Mälardalen University Press Dissertations
No. 76

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

Erik Olsson

2009

School of Innovation, Design and Engineering

ISSN 1651-4238
ISBN 978-91-86135-32-4
Printed by Mälardalen University, Västerås, Sweden

Mälardalen University Press Dissertations
No. 76

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

MÄLARDALEN UNIVERSITY
SWEDEN

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.

ISSN 1651-4238

ISBN 978-91-86135-32-4

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

Publications

Publications included in the thesis

E. Olsson, P. Funk and N. Xiong. Fault Diagnosis in Industry Using Sensor Readings and Case-Based Reasoning. Journal of Intelligent & Fuzzy Systems, Vol.15, No.1, pages 41–46, 2004.

E. Olsson, P. Funk and M. Bengtsson. Case-Based Diagnosis of Industrial Robots Based on Acoustic Signals. Proceedings of the European Conference on Case-Based Reasoning, pages 686–701. ECCBR, Madrid, August 2004.

E. Olsson and R. Stolyarchuk. Dynamic Modeling and Sound (Noise) Diagnostics of Robot Gearboxes for Fault Assessments. Scandinavian Conference on Simulation and Modeling. SIMS, Trondheim, October 2005.

E. Olsson. Identifying Discriminating Features in Time Series Data for Diagnosis of Industrial Machines. The 24th Annual Workshop of the Swedish Artificial Intelligence Society SAIS, Borås, May 2007.

E. Olsson. Using Cased-Based Reasoning Domain Knowledge to Train a Back Propagation Neural Network in order to Classify Gear Faults in an Industrial Robot. 21st International Congress and Exhibition. Condition Monitoring and Diagnostic Engineering Management. COMADEM 2008. Prague, JUNE, 2008.

E. Olsson and P. Funk. Agent-Based Monitoring using Case-Based Reasoning for Experience Reuse and Improved Quality. Journal of Quality in Maintenance Engineering (ISSN 1355-2511), Vol.15, No.2, pages 179–192, 2009.

Publications not included in the thesis

E. Olsson and P. Funk. Diagnosis of Industrial Equipment using Case-Based Reasoning and Sound Comparison. Proceedings of the AILS-04 Workshop, pages 93–100. Lund, April 2004.

M. Bengtsson, E. Olsson, P. Funk and M. Jackson. Technical Design of Condition Based Maintenance System - A Case Study using Sound Analysis and Case-Based Reasoning. Proceedings of the 8th Maintenance and Reliability Conference. Knoxville, May 2004.

E. Olsson. A Survey of Case-Based Diagnostic Systems for Machines. Seventh International Conference on Enterprise and Information Systems, pages 381–385. ICEIS, Miami, May 2005.

P. Funk, E. Olsson, M. Bengtsson and N. Xiong, Case-Based Experience Reuse and Agents for Efficient Health Monitoring, Prevention and Corrective Actions, proceedings of the 19th International Congress on Condition, COMADEM 2006, p 445-453, Lule, Sweden, Editor(s):Kumar, U., Parida A., Rau R.B.K.N., June, 2006

M. U. Ahmed, E. Olsson, P. Funk and N. Xiong, A Case-Based Reasoning System for Knowledge and Experience Reuse, Proceedings of the 24th annual workshop of the Swedish Artificial Intelligence Society, p 70-80, Bors, Sweden, Editor(s):Lfstrm et al., May, 2007

M. U. Ahmed, E. Olsson, P. Funk and N. Xiong, Efficient Condition Monitoring and Diagnosis Using a Case-Based Experience Sharing System, The 20th International Congress and Exhibition on Condition Monitoring and Diagnostics Engineering Management, Comadem 2007, Faro, Portugal, June, 2007

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, Comadem 2007, Faro, Portugal, June, 2007

E. Olsson, M. Hedelind, M. U. Ahmed and P. Funk, Experience Reuse Between Mobile Production Modules - An Enabler for the Factory-in-a-Box Concept, The Swedish Production Symposium, Gothenburg, Sweden, August, 2007

C. Karlsson, E. Olsson, P. Funk, HYBRID EARLY WARNING SYSTEMS, COMADEM 2009 (in press), p 6, Spain, Editor(s):R.B.K.N. Rao, A. Arnaiz, June, 2009

Contents

I

Thesis

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 contains 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 classification 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

Parameter	Sensor
Temperature	Temperature detector
Vibration	Accelerometer
Acoustic Emission	Microphone
Electrical current	Ammeter, voltmeter

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].

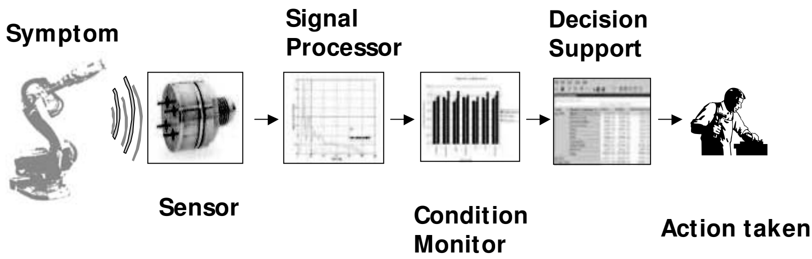

Figure 2.1: Four of the OSA-CBM standard modules for machine monitoring and fault diagnosis.

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

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.

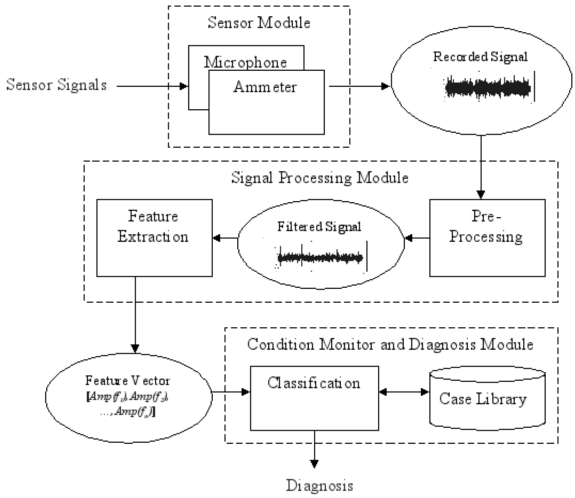

Figure 2.2: fault diagnosis framework based upon sensor signals

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 to 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

- Speed
- Direction

The frequency is given by:

$$f = \frac{1}{T}, T = \text{time of 1 period} \tag{2.1}$$

A more accurate description is given by:

$$f = \frac{v}{\lambda}, v = speed, \lambda = wavelength \tag{2.2}$$

Wavelength λ is inverse proportional with the frequency.

$$\lambda = \frac{v}{f} \tag{2.3}$$

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.

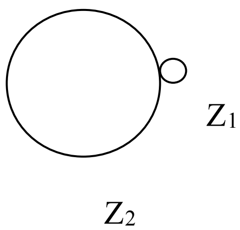

Figure 2.3: A part of a simple drive train.

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

$$i = \frac{Z_2}{Z_1} \tag{2.4}$$

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]:

$$f_{s1} = \frac{N_1}{60} \quad (2.5)$$

$$f_{s2} = \frac{N_2}{60} = f_s 1 \frac{Z_1}{Z_2} \quad (2.6)$$

$$f_m = f_{s1} Z_1 \quad (2.7)$$

Where:
f_{s_1} = driving shaft frequency, Hz
f_{s_2} = driven shaft frequency, Hz
f_{m_1} = 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:

$$f_m = f_{s1} Z_1 = 43 * 14 = 602 Hz$$

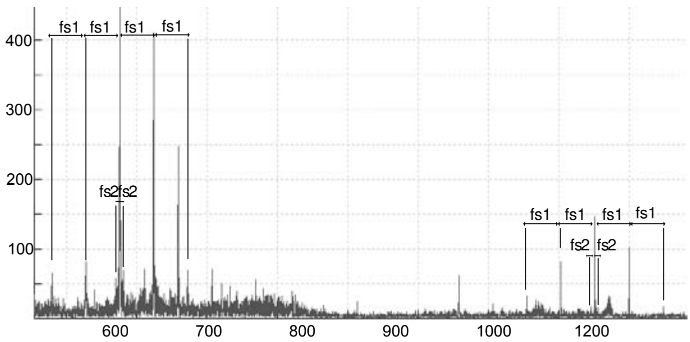

Figure 2.4: An FFT spectrum from an industrial robot.

As depicted in 2.4, the shaft frequencies can often be read from the sidebands; fs_1 corresponds to the driving shaft rotational frequency and fs_2 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 $8bit = 2^8 = 256$

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 Nyqvist 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.

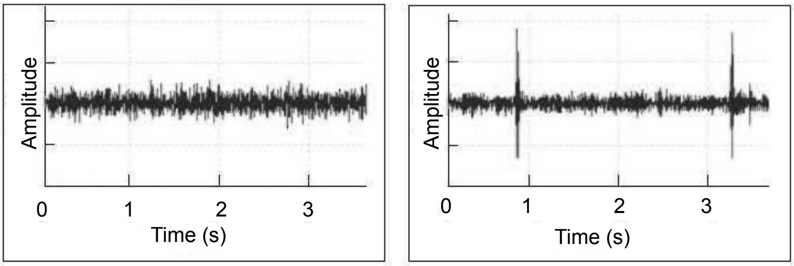

Figure 2.5: A normal and a faulty recording from an industrial robot

By measuring the time t between two repeating impulses the shaft speed can be obtained (see 2.5 and 2.6) using the formula:

$$f = \frac{1}{t} = \frac{N}{60} \tag{2.9}$$

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].

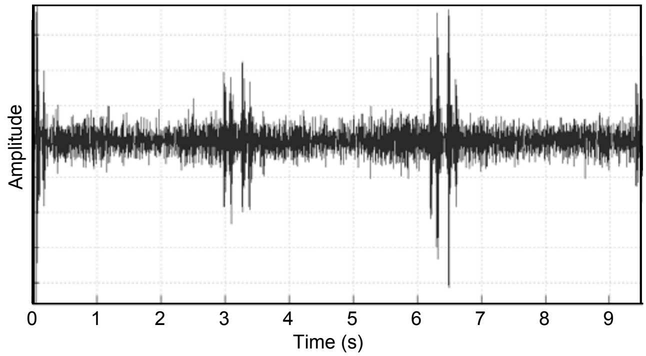

Figure 2.6: Filtered gear noise with play fault.

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:

$$M_c = M_t * c \tag{2.10}$$

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

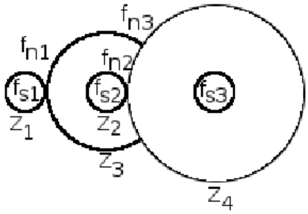

Figure 2.7: Gearbox drive train

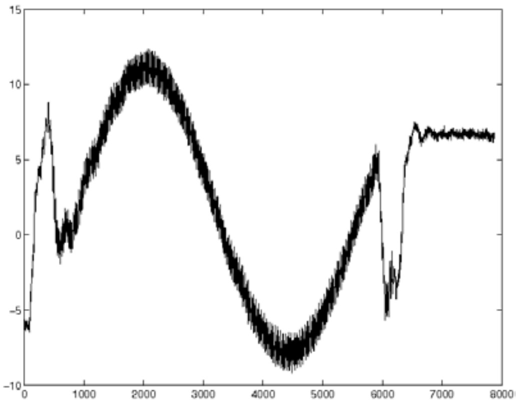

Figure 2.8: Current measurement

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:

$$f_{sn} = f_{sn-1} * \frac{Z_n}{Z_{n+1}} \tag{2.11}$$

$$f_m = f_{sm} * Z_{m+1} \tag{2.12}$$

Figure 2.9 shows two filtered current measurements of gearboxes of industrial 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.

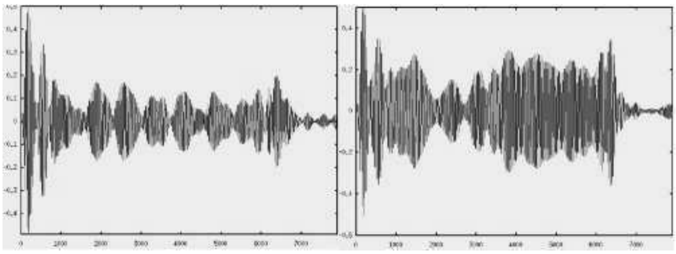

Figure 2.9: Current measurements of a normal and a faulty gearbox

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

- High pass
- Low pass

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:

$$\frac{a_0}{2} + \sum_{n\in N} a_n \cos nt + \sum_{n\in N} b_n \sin nt \tag{2.13}$$

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:

$$e^{int} = \cos(nt) + i \sin(nt) \tag{2.14}$$

we can represent f as a sum of Fourier coefficients:

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{int} \tag{2.15}$$

Now suppose f is defined for all real t without truncation to a finite interval e.g. $-\pi$ to π. Instead we integrate over $\Re$. 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:

$$\hat{f}(\omega) \approx \sum_{k=1}^{N-1} f(t_k) e^{i\omega t_k} \Delta t \tag{2.16}$$

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):

$$Df(n) = \sum_{k=0}^{N-1} f(k) w^{-nk}, where\ w = e^{2\pi i/N} \tag{2.17}$$

And by viewing f and Df as vectors then $Df = M_N f$ where:

$$f = \begin{pmatrix} f(0) \\ \vdots \\ f(N-1) \end{pmatrix} \tag{2.18}$$

And:

$$Df = \begin{pmatrix} Df(0) \\ \vdots \\ Df(N-1) \end{pmatrix} \tag{2.19}$$

And:

$$M_N = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & e^{-1\cdot 2\pi i/N} & e^{-2\cdot 2\pi i/N} & \dots & e^{-(N-1)\cdot 2\pi i/N} \\ 1 & e^{-2\cdot 2\pi i/N} & e^{-2\cdot 2\pi i/N} & \dots & e^{-2(N-1)\cdot 2\pi i/N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & e^{-(N-1)\cdot 2\pi i/N} & e^{-2(N-1)\cdot 2\pi i/N} & \dots & e^{-(N-1)^2\cdot 2\pi i/N} \end{pmatrix} \tag{2.20}$$

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:

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

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

$$\psi_{(sl)}(t) = 2^{\frac{-s}{2}}\psi\left(2^{-s}t - l\right) \tag{2.23}$$

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:

$$Peak = |x|_{max} \tag{2.25}$$

$$Mean = \frac{1}{n} sum_{i=1}^{n} x_i \tag{2.26}$$

$$RMS = \sqrt{\frac{1}{n} sum_{i=1}^{n} x_i^2} \tag{2.27}$$

$$Peak - to - peak = |x|_{max} + |x|_{min} \tag{2.28}$$

$$CrestFactor(CF) = \frac{S_{max}}{RMS} \tag{2.29}$$

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.

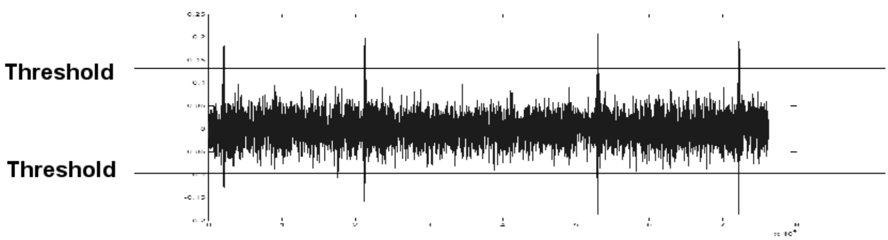

Figure 2.10: Thresholding impulse peaks

Technical details about signal tresholding 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:

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

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

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

A graphic representation of a standard deviation "bell" curve in combination with thresholds δ 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].

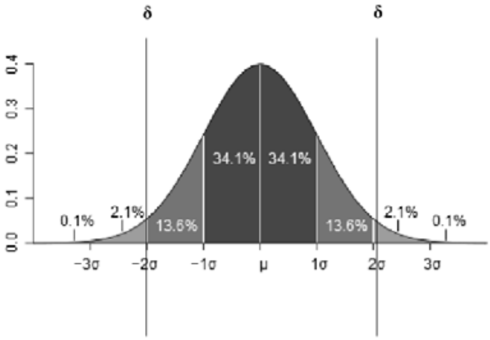

Figure 2.11: δ in standard deviation

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]:

$$FV = [Amp(f_1), Amp(f_2), ..., Amp(f_n)] \tag{2.32}$$

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:

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

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 it's 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.

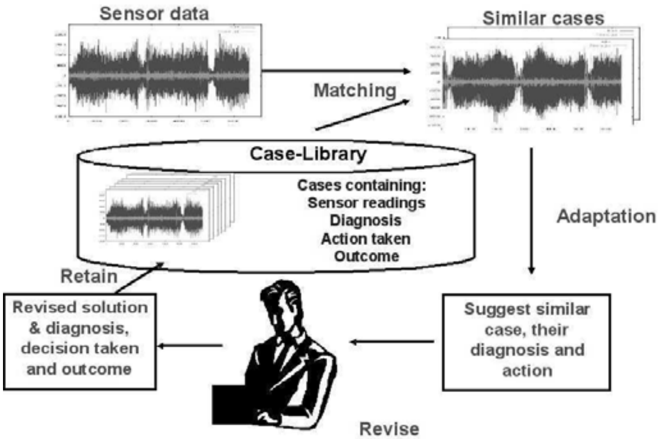

Figure 2.12: The CBR process.

Case Retrieval

To retrieve cases similar to a new problem the system needs a matching function able do 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.

$$Similarity(N,R) = \sum_{i=1}^{n} w_i \times f(N_i, R_i) \tag{2.34}$$

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. McCullogh 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 McCullogh 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:

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

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

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

$h(x)$ 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 $\frac{1}{1+e^{-x}}$ 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:

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

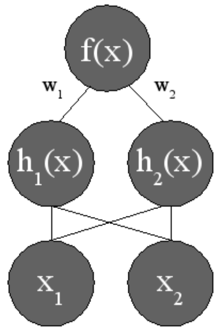

Figure 2.13: A two-layer neural network

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:

$x_1 \in class_1$
$x_2 \in class_0$
$x_3 \in class_0$

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

$f'(x_1) = f(x_1) = w_1 * h_1(x_1) + w_2 * h_2(x_1) = 1$
$f'(x_2) = f(x_2) = w_1 * h_1(x_2) + w_2 * h_2(x_2) = 0$
$f'(x_3) = f(x_3) = w_1 * h_1(x_3) + w_2 * h_2(x_3) = 0$

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:

$$w_j \leftarrow w_j + \alpha \times \mathrm{Err} \times g'(in) \times x_j \tag{2.38}$$

where $Err = y - hw(x)$ for true output y minus network output $hw(x)$ and g' is the derivate of the activation function and α 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:

$$\frac{\left[\sum w_c\right]^2}{\left[\sum w_s\right]\left[\sum w_n\right]} \tag{3.1}$$

Where:

w_c = weights in common clusters between stored and new case
w_s = weights of clusters in stored case
w_n = 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

Table 3.1: Case Structure. The Measurement Part.

Case id	Measure1	Measure2	...	MeasureN
Case i	M1	M2	...	MN

Table 3.2: Case Structure Fault Part

Class	Comp.	Deviation	Hierarchy
Class	Comp.	X%	$M_i L_i$

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.

	M1	M2	M3	Comp	Devi
New Case	0.6	0.7	0.2	C_1	75
Neighbor1	0.6	0.7	1.1	C_1	23
Neighbor2	0.7	0.4	1.3	C_1	24
Neighbor3	0.7	0.4	1.3	C_2	11

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 3.4: structure of a satellite case (problem part).

Case id	Length of time series	Sampling rate	Lower bound	Upper bound
1234	1000	45	-3	10
2345	2000	60	0	10

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:

$$d_g\left(R,W\right) \quad = \quad \frac{1}{k}\sum_{i=1}^{k} d_i\left(R_i,W_i\right) \tag{3.2}$$

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

1. A similarity threshold determines if the new case should be considered 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.

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.

Serial Number	Type	Fault	Diagnosis and Repair	Features 1-n
45634	4500	2	...	...

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 retaining (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 implemented 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 prototypes. 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.

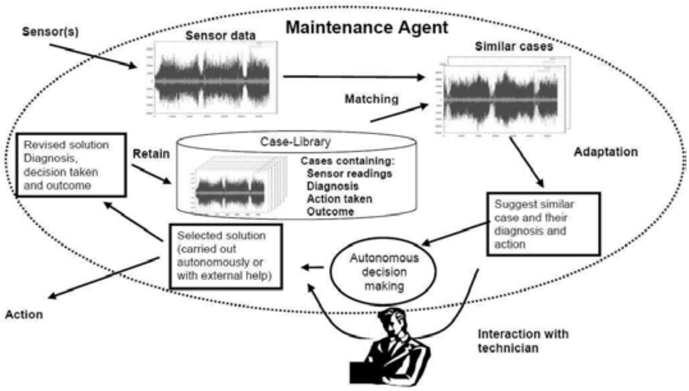

Figure 4.1: Outline of a maintenance agent in its environment [paper F].

