previously saved cases of experience from other technicians and maintenance agents. Except from the general experience located in the maintenance agent experience can also be saved in the form of fault and maintenance libraries describing symptoms, diagnosis, actions, prognosis etc of various failure modes that can occur.
Chapter 5

Paper Contributions

This thesis includes six papers. All papers were written within the frames of the EXACT project [35] initiated in 2003, the Factory-in-a-Box project [36] initiated in 2005 and the Eken project [37] initiated in 2006. The first paper, paper A; Fault Diagnosis in Industry using Sensor Readings and Case-Based Reasoning is largely based on my master’s thesis. The paper contains additional research results and is largely rewritten to follow the style of a journal publication. It was published in the Intelligent & Fuzzy Systems Journal, volume 15, number 1, 2004. Paper B; Fault Diagnosis of Industrial Robots using Acoustic Signals and Case-Based Reasoning presents an exhaustive study of the various stages of a proposed system used in the application of diagnosis of industrial robots using acoustic signals. The paper was presented at the 7th European conference on Case-Based Reasoning, Madrid in August 2004. Paper C; Dynamic Modeling and Sound (Noise) Diagnostics of Robot Gearboxes for Fault Assessments, was presented at the Scandinavian Conference on Simulation and Modeling (SIMS 2005) in Trondheim, Norway. Paper D; Identifying Discriminating Features in Time Series Data for Diagnosis of Industrial Machines was the result of my work to classify induction motor current readings driving faulty and normal gearboxes on industrial robots. The paper was presented at the 24th annual workshop of the Swedish Artificial Intelligence Society, May 2007 in Borås, Sweden. Paper E; Using Cased-Based Reasoning Domain Knowledge to Train a Back Propagation Neural Network in order to Classify Gear Faults in an Industrial Robot presented the results of using a neural network for
fault classification of unfiltered acoustic signals from faulty and normal gearboxes on industrial robots. And finally, the last paper, paper F; *Agent-Based Monitoring using Case-Based Reasoning for Experience Reuse and Improved Quality* was published in the Journal of Quality in Maintenance Engineering volume 15, number 2, 2009. It presents a system integration into a “intelligent maintenance agent” concept.

### 5.1 Paper A

Paper A presents an innovative approach to the fault diagnosis of industrial robots by using sensor signals (sound recordings) combined with CBR. The end-testing of industrial robots plays a very important part in the assembly line in a robot factory. As a part of this end-test the robots are set up and an automatic run-in program is executed. The robot is driven back and forward in all its degrees of freedom during this run-in cycle. The run-in cycle is primarily used for the run-in of the robot gearboxes but it also functions as a check to ensure that the robot is fully operational and without defects in its gearboxes, electric motors, cables etc. This paper represents an approach to the automatic detection of any problems during this cycle by means of sound recording and CBR; sound from the gearboxes is recorded during the run-in cycle. A system that inputs this sound, extracts features from it and uses CBR as a means of making a diagnosis on the basis of the sound recording is outlined. Such a system has many advantages as compared with a manual analysis performed by the testing personnel. It not only performs a diagnosis of the gearbox but also enables the storage for reuse of experience gained in machine diagnosis by connecting the symptom, diagnosis, corrective action and follow-up of the machine by packaging as a case.

Erik Olsson is the main author of the paper and Peter Funk contributed with valuable ideas and comments. Ning Xoing added to the paper with expert knowledge in Fuzzy systems and sensor fusion.

### 5.2 Paper B

This paper presents an exhaustive study of the various stages of a proposed fault diagnosis system used in the application of diagnosis of indus-
trial robots using acoustic signals. The paper proposes a CBR approach to collect, preserve and reuse the available experience for diagnosis of industrial robots. Sounds from normal and faulty robots are recorded and stored in a case library together with their diagnosis results. Given an unclassified sound signal, the relevant cases are retrieved from the case library as reference for deciding the fault class of the new case. Adding new classified sound profiles to the case library improves the systems performance. The system has been applied to the testing environment for industrial robots. Results demonstrate that such a system is able to preserve and transfer related experience among technicians and shortens the overall testing time.

Erik Olsson is the main author of this paper. Peter Funk contributed with valuable ideas and comments. Marcus Bengtsson added to the paper with expert knowledge in the area of condition-based maintenance.

5.3 Paper C

This paper builds upon previous work on the classification of sound recordings from industrial robots. The paper presents a model of a gearbox of an industrial robot. The model was made with the Modelica mechanical library using Dymola graphical tools. The model was used for simulation of the gearbox and was run under different load conditions in order to detect correlations between vibrations on the force level extracted from the model during simulation and previously obtained sound recordings from real gearboxes. These vibrations were projected onto the sound recordings with a statistical vibration diagnostic parameter known as the Crest Factor.

Erik Olsson and Rostyslav Stolyarchuk contributed equally to this paper. Rostyslav, from the State Scientific and Research Institute of Information Infrastructure, Lviv, Ukraine worked as a guest researcher at Mälardalen University during the time this paper was written. The authors are listed in alphabetical order.
5.4 Paper D

Paper D was the result of my work to classify induction motor current readings driving faulty and normal gearboxes on industrial robots. Reducing the inherent high dimensionality in time series data such as induction motor current readings is the goal of this paper. An algorithm is presented using a time series case base containing previously classified time series measurements. Feature vectors for time series measurements is selected with respect to their discriminating power using an unsupervised feature discrimination approach incorporating statistical feature discrimination. For evaluation, previously classified current measurements from an electrical motor driving a gearbox on an industrial robot were used. Results showed that the presented algorithm was able to correctly classify measurements from healthy and unhealthy gearboxes.

Erik Olsson is the single author of the paper.

5.5 Paper E

This paper presented the results of using a neural network for fault classification of unfiltered acoustic signals from faulty and normal gearboxes on industrial robots. Domain knowledge stored in the case base of a previously proposed fault diagnosis system [paper A,B] are used in order to train a back propagation neural network to classify gear faults in an industrial robot. The approach is to compile domain knowledge from the case base using attributes from previously stored cases. These attributes holds vital information usable in the training process. The paper shows that this method successfully can be used to train back propagation neural networks on noisy sound recordings in order to classify gear faults that generates impact sounds caused by a broken gear tooth.

Erik Olsson is the single author of the paper.

5.6 Paper F

Presents a system integration utilizing the “intelligent maintenance agent” concept of case-based experience reuse in production. An intelligent maintenance agent using a CBR approach to collect, preserve and reuse
available experience in the form of sound recordings exemplifies the concept. The main focus of this paper is to show how to perform efficient experience reuse in modern production industry to improve quality of products using two approaches; a case-study describing an example of experience reuse in production using a fault diagnosis system recognizing and diagnosing audible faults on industrial robots and an efficient approach on how to package such a system using the agent paradigm and agent architecture.

Erik Olsson and Peter Funk contributed equally to this paper.


Bibliography


