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, editors: Peter Funk, Pedro Gonzales, pages 686-701 Springer LNCS, June 2004.

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

Erik Olsson[1], Peter Funk[1], Marcus Bengtsson[2]
[1]Department of Computer Science and Engineering
[2]Department of Innovation, Design and Product Development
Mälardalen University, Västerås, Sweden
{erik.olsson; peter.funk; marcus.bengtsson}@mdh.se,

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 degraded product quality due to mistakes in judgments. The acquired experience is also difficult to preserve and transfer and it often gets lost if the corresponding personnel leave the task of testing. We propose herein a Case-Based Reasoning 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 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 system's performance. So far the developed system has been applied to the testing environment for industrial robots. 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.

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 CDs 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

* This research is part of the ExAct project, funded by The Swedish Foundation for Strategic Research (ProViking) and ABB.

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. Math Lab or more sound and vibration profiled tools such as the Plato toolbox [6]) 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 [7] 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 [1].

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 2 gives a brief overview of the sound classification technique. Section 3 describes the model used in this paper to classify sound recordings. Sections 4, 5 and 6 describe the implementation of the prototype classification system based on the model. Section 7 discusses system evaluation with a case study. Section 8 gives an experimental comparison of FFT and wavelet analysis and finally section 9 concludes this paper with summary and conclusions.

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 recordings.

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 [2, 3]). 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.

2.2 Features and Feature Vector

When experienced technicians are classifying robot sound they listen for abnormalities 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 well-adopted 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.

$$\left[\frac{max\_value \;\; 45}{300 \;\; HZ}, \frac{max\_value \;\; 18}{520 \;\; HZ}, \frac{max\_value \;\; 89.6}{745 \;\; HZ} \right] \qquad \textbf{(1)}$$

The adoption of frequency-based features in this context is motivated by the awareness 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 [4] 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 4.2.

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.

3 Case-Based Categorization of Machine Sound

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 4, 5 and 6, respectively. Sound is obtained from the robot to be diagnosed via a microphone as shown at the top left in Fig. 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. 1 is responsible for filtering and removal of unwanted noise. It also extracts period information from the sound.

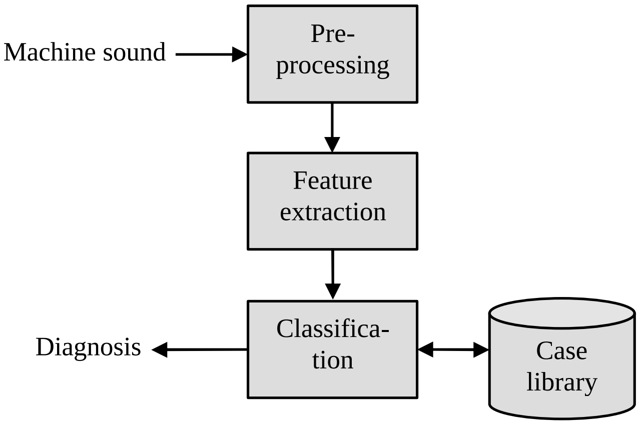

Fig. 1. Schematic picture of the system

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. 1. After a new sound has been classified it is added to the case library. The classification process will be described in section 6. 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.

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) [9]. This architecture is seen as a proposed standard for Condition Based Maintenance (CBM) system which is recommended to consist of seven modules [10], including sensors, signal processing, condition monitoring, diagnosis, prognosis, decision support, and presentation (see Fig. 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.