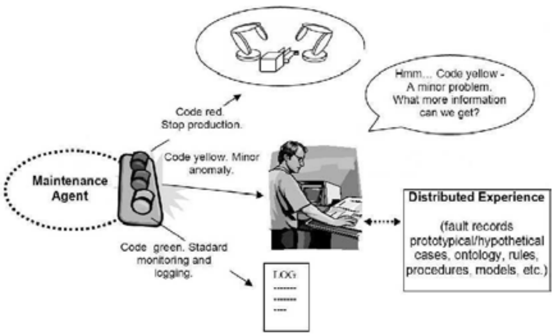

Figure 4.2: A scenario demonstrating operation of an agent system.

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 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 industrial 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

[1] Coetzee Jasper L. A holistic approach to the maintenance "problem". *Journal of Quality in Maintenance Engineering*, 5(3):276–280, 1999.

[2] M. Bengtsson, E. Olsson, P. Funk, and M. Jackson. Technical design of condition based maintenance system - a case study using sound analysis and case-based reasoning. *Proceedings of the 8th Maintenance and Reliability Conference*, 2004.

[3] Peter Funk, Erik Olsson, Marcus Bengtsson, and Ning Xiong. Case-based experience reuse and agents for efficient health monitoring, prevention and corrective actions. pages 445–453, June 2006.

[4] M. Nilsson. *Retrieve and Classify.* Ph D Thesis, Mlardalen University Press, 2005.

[5] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Com*, 7:39–59, 1994.

[6] M. Nilsson and M. Sollenborn. Advancements and trends in medical case-based reasoning: An overview of systems and system development. *Proceedings of the 17th International Flairs Conference (Flairs'04)*, May 2004.

[7] Shahina Begum, Mobyen Uddin Ahmed, Peter Funk, Ning Xiong, and Bo von Schele. A case-based decision support system for individual stress diagnosis using fuzzy similarity matching. *Computational Intelligence (CI)*, in press, December 2008.

[8] Mobyen Uddin Ahmed, Shahina Begum, Peter Funk, Ning Xiong, and Bo von Schele. Case-based reasoning for diagnosis of stress using enhanced cosine and fuzzy similarity. *Transactions on Case-Based Reasoning on Multimedia Data*, Volume 1(Number 1), October 2008.

[9] Sentenitia Software Inc. Case advisor.

[10] P. W. Grant, P. M. Harris, and L. G. Moseley. Fault diagnosis for industrial printers using case-based reasoning. *Engineering Applications Artificial lntelligence*, 9:163–173, 1995.

[11] M. G. Thurston. An open standard for condition-based maintenance systems. *AUTOTESTCON Proceedings*, pages 401–415, 2001.

[12] A. Barber. *Handbook of Noise and Vibration Control. 6th Edition.* Elsevier Advanced Technology, 1992.

[13] M. Bengtsson. Condition based maintenance system technology - where is development heading. *Proceedings of the 17th European Maintenance Conference*, 2004.

[14] D. P. Townsend. *Dudleys Gear Handbook - The Design, Manufacture and Application of Gears.* Mc-Graw Hill, 1992.

[15] E. Beckenstein G. Bachman and L. Narici. *Fourier and wavelet analysis.* Springer, 2000.

[16] Shannon C.E. Communication in the presence of noise. *Proceedings of the IEEE*, 86(2):447–457, 1998.

[17] J. Lin. Feature extraction of machine sound using wavelet and its application in fault diagnosis. *NDTE International*, 34:25–30, 2001.

[18] Thomas M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[19] I. Watson. *Applying Case-Based Reasoning: Techniques for Enterprise Systems.* Morgan Kaufmann, 1997.

[20] J. Kolodner. *Case-Based Reasoning.* Morgan Kaufmann, 1993.

[21] B. Mark. Case-based reasoning for autoclave management. *Proceedings of the Case-Based Reasoning Workshop*, 1989.

[22] Warren S. McCulloch and Walter Pitts1. A logical calculus of the ideas immanent in nervous activity. *Journal Bulletin of Mathematical Biology, Springer New York*, 86(2):99–115, 1943.

[23] D.O. Hebb. *Retrieve and Classify.* Lawrence Erlbaum, 1949.

[24] Frank Rosenblatt. Principles of neurodynamics. *Parallel Distributed Processing*, 1,2, 1962.

[25] S. ussel and P. Norvig. *Artificial Intelligence A Modern Approach Second Edidtion.* Prentice Hall International, 2003.

[26] C. Pous, J. Colomer, and J. Melendez. Extending a fault dictionary towards a case-based reasoning system for linear electronic analogue circuits diagnosis. *Proceedings of the European Conference on Case-Based Reasoning*, pages 748–762, 2004.

[27] A. Varma. Icarus: Design and deployment of a case-based reasoning system for locomotive diagnostics. *Third International Conference on Case-Based Reasoning*, pages 581–596, 1999.

[28] D. Wilson and T. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38:257286, 2000.

[29] K. K. Penta and D. Khemani. Satellite health monitoring using cbr framework. *Seventh European Conference*, 21:732–747, 2004.

[30] T. Han B. Yang and Y. Kim. Integration of art-kohonen neural network and case-based reasoning for intelligent fault diagnosis. *Expert Systems with Applications*, 26:387395, 2004.

[31] G. A. Carpenter and S. Grossberg. The art of adaptive pattern recognition by a self-organizing neural network. *IEEE Transactions on Computers*, 21:77–88, 1988.

[32] T. Kohonen. *Self-organizing maps.* Springer-Verlag, 1995.

[33] D. W. Aha, D. Kibler, and M. Albert. Instance based learning algorithms. *Machine Learning*, 6:3766, 1991.

[34] Erik Olsson, Mikael Hedelind, Mobyen Uddin Ahmed, and Peter Funk. Experience reuse between mobile production modules - an enabler for the factory-in-a-box concept. In *The Swedish Production Symposium*, August 2007.

[35] P. Funk and M. Jackson. Experience based diagnostics and condition based maintenance within production systems. *COMADEM 2005, The 18th International Congress and Exhibition on Condition Monitoring and Diagnostic Engineering Management*, August 2005.

[36] M. Jackson. Factory in a box, 2005.

[37] P. Funk. Eken-efficient knowledge and experience reuse within the business world, 2006.

II

Included Papers

Chapter 6

Paper A: Fault Diagnosis in Industry Using Sensor Readings and Case-Based Reasoning

E. Olsson, P. Funk and N. Xiong. Fault Diagnosis in Industry Using Sensor Readings and Case-Based Reasoning. Journal of Intelligent & Fuzzy Systems, 15, pages 41–46, 2004.

Abstract

Fault diagnosis of industrial equipments becomes increasingly important for improving the quality of manufacturing and reducing the cost for product testing. Developing a fast and reliable diagnosis system presents a challenge issue in many complex industrial scenarios. The major difficulties therein arise from contaminated sensor readings caused by heavy background noise as well as the unavailability of experienced technicians for support. In this paper we propose a novel method for diagnosis of faults by means of case-based reasoning and signal processing. The received sensor signals are processed by wavelet analysis to filter out noise and at the same time to extract a group of related features that constitutes a reduced representation of the original signal. The derived feature vector is then forwarded to a classification component that uses case-based reasoning to recommend a fault class for the probe case. This recommendation is based on previously classified cases in a case library. Case-based diagnosis has attractive properties in that it enables reuse of past experiences whereas imposes no demand on the size of the case base. The proposed approach has been applied to fault diagnosis of industrial robots at ABB Robotics and the results of experiments are very promising.

Key Words: case-based reasoning, fault diagnosis, feature extraction, signal filtering, wavelet analysis

6.1 Introduction

A fault is an abnormal state of a machine or a system such as dysfunction or malfunction of a part, an assembly, or the whole system. As machines become larger and more complex with industrial development, the costs and technical know-how required for system maintenance increases substantially. Fast and precise identification of faults and problems in equipments makes a crucial contribution to the enhancement of reliability in manufacturing and efficiency in product testing.

For monitoring purpose, streams of data are gathered by various sensors on-board equipments. Such sensor recordings can be regarded as evidence of origin for recognizing the working conditions of a machine (e.g. normal operation, loose rear wheel, damaged gear). Although experienced key persons can make proper judgment of failures by inspection of the measured signals in many circumstances, it would be fairly hard to do so by moderate staff. Trouble might arise when a fault occurs whereas the experienced personnel are not around due to some reasons like vacation and sickness to mention a few. Things turn still tougher with those sensor signals containing heavy measurement noise such that even skilled operators fail to distinguish faults without supporting tools.

Construction of automatic diagnosis systems based on Artificial Intelligence (AI) methods and techniques receives increasing attention for extending the capability of key personnel and reducing human costs connected with equipment maintenance. Expert systems [1] provide a useful means to acquire diagnosis knowledge directly from key personnel and transform their expertise into production rules. However, the knowledge acquisition and verification processes are difficult and complicated and sometimes experienced technicians even have no idea of how to express their strategies explicitly and accurately. Rule induction [2, 3] and neural network models [4, 5] are data mining methodologies that can be applied to find out fault classification knowledge using previous known examples. They show strong ability in discovering important knowledge from historic data but require a sufficiently large training set to ensure promising outcome and overcome the risk of over-fitting. Unfortunately, in many practical scenarios, merely a very low number of examples are available in support of machine learning.

Case-based reasoning [6] (CBR) offers another alternative to implement intelligent diagnosis systems for real-world applications [7]. Motivated by the doctrine that similar situations lead to similar outcomes, CBR fits well to classify the current new sensor signals based on experiences of past categorizations. The main strength lies in the fact that it enables directly reusing concrete examples in history and consequently eases the knowledge acquisition bottleneck. It also creates the opportunity of learning from experiences but skipping the step of data training such that the over-fitting problem no longer exists. We believe that CBR techniques are of particular application value for diagnosis in real industrial environments where the acquirement of adequate training examples in advance is mostly not realistic if not impossible.

This paper aims to investigate the utility of CBR techniques for diagnosis of industrial equipments based on streams of sensor recordings. The received signals are processed by wavelet analysis to filter out noise and at the same time to extract a group of related features that constitutes a reduced representation of the original signal. The derived feature vector is then compared with the known cases in the case library with its neighboring cases sorted out, and subsequently the new situation is classified by combining the outcomes of those similar cases retrieved. Our presented approach has been applied to fault diagnosis of industrial robots produced by ABB Robotics in Västerås (Sweden) and the preliminary results of evaluation are very promising.

The paper is organized as follows. Section 6.2 gives a general structure for fault classification starting from streams of sensor readings. Signal analysis and feature extraction is addressed in Section 6.3, followed by an outline of necessary details of performing case-based classification using extracted features in Section 6.4. Section 6.5 gives a case study applying the proposed approach to fault diagnosis of industrial robots and some experiment results are demonstrated. Finally the paper is concluded in Section 6.6 with a short summary and remarks.

6.2 Fault Diagnosis Based on Sensor Signals

Abnormality of industrial machines can be reflected by some key states during their operation. Using sensor technology it is possible to detect and measure the values of these system states and their profiles. We can then process and analyse the collected sensor recordings in order to find out hidden symptoms. The system can, based on the symptoms, reason about the class of fault associated with the machine or make prediction about what potential problem is likely to occur in a near future. A general system structure for this purpose is illustrated in Figure 6.1, which includes signal filtering, feature extraction, and pattern classifier as its important components.

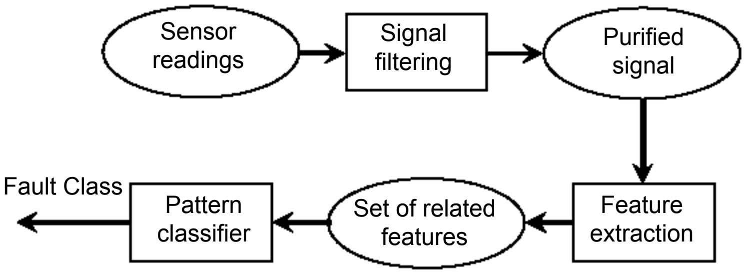

Figure 6.1: Fault diagnosis based upon sensor signals.

Signal filtering is used to purify original sensor readings by removing the noises contained in the signals such that more reliable diagnosis results will be warranted. Usually there are two kinds of noises involved in the perceived signals; one is measurement noise due to intrinsic imprecision of sensors and the other is external noise caused by disturbance from surroundings and which is added to the sensor data received. Signal recovery from external background noise has been well dealt with by applying signal processing methods like wavelet analysis and time domain averaging (see [8, 9]). The reduction of measurement errors is outside the scope of this paper, but interested readers can refer to sensor fusion systems in which Bayesian based filtering approaches such as Kalman filtering [10] and particle filtering [11] merit to be used to obtain more accurate estimates of related states.

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 which the underlying system is dynamic. If so, the measurements of a state generally change with the time rather than constantly staying at a static level. This means that the observations of the system are continuously varying which makes it hard to handle them directly in diagnosis. In order to supply the pattern classifier (in Figure 6.1) with a moderate number of inputs for effective analysis and reasoning, representative features from the sensor signals have to be extracted. Our point is that for many tasks the collection of extracted features ought to be adequate to give a concise and complete description of the condition of the system to diagnose.

Regarding fault classification a number of different methodologies can be considered. Expert systems were developed in support of gathering, representing and utilizing human expert knowledge for problem solving but they suffer from the knowledge acquisition bottleneck. Regression functions fit themselves into defining linear classification boundaries using a low number of attributes as function variables. For complex diagnosis situations with nonlinear boundaries and many relevant features a classifier based on artificial neural network 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 contrast CBR has the advantages of entailing no training beforehand but still exhibiting the ability for incremental learning if new useful cases are properly injected into the case library. This motivates us to develop a case-based classifier of fault patterns in this paper. We 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.

6.3 Case-Based Classification using Extracted Features

As mentioned before, the measurements from a dynamic industrial system constitute time-varying data streams that are not suitable for immediate usage. Hence we need to dig out representative features hidden in the signal profiles prior to fault classification. The features extracted are delivered to the fault classifier as a probe case. 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. Typical features of this kind can be peak value, start time, overshoot, rising time, mean value, integral, standard deviation, etc. In practice what features to derive from the time domain is commonly ad-hoc and problem dependent. An example of using time-based features for case-based circuit diagnosis is illustrated in [12].

Frequency-based features characterize sensor signals according to their amplitudes under significant frequencies. As many fundamental signal analysis methods are available to yield frequency spectra, we seem to have more solid basis for extracting features based on frequency than for deriving time-based features. We thus adopt frequency-based features as descriptors of condition parts of cases in our research. Generally a vector of frequency-based features is formulated as

$$FV = [Amp(f_1), Amp(f_2), ..., Amp(f_n)] \tag{6.1}$$

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

Wavelet analysis [13] 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 [9]. Technical details of wavelet analysis for feature extraction are discussed in [14], wherein a comparative study was also performed between wavelet analysis and Fourier transform demonstrating the superiority of the wavelet approach in producing high quality features for case-based classification.

After the features have been extracted from the sensor signals, we perform case-based reasoning to make a classification of the current fault using known cases in the case library. Figure 6.2 gives an overall illustration of this procedure, which consists of the following two steps:

1. Retrieval: compare the feature vector with the known cases in the library by means of similarity calculation and subsequently select the k nearest cases exhibiting the highest similarity degrees;

2. Solution fusion: determine the fault class associated with the current feature vector in terms of both the classes of the retrieved cases and their similarity values with respect to the probe case.

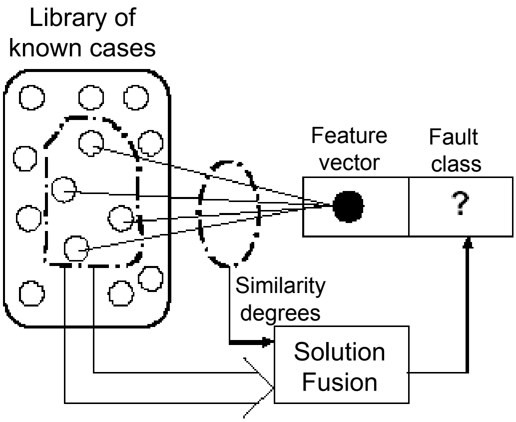

Figure 6.2: Case-based fault classification.

Given a feature vector $Y = (y_1, y_2, \ldots, y_n)$, its similarity degree with case C in the case library is defined as

$$Similarity(Y, C) = \sum_{i=1}^{n} w_i \times (1 - |norm(y_i) - norm(c_i)|) \qquad (6.2)$$

where $w_1, w_2, \ldots, w_n$ are attribute weights reflecting different importance of individual features, c_i represents the ith feature of case C, and $norm(y_i)$ and $norm(c_i)$ denote the normalized values of y_i and c_i respectively.

In the step of solution fusion we can easily judge a fault class if all the retrieved cases have that class as their outcomes. Otherwise voting is launched among the classes that exist in the retrieved cases. For every such class B_j we calculate its voting score as

$$VS(B_j) = \sum_{P \in Rs} \begin{cases} \text{Similarity(Y,P), if P has class } B_j \\ \text{0 otherwise} \end{cases} \tag{6.3}$$

where Rs denotes the set of retrieved cases and P is the current feature vector. Finally the fault is classified into the class that has the largest voting score.

6.4 Application to Fault Diagnosis for Industrial Robots

As a case study we applied the proposed approach to diagnosis of industrial robots manufactured by ABB Robotics in Västerås, Sweden. The prototype system developed for this purpose is shown in Figure 6.3 Sound signals are gathered from the robot to be tested via a microphone device and then transmitted to the computer for pre-processing. The pre-processing is tasked to filter out or remove unwanted noise as well as identify period information from a sound profile. Subsequently sound features are extracted from the frequency domain and they are assembled into a feature vector as a condensed representation of the original sound signal. Classification of the feature vector is performed based upon previously classified sound descriptions in the case library. The experiments have shown that this system is able to successfully diagnose faults in an industrial robot based on a low number of previous examples.

It is worth mentioning that the above prototype system has some similarities with the Open System Architecture for Condition Based Maintenance (OSA-CBM) [15]. That architecture suggests that a Condition Based Maintenance (CBM) system be divided into seven modules [16] including sensors, signal processing, condition monitoring, diagnosis, prognosis, decision support, and presentation. The system presented here in this paper has a microphone as sensor module and pre-processing & feature extraction steps as signal processing module in correspondence to the OSA-CBM architecture. In addition, the case-based classification in

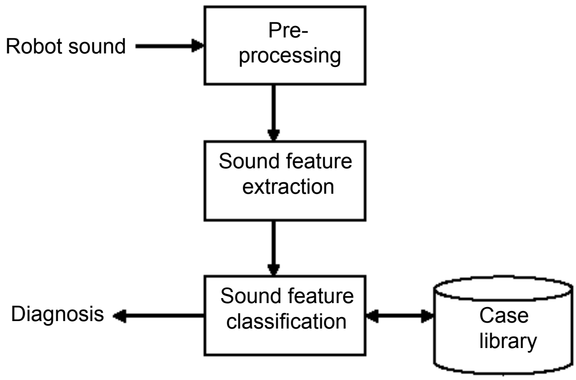

Figure 6.3: Schematic outline of the prototype system.

Figure 6.3 also serves condition monitoring by detecting and identifying deviations in sound profiles.

6.4.1 Pre-processing and Feature Extraction

Sounds of robots in industrial environments typically contain unwanted noise from background. A robot fault is often indicated by the presence or increase of impulsive elements in the sound. The detection of these impulsive sound elements can be hard. This is owing to the various sporadic background noises prevalent in industrial environments and they are added to the received sound signals. Before the attempt of classification, the sound from the robot has to be pre-processed in order to remove as much unwanted noise as possible. In Figure 6.4 the two pre-processing steps are shown which are termed as period extraction (left box) and time domain averaging (right box).

In order to obtain time information about the robot arm movement, period has to be detected from the sound profile. A period refers to the duration within which the robot arm rotates from the start position to its destination. Commonly sounds from the robot are recorded in a time span with a few periods. Each period for the robot arm movement is characterized by a continuous sound followed by a short time of silence. After getting period information a mean length for periods is calculated from a number of successive periods of the robot sound, thereby eliminat-

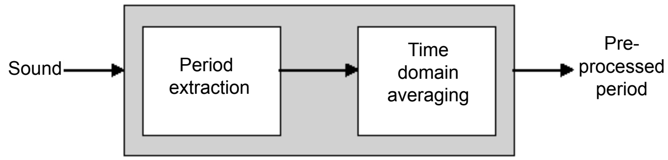

Figure 6.4: Pre-processing of sound data in the prototype system.

ing sporadic impulsive elements from unwanted sources and enhancing repeating impulse sound normally related with robot faults.

After identifying period information a set of important features must be extracted from the sound signal within a single period. Wavelet analysis is applied herein to find out such features for sound classification. In a related paper [14] we experimentally verified that, under certain circumstances of strong background noise, wavelet outperforms Fourier transform in supplying distinguishable feature vectors between different faults for case-based classification.

