NONSTATIONARY FEATURE EXTRACTION TECHNIQUES FOR AUTOMATIC CLASSIFICATION OF IMPACT ACOUSTIC SIGNALS

Yasemin BEKIROGLU

2008

Table of Contents

Abstract

Condition monitoring of wooden railway sleepers applications are generally carried out by visual inspection and if necessary some impact acoustic examination is carried out intuitively by skilled personnel. In this work, a pattern recognition solution has been proposed to automate the process for the achievement of robust results. The study presents a comparison of several pattern recognition techniques together with various nonstationary feature extraction techniques for classification of impact acoustic emissions. Pattern classifiers such as multilayer