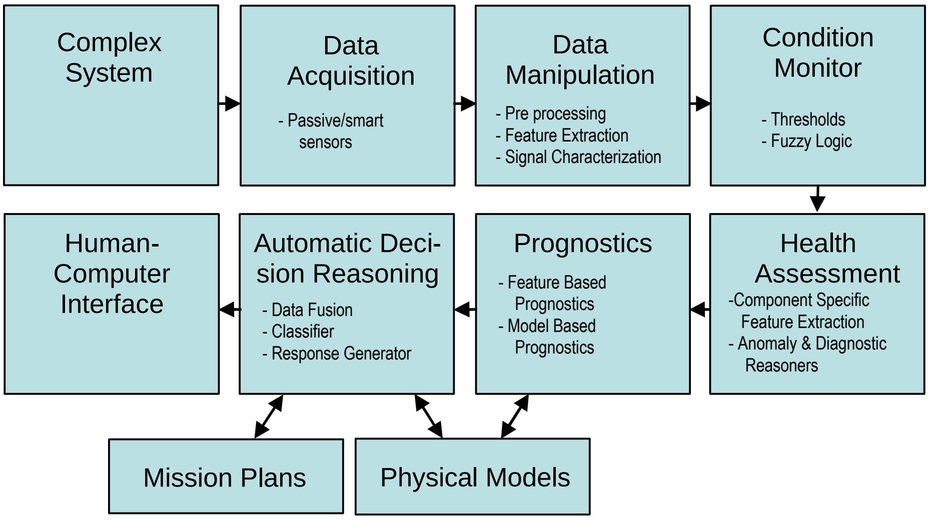

Fig. 2. The standard OSA-CBM architecture proposed in [9]

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.

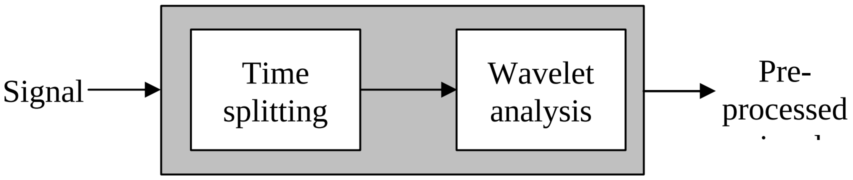

Fig. 3. Pre-processing of the signal in the system

Fig. 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. 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.

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>=2. This is due to the way the wavelet packet algorithm is implemented.

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 8). The signal f(t) will be represented as a weighted sum of the wavelets $y(t)$ and the scaling function $j(t)$ by

$$f(\mathrm{t}) = A_1 j(t) + A_2 y(t) + \sum_{\substack{n\in +Z,\\ m\in Z}} A_{n,m} y(2^n t - m) \qquad \textbf{(1)}$$

where $y(t)$ is the mother wavelet and $j(t)$ is the scaling function.

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

$$\int_{-\infty}^{\infty} y\,(t)dt = 0 \qquad \textbf{(2)}$$

Dilations and translations of the mother wavelet function define an orthogonal basis of the wavelets as expressed by

$$y_{\,(sl)}(t) = 2^{\frac{-s}{2}}\, y\,(2^{-s}t - l) \qquad \textbf{(3)}$$

where variables *s* and *l* are integers that scale and dilate the mother function $y\,(t)$ 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

$$j\,(t) = \sum_{k=-1}^{N-2} (-1)^k c_{k+1} y\,(2t + k) \qquad \textbf{(4)}$$

where $j\,(t)$ is the scaling function for the mother function $y\,(t)$, 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.4). 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. 4 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

Fig. 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).

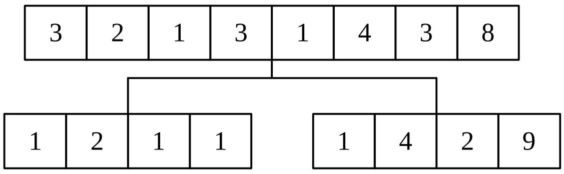

Fig. 5. Smoothed data vectors

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 6).

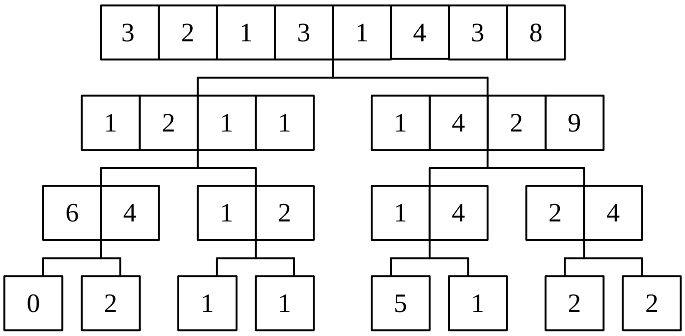

Fig. 6. The result of the pyramidal algorithm

This system uses the wavelet packet transform algorithm. It is a computer implementation of the Discrete Wavelet Transform (DWT). It uses the Daubecies mother wavelet, scaling function and wavelet coefficients [11].