6.5 Sound Classification and Results

Sounds from 24 healthy robots and 6 faulty robots were collected to enable case-based classification of conditions of robots. Two types of faults need to be recognized in the experiments hereafter called Fault 1 and Fault 2. A notch on the big gear wheel in the gearbox causes Fault 1. This fault is hearable and is characterized by a low frequency impulse sound in the middle of the rotation of the axis. Fault 2 is caused due to a slack between the gear wheels in the gearbox and can be heard as bumps at the end of each rotation.

A feature vector is assembled with peak wavelet coefficients taken from different depths in a wavelet package tree [13] and it is then matched with the previously inserted cases in the case library. The prototype system demonstrated quite good performance by making right judgements in 91% of the all tests (see further down). Table 6.1 displays a ranked list of the three best matching cases in the case library according to the similarity values calculated. As can be seen from the table, a previously diagnosed notch fault recording is deemed to be the most similar case thereby making the strongest recommendation to classify the probe situation into notch fault. The cases ranked the second (case #12) and the third (case #4) are descriptions classified as normal in the case library. This list of the most similar cases can be presented to human operators as decision support.

Table 6.1: A ranking of the most similar cases for the sound profile.

Case name	Similarity	Case ranking
Notch fault #2	98%	1
Normal case #12	84%	2
Normal case #4	83%	3

We also investigated the classification accuracy in relation with different feature vector sizes in order to assess the smallest number of features that still produce good classification performance. The diagram in Figure 11.11 indicates the relation between the classification error rate and the number of features. The upper curve in the figure shows the results when only top 1 case was considered for solution fusion. The curve below in the diagram shows the classification results when the top three cases were considered. When only the nearest case was considered, the system produced a classification rate of 91%. When the three nearest cases were considered, the classification rate of the system rose to 99%.

6.6 Conclusions

This paper presents a new approach to fault diagnosis of industrial equipments using case-based reasoning and sensor data. Wavelet analysis is advocated as an effective means to remove noise and extract a set of good quality features. The assembled feature vector serves as condition description of a case. Case-based fault classification gives considerable benefits in numerous practical applications. They include:

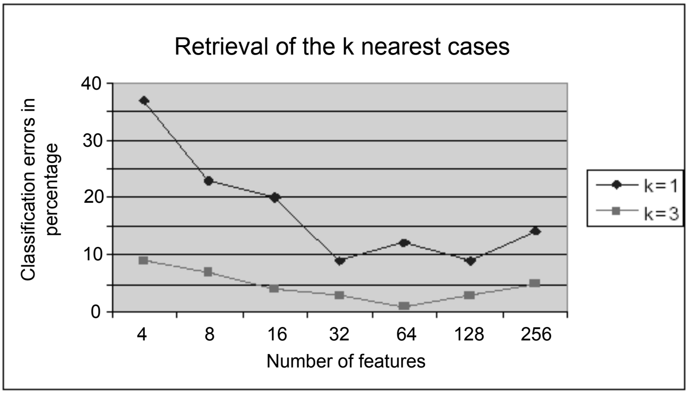

Figure 6.5: Relation between classification performance and the number of features.

It fosters experience reuse and sharing in the sense that classified signal descriptions from different sources can be easily added to a common library.

- It does not require a complete case library for functioning properly. As no training of known cases is needed, there exists no over-fitting risk any more.
- It enables improving classification performance as long as newly classified signal descriptions are injected into the case library.
- It entails case retrieval, giving intermediate results that are user-friendly and offer a sort of decision support for human operators in diagnosis.

6.7 Acknowledgement

This research is part of the ExAct project, funded by The Swedish Foundation for Strategic Research (ProViking) and ABB Robotics, Volvo, SKF and Svenska Underhållsföretagen (SCEMM).

Bibliography

[1] Y. Zhang et al. Fuzzy production rule based expert systems. *Fuzzy Sets and Systems*, 44:391–403, 1991.

[2] P. Clark and R. Boswell. Rule induction with cn2: Some recent improvements. *Proceedings of the European Working Session Learning*, pages 151–163, 1991.

[3] J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[4] J. P. Bigus. *Data Mining with Neural Nets.* McGraw-Hill, 1996.

[5] R. P. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Mag*, 4:4–22, 1987.

[6] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Com*, 7:39–59, 1994.

[7] M. Nilsson, P. Funk, and M. Sollenborn. Complex measurement in medical applications using a case-based approach. *Workshop Proceedings of the International Conference on Case-Based Reasoning*, pages 63–73, 2003.

[8] S. K. Lee. and White P. R. The enhancement of impulse noise and vibration signals for fault detection in rotating and reciprocating machinery. *Journal of Sound and Vibration*, 217:485–505, 1998.

[9] J. Lin. Feature extraction of machine sound using wavelet and its application in fault diagnosis. *NDTE International*, 34:25–30, 2001.

[10] Y. Bar-Shalom and X. Li. *Estimation and Tracking: Principles, Techniques, and Software.* Artech House, 1993.

[11] A. Marrs N. Gordon and D. Salmond. *Nonlinear and Nonstationary Signal Processing.* Cambridge University Press, 2001.

[12] C. Pous, J. Colomer, and J. Melendez. Extending a fault dictionary towards a case-based reasoning system for linear electronic analogue circuits diagnosis. *Proceedings of the European Conference on Case-Based Reasoning*, pages 748–762, 2004.

[13] E. Beckenstein G. Bachman and L. Narici. *Fourier and wavelet analysis.* Springer, 2000.

[14] E. Olsson, P. Funk, and M. Bengtsson. Case-based diagnosis of industrial robots based on acoustic signals. *Proceedings of the European Conference on Case-Based Reasoning*, pages 686–701, 2004.

[15] M. G. Thurston. An open standard for condition-based maintenance systems. *AUTOTESTCON Proceedings*, pages 401–415, 2001.

[16] M. Bengtsson. Standardization issues in condition based maintenance. *Proceedings of the 16th International Congress on Condition Monitoring and Diagnostic Engineering Management*, pages 651–660, 2004.

Chapter 7

Paper B: Fault Diagnosis of Industrial Robots using Acoustic Signals and Case-Based Reasoning

Erik Olsson, Peter Funk, Marcus Bengtsson. Fault Diagnosis of Industrial Robots using Acoustic Signals and Case-Based Reasoning, In the 7th European Conference on Case-Based Reasoning, Madrid, editors: Peter Funk, Pedro Gonzales, pages 686-701 Springer LNCS, June 2004.

Abstract

Abstract. In industrial manufacturing rigorous testing is used to ensure that the delivered products meet their specifications. Mechanical maladjustment or faults often show their presence through abnormal acoustic signals. This is the same case in robot assembly - the application domain addressed in this paper. Manual diagnosis based on sound requires extensive experience, and usually such experience is acquired at the cost of reduced production efficiency or de- graded product quality due to mistakes in judgments. The acquired experience is also difficult to preserve and transfer and it often gets lost if the correspond- ing personnel leave the task of testing. We propose herein a Case-Based Rea- soning approach to collect, preserve and reuse the available experience for robot diagnosis. This solution enables fast experience transfer and more reliable and informed testing. Sounds from normal and faulty robots are recorded and stored in a case library together with their diagnosis results. Given an unclassi- fied sound signal, the relevant cases are retrieved from the case library as refer- ence for deciding the fault class of the new case. Adding new classified sound profiles to the case library improves the systems performance. So far the de- veloped system has been applied to the testing environment for industrial ro- bots. The preliminary results demonstrate that our system is valuable in this application scenario in that it can preserve and transfer the related experience among technicians and shortens the overall testing time.

7.1 Introduction

Mechanical faults in industrial robots often show their presence through abnormal acoustic signals compared with the normal ones. Correct classification of the robot sound may be a very critical part of the end-test. An incorrect classification of the sound can result in the delivery of a faulty robot to the customer. A technician needs rich experience to make a reliable diagnosis of robots. The importance of fault detection based on sound is confirmed by a current activity of Volkswagen which sells Cd's containing recordings of different faults in equipments to aid technicians in classifying audible faults. The use of sound and vibration measurements for the purpose of fault detection in end-testing of industrial equipments is today most commonly practiced by gearbox manufacturers. The measurements are shown graphically and analysed manually by a technician via careful observations of the measurements (normal/high amplitude level, frequency distribution etc.). Some toolbox systems exist (e.g. Mathlab or more sound and vibration profiled tools such as the Plato toolbox [1] that offer a variety of aids enabling experts to analyse and visualise data in different ways. Some additional modules are offered able to classify a measurement as pass/failure or compare it with a library of faults. These systems are semiautomatic, large and run on PC computers. Some diagnostic systems use neural nets, such as Dexter [2] employing probabilistic neural net for classification.

We propose the use of a Case-Based Reasoning (CBR) system resorting to a nearest neighbour approach for a lightweight solution of recognising and diagnosing audible faults in industrial robots. Sound is recorded with a microphone and compared with previous recordings; similar cases are retrieved and shown to the user with correspondence to relevant diagnosis results in history. A prototype system for this purpose has been developed.

AI techniques such as Case-Based Reasoning (CBR) have some advantages in this category of applications. The fundamental idea of CBR applying old knowledge of problem solving to solve new problems is very feasible for industrial applications. Implementing this technique in industrial applications preserves experience that would be often lost if skilled personnel leave their employments. The system aids technicians in making a correct diagnosis of industrial robots based on earlier classifications of similar sounds. It also eases the knowledge acquisition bottleneck [3].

This paper gives an overview of the CBR system for robotic fault classification and describes the implemented prototype system as well as some initial evaluation results. The system is able to successfully diagnose faults in an industrial robot based on sound recordings (4 recordings from faulty robots and 20 recordings from normal robots are used in the evaluation). The system elicits classifiable features from the sound recordings and makes a diagnosis according to prior knowledge.

The paper is organized as follows. Section 11.3 gives a brief overview of the sound classification technique. Section 11.4 describes the model used in this paper to classify sound recordings. Sections 11.5, 11.5.1 and 11.5.2 describe the implementation of the prototype classification system based on the model. Section 11.5.3 discusses system evaluation with a case study. Section 11.5.4 gives an experimental comparison of FFT and wavelet analysis and finally section 11.6 concludes this paper with summary and conclusions.

7.2 Classifying Sound Recordings

This section gives short background knowledge for sound classification and outlines some of the methods and techniques used to filter, analyse and classify sound re- cordings.

7.2.1 Filtering and Pre-processing

Filtering is used to eliminate unwanted components in the signal by removing noise and distortions. A number of different techniques, such as adaptive filters, wavelet analysis and time domain averaging have been developed for signal filtering and noise reduction (see [4] [5]). The filtering process may be complicated in some scenarios because of heavy background noise. After a successful pre-processing the signal will have an increased Signal to Noise Ratio (SNR), which makes it more amenable for further processing such as feature extraction.

7.2.2 Features and Feature Vector

When experienced technicians are classifying robot sound they listen for abnormali- ties in the sound. An indication of an abnormal sound can be the presence or absence of certain acoustic features. Using feature vector as the signature for sound is a welladopted method to detect and identify faults in machinery. It is also commonly used in CBR systems. A simplified example for feature vector from a sound profile is shown below where the elements above the sign are signal amplitude values and those under denote the corresponding frequencies.

$$\left[\frac{max\_value45}{300Hz}, \frac{max\_value18}{520Hz}, \frac{max\_value89.6}{745Hz}\right] \tag{7.1}$$

The adoption of frequency-based features in this context is motivated by the aware- ness of resonant frequency of each mechanical part that depends on its mass and rigidity. Hence the faults occurring in different parts will result in different frequency spectra. Experienced technicians often listen for such features on an intuitive basis in order to propose a diagnosis in terms of his/her experience. However technicians may not always be able to point out these features that he/she uses to classify sounds.

Wavelet analysis [6] is a powerful technique for filtering out noises and transforming analogue signals to frequency diagrams. It is hence adopted in our research to establish frequency-dependent features from polluted acoustic signals collected from environments with strong background noise. Extraction of sound features based on wavelet will be detailed in section 11.5.

7.2.3 Classification Process

A number of different methods are available for the classification of machine sound. The selection of classification method is based on the nature of the task. A simple classification may only require a single test with a threshold (e.g. amplitude above or below 10) for a complete classification.

A different approach to the classification of feature vectors is to use Artificial Neural Nets (ANN). Reliable classification using the ANN approach requires prior training of the network with a sufficient number of classified examples. Moreover, once a new important case is recognized, the old network has to be retrained in order to assimilate this new acquired experience. However, in our task of robot fault diagnosis, sufficient samples of classified sound recordings required for training are frequently not available.

7.3 Classifying Sound Recordings

This section gives an overview and introduction to the case-based classification of machine sound. The different steps, pre-processing, feature identification and classification are described in sections 11.5, 11.5.1 and 11.5.2, respectively. Sound is obtained from the robot to be diagnosed via a microphone as shown at the top left in Fig. 7.1. The sound is recorded to a computer and the recording is taken as input to the pre-processing step. The pre-processing component in Fig. 7.1 is responsible for filtering and removal of unwanted noise. It also extracts period information from the sound.

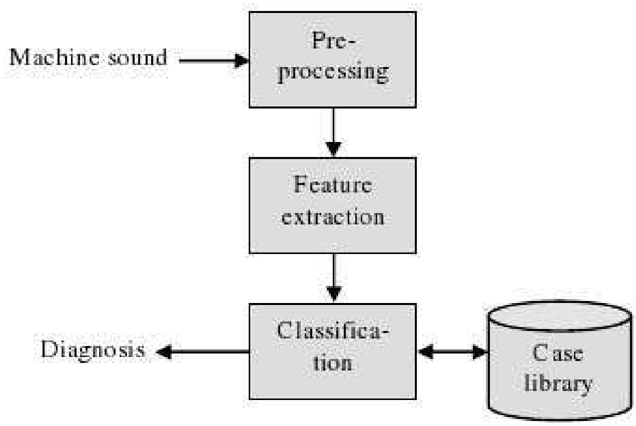

Figure 7.1: Schematic picture of the system

In the feature identification process, the system uses a two-pass model, first identifying features and then creating a vector with the extracted features. Once the features are identified, the system classifies the feature vector. The classification is based on previously classified measurements (case library) in Fig. 7.1. After a new sound has been classified it is added to the case library. The classification process will be described in section 11.5.2. A diagnosis based on the result of the classification is shown to the technician. In the research prototype a ranked list of the most similar cases based on a nearest neighbour function is presented as decision support to the technician.

7.3.1 Comparison to the OSA-CBM Architecture

The design of the system described in this paper has some similarities with the Open System Architecture for Condition Based Maintenance (OSA-CBM) [7]. This architecture is seen as a proposed standard for Condition Based Maintenance (CBM) system which is recommended to consist of seven modules [8], including sensors, signal processing, condition monitoring, diagnosis, prognosis, decision support, and presentation (see Fig. 7.2). In the system presented in this paper the microphone can be regarded as a sensor module. The pre-processing and feature extraction components play the role of signal processing. The classification (with the case-library) component performs both condition monitoring and diagnosis as it can both detect deviations in the sound profiles and classify different sound profiles into different fault modes.

7.4 Pre-Processing

Robot sound typically contains unwanted noise. The presence of a fault is often indicated by the presence, or increase in impulsive elements in the sound. The detection of these impulsive sound elements can be hard. This is owing to the mixture of signals from normal running of the robot and from various sporadic background noises normally existing within an industrial environment. Before a classification attempt is made, the machine sound is pre-processed in order to remove as much unwanted noise as possible. In this system wavelets are used to purify the raw signal and transform the incoming sound into a series of wavelet coefficients. Selected wavelet values are then used as features.

Fig. 7.3 shows the pre-processing process. It contains two steps; splitting and wavelet analysis. In the first step the signal is split to windows of discrete time steps. The length of each window can be arbitrary.

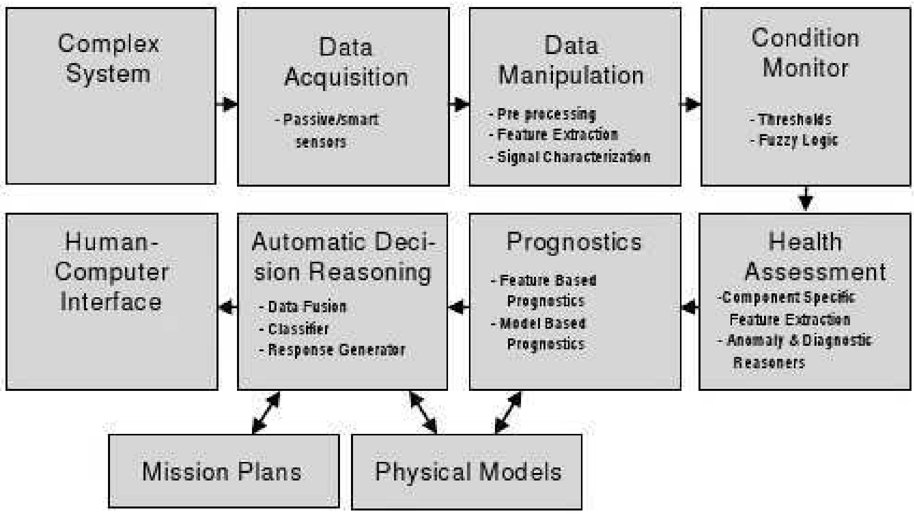

Figure 7.2: The standard OSA-CBM architecture proposed in [9]

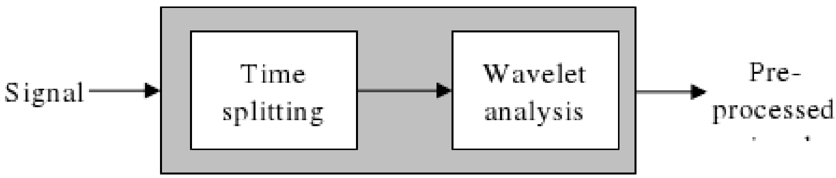

Figure 7.3: Pre-processing of the signal in the system

Each window is then sent to the wavelet analysis algorithm (step #2). The output from the wavelet analysis and from the pre-processing step is a series of wavelet values. Below, the function of each step is further explained.

7.4.1 Time splitting

Only a part of the input signal can be analysed each time conducting the wavelet algorithm. Due to this fact the signal is divided into windows of discrete time steps. The length of each window can be arbitrary but its data size must be 2^n where $n \geq 2$. This is due to the way the wavelet packet algorithm is implemented.

7.4.2 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 (see chapter 11.5.4. The signal $f(t)$ will be represented as a weighted sum of the wavelets $\psi(t)$ and the scaling function $\phi(t)$ by

$$f(t) = A_1\phi(t) + A_2\psi(t) + \sum_{n\in+Z, m\in Z} A_{n,m}\psi(2^n t - m) \tag{7.2}$$

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

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

$$\int_{-\infty}^{\infty} \psi(t)\, dt = 0 \tag{7.3}$$

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

$$\psi_{(sl)}(t) = 2^{\frac{-s}{2}}\psi\left(2^{-s}t - l\right) \tag{7.4}$$

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

$$\phi(t) = \sum_{k=-1}^{N-2} (-1)^k c_{k+1}\psi(2t + k) \tag{7.5}$$

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 (see Fig. 11.6). 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 wavelet coefficient matrix is applied to the input data vector. The matrix is applied in a hierarchical algorithm, sometimes called a pyramidal algorithm. The wavelet data values are arranged so that odd rows contain an ordering of wavelet data values that act as the smoothing filter, and the even rows contain an ordering of wavelet coefficients with different signs that act to bring out the data's detail. The matrix is first applied to the original, full-length vector. Fig. 11.6 shows an example of a data vector consisting of 8 samples. The samples can be any type of data; sensor signals from various process applications, stock market curves etc. In this paper the samples are acoustic signals from a gearbox of an industrial robot.

3	2	1	3	1	4	3	8

Figure 7.4: Original signal consisting of 8 samples

The data vector is smoothed and decimated by half and the matrix is applied again (see Fig. 5).

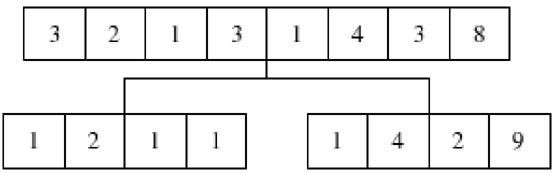

Figure 7.5: Smoothed data vectors

Then the smoothed, halved vector is smoothed, and halved again, and the matrix applied once more. This process continues until a trivial number of ”smooth-smooth- smooth...” data remain (see Fig 11.8).

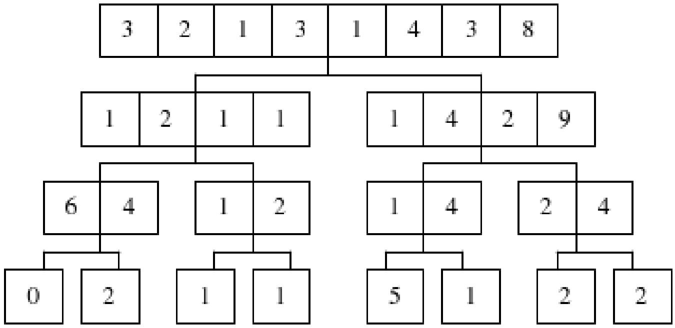

Figure 7.6: The result of the pyramidal algorithm

This system uses the wavelet packet transform algorithm. It is a computer imple- mentation of the Discrete Wavelet Transform (DWT). It uses the Daubecies mother wavelet, scaling function and wavelet coefficients [9].

The result of the pyramidal algorithm is a tree of smoothed data values (see Fig.11.8). Each collection of smoothed data values (node in the tree) can be seen as a time-frequency-packet. Each time-frequency-packet can be seen as a filtered version of the original data samples. As an example, the left packet in Fig. 11.7 can be seen as a low pass filtered version of the original data and the right packet in Fig. 11.7 can be seen as a high pass filtered version of the original data. The leaves of the tree can be seen as high and low pass units of length 20.

The depth of the tree is determined from the length of the input data. If the input data are of length 2^n the depth of the tree will be n. A suitable collection of time-frequency-packets can be selected by taking a cross section of the tree at an arbitrary depth. Each sibling in the cross section of the tree is spanning the entire time of the original data set. This means that going deeper in the tree produces at better resolution in frequency but a poorer resolution in time. The best compromise between time and frequency resolution is to take a cross section in the tree were the length of each Sibling is the same as the number of siblings in the cross section. At a given depth n and with original data size S, the length of a sibling (or leaf) is $\frac{S}{2^n}$ and the number of siblings is 2^n.

The wavelet packet algorithm offers the basis for the Pre-processing process. The input signal is first divided into windows of discrete time steps. Each window is then passed to the wavelet packet algorithm resulting in a wavelet packet tree as pictured in Fig 11.8. The wavelet data values from a cross section of the wavelet packet tree are then passed to the Feature Extraction process.

7.5 Feature Extraction Process

It is necessary to find a suitable form in which to represent and compress the sound data while storing enough information to be able to classify the sound correctly. The feature extraction component uses a two-pass model to achieve this. First, wavelet data values obtained from pre-processing are fed as inputs to the feature extraction component which extracts features from these coefficients (left box in Fig. 7.7). The extracted features are then stored in a feature vector (right box in Fig. 7.7).

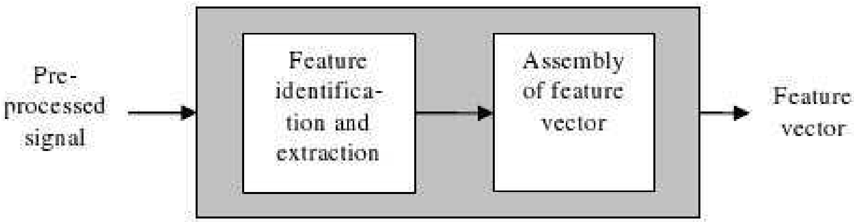

Figure 7.7: Feature identification in the system

7.5.1 Feature Identification

Our system uses normalized wavelet data values as features. The values are selected from a cross-section of the wavelet packet tree. Gear defects often show their presence as sharp peaks or dips in the sound. Such peaks or dips can be spotted in some dominant wavelet data values in certain packets in the cross section of the wavelet packet tree. The feature extraction component examines the wavelet data values and extracts one dominant value from each packet in a cross section at an arbitrary depth. In Fig. 11.9 the grey area shows a cross section at level 2 in the tree. The chosen coefficients are those that are marked as bold. They are chosen because they are the dominant values in each packet in that cross section.

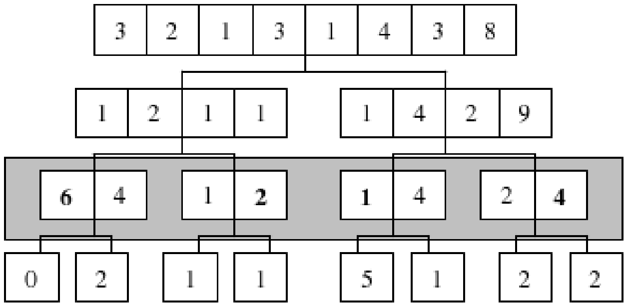

Figure 7.8: Feature identification from wavelet data values

7.5.2 Assembly of a Feature Vector

A feature vector is assembled from these dominant wavelet values. A feature vector forms a cross section of wavelet data values at level n in the wavelet packet tree containing 2^n features. This system is dynamic and can assemble vectors from all depths of the tree. The feature vector assembled from Fig. 11.9 is $[6, 2, 1, 4]$.

In order to purify sounds from various sporadic background noises normally existing within an industrial environment - several cross sections of the wavelet packet tree from a series of windows are passed from the Pre-processing component to the feature extraction component. The amount of cross sections passed to feature extraction is dependent on the length of the recorded sound and the size of the window. We denote the vector produced from window i by X_i. Then a mean vector x is calculated by

$$\bar{x} = \frac{(X_1, X_2, ..., X_w)}{w} \tag{7.6}$$

Here w is the number of windows and $\bar{x}$ is the final feature vector that will be used as condition description of a case.

Apart from the final feature vector, a case contains information about the robot being diagnosed. Typical information contained in a case is the serial number, model number of the robot and a field that can be manually filled with experts classification. Each case also contains a weight vector of the same dimension as the feature vector. The weight vector is used to adjust and suppress unwanted features in the feature vector in the matching process (explained in the next section). A typical case data structure is displayed in Fig 7.9. The data structure can be extended to contain more information if wanted. Other useful information could be graphs of the sound, the sound itself etc.

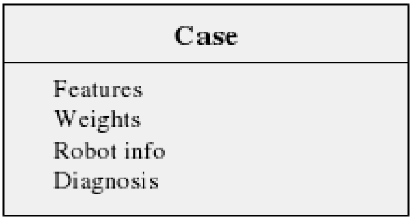

Figure 7.9: Data structure for stored cases in the case library

7.6 Fault Classification

When a feature vector for a new case is assembled from the robot sound, it is com- pared with known cases that were previously stored in the case library. The compari- son is called matching and is based on a nearest neighbour algorithm.

The matching algorithm calculates the Euclidean distance between the case that is to be classified and those cases previously stored in the case library. The distance function uses the feature vectors along with a set of weights defined on the features. Such weights c_j are incorporated into the distance calculation, as indicated in 11.6, to reflect different importance of different features.

$$\sum_{j=d}^{d} |a_j - b_j| * c_j, a, b, c \in \Re^d \tag{7.7}$$

The classification of robot sound is based on the above matching function. The result of matching yields a scored list of the most similar

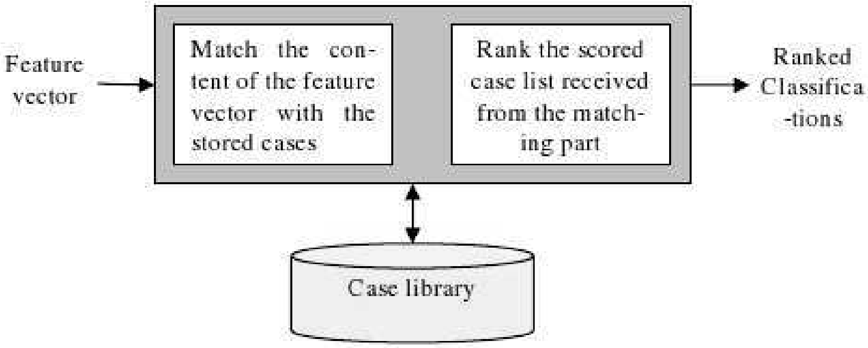

Figure 7.10: Case-based classification as decision support

cases. This list can be presented to responsible technicians as decision support for their further evaluation. An alternative is to derive a voting score for every class involved in the retrieved list of similar cases and then the final decision is settled upon the class exhibiting the highest voting score [10].

It is worthwhile to mention that the performance of our CBR system is improved each time when a new classified case is injected into the case library. The system can thereafter be tested with sounds from other robots previously classified by experts so as to estimate its accuracy. If the accuracy is estimated to be adequate, this CBR system can then be applied to diagnosing robot faults for practical usage.

7.7 Evaluation

Sounds from 20 robots have been recorded. All recordings were obtained during the end-test of the robots. The end-test involves a separate axis test. In the separate axis test, all axes on the robot were individually tested. Each individual axis was tested twice with and without a payload attached to it. A microphone was mounted close to the axis of the industrial robot that was going to be measured. The robot was set to separate axis tests and the signals from axis 4 has been chosen for analysis.

Ten recordings were performed on robots not equipped with payloads and 10 recordings were performed on robots equipped with payloads. The sound from a robot equipped with a payload differs a bit from that without a payload.

Two types of faults have been recorded, hereafter called Fault #1 and Fault #2. Fault #1 is caused by a notch on the big gear wheel in the gearbox of axis 4. It is characterized by a low frequency impulse sound in the middle of the rotation of the axis. Fault #2 is due to a slack between the gear wheels in the gearbox. This fault can be heard as a few low frequency bumps at the end of each rotation of the robot arm. Two robots with Fault #1 (hereafter called Fault #1a and Fault #1b) and two robots with Fault #2 (hereafter called Fault #2a and Fault #2b) were recorded.

Below, Figs. 7.11, 7.12, 7.13 and 7.14 display the sound signals gathered from robots Fault #1a, #1b, #2a and #2b respectively. The black plots show s the unfiltered original sound profiles and the wavelet filtered sounds are represented by grey plots. The span of the frequency of the filtered sounds is from 384Hz to 512Hz.

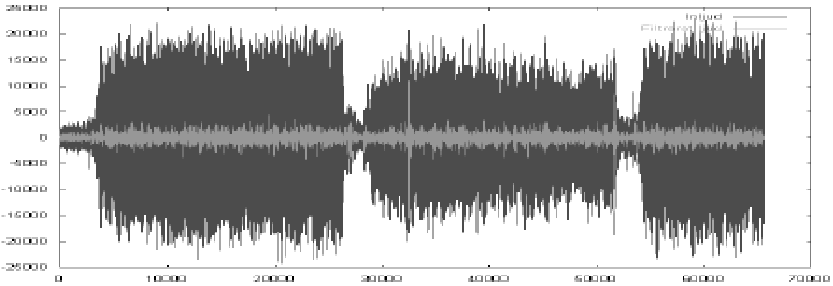

Figure 7.11: Sound signals for robot Fault #1a

All recordings were analysed in the system and transformed to cases and inserted into the case library. Because of dramatic differences between sounds with and without payloads, recordings in both situations were collected and added to the case library. The number of features extracted equals 64.

The cases were first manually analysed. The cases from normal robots were compared to other cases from faulty recordings. The analysis betrays that feature 4 seems a strong attribute for distinguishing abnormality from normal ones. This is obvious to perceive by observing the

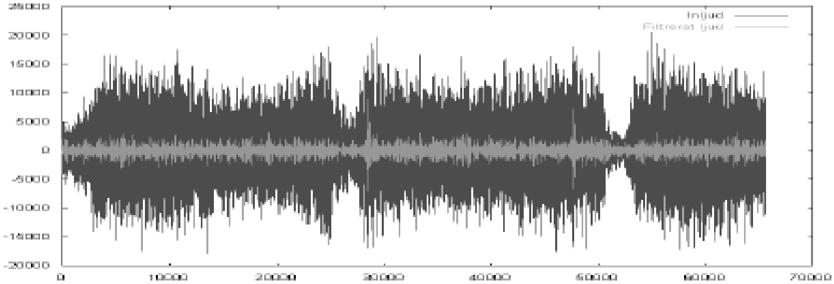

Figure 7.12: Sound signals for robot Fault #1b

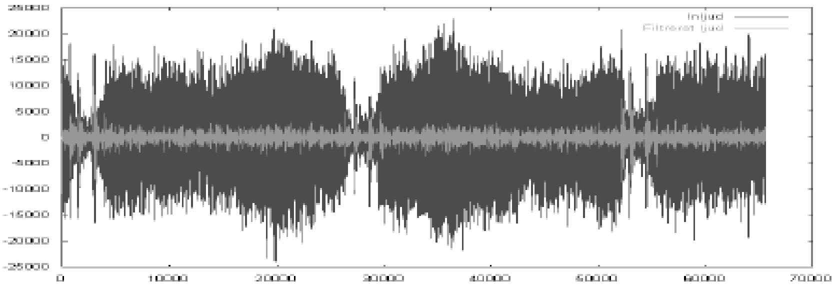

Figure 7.13: Sound signals for robot Fault #2a

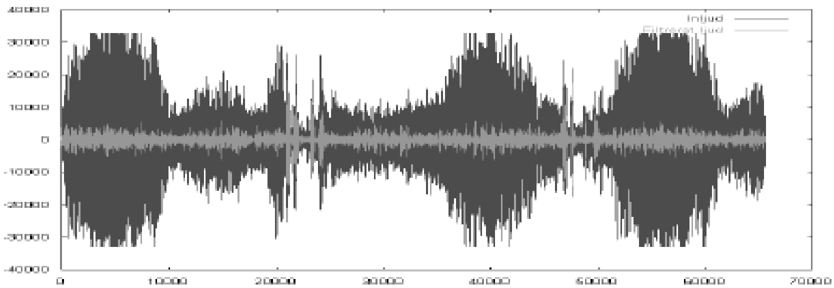

Figure 7.14: Sound signals for robot Fault #2b

following two figures. Fig. 7.15 shows the distribution of feature 4 extracted from the sound signals of the robots not equipped with payloads. Fig. 7.16 shows the distribution of feature 4 for the robots equipped with payloads. The feature in both figures is a normalised absolute value of the dominant wavelet coefficient at a frequency between 384Hz and 512Hz. Likewise we can use the same method to assess the discriminating capabilities of other features.

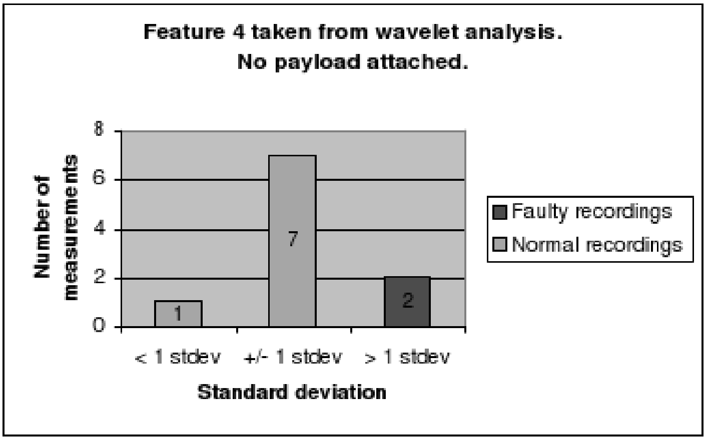

Figure 7.15: Distribution of feature 4 for robots not equipped with a payload

7.8 Example of Case Retrieval

In Fig. 7.11 the sound of the notch can be seen as two repeated prominent peaks in the filtered sound in the middle of the figure. The frequency of the filtered sound spans from 384 HZ to 512 Hz. This figure also indicates three successive rotations of the robot arm. A feature vector with 64 features is assembled from the sound and matched with the previously inserted cases in the case library. Table 11.1 illustrates a ranked list of the most similar cases retrieved. As can be seen form table 1, a previously diagnosed notch fault is regarded to be the closest

to the current recording, thus making the strongest recommendation for diagnosis as Fault #1a. The cases ranked as the second candidate

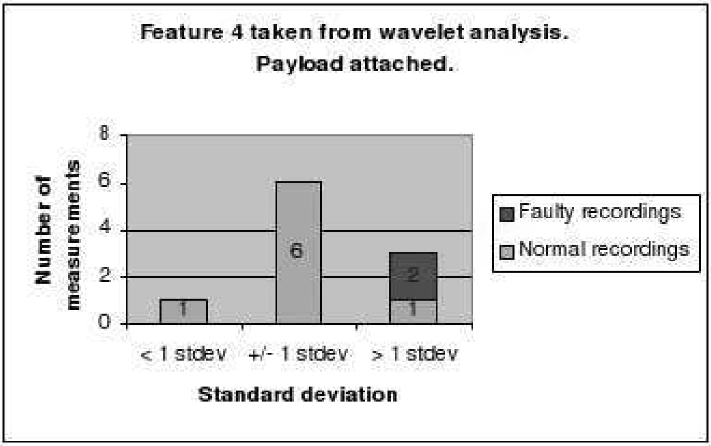

Figure 7.16: Distribution of feature 4 for robots equipped with a payload

Table 7.1: The three most similar cases in the case library

Case name	Similarity	Case ranking
Fault #1a	99.1%	1
Normal case #3	84.1%	2
Normal case #10	83.2%	3

(case #3) and the third candidate (case #10) comes from normal recordings in the case library.

The above matching and classification process involves prior specification of the weights for individual features by means of available background knowledge and/or preliminary analysis of extracted features from pre-diagnosed sound recordings (as what is done in Figs. 7.15 and 7.16). One other method for weighting is to automate the process using machine learning technique [11]. The matching process can also be extended with a neural net classifier.

7.9 How about FFT in This Context

FFT analysis is another common method for feature extraction from signals and it has been shown to be useful in some classification tasks. In this section the performance of FFT is highlighted to explain why it is not employed in our context. An FFT analysis with a Hanning window of length 512 was conducted on the recordings. The FFTspectrum was broken down into 64 features and a feature vector was assembled from the features as described in section 11.5.1. A manual analysis of the FFT-spectrum and of the feature vectors was made in order to find out if any difference between faulty and normal recordings in the frequency spectrum could be spotted. Figs. 7.17 and 7.18 show the results of a standard deviation calculation for feature 4 in the feature vectors.

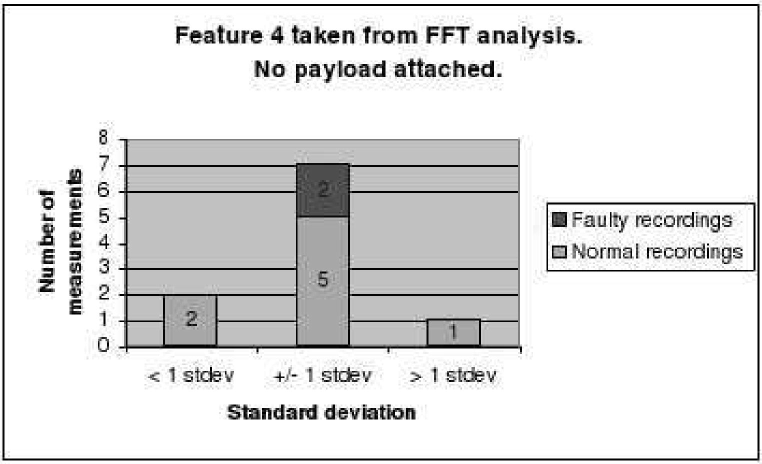

Figure 7.17: Distribution of feature 4 for robots not equipped with a payload

As can be seen in the distributions in Figs. 7.17 and 7.18, feature values from faulty recordings end up amongst those from normal recordings, making it impossible to separate features between faulty and normal signals. This is true when performing such like analysis on any other features. The connotation is that FFT does not offer well distinguishable features for case-based classification in our context. Unlike wavelet analysis, FFT does not clean raw signals and thus is not able to discriminate

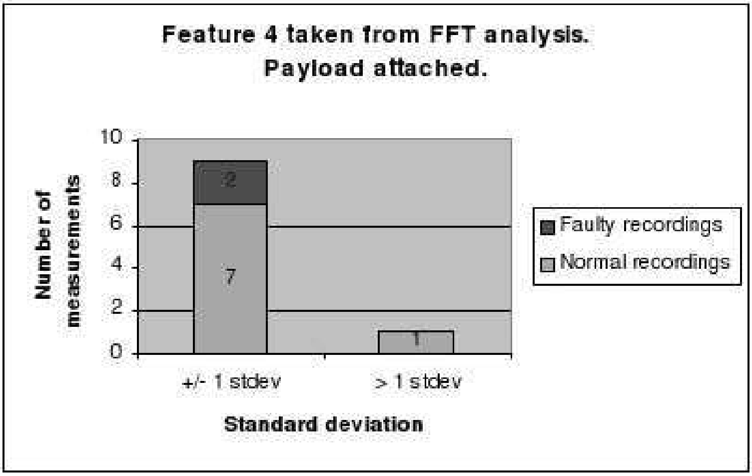

Figure 7.18: Distribution of feature 4 for robots equipped with a payload

different kinds of robot sounds that are overwhelmed by even stronger background noise.

7.10 Conclusions

Case-Based Reasoning is a feasible method to identify faults based on sound re- cordings in robot fault diagnosis. Sound recordings were made under realistic indus- trial conditions. The proposed CBR system has a number of benefits as an industrial diagnostic tool:

- New cases are easy to be added to the library, one sound recording is sufficient.
- The method is easily accepted by technicians and is seen as a tool enabling them to perform better.
- It transfers experience; technicians are able to listen to different sounds and make manual comparisons.
- The system does not need to be complete from the beginning. A list of similar sounds and their classifications can be shown to

technicians as decision support.

- System performance increases continuously. If a new abnormal sound is re- corded but cannot be classified by the system, the technician contributes to the system experience by classifying the sound after the fault has been identified and corrected.