The result of the pyramidal algorithm is a tree of smoothed data values (see Fig. 6). 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. 5 can be seen as a low pass filtered version of the original data and the right “packet” in Fig. 5 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 2^0.

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 $S/2^n$ 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 6. The wavelet data values from a cross section of the wavelet packet tree are then passed to the Feature Extraction process.

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). The extracted features are then stored in a feature vector (right box in Fig. 7).

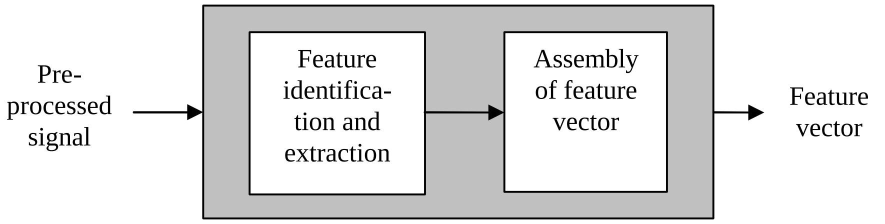
Pre-processed signal
Feature identifica-tion and extraction
Assembly of feature vector
Feature vector

Fig. 7. Feature identification in the system

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. 8 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.

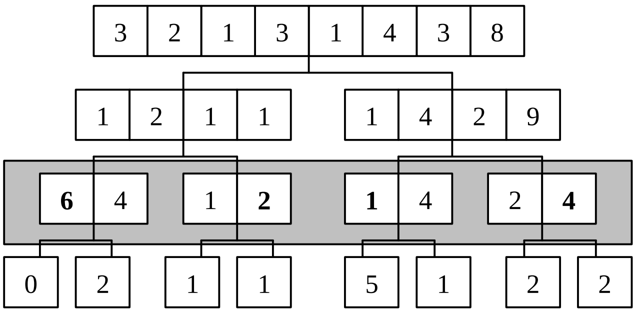

Fig. 8. Feature identification from wavelet data values

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. 8 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 $\bar{x}$ is calculated by

$$\bar{x} = \frac{(X_1 + X_2 + ... + X_w)}{w} \tag{6}$$

Here w is the number of windows and $\bar{x}$ is the final feature vector that will 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 expert's 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 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.

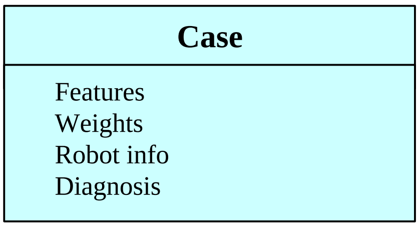

Fig. 9. Data structure for stored cases in the case library

6 Fault Classification

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

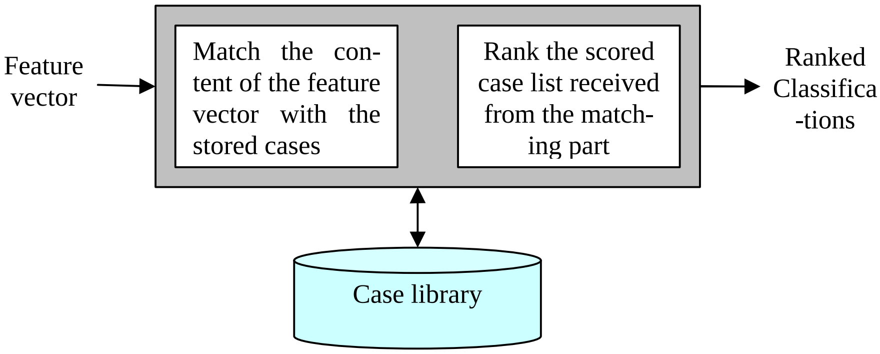
Feature vector
Match the content of the feature vector with the stored cases
Rank the scored case list received from the matching part
Ranked Classifica -tions
Case library

Fig. 10. Case-based classification as decision support

The matching algorithm calculates the Euclidian 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 (7), to reflect different importance of different features.

$$\sum_{j=1}^{d}\left|a_j - b_j\right| * c_j, a, b, c \in \Re^d \qquad \textbf{(7)}$$

The classification of robot sound is based on the above matching function. The result of matching yields a scored list of the most similar 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 [13].

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 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. 11, 12, 13 and 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.

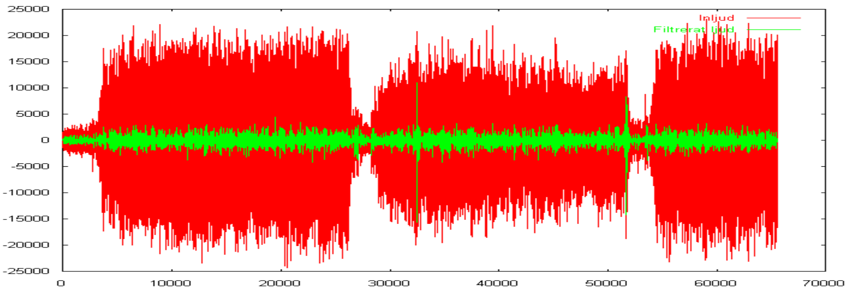

Fig. 11. Sound signals for robot Fault #1a

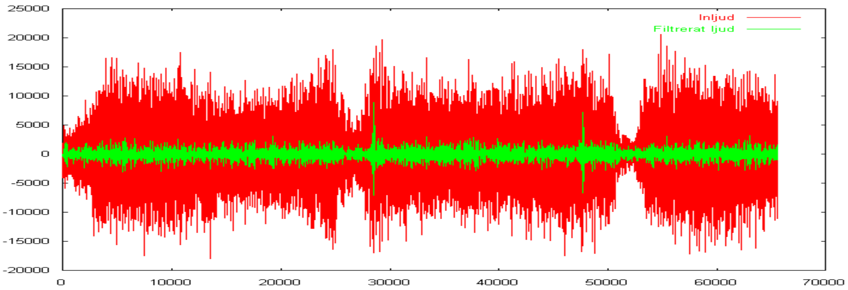

Fig. 12. Sound signals for robot Fault #1b

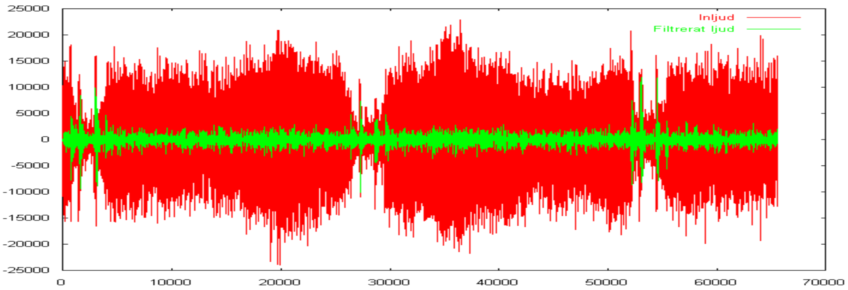

Fig. 13. Sound signals for robot Fault #2a

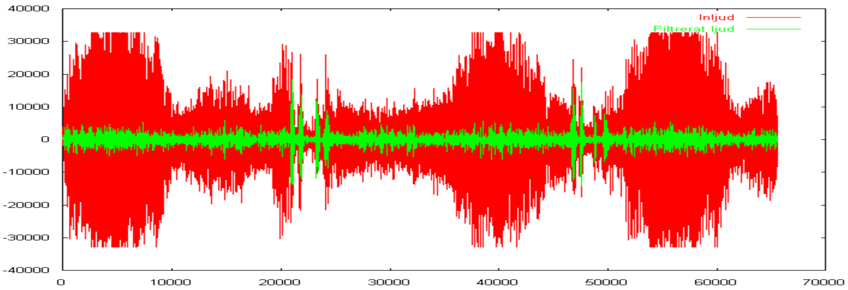

Fig. 14. Sound signals for robot Fault #2b

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 following two figures. Fig. 15 shows the distribution of feature 4 extracted from the sound signals of the robots not equipped with payloads. Fig. 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.

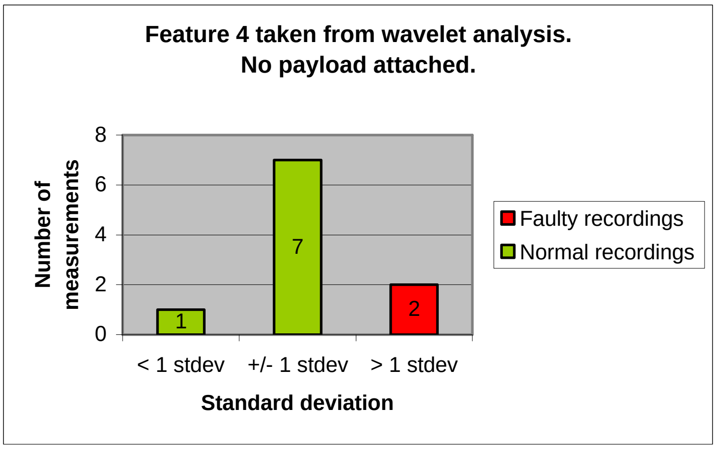

Fig. 15. Distribution of feature 4 for robots not equipped with a payload

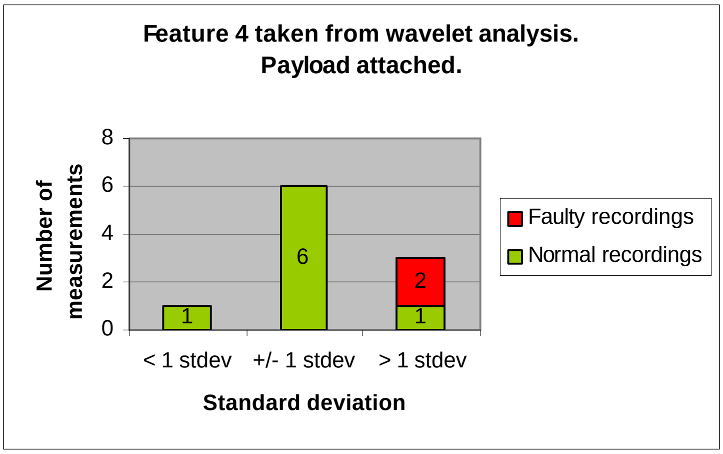

Fig. 16. Distribution of feature 4 for robots equipped with a payload

Example of Case Retrieval

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

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

Table 1. The three most similar cases 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. 15 and 16). One other method for weighting is to automate the process using machine learning technique [12]. The matching process can also be extended with a neural net classifier.

8 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 FFT–spectrum was broken down into 64 features and a feature vector was assembled from the features as described in section 5. 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. 17 and 18 show the results of a standard deviation calculation for feature 4 in the feature vectors.

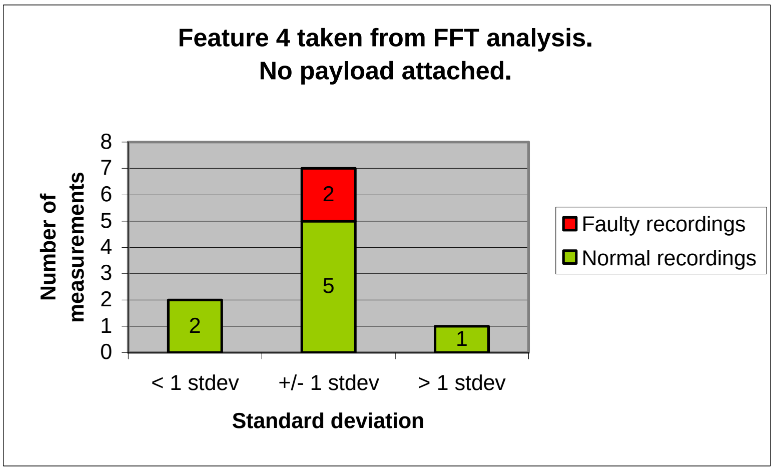

Fig. 17. Distribution of feature 4 for robots not equipped with a payload

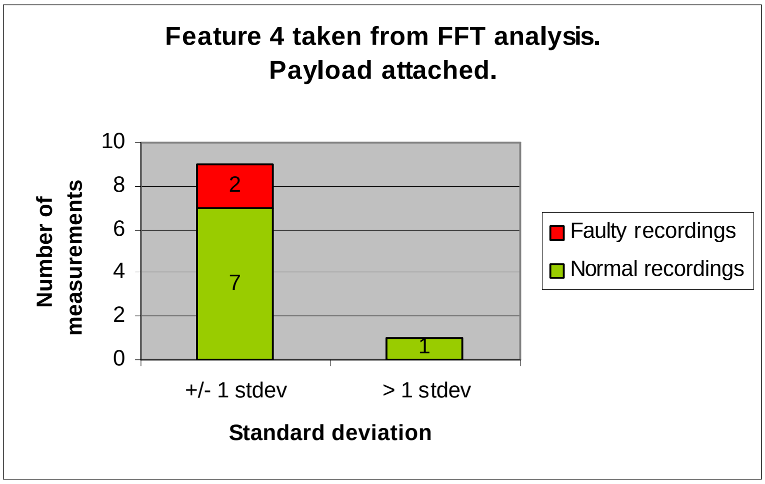

Fig. 18. Distribution of feature 4 for robots equipped with a payload

As can be seen in the distributions in Figs. 17 and 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 suchlike 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 different kinds of robot sounds that are overwhelmed by even stronger background noise.

9 Conclusions

Case-Based Reasoning is a feasible method to identify faults based on sound recordings in robot fault diagnosis. Sound recordings were made under realistic industrial 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 recorded 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.

References

1. Aamodt A., Plaza E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. Artificial Intelligence Com. Vol 7:1, pp. 39-59. IOS Press (1994).
2. Lee S. K., White P. R.: The Enhancement Of Impulse Noise And Vibration Signals For Fault Detection In Rotating and Reciprocating Machinery. Journal of Sound and Vibration (1998), pp. 485-505
3. Lin J.: Feature Extraction of Machine Sound using Wavelet and Its Application in Fault Diagnosis. NDT&E International 34 (2001), pp 25-30.
4. Bachman G. Fourier and Wavelet Analysis. New York, Springer (2000). ISBN:387-98899
5. Nilsson M., Funk P., Sollenborn M.: Complex Measurement in Medical Applications Using a Case-Based Approach. ICCBR (2003).
6. Intelligent Noise/Vibration Measurement and Analysis http://www.abd.uk.com/ 2004-01.
7. Kevin P. Logan, Prognostic Software Agents for Machinery Health Monitoring, MACSEA Ltd. http://www.dexteragents.com/pdf/dexter_wp1a.pdf, (2002).
8. Zhao Z et al: Application of principal component analysis in gear fault diagnosis. Modal Analysis, Modeling, Diagnostics, and Control - Analytical and Experimental, 1991, p 159-163
9. Thurston, M. G., An Open Standard for Web-Based Condition-Based Maintenance Systems. AUTOTESTCON Proceedings (2001). USA, Valley Forge, PA, pp. 401-415.
10. Bengtsson, M., Standardization Issues in Condition Based Maintenance. COMADEM - Proceedings of the 16th International Congress (2003). Växjö University, Sweden. Edited by Shrivastav, O. and Al-Najjar, B. Växjö University Press, ISBN 91-7636-376-7.
11. I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets," *Comm. Pure Appl. Math.*, Vol 41, 1988, pp. 906-966.
12. A. Stahl, Learning Feature Weights from Case Order Feedback, in: Proceedings of the 4th International Conference on Case-Based Reasoning, Springer, 2001.
13. E. Olsson, P. Funk and N. Xiong, Fault Diagnosis in Industry Using Sensor Readings and Case-Based Reasoning, accepted for publication by Journal of Intelligent and Fuzzy Systems, IOS Press.