In the validation we have shown that one recording is sufficient for identification of a similar sound in the case library. Also a straightforward feature vector extracted from the original sound recording is sufficient for good results in the matching based on nearest neighbour algorithm. The feature vector and matching process has good potential for improvement. The selected features in the tests are peak wavelet values. Potential users have been interviewed and their reaction to our research prototype tool is very positive and they all consider that it would improve their performance and productivity.

Bibliography

[1] Intelligent noise/vibration measurement and analysis, 2004.

[2] MACSEA Ltd Kevin P. Logan. Prognostic software agents for machinery health monitoring, 2002.

[3] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Com*, 7:39–59, 1994.

[4] S. K. Lee. and White P. R. The enhancement of impulse noise and vibration signals for fault detection in rotating and reciprocating machinery. *Journal of Sound and Vibration*, 217:485–505, 1998.

[5] J. Lin. Feature extraction of machine sound using wavelet and its application in fault diagnosis. *NDTE International*, 34:25–30, 2001.

[6] E. Beckenstein G. Bachman and L. Narici. *Fourier and wavelet analysis*. Springer, 2000.

[7] M. G. Thurston. An open standard for condition-based maintenance systems. *AUTOTESTCON Proceedings*, pages 401–415, 2001.

[8] M. Bengtsson. Standardization issues in condition based maintenance. *Proceedings of the 16th International Congress on Condition Monitoring and Diagnostic Engineering Management*, pages 651–660, 2004.

[9] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 41:906–966, 1988.

[10] E. Olsson, P. Funk, and N. Xiong. Fault diagnosis in industry using sensor readings and case-based reasoning. *Journal of Intelligent and Fuzzy Systems*, 15:41–46, 2004.

[11] A. Stahl. Learning feature weights from case order feedback. *Proceedings of the 4th International Conference on Case-Based Reasoning, Springer*, 2001.

Chapter 8

Paper C: Dynamic Modeling and Sound (Noise) Diagnostics of Robot Gearboxes for Fault Assessments

E. Olsson and R. Stolyarchuk. Dynamic Modeling and Sound (Noise) Diagnostics of Robot Gearboxes for Fault Assessments. Scandinavian Conference on Simulation and Modeling. SIMS, Trondheim, October 2005.

Abstract

Some gear faults in industrial robots can during operation be recognized as abnormal noise peaks coming from the gearbox. A library of such recordings has been assembled in order to automate fault diagnosis of the robots. A computer records sound from the gearbox and compare the new recordings with recordings stored in the library. The result of the comparison is a diagnosis of the condition of the robot. This paper proposes an extension of the sound library by incorporating model based reasoning. A dynamic model of the gearbox in the drive system has been constructed and gear vibrations on the force level are extracted from the model. These vibrations are projected onto the sound recordings with a statistical vibration diagnostic parameter known as the Crest Factor (CF).

8.1 Introduction

A case-based prototype system that makes a diagnosis based on recordings of noise from an industrial robot has previously been implemented [1]. The prototype system analyzes the recordings using Fast Fourier Transform (FFT) [2] for feature extraction and case-based reasoning [3] to make a diagnosis of the condition of the gearbox of the robot.

Gearbox dynamics often have a strong impact on the performance of the system vibrations. In this paper we use the Modelica.Mechanics.Rotational Library [4] to simulate gearbox torques, especially for the output shaft with an applied payload. The simulations were then compared with sound recordings from one normal and two faulty gearboxes.

It is difficult to simulate gearbox effects and to get a reasonable agreement between the measurements and the dynamic simulation. To solve this problem we represented the simulation results and sound recordings by means of the Crest Factor (CF) [5]. CF is defined as the maximum value of a signal normalized by the RMS value. CF aids the comparative study between noise measurements of normal and faulty gearboxes as well as providing a mean to compare these noise measurements against the simulation results measured on the input and output shaft of the model.

The gearbox model is created in Dymola and is characterized by tooth contact stiffness, backlash and efficiency. The model has a correct representation of the relation between force and vibration. Several parameters can be altered in order to produce different simulation results. The results of the simulation represent an oscillation of torque on the input and output shaft of the gearbox model.

In the majority of cases the regular vibrations and noise effects in the gear sets have been predicted theoretically as well as experimentally by measurements. The theoretical description of the gear noise phenomena has been based mainly on force analysis of multibody models undergoing non-linear tooth and bearing contact conditions, inertia masses and the influences of the applied excitation torques. Those directions are highly complicated and involve problem identification, a mathematical formulation and numerical methods. To reduce the gearbox simulation problem to its simplest form we can use modern software e.g. the Dymola tools and Modelica Mechanical Library [4, 6].

8.2 Sources of Gear Noise

An ideal gearbox with rigid equally spaced gears, accurate teeth and good lubrication would transmit minimal noise and vibrations. All kinds of deviations from this ideal gearbox cause an increase in vibrations and noise. In the majority of cases the source of noise and vibrations is transmission errors introduced during manufacture. These errors can e.g. be geometry inaccuracies and eccentricities which both result in impact noises [7]. Other sources of impact noises can be gear rattle. Gear rattle is caused by a combination of backlash and unloaded gears. Friction and pitting due to gear fatigue is also a source of noise [8].

Most modern techniques for gear diagnostics are based on the vibration signal picked up by an accelerometer from the gearbox casing. The vibration signal is normally filtered by time synchronous averaging (TSA) and analyzed in the frequency domain with methods such as the Wavelet Transform (WT) or the FFT. A similar approach is to use a microphone instead of an accelerometer and record the noise from the gearbox. This method was used to detect faults in industrial robots [1] and further work on noise recordings is also presented in this paper.

The expected noise spectrum from a gearbox should contain the gear meshing frequencies and integer multiples of it. There is also common with harmonics and sidebands due to gear eccentricities and geometric errors. Figure 8.1 shows an FFT of the recorded noise of an industrial robot.

The recordings are first pre-processed in order to remove unwanted noise. In this case the recordings are filtered with a low pass filter that removes all noise above 200 Hz. The result contains the most important meshing frequencies, which localizes the amplitude increments during a rotation period. These amplitude increments arise from the transient force effects introduced by the cracked tooth in the driven gear wheel.

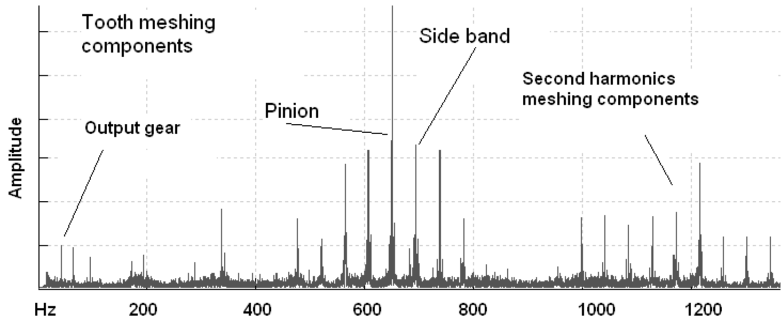

Figure 8.1: FFT spectrum of gear noise analysis.

8.3 Simulation of a Drive Model in Dymola / Modelica

A dynamical model enables a visualization of how a typical design of a multibody system performs with emphasis on our target. This was achieved with the Dymola tools and the Modelica Library [4, 6]. The components in the Modelica Mechanical Rotational package was developed for the fundamental units of a mechanical system e.g. inertia, gear, gear efficiency, friction in bearings, clutches, brakes, external torques, backlash, cut of torque of a flange and others. Every basic mechanical component from the Modelica Library has at least one interface to connect the element rigidly to other mechanical elements. The underlying feature of this library is the component-oriented modeling, which is based on the solution of mixed continuous/discrete systems of equations, or DAE's equations. Figure 8.2 presents a structural model of the gearbox drive train where T_{1-2} stands for input and output torque, f_{1-2} represents the rotational speed of the input and the output shaft and Z_{1-4} represents the number of teeth on each gear.

Figure 8.3 presents a composition diagram of a sample system build in the Dymola environment with icons from Modelica. It is a composite model, which specifies the topology of the system to be modeled in terms of components and connections between the components.

The following setup parameters and assumptions are applied to the model simulation: $I_1 = 0,6kgm^2$ is a motor inertia (pos.3 on diagram) that is driven by a sinewave motor torque T_1 (pos.1 and 2 on fig.3). The torque sinusoidal signal is provided by the values: torque amplitude $T_a = 12nm$ and simulation (case-study) frequency $fr = 0,4; 0,5Hz$ (rps). These frequencies are obtained from a real time rotation period of a robot arm. The rotation period of the robot arm is $\tau = 2-2,2sec$. Via a gearbox (pos.4) the rotational energy is transmitted to a load (inertia) I_2 (pos.5). For simulation purpose we used the following variable cases of the inertia of the load: $I_2 = 20; 40; 50kgm^2$.

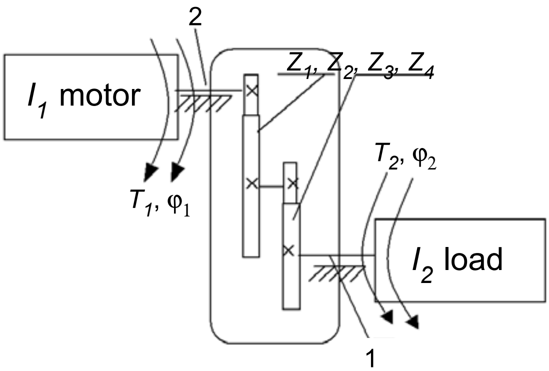

Figure 8.2: Dynamical model of the gearbox drive.

The library gearbox model is specified by the statement $Gear2\_i = 100$ (see Figure 8.3). It is a component assembly model of several components taking essential effects of gear vibration and noise into account. This leads to different faults between gears teeth. In particular, component *lossyGear* defines gear efficiency due to friction between teeth and bearing friction and component *elastoBacklash* defines gear elasticity, damping and backlash.

For simulation purposes we tried to adjust the parameters of the simulated gearbox as close to the parameters of the actual gearbox as possible. The parameters set were:

- Transmission ratio
- Bearing friction

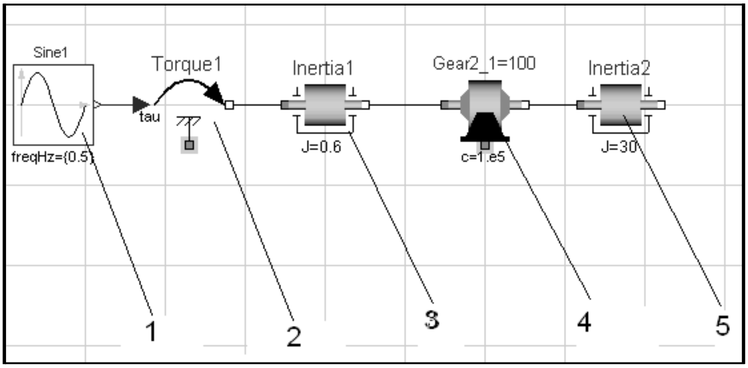

Figure 8.3: Composition diagram of the gearbox drive in Dymola/Modelica.

- Gear elasticity
- Total backlash

We simulated the model with three different payloads. Each simulation was run for 30 seconds. One simulation case is shown in Figure 8.4. Figure 8.4 presents a plot of the behavior of the internal torques on the driven shaft by the variable Inertia2 *flange_tau*. We then applied the CF formula on all obtained data. All calculation results are prepared in table 1.

8.4 Noise Experimental Setup

The gearbox of an industrial robot was used to perform the testing. The robot was mounted in a test cage and a microphone was attached to the gear housing of the axis.

The tested gearbox consists of a common drive train. The drive train has two helical gears driven by a pinion gear that is mounted on the shaft of an electrical motor. The output gear is directly mounted on the robot axis.

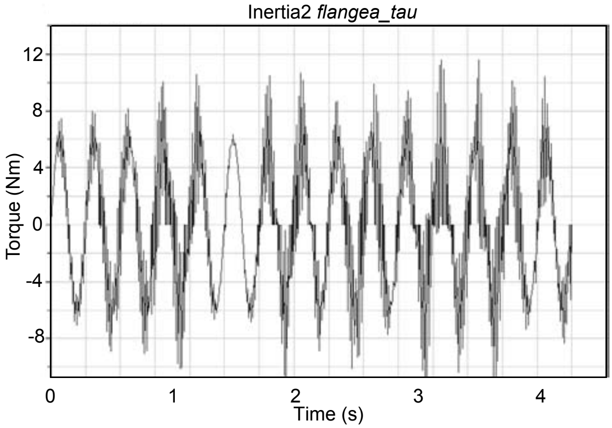

Figure 8.4: Torque on driven gearbox shaft. Time history. Fr=0,5 Hz; J2=30 kgm2.

The gear ratio of the gearbox is 100. It means that one revolution on the output gear corresponds to 100 revolutions on the pinion gear.

The tested gearbox is protected by a housing on which a microphone was attached. The location of the microphone was selected in order to get it as close to the gear drive train as possible. A magnet was used to attach the microphone. The microphone is of a common capacitor type and was connected to the sound card of a computer. The sampling frequency was 8 kHz.

8.5 Recording of Noise

The axis was run back and forward with a driver pinion speed of 270 rad/s during the recordings resulting and in an output (driven) shaft speed of about 2.7 rad/s. The recorded unfiltered sound is shown in Figure 8.5.

The figure shows three periods of rotations of the output axis. The rotational speed of the output axis is 2.7 rad/s. Two types of faults were observed and recorded with the procedure described above. The sources

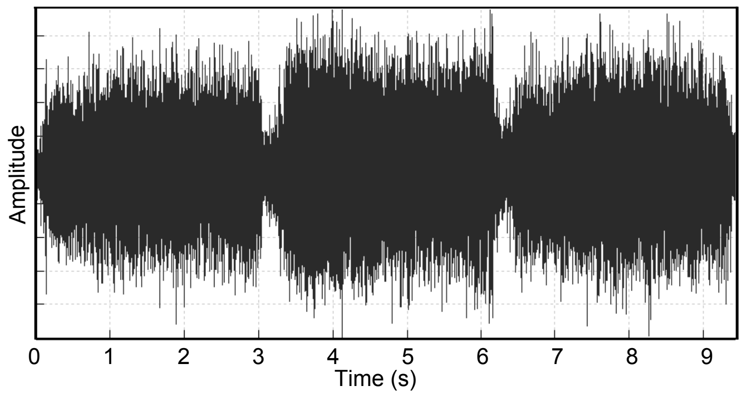

Figure 8.5: Unfiltered noise.

of the faults were:

1. A notch on the output gear
2. Play between the transfer gear and the output gear

The noise signal from the gearbox needs to be pre-processed in order to extract information about the condition of a specific gear wheel. As can be seen in fig. 1 the meshing frequency of the output gear is below 200 Hz and thus all frequencies above 200 Hz was removed by a low pass filter leaving only frequencies from 0-200 Hz in order to reveal the impulse peaks from the noisy sound recordings. A filtered recording of a fault caused by a notch on the driven gear is shown in Figure 8.6.

The peaks at time 3.9 and 6.3 seconds in Figure 8.6 is the result of a small notch on the output gear. The notch is only visible in one direction of rotation and thus leaves the two surrounding periods uninfluenced. The notch is repeated every full rotation of the gear with the same frequency as the rotation speed of the gear.

In Figure 8.7 there are peaks visible in the end and the beginning of each rotation of the gear. These peaks are the results of play between the transfer gear and the output gear. At the end of each rotation the force between the transfer gear and the output gear is radically increased

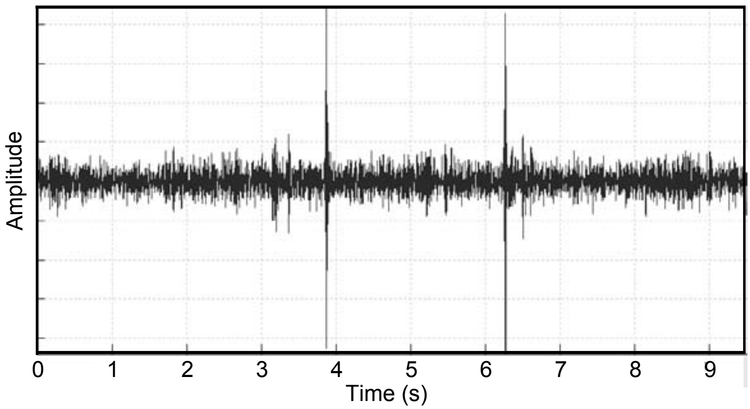

Figure 8.6: Filtered sound with notch fault.

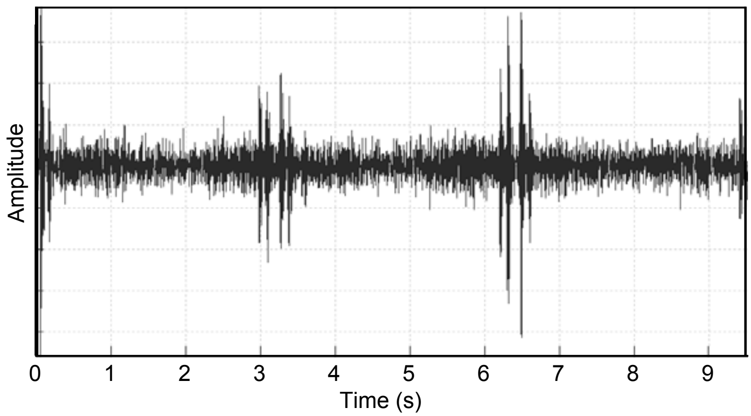

Figure 8.7: Filtered noise with play fault.

causing a backlash with a resulting impulse noise.

8.6 Crest Factor and Results Comparison

In order to make a comparison between the previously explained simulation results and the obtained sound recordings CF was introduced and calculated for each recorded fault and for each simulated fault. CF is based on the Root Mean Square (RMS) value of a signal. RMS is a simple measurement of the fluctuating effect of the signal. RMS is defined to be the square root of the average of the sum of squares:

$$RMS = \sqrt{\frac{1}{N}\left[\sum_{i=1}^{N}(S_i)^2\right]} \tag{8.1}$$

CF is calculated by dividing the peak value of a signal with the RMS of the signal (see 8.2). CF is based on the simple assumption that a signal with a few high amplitude peaks would produce a greater CF than a smooth signal. CF is a normalized parameter suitable for comparison between different measurements results.

$$CF = \frac{S_{max}}{RMS} \tag{8.2}$$

The results of the calculations of CF for the filtered recording of the gear notch fault and the gear play fault are shown in Figure 8.8 and in Figure 8.9 respectively.

The CF produces prominent peaks at each notch. The energy of the peaks is about seven times the average value of the CF.

The CF produces prominent peaks at each change of rotation of the axis. The energy of the most prominent peaks is more than four times the average value of the CF. Results from calculations of the CF parameter can be seen in table 8.1.

The CF was calculated on two types of data:

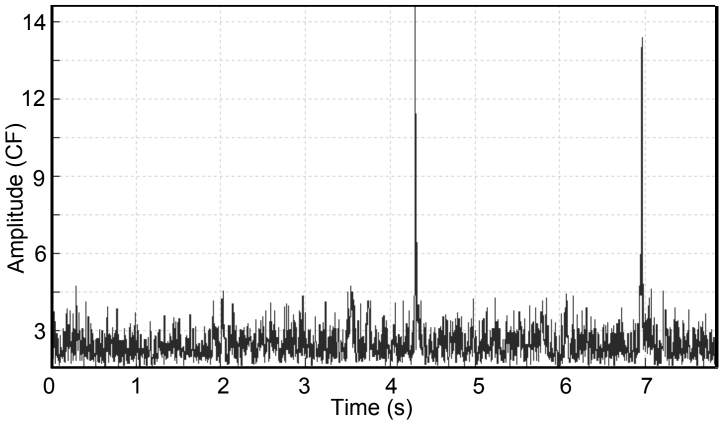

Figure 8.8: CF on notch fault.

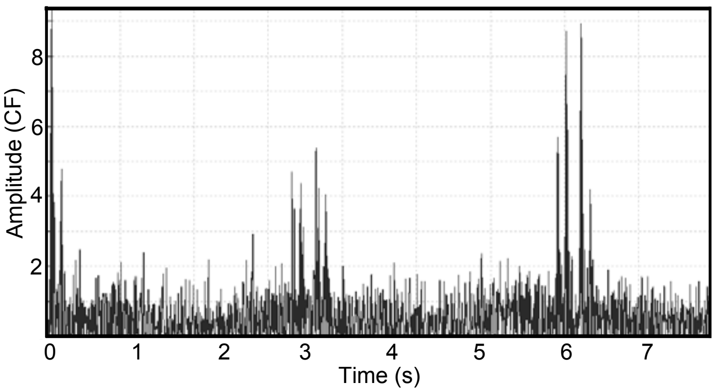

Figure 8.9: CF on play fault.

Table 8.1: CF parameter value from simulation and noise recordings.

Test type	Variable parameters	Mean CF	Peak CF
Torque Sim.	Applied payload 10kg	1.16	1.18
Simulation	Applied payload 125kg	1.14	1.15
Simulation	Applied payload 200kg	1.15	1.17
	Faults type		
Filtered Noise	Gearbox in normal cond.	2.51	3.43
Filtered Noise	Gearbox with play fault	2.94	9.35
Filtered Noise	Gearbox with notch fault	2.92	14.6

1. On low pass filtered noise signals. Recorded from the gearbox of an industrial robot

2. On the simulated torque from the input and output shaft of a dynamical model of the gearbox.

8.7 Conclusions

CF is able to make a normalized parameter from a low pass filtered noise spectrum that can be useful for fault monitoring of the gearbox. The CF increased with more than 200% on sound recordings in faulty case gearboxes compared to recordings of normal gearboxes.

The simulation results are available for engineering design. They can predict the tendency of faults development during the operating period while the design is subjected to varying parameters such as inertia, external torque and frequency/speed. Normal gearboxes with different payload setups were simulated in the component model. They resulted in a low and stable CF. Those results are closer to the calculations of CF from the recording of the normal gearbox than to the CF of the noise recordings of the faulty gearboxes. The CF obtained from the simulation and the experimental noise spectrum from the normal case gearbox is correlated.

Other useful results from this work consist in the following: for a comparative study of the dynamical behavior and vibration effects in gearboxes the statistical methods and factors are reasonable for faults detection.

Bibliography

[1] E. Olsson, P. Funk, and N. Xiong. Fault diagnosis in industry using sensor readings and case-based reasoning. *Journal of Intelligent and Fuzzy Systems*, 15:41–46, 2004.

[2] E. Beckenstein G. Bachman and L. Narici. *Fourier and wavelet analysis*. Springer, 2000.

[3] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Com*, 7:39–59, 1994.

[4] MODELICA HOMEPAGE. Modelica. http://www.Modelica.org.

[5] H. J. Decker. Crack detection for aerospace quality spur gears. *NASA/TM*, 2002.

[6] Dynasim AB. Dymola. http://www.dymasin.se.

[7] A. Barber. *Handbook of Noise and Vibration Control. 6th Edition.* Elsevier Advanced Technology, 1992.

[8] P. J. Dempsey. A comparison of vibration and oil debris gear damage detection methods applied to pitting damage. *Proceedings of COMADEM*, 2000.

Chapter 9

Paper D: Identifying Discriminating Features in Time Series Data for Diagnosis of Industrial Machines

E. Olsson. Identifying Discriminating Features in Time Series Data for Diagnosis of Industrial Machines. The 24th Annual Workshop of the Swedish Artificial Intelligence Society SAIS, BorÅs, May 2007.

Abstract

Reducing the inherent high dimensionality in time series data is a desirable goal. Algorithms used for classification can easily be misguided if presented with data of to high dimension. E.g. the k-nearest neighbor algorithm which is often used for classification performs best on smaller dimensions with less than 20 attributes. In this paper we address the problem 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. The results were promising and we managed to correctly classify measurements from healthy and unhealthy gearboxes.

9.1 Introduction

Selecting adequate features for classification of time series data can be a time-consuming task that requires good domain knowledge and a tedious manual inspection of the data. Even if adequate domain knowledge is present it may not always be directly applicable due to a noisy sensor environment. Using the original high dimensional and presumably noisy data for classification may cause the "curse of the high dimensionality problem" [1] and result in a misguided matching process due to unwanted computation of similarities between irrelevant features. Individual weighting of important features [2] may be a solution to this problem but it often requires expert knowledge about the relevance of each feature and its impact in the matching process. In this paper we present an unsupervised feature selection algorithm which requires no expert knowledge and no individual weighting of features. It uses a time series case base and a feature discrimination approach incorporating an unsupervised function based on statistical feature discrimination finding features with maximum discriminating power. Feature vectors for time series data measurements is assembled from these features. For evaluation, previously classified current measurements from an electrical motor driving a gearbox on an industrial robot were used. The results were promising and we managed to correctly classify measurements from healthy and unhealthy gearboxes. The paper is organized as follows; section 9.2 gives some background and related work, section 9.3 and 9.4 presents our solution to the problem, section 9.5 presents an evaluation on real world time series data and section 9.6 concludes the paper with a discussion and a proposal for future work.

9.2 Background and Related Work

9.2.1 Feature Discrimination

Feature discrimination relies on the fact that certain features in time series data have a stronger discriminating power than others. By letting the features with the strongest discriminating power represent the time series we have hopefully achieved a great reduction in dimension and a more qualitative knowledge representation of the data. The reduced representation will thereby stand a better chance to perform well in applications for classification of time series data. E.g. the k-nearest neighbor algorithm which is often used for case-based classification [3] performs best on smaller dimensions with less than 20 attributes. Feature discrimination usually relies on on a criterion function and a search strategy. The search strategy is used to select features and the criterion function is used to evaluate whether a selected feature is better than another. Bayesian probability estimation has been successfully used for criterion [4] and key sequences in synthesized data was found with great accuracy. In [5] several approaches of feature discrimination is discussed. Also the use of a neural network for simultaneous clustering and feature discrimination has been proven useful [6].

9.3 Computing Feature Vectors for Time-Series Data

We address the dimensionality problem using a time series case base CB containing cases with previously classified time series measurements. Each measurement is first transformed into a time/frequency representation of the original time series data by computing a time FFT [7] transformation. A feature extraction function $FV = GetFeatures(CB_k, \delta)$ is then applied on each time/frequency representation that for a given case CB_k returns a subset of time/frequency elements in feature vector FV representing CB_k in a reduced dimension form. δ is a threshold value defining the criterion for discriminating power of CB_k. δ is found by search and criterion function $\delta = GlobalMaximun(N(CB_k))$. Function $N(CB_k)$ returns the number of fully discriminated cases by CB_k with respect to CB and function $GlobalMaximun(N(CB_k))$ returns the value of δ where $N(CB_k)$ has its global maximum thus representing a maximum of discriminating power of CB_k with respect to CB.

9.3.1 Extracting Discriminating Features for Case Indexing

Definitions

Definition 1. *We define a time series X of dimension n as a sequence of data points x ordered in time as $X = \{x_1, x_2, ..., x_i, ..., x_n\}$ where x_i refers to a data point at position i.*

Definition 2. *We define* $Transform$ *to be a function* $f : \Re \rightarrow \Re^2$ *mapping time series* X *to a time/frequency matrix* A

$$A = \begin{pmatrix} a_{11} & a_{12} \ldots & a_{1j} \ldots & a_{1n} \\ a_{21} & a_{22} \ldots & a_{2j} \ldots & a_{2n} \\ \vdots & \ddots & \vdots & \ddots \vdots \\ a_{i1} & a_{i2} \ldots & a_{ij} \ldots & a_{in} \\ \vdots & \ddots & \vdots & \ddots \vdots \\ a_{m1} & a_{m2} \ldots & a_{mj} \ldots & a_{mn} \end{pmatrix} \tag{9.1}$$

In A *each element* a_{ij} *represents a discrete time/frequency element with time* j *and frequency* i.

Definition 3. *A time series case base* CB *contains a number of cases where each case is represented by a vector of triplets* $\{X, A, C\}$ *where* X *is the original time series measurement,* A *is the time/frequency representation of the original time series data* X *and* C *represents its class.* CB_k *represents a case* k *in* CB *and* $CB_{k_{a_{ij}}}$ *represents a time/frequency element in case* k.

Definition 4. *We let the function* $std(a, b)$ *denote the standard deviation function returning the standard deviation of time/frequency elements* a *and* b *and we define the function* $Threshold(a, b)$ *to be:*

$$Threshold(a, b) = \begin{cases} 1 & \text{if } std(a, b) < \delta \\ 0 & \text{otherwise} \end{cases} \tag{9.2}$$

Definition 5. *We define case* CB_k *to be fully discriminated by case* CB_l *(and the opposite) if there exists a* $\delta > 0$ *such as*

$$\left[\sum_{i=1}^{m} \sum_{j=1}^{n} Threshold(CB_{k_{a_{ij}}}, CB_{l_{b_{ij}}}) \right] = 0, k \neq l \tag{9.3}$$

Definition 6. *We let the function* $Discriminate(CB_k, CB_l)$ *to be defined as*

$$Discriminate(CB_k, CB_l) = \begin{cases} 1 & \text{if } CB_k \text{ is fully discriminated by } CB_l \\ 0 & \text{otherwise} \end{cases} \tag{9.4}$$

Definition 7. *We define a measurement of discriminating power of a case CB_k to be the sum of all the cases in the case base it fully discriminates*

$$N(CB_k) = \sum_{l=1}^{n} Discriminate(CB_k, CB_l), k \neq l \tag{9.5}$$

where $N(CB_k)$ denotes the number of fully discriminated cases by CB_k with respect to CB.

We now want to extract the time/frequency elements from CB_k that represents the strongest discriminating power with respect to CB. The first step is to find δ where $N(CB_k)$ has its global maximum. By definition, a global maximum must be either a local maximum in the interior of the domain of $N(CB_k)$ or it must lie on the boundary of its domain [8]. The domain of $N(CB_k)$ is all positive real values $\delta > 0$ but we can limit the domain to $\delta = (c, d)$ where $c = 0, d \geq c, \delta = d \rightarrow N(CB_k) = 0$. We solve this with the search and criterion function $\delta = GlobalMaximun(N(CB_k))$.

```
SET max=0
SET ret=0
FOR δ = c TO d
      IF N(CB_k) > max
          SET max = N(CB_k)
          SET ret = δ
      END
END
RETURN ret
```

Figure 9.1: Code for finding the global maximum of $N(CB_k)$

Definition 8. *If we let the function $GetFeatures(CB_k, \delta)$ be the function returning a set of time/frequency elements from case CB_k that for each case in CB satisfies (9.6).*

$$GetFeatures(CB_k, \delta) = \begin{cases} CB_{k_{a_{ij}}} & \textit{if } std(CB_{k_{a_{ij}}}, CB_{l_{b_{ij}}}) > \delta, k \neq l \\ 0 & \textit{otherwise} \end{cases} \tag{9.6}$$

then

$$FV = GetFeatures(CB_k, GlobalMaximun(N(CB_k))) \tag{9.7}$$

will produce a feature vector $FV = \{F_1, F_2, ..., F_i, ..., F_m\}$ *representing all time/frequency elements with the discriminating power* δ*.*

9.4 Case Indexing

In order to use FV for case indexing we want an appropriate representation of the features. We use a naive binary structure [4] in combination with sequence appearance numbers. In this case we reduce the time dimension and save only frequency information in our vector. We recalculate FV given in (11.6) and represents it as in (9.8)

$$FV = \{b_1 * f_1, b_2 * f_2, ..., b_i * f_i, ..., b_m * f_m\} \tag{9.8}$$

where b_i denotes the number of occurrences of similar frequency elements and f_i denotes the frequency.

For similarity measure between two time series we use the Euclidean distance function defined as

$$sim(FV_1, FV_2) = \sqrt{\sum_{i=1}^{m}(FV_{1i} - FV_{2i})^2} \tag{9.9}$$

9.5 Example Implementation and Evaluation

In order to evaluate our framework we tested it on pre-classified current time series data from an electrical motor driving the gearbox of axis 4 on an industrial robot. A total of 40 classified measurements were used in the evaluation. Our goal was to compute feature vectors that were able to discern a healthy gearbox from an unhealthy gearbox.

9.5.1 Measuring Current Time-Series

The robot control cabinet can log current signals from the electrical motors driving the gearboxes on the robot. We programmed the cabinet to log current signals from the electrical motor driving the gearbox of axis 4. The signal is passing through the robot control system and is an indirect measurement of current [9] derived from the motor torque. Because of its indirect nature, basic theory of feature selection usually applied to current measurements is difficult to apply here. We thereby find this time-series data especially suitable for evaluation of our framework.

Each time-series measurement is approximately 3.8 seconds long and involves a full rotation of the robot arm. The sampling rate is 2 kHz which result in a time-series measurement containing approximately 7600 samples (see Fig. 9.2).

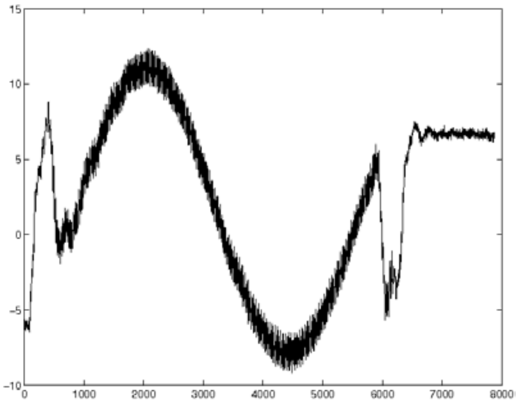

Figure 9.2: Motor Current

9.5.2 Classification of Time-Series

During the end test of the industrial robots we logged current signals from 34 normal gearboxes and 6 noisy gearboxes. All robots were classified by experts. Table 9.1 presents the experts classifications.

Based on the information in table 9.1 we classified our measurements in five classes; C_{Normal}, $C_{RepeatedKnocks}$, C_{Knock}, C_{Burr} and C_{Noise} as for normal, repeated knocks, a single knock, burr and noise respectively.

Table 9.1: Robots classified by human expert.

Class	Number of robots
No symptom (normal)	34
Repeated Knocks	3
Knock	1
Burr	1
Noise	1

Table 9.2: Robots classified by system.

Class	Percentage of correct hits
No symptom (normal)	100
Repeated Knocks	100
Knock	0
Burr	0
Noise	0

Classified cases were created for all measurements and inserted into the case library.

9.5.3 Computing Feature Vectors

Before computing feature vectors for the classified cases we pre-processed the data in each case computing a time FFT [7] matrix A (see Defn. 2) on each time series current measurement X (see Defn. 1). The time FFT was computed with a precision of 46 time-segments and 169 frequency steps for each measurement resulting in an approximate time/frequency resolution of 83 ms/segment and 7 Hz/step.

After computing feature vectors for all cases as explained in section 3 and performing a leave-one-out k-nearest neighbor evaluation [10] with $k = 3$ on all cases as explained in section 4 (see Fig. 9.3). The result is presented in table 9.2.

We managed, with 100 percent accuracy, to correctly classify all cases

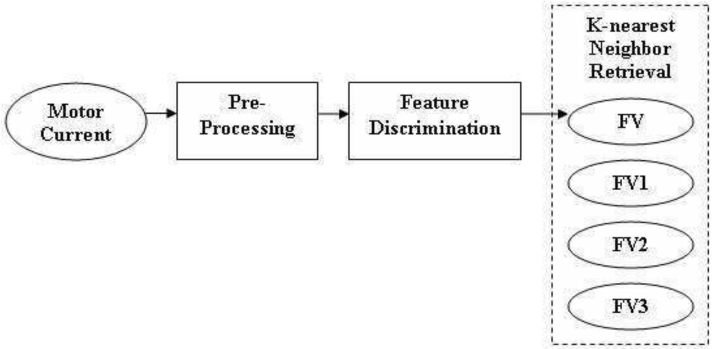

Figure 9.3: System for k-nearest neighbor evaluation

with class C_{Normal} and $C_{RepeatedKnocks}$. We failed to classify C_{Knock}, C_{Burr} and C_{Noise}. A reason for that is given in the next chapter.

9.6 Conclusions and Future Work

Our approach of feature selection by feature discrimination proves to be useful for machine sensor time series. It shows that the method can be valuable on already classified time series which lacks of other useful domain knowledge or where domain knowledge is hard to apply. We managed to compute discriminating feature vectors and correctly classify the two dominating classes C_{Normal} and $C_{RepeatedKnocks}$. We failed to classify C_{Knock}, C_{Burr} and C_{Noise} because we had no similar cases in the case base. The leave-one-out k-nearest neighbor evaluation approach demands several cases of similar class in order to successfully classify all cases. A larger case base with more cases of class C_{Knock}, C_{Burr} and C_{Noise} would probably perform better in classification. Some suggestions for future work is stated below.

1. Expand the case base with more classified cases.

2. Evaluate the performance of the algorithm on other kinds of time

series data.

3. Test other classification algorithms such as Self Organizing Maps (SOM) [11], the cosine matching function [12] etc.

Bibliography

[1] Thomas M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[2] D. Wettschereck D., Aha. Weighting features. *Proceedings of the 1st International Conference on Case-based Reasoning. Springer*, pages 347–358, 1995.

[3] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Com*, 7:39–59, 1994.

[4] Xiong N. Funk P. Knowledge about key sequences for indexing time series cases in medical applications. *Advances in Case-Based Reasoning*, 2006.

[5] A. Liakat M. Feature selection of dna microarray data. Seminar and Computing Tools, 2005.

[6] Soon-Ann L. Hichem F., Nozha B. Unsupervised clustering and feature discrimination with application to image database categorization. *Digital Object Identifier*, pages 401–406, 2001.

[7] E. Beckenstein G. Bachman and L. Narici. *Fourier and wavelet analysis.* Springer, 2000.

[8] Adams R. *Calculus: A Complete Course.* Addison-Wesley, 1995.

[9] K. Wallin. Matdatorer. ABB internal report, 1997.

[10] Belur V. *Nearest Neighbor (NN) Norms: NN Pattern Classification.* Ieee Computer Society, 1991.

[11] T. Kohonen. *Self-organizing maps.* Springer-Verlag, 1995.

[12] Lopez de Mantaras R. Plaza E. A case-based apprentice that learns from fuzzy examples. *Methodologies for Intelligent Systems. Elsevier*, 5:420–427, 1990.

Chapter 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

E. Olsson. Using Cased-Based Reasoning Domain Knowledge to Train a Back Propagation Neural Network in order to Classify Gear Faults in an Industrial Robot. 21st International Congress and Exhibition. Condition Monitoring and Diagnostic Engineering Management. COMADEM 2008. Prague, JUNE, 2008.

Abstract

The classification performance of a back propagation neural network classifier highly depends on its training process. In this paper we use the domain knowledge stored in a Case-based reasoning system in order to train a back propagation neural network to classify gear faults in an industrial robot. Our approach is to compile domain knowledge from a Case-based reasoning system using attributes from previously stored cases. These attributes holds vital information usable in the training process. Our 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 large Case-based reasoning 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.

Key Words: Case-Based Reasoning, Neural Network, Sound recordings, Fault classification

10.1 Introduction

In this paper we use the domain knowledge already stored in a Case-based reasoning (CBR) [1] system in order to train a back propagation neural network (NN) to classify gear faults in an industrial robot. Our approach may be usable when a simple classifier is wanted due to e.g. lack of computing power, ease of use or when only a part of the knowledge stored in the case base of a large CBR system is needed. CBR offers a method to implement intelligent diagnosis systems for real-world applications [2]. Motivated by the doctrine that similar situations lead to similar outcomes, CBR is able to classify sensor signals based on experiences of past categorizations saved as cases in a case-base. This paper is based on a CBR system used to diagnose audible faults in industrial robots [3] as mechanical fault in industrial robots often show their presence through abnormal acoustic signals. The system uses CBR and acoustic signals as a proposed solution of recognizing audible deviations in the sound. 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. The system uses three different steps in its classification process; pre-processing, feature identification and classification. The pre-processing process is responsible for filtering and removal of unwanted noise. In the feature identification process, the system uses a two-pass model, first identifying features and then creating a vector with features. Features are extracted using methods such as FFT and wavelet analysis [4], [5]. A feature in the case is a normalized peak value at a certain frequency and time offset. Once the features are identified, the system classifies the feature vector. The classification is based on previously classified measurements stored as cases in a case base. 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. When a new sound has been classified, the system learns by adding it as a new case to the case-base. At recent time, the system stores classified cases of recordings of gearboxes from 24 healthy and 6 faulty robots. We have used CBR domain knowledge from two of those cases in order to train a NN classifier to classify one type of gear fault. Our approach may be usable when only a small and simple classifier is wanted that might use only a part of the knowledge stored in a CBR system. Further, no use of the usual sensor signal classification steps involving filtering and feature extraction are needed. Once successfully trained, the neural network classifier can be directly applied on noisy sensor data and it will represent the part of the case-base used in its training process and it will respond accordingly e.g. it might act as a red/green light in response to its input, signaling a failed/normal gearbox. The paper is organised as follows: section 10.2 gives a formal description of the CBR system that is used as the source for domain knowledge. Section 10.3 presents the method used to extract this domain knowledge. Section 10.4 describes how to train a simple NN classifier using the extracted domain knowledge. Section 10.5 presents an evaluation of the classification performance of the neural network classifier and section 10.6 gives a brief conclusion of the paper.

10.2 The CBR System

The CBR system consists of the tuple (CB, sim) [6] where CB denotes its case-base and sim is a similarity function that classifies a case by searching for similar cases already processed and stored in the case base. A case base (CB) contains a sequence of n cases. The cases are indexed in a flat hierarchy with $CB = (X_1, X_2, ..., X_i, ..., X_n)$. A case X is a triple $(x, FV, class\,(FV))$ where x is the unprocessed sensor signal, FV is a feature vector containing m features describing the nature of the sound recording and $class\,(FV)$ is the class of X. In our system, each feature F in FV is a triplet (A, t, f) describing a peak in sensor data x with amplitude A, frequency f and location offset at time t. Features are extracted from the sensor data by means of various methods such as FFT, wavelet analysis [4][5] etc.

The class of X is determined by the similarity function

$$sim(FV_1, FV_2) = \sqrt{\sum_{i=1}^{m}(FV_{1i} - FV_{2i})^2} \qquad (10.1)$$

measuring the Euclidean distance between two feature vectors FV_1 and FV_2. The k nearest neighbours of X indexed by FV_1 are retrieved and the class of X is determined by the class of these neighbours.

10.3 Extracting Domain Knowledge

The domain knowledge for $class_i$ can be seen as the information stored in the cases contained in the cluster CB_i formed by all cases X_i having feature vector FV where $class\,(FV) = i$ and consequently, domain knowledge for $class_j$ can be seen as the information stored in the cases X_j contained in the cluster CB_j formed by all cases X_j having feature vector FV where $class\,(FV) = j$ etc. Our approach is to use cluster CB_i to train a stand-alone NN classifier to classify sensor data of $class_i$ and consequently, a cluster CB_j can be used to train the same stand-alone NN classifier to classify sensor data of $class_j$ etc. In this manner, explicit domain knowledge from a case base can be transferred and transformed into implicit domain knowledge inside a NN classifier.

We have used CBR domain knowledge in the form of time offsets from two cases X_i and X_j in order to train a two-layer back propagation neural network [7]. The cases contain sound recordings from a normal ($class_i$) and a broken ($class_j$) gearbox originating from two industrial robots. The output gear in the broken gearbox had a broken gear tooth generating impact sounds [8] whereas the normal gearbox did not generate any impact or any other abnormal sound whatsoever. Both sound recordings were contaminated with noise originating from the gearboxes themselves and from the noisy factory environment the robots was situated in. Case X_i and X_j are described as follows

$$X_i = \left(x_i, FV\,(NULL)\,, class\,(FV) = i\right) \tag{10.2}$$

$$X_j = \left(x_j, FV\,\left(\left(\left[A\,(0.5)\,, t\,(3.1)\,, f\,(50)\right], \left[A\,(0.4)\,, t\,(5.1)\,, f\,(50)\right]\right)\right), class\,(FV) = j\right) \tag{10.3}$$

Where $caseX_i$ contains no description of an impact or any other abnormal sound whatsoever and case $caseX_j$ reveals two impulse sound peaks; $peak_1$ and $peak_2$, caused by a broken gear tooth on the output gear. $peak_1$ is located at time offset $t\,(3.1)$ with an amplitude of $A\,(0.5)$ and a frequency of $f\,(50)$. $peak_2$ is located at time offset $t\,(5.1)$ with an amplitude of $A\,(0.4)$ and a frequency of $f\,(50)$. Figure 10.1 depicts the unprocessed sound signal as it is stored in case and Figure 10.2 shows the same sound recording as depicted in figure 1 but filtered at frequency in order to reveal impulse sound $peak_1$ and $peak_2$ caused by the broken gear tooth on the output gear.

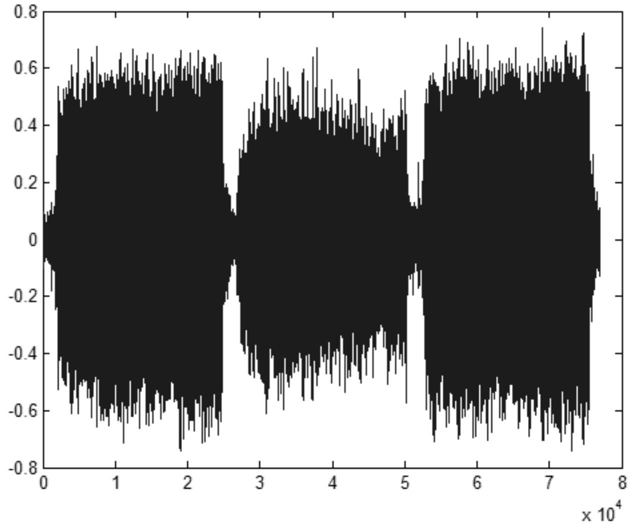

Figure 10.1: Unfiltered sound recording from the gearbox

10.4 Training a Neural Network Classifier

A two-layer [7] NN classifier consisting of layers:
$[input = 12hidden = 5output = 1]$, with a tan-sigmoid transfer function [9] in the hidden layer and a linear transfer function in the output layer was created using Matlab [10]. We form cluster CB_k by extracting training examples from sensor data x_k by using time offset t from feature triplets (A, t, f) in X_k. Cluster CB_j is formed by picking 400 training examples representing $peak_1$ from sensor signal x_j at location time offset $t\,(3.1)$ and by picking 400 additional training examples representing $peak_2$ from sensor signal x_j at location time offset $t\,(5.1)$. Training examples are extracted from unprocessed sound data. Equally we form cluster CB_i by extracting training examples from unprocessed sound data x_i. As X_i contains no feature triplets, cluster CB_i is formed by

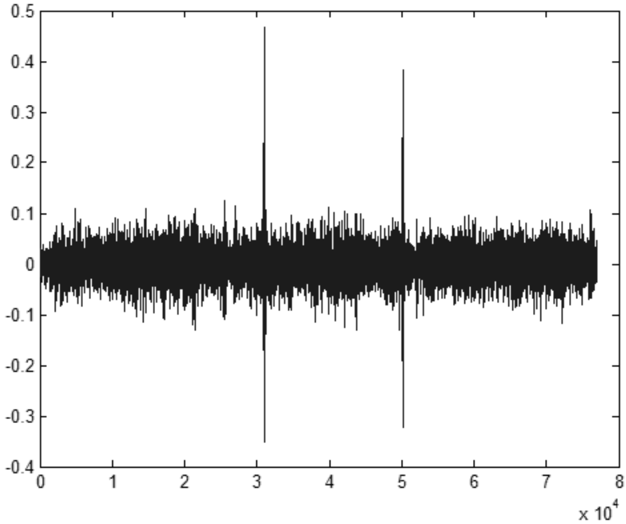

Figure 10.2: Impulse sound peaks hidden in the noisy sound recording

picking 4000 training examples from randomly chosen time offsets from the sound signal x_i representing the sound from a healthy gearbox. We used a sliding window approach [11] when picking training examples. The window was of length 12 relating to the number of input neurons in the network and it was shifted one sample to the right each time a new sequel training example was to be obtained from a time offset. We then equally distributed cluster CB_i into sub clusters CB_{it}, CB_{iv} and CB_{ie} where t, v, e stands for training, validation and evaluation consequently. In the same manner, we make sub clusters CB_{jt}, CB_{jv} and CB_{je} from CB_j. The network was trained using supervised training and it was trained to output 1 when exposed to examples from cluster CB_j (sound data containing impact sounds) and 0 when exposed to examples from cluster CB_i (sound data not containing any impact sounds or any other abnormal sounds whatsoever).

10.5 Evaluation

The NN classifier was trained using clusters as described in 10.4 and then evaluated. In the evaluation, we focused on the ability of the NN classifier to separate between faulty and normal sound recordings. As a result of the training, the NN classifier should ideally output values closer to 1 when exposed to impact sounds and otherwise values closer to 0. However, the NN classifier are not likely to have quantized outputs. So, a simple post-processing algorithm depicted in figure 10.3 was applied.

```
FOR all values y in NN output:
    y=round(y)
END
```

Figure 10.3: Post-processing algorithm

We evaluated the classification performance of the NN classifier by exposing it to unprocessed sound recordings from similar gearboxes recorded during similar conditions. Sound recordings from 6 gearboxes were used for evaluation. 2 sound recordings were obtained from gearboxes containing similar faults as described in section 10.2 and 4 sound recordings were obtained from normal gearboxes containing no prominent impact sounds whatsoever. The NN classifier managed to achieve a correct classification score of 100

10.6 Conclusions

We have shown that our 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. Our approach may be usable when a simple classifier is wanted due to e.g. lack of computing power, ease of use or when only a part of the knowledge stored in the case base of a large Case-based reasoning system is needed. Further, no use of the usual sensor signal classification steps involving filtering, feature extraction and classification are needed once the neural network classifier is successfully trained. We find there is no reason not to believe that our approach would be successful for similar classification tasks.

Bibliography

[1] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Com*, 7:39–59, 1994.

[2] M. Nilsson, P. Funk, and M. Sollenborn. Complex measurement in medical applications using a case-based approach. *Workshop Proceedings of the International Conference on Case-Based Reasoning*, pages 63–73, 2003.

[3] E. Olsson, P. Funk, and M. Bengtsson. Case-based diagnosis of industrial robots based on acoustic signals. *Proceedings of the European Conference on Case-Based Reasoning*, pages 686–701, 2004.

[4] S. K. Lee. and White P. R. The enhancement of impulse noise and vibration signals for fault detection in rotating and reciprocating machinery. *Journal of Sound and Vibration*, 217:485–505, 1998.

[5] J. Lin. Feature extraction of machine sound using wavelet and its application in fault diagnosis. *NDTE International*, 34:25–30, 2001.

[6] P. Perner. Case-based reasoning and the statistical challenges. *Seventh Annual ENBIS Conference, Dortmund*, 2007.

[7] M. Bishop, C. *Neural Networks for Pattern Recognition.* Oxford : Clarendon : Oxford Univ. Press, 1995.

[8] A. Barber. *Handbook of Noise and Vibration Control. 6th Edition.* Elsevier Advanced Technology, 1992.

[9] S. ussel and P. Norvig. *Artificial Intelligence A Modern Approach Second Edidtion.* Prentice Hall International, 2003.

[10] Mathworks. Matlab.

[11] F.M et al Dias. A sliding window solution for the on-line implementation of the levenberg-marquardt algorithm. *Engineering Applications of Artificial Intelligence, Pergamon*, 2006.

Chapter 11

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

E. Olsson, P. Funk. Agent-Based Monitoring using Case-Based Reasoning for Experience Reuse and Improved Quality. Journal of Quality in Maintenance Engineering (ISSN 1355-2511), Vol.15, No.2, pages 179–192, 2009.

Abstract

Purpose The purpose with this paper is to propose an agent-based condition monitoring system for use in industrial applications. An intelligent maintenance agent is described that is able to autonomously perform necessary actions and/or aid a human in the decision making process. An example is presented as a case-study from manufacturing of industrial robots.

Design/methodology/approach The paper is mainly based on a case-study performed at a large multi-national company aiming to explore the usefulness of case-based experience reuse in production.

Findings This paper presents a concept of case-based experience reuse in production. A maintenance agent using a Case-Based Reasoning approach to collect, preserve and reuse available experience in the form of sound recordings exemplifies this concept. Sound from normal and faulty robot gearboxes are recorded during the production end test and stored in a case library together with their diagnosis results. Given an unclassified sound signal, relevant cases are retrieved to aid a human in the decision making process. The maintenance agent demonstrated good performance by making right judgments in 91% of all the tests, which is better than an inexperienced technician.

Originality/value The main focus of this paper is to show how to perform efficient experience reuse in modern production industry to improve quality of products. Two approaches are used: 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

Paper type Research paper

Key Words: Experience Reuse, Decision Support Systems, Condition Monitoring, Intelligent Agents, Case-Based Reasoning, Quality Improvement

11.1 Practical implications

Experienced staffs acquire their experience during many years of practice and sometimes also through expensive mistakes. The acquired experience is difficult to preserve and transfer and it often gets lost if the corresponding personnel leave their job due to retirements etc. The proposed Case-Based Reasoning approach to collect, preserve and reuse the available experience enables a large potential for time and cost savings, predictability and reduced risk in the daily work. The paper exemplifies experience reuse for quality improvement in production using a number of methods and techniques from Artificial Intelligence.

11.2 Introduction

Society has moved from the industrial age to the knowledge age, where information and experience are one of the most valuable resources. This is also becoming increasingly obvious in production industry, how well a production company reuse experience and integrate new knowledge and research results into their production may both be one of the key factors for success as well as a major long term survival factor. Research is rapidly moving forward in the area of condition monitoring and diagnostics [1] but production companies not following progress in research and integrating research results into their production when results are ripe for commercial deployments take a serious risk. A large part of a companys knowledge and experience is also produced by human mistakes and often costly incidents. When engineers need to make a decision to prevent or correct, they often encounter the information overload problem [2]. Human information overload is also a serious issue, especially when quality of information is difficult to judge, faulty, or missing. Although valuable experience in the same or similar situation may be available from a phone call away, such information is rarely found when needed. Bengtsson et al. [3] states that experience reuse, in even the simplest form of e.g. checklists, and subjective monitoring is helpful. It is increasingly difficult to meet customers demands, requiring increased production quality, flexibility, reliability and fast delivery times is a major challenge for industry. These requirements make product development and manufacturing increasingly complex.

In this research we explore how methods and techniques from artificial intelligence can be used to enable systematize in production industry. Large companies have reduced their technical support costs with up to 33% by deploying methods and techniques from Artificial Intelligence [4].
In this study we focus on Agents based technology and Case-Based reasoning. Intelligent agents offers a concept and framework enabling flexible interactions with agents, systems and humans. An agent may also have knowledge on what the engineers currently are working on and what knowledge they have in order to offer better support and identify relevant knowledge and experience e.g. past solved cases, documentations, PMs, etc.

Case-Based Reasoning is an efficient method to identify past experience in the form of cases that may help to solve the current problem. Intelligent agents deploying case-based reasoning 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. 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. Case-Based Reasoning solutions may also reduce costs for technical support since technicians often need considerably less time the second time they encounter a similar problem. But even access to the a case containing the solution to a similar problem saves often considerable time according to the technicians.

Using intelligent agents for monitoring is an important path to the next generation of monitoring systems [2]. We suggest that the following reasons are valuable properties in an agent based approach for condition monitoring:

1. Agents enable decentralization of decisions and reduction of complexity.

2. Agents enable localized expertise and experience reuse.

3. Agents enable learning and experience sharing between agents with similar tasks.

4. Agents can be implemented to collaborate with other agents and humans.
5. Agents are able to make informed decisions on what actions to take or when to verify actions or interact with humans.

There are already research prototypes demonstrating the value of agents based monitoring systems in industrial applications, e.g. [5].

11.3 Intelligent Agents

In this section we give a brief introduction to agents and also motivate why the agent concept is suitable for the domain of condition monitoring. There are a number of different definitions of agents, but the difference for practical applications are often less important, the definitions are often based on the fact that agent research often is application oriented and having a particular applications in mind may lead to slightly different definitions [6]. Russel and Norvig [7] state that an agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors. A pure reactive system like a thermostat would meet this basic definition, but most definition also includes social skills (communication) and some intelligent behaviour, e.g. ability to pro-activeness, learning and predicting. An agents goals or desires may be to decrease maintenance costs without reducing reliability and life expectancy. Wooldridge and Jennings [8] define agents to be computer systems that have properties such as:

- autonomy
- social abilities, and
- reactivity and pro-activeness

In the context of condition monitoring the agent perceives the environment through one or more sensors. Additional information about the environment may also be acquired through communication with other agents or systems (a system may be given an agent wrapper to enable uniform communication). An agents ability to influence its environment may in the context of condition monitoring be to operate a valve/switch or adjust a process. An action may also be to communicate with some other agents or human, e.g. a technician close by and ask for help to carry out some preventive or corrective needs.

11.3.1 The Maintenance Agent

A maintenance agent is specialized in interpreting data from the device it is connected to. Figure 11.1 presents the outline of a maintenance agent in its environment. 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 [9]. Handling groups of sensors with a dependency between measurements enabling sensor agents to collaborate and learn from experience, resulting in more reliable performance. Maintenance agents may also improve their performance, e.g. recalibrate sensors if needed, or determine if sensors are faulty. Similar sensors may also share experience enabling them to avoid repetition of similar failures or make estimates on their reliability.

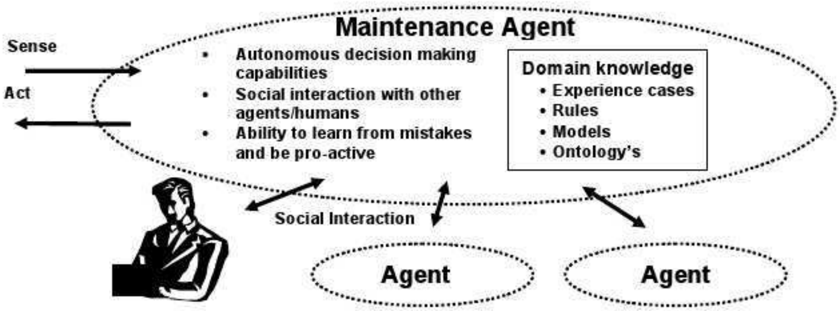

Figure 11.1: Outline of the Maintenance Agent in its environment.

11.4 Factors Affecting Decisions by Agents

In an agent based approach a critical issue is the agents decision whether to take a certain action autonomously or to collaborate with humans (technician/economist) in problem solving. For this decision a number of factors have to be considered. Some of them are domain dependent and also dependent on the reasoning method applied by the agent, since different methods and techniques enable more or less informed decision making. Case-Based Reasoning Systems have some desirable properties enabling an informed decision. These systems make reasoning in terms of similarity, confidence and usefulness. These are important for agents as well as humans during a decision making process. The most central measurement in Case-Based Reasoning systems is similarity, i.e., how similar a previous case is to the current situation and also reflecting how well the previous solution can be reused in the current situation. Some adaptation of the solution may be needed. If all the symptoms and conditions are identical between the current situation and the case from the case library, the same solution can be reused without modifications. Otherwise, even if a good matching case is found, other factors may influence the decision and the agent may decide to either doing nothing or applying/adapting a solution case further down the similarity ranking list or apply multiple solutions. Other factors the agent may take into account to make an informed and final decision are given below. Some of the most central factors in the context of maintenance are (not given in priority order):

1. How similar is the case to the current situation
2. Track record of how successful the case was in the past
3. The confidence in the case and its solution given a current situation
4. The benefit of a case (how efficiently the solution solves the problem)
5. The cost of implementing the solution correctly
6. The consequences/costs, short term/long term if the agent/human is idle by taking no action
7. The consequences/costs if the proposed solution in the case is wrongly deployed

The track record of a case (factor 2) is important since a good matching case may have a less good track record and other statistics appearing against it. The confidence in a case (factor 3) may depend on the nature of the cases in the case library that are similar to the current situation.

E.g. if there are many similar cases nearby the same solution, the confidence grows. But if there are surrounding cases with a very different solution, the confidence in the most similar case may be reduced. The benefit of a case (factor 4) may also have a large influence on which case is selected, e.g. one case may have a solution fixing the problem temporarily while other cases offer long term solution. Factor 5-7 are cost and risk related factors. They have a major influence on the selection of the final case and also if an expert should be included in the decision process. These factors should reflect company policy, current economical situation and also laws and regulations. An example would be that a specific well matching case is the best matching case, and there is a high confidence and success rate connected to the case, but the solution in the case increases the risk of hazardous leakage damaging the environment, and environmental regulations may forbid this and may give the company unacceptable negative publicity. It may be argued that some of these factors may be included into the case descriptions and similarity measurement, but since many of them change with time and are complex in themselves, we propose that for monitoring and diagnostic tasks in industry it may be an advantage and lead to reduced complexity if these factors are handled separately in the decision process or decision support process instead of completely integrating them into the Case-Based Reasoning system.

11.5 Designing and Building Agent-Based Systems using Artificial Intelligence

In this section we give an example of an agent based maintenance system both able to perform corrective maintenance, preventive maintenance, and condition based maintenance (possibly with help of other agents and technicians).

Agents are being implemented with a wide variety of different techniques; both using traditional software engineering methods and techniques such as object oriented programming and Artificial Intelligence methods and techniques such as Artificial Neural Networks or Case-Based Reasoning. Case-based reasoning [10] (CBR) offers an alternative to implement intelligent diagnosis systems for real-world applications [11]. Motivated by the doctrine that similar situations lead to similar outcomes, CBR fits well to classify the current new sensor signals based on experiences of past categorizations. The main strength lies in the fact that it enables directly reusing concrete examples in history and consequently eases the knowledge acquisition bottleneck [10]. In figure 11.2 an example of an agent using case-based reasoning is shown. The Sensor signal is preprocessed, compared with the case library based on domain dependent similarity matching; the best matching cases are adapted to the current situation and suggested as solution to a human.

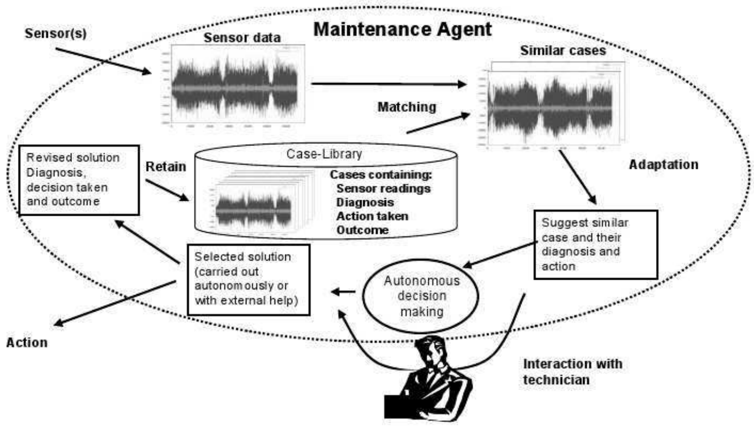

Figure 11.2: An example of an agent based system using case-based reasoning.

11.5.1 Prototype Agent-Based Fault Diagnosis System Based on Sensor Signals and Case-Based Reasoning A Case Study

Abnormality of industrial machines can be reflected by some key states during their operation. Using sensor technology it is possible to detect and measure the values of these system states and their profiles. We can then process and analyse the collected sensor signals in order to find out hidden symptoms. The received signals are processed by wavelet analysis [12] in order to filter out noise and at the same time to extract a group of related features that constitutes a reduced representation of the original signal. The derived feature vector is then compared with the known cases in the case library with its neighboring cases sorted out. The agent system can, based on the symptoms, reason about the class of fault associated with the machine or make prediction about what potential problem is likely to occur in a near future. A general system structure for this purpose is illustrated in figure 3, which includes signal filtering, feature extraction, and pattern classifier as its important components.

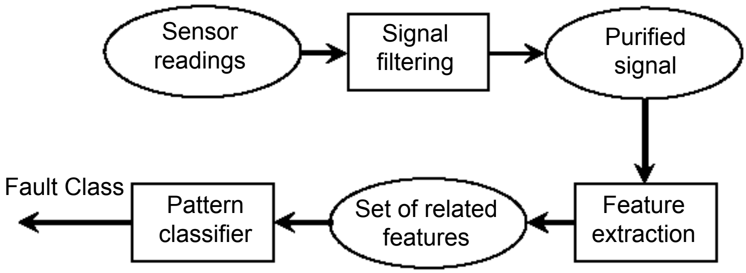

Figure 11.3: Fault diagnosis based upon sensor signals.

As a case study we applied the proposed approach to diagnosis of industrial robots manufactured by a large company in Sweden. The prototype system developed for this purpose is shown in figure 11.4. Sound signals are gathered from the robot to be tested via a microphone device and then transmitted to the computer for pre-processing. The pre-processing is tasked to filter out or remove unwanted noise as well as identify period information from a sound profile. Subsequently sound features are extracted from the frequency domain and they are assembled into a feature vector as a condensed representation of the original sound signal. Classification of the feature vector is performed based upon previously classified sound descriptions in the case library. The experiments have shown that this system is able to successfully diagnose faults in an industrial robot based on a low number of previous examples. The system is able to successfully diagnose faults in an industrial robot based on sound recordings (6 recordings from faulty robots and 24 recordings from normal robots are used in the evaluation).

It is worth mentioning that the above prototype system has some similarities with the Open System Architecture for Condition Based Maintenance (OSA-CBM) [13]. That architecture suggests that a Con-

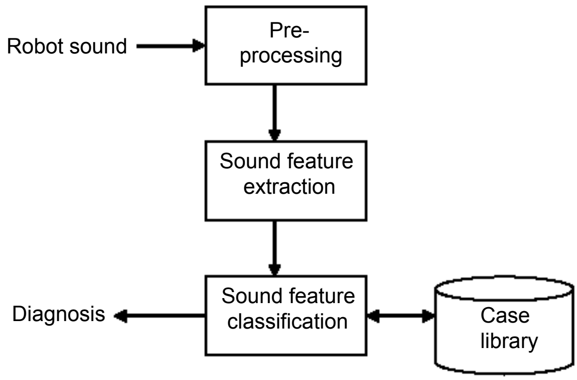

Figure 11.4: Schematic of system logic.

dition Based Maintenance (CBM) system be divided into seven modules [14] including sensors, signal processing, condition monitoring, diagnosis, prognosis, decision support, and presentation. The system presented here in this paper has a microphone as sensor module and pre-processing & feature extraction steps as signal processing module in correspondence to the OSA-CBM architecture. In addition, the case-based classification in figure 4 also serves as a condition monitoring module by detecting and identifying deviations in sound profiles. Figure 11.5 depicts the system transfered to OSA-CBM standard.

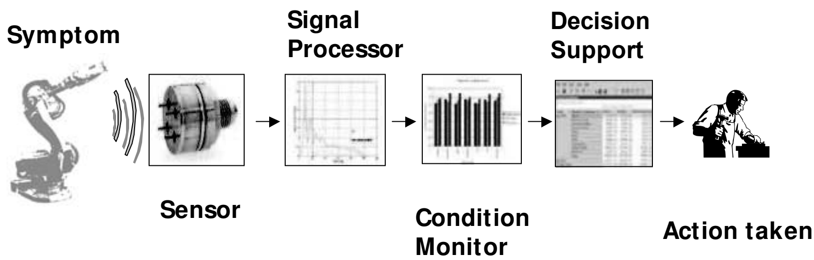

Figure 11.5: System transfer to OSA-CBM standard.

11.5.2 Signal Pre-Processing and Feature Extraction

Sounds of robots in industrial environments typically contain unwanted noise. Signal pre-processing is used to purify the original signal 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 sensor data received. 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 pattern classifier with a moderate number of inputs for effective analysis and reasoning, representative features from the sensor signals have to be extracted. Algorithms used for classification can easily be misguided if presented with data of to 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 [15]. We deal with signal pre-processing and feature extraction by applying wavelet analysis [12]. 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. In a related paper [16] we experimentally verified that, under certain circumstances of strong background noise, wavelet outperforms Fourier transform in supplying distinguishable feature vectors between different faults for case-based classification. The signal $f(t)$ will be represented as a weighted sum of the wavelets $\psi(t)$ and the scaling function $\phi(t)$ by

$$f(t) = A_1\phi(t) + A_2\psi(t) + \sum_{n\in+Z,m\in Z} A_{n,m}\psi(2^n t - m) \tag{11.1}$$

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

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

$$\int_{-\infty}^{\infty} \psi(t)\, dt = 0 \tag{11.2}$$

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

$$\psi_{(sl)}(t) = 2^{\frac{-s}{2}} \psi\left(2^{-s}t - l\right) \tag{11.3}$$

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

$$\phi(t) = \sum_{k=-1}^{N-2} (-1)^k c_{k+1} \psi(2t + k) \tag{11.4}$$

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 (see Fig. 11.6). 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 wavelet coefficient matrix is applied to the input data vector. The matrix is applied in a hierarchical algorithm, sometimes called a pyramidal algorithm. The wavelet data values are arranged so that odd rows contain an ordering of wavelet data values that act as the smoothing filter, and the even rows contain an ordering of wavelet coefficients with different signs that act to bring out the data's detail. The matrix is first applied to the original, full-length vector. Fig. 11.6 shows an example of a data vector consisting of 8 samples. The samples can be any type of data; sensor signals from various process applications, stock market curves etc. In this paper the samples are acoustic signals from a gearbox of an industrial robot.

The data vector is smoothed and decimated by half and the matrix is applied again (see Fig. 5).

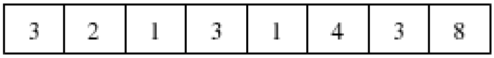

Figure 11.6: Original signal consisting of 8 samples

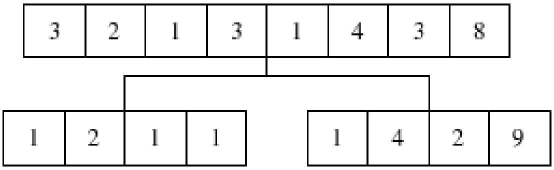

Figure 11.7: Smoothed data vectors

Then the smoothed, halved vector is smoothed, and halved again, and the matrix applied once more. This process continues until a trivial number of "smooth-smooth- smooth..." data remain (see Fig 11.8).

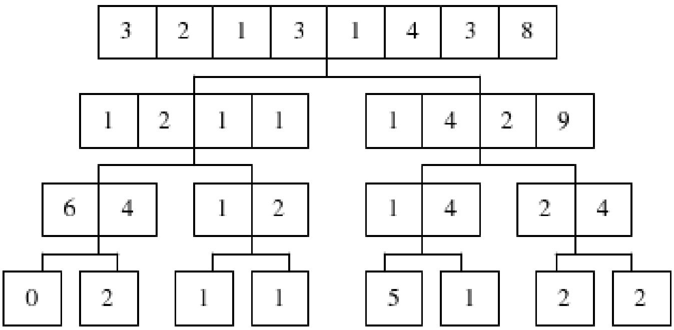

Figure 11.8: The result of the pyramidal algorithm

This system uses the wavelet packet transform algorithm. It is a computer imple- mentation of the Discrete Wavelet Transform (DWT). It uses the Daubecies mother wavelet, scaling function and wavelet coefficients [17].

The result of the pyramidal algorithm is a tree of smoothed data values (see Fig.11.8). Each collection of smoothed data values (node in the tree) can be seen as a time-frequency-packet. Each time-frequency-packet can be seen as a filtered version of the original data samples. As an example, the left packet in Fig. 11.7 can be seen as a low pass filtered version of the original data and the right packet in Fig. 11.7 can be seen as a high pass filtered version of the original data. The leaves of the tree can be seen as high and low pass units of length 20.

The depth of the tree is determined from the length of the input data. If the input data are of length 2^n the depth of the tree will be n. A suitable collection of time-frequency-packets can be selected by taking a cross section of the tree at an arbitrary depth. Each sibling in the cross section of the tree is spanning the entire time of the original data set. This means that going deeper in the tree produces at better resolution in frequency but a poorer resolution in time. The best compromise between time and frequency resolution is to take a cross section in the tree were the length of each sibling is the same as the number of siblings in the cross section. At a given depth n and with original data size S, the length of a sibling (or leaf) is $\frac{S}{2^n}$ and the number of siblings is 2^n.

The wavelet packet algorithm offers the basis for the Pre-processing process. The input signal is first divided into windows of discrete time steps. Each window is then passed to the wavelet packet algorithm resulting in a wavelet packet tree as pictured in Fig 11.8. The wavelet data values from a cross section of the wavelet packet tree are then passed to the Feature Extraction process.

Our system uses normalized wavelet data values as features. The values are selected from a cross-section of the wavelet packet tree. Gear defects often show their presence as sharp peaks or dips in the sound. Such peaks or dips can be spotted in some dominant wavelet data values in certain packets in the cross section of the wavelet packet tree. The feature extraction component examines the wavelet data values and extracts one dominant value from each packet in a cross section at an arbitrary depth. In Fig. 11.9 the grey area shows a cross section at level 2 in the tree. The chosen coefficients are those that are marked as bold. They are chosen because they are the dominant values in each packet in that cross section.

Frequency-based features characterize sensor signals according to their amplitudes under significant frequencies. As many fundamental signal analysis methods are available to yield frequency spectra, we seem to

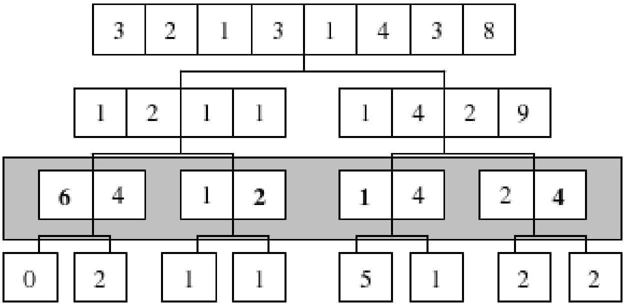

Figure 11.9: Feature identification from wavelet data values

have more solid basis for extracting features based on frequency than for deriving time-based features. We thus adopt frequency-based features as descriptors of condition parts of cases in our research. Generally a vector of frequency-based features is formulated as

$$FV = [Amp(f_1), Amp(f_2), ..., Amp(f_n)] \tag{11.5}$$

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 feature vector is assembled from these dominant wavelet values. A feature vector forms a cross section of wavelet data values at level n in the wavelet packet tree containing 2^n features. This system is dynamic and can assemble vectors from all depths of the tree.

11.5.3 Case-Based Classification Using Extracted Features

After the features have been extracted from sensor signals, we perform case-based reasoning to make classification of the current fault using known cases in the case library. Figure 11.10 gives an overall illustration for this procedure, which consists of the following two steps:

1. Retrieval: compare the feature vector with the known cases in the library by means of similarity calculation and subsequently select the k nearest cases exhibiting the highest similarity degrees;

2. Solution fusion: determine the fault class associated with the current feature vector in terms of both the classes of the retrieved cases and their similarity values with respect to the probe case.

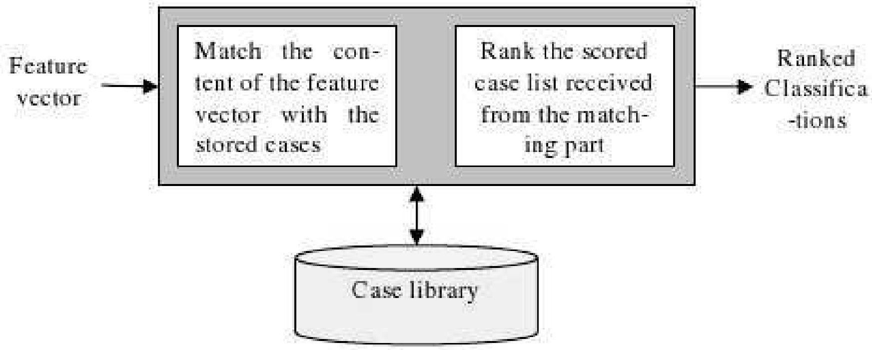

Figure 11.10: Case-based classification as decision support

The matching algorithm calculates the Euclidean distance between the case that is to be classified and those cases previously stored in the case library. The distance function uses the feature vectors along with a set of weights defined on the features. Such weights c_j are incorporated into the distance calculation, as indicated in 11.6, to reflect different importance of different features.

$$\sum_{j=d}^{d} |a_j - b_j| * c_j, a, b, c \in \Re^d \tag{11.6}$$

The classification of robot sound is based on the above matching function. The result of matching yields a scored list of the most similar cases. This list can be presented to responsible technicians as decision support for their further evaluation. An alternative is to derive a voting score for every class involved in the retrieved list of similar cases and then the final decision is settled upon the class exhibiting the highest voting score [16]. It is worthwhile to mention that the performance of our CBR system is improved each time when a new classified case is injected into the case library. The system can thereafter be tested with sounds from other robots previously classified by experts so as to estimate its accuracy. If the accuracy is estimated to be adequate, this CBR system can then be applied to diagnosing robot faults for practical usage.

11.5.4 Sound Classification and Results

Sounds from 24 healthy robots and 6 faulty robots were collected to enable case-based classification of conditions of robots. Two types of faults need to be recognized in the experiments hereafter called Fault 1 and Fault 2. A notch on the big gear wheel in the gearbox causes Fault 1. This fault is hearable and is characterized by a low frequency impulse sound in the middle of the rotation of the axis. Fault 2 i caused due to a slack between the gear wheels in the gearbox and can be heard as bumps at the end of each rotation. A feature vector is assembled with peak wavelet coefficients taken from different depths in a wavelet package tree [12] and it is then matched with the previously inserted cases in the case library. The prototype system demonstrated quite good performance by making right judgements in 91% of the all tests (see further down). Table 11.1 displays a ranked list of the three best matching cases in the case library according to the similarity values calculated. As can be seen from the table, a previously diagnosed notch fault recording is deemed to be the most similar case thereby making the strongest recommendation to classify the probe situation into notch fault. The cases ranked the second (case #12) and the third (case #4) are descriptions classified as normal in the case library. This list of the most similar cases can be presented to human operators as decision support.

Table 11.1: A ranking of the most similar cases for the sound profile.

Case name	Similarity	Case ranking
Notch Fault #2	98%	1
Normal case #12	84%	2
Normal case #4	83%	3

We also investigated the classification accuracy in relation with different feature vector sizes in order to assess the smallest number of features that still produce good classification performance. The diagram in figure 11 indicates the relation between the classification error rate and the number of features. The upper curve in the figure shows the results when only top 1 case was considered for solution fusion. The curve below in the diagram shows the classification results when the top three cases were considered. When only the nearest case was considered, the system produced a classification rate of 91%. When the three nearest cases were considered, the classification rate of the system rose to 99%.

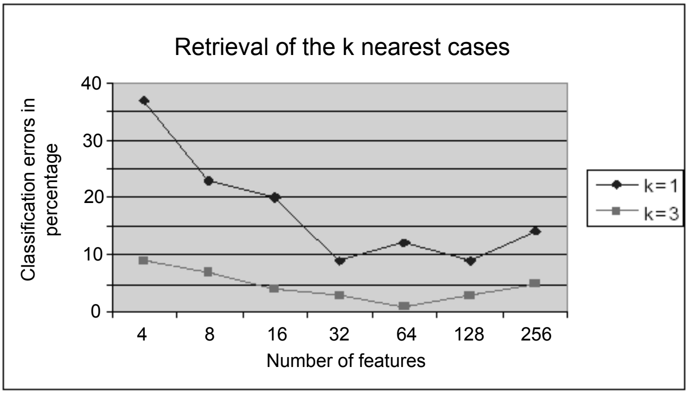

Figure 11.11: Relation between classification performance and the number of features.

11.6 Conclusions

Artificial Intelligence offers a number of methods and techniques, which offers potential benefits if harnessed properly. The Agent paradigm and agent architecture is one of these. In combination with other methods such as case-based reasoning the potential for building valuable systems for monitoring and maintenance is argued to be large and enable valuable properties, sometimes difficult to achieve with a traditional software architectures and engineering. This is exemplified with a system that we have implemented in the area of industrial machine diagnostics. The case-study shows that the concept of intelligent agents in combination with Case-based reasoning offer a potential of increased production quality, flexibility, reliability and fast delivery times in production industry. Agents may enable improved decision making since they can be designed to learn from experience, use and transfer experience relevant to the current situations, collaborate with other agents, systems and humans. This enables flexible and modular maintenance systems where different suppliers can deliver agents, developed to be experts in a specific tasks. Most of the different parts needed for an agent based maintenance approach have been shown to work in different research projects and case studies. To bring the benefits of agent based maintenance systems to industry the first issue is to spread knowledge and understanding of the agent paradigm and its strengths and weaknesses. Also spreading information on successful research projects where the agent paradigm is used, or where the agent paradigm has been used as a thought concept for ideas, implementation and functionality is important. But industry does not need to wait for an industry standard framework for maintaining agents; thinking in terms of agents already gives a number of advantages in system design and development, and may lead to valuable features and functionality that may not have been thought of without this framework of thoughts.

11.7 Acknowledgement

This work is sponsored by the Swedish Foundation for Strategic Research, ProViking, and is a part of the ExAct and Factory in a Box projects.

Bibliography

[1] Mba D. and Rao R. B. K. N. Condition monitoring and diagnostic engineering management. *COMADEM, Proceedings of the 18th International Congress*, 2005.

[2] E. Mangina. Intelligent agent-based monitoring platform for applications in engineering. *International Journal of Computer Science and Applications*, 2(1):38–48, 2005.

[3] M. Bengtsson, S. Elfving, and M. Jackson. Maintenance as an enabler of production system concepts - the factory-in-a-box concept. *Journal of Quality in Maintenance Engineering*, 2007.

[4] W. Cheetham. Case-based reasoning for general electric appliance customer support. *forthcoming paper, contact cheetham@research.ge.com*, 2006.

[5] T. Pirttioja, Pakonen A., I Seilonen, Halmel A., and Koskinen K. Multi-agent based information access services for condition monitoring in process automation. *3rd IEEE International Conference on Industrial Informatics (INDIN)*, pages 240–245, 2005.

[6] S. Franklin and A. Graesser. Is it an agent, or just a program: A taxonomy for autonomous agents. *Lecture Notes in Computer Science*, page 1193:2135, 1997.

[7] S. ussel and P. Norvig. *Artificial Intelligence A Modern Approach Second Edidtion.* Prentice Hall International, 2003.

[8] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115152, 1995.

[9] Peter Funk, Erik Olsson, Marcus Bengtsson, and Ning Xiong. Case-based experience reuse and agents for efficient health monitoring, prevention and corrective actions. pages 445–453, June 2006.

[10] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Com*, 7:39–59, 1994.

[11] M. Nilsson, P. Funk, and M. Sollenborn. Complex measurement in medical applications using a case-based approach. *Workshop Proceedings of the International Conference on Case-Based Reasoning*, pages 63–73, 2003.

[12] E. Beckenstein G. Bachman and L. Narici. *Fourier and wavelet analysis.* Springer, 2000.

[13] M. G. Thurston. An open standard for condition-based maintenance systems. *AUTOTESTCON Proceedings*, pages 401–415, 2001.

[14] M. Bengtsson. Standardization issues in condition based maintenance. *Proceedings of the 16th International Congress on Condition Monitoring and Diagnostic Engineering Management*, pages 651–660, 2004.

[15] Thomas M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[16] E. Olsson, P. Funk, and M. Bengtsson. Case-based diagnosis of industrial robots based on acoustic signals. *Proceedings of the European Conference on Case-Based Reasoning*, pages 686–701, 2004.

[17] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 41:906–966, 1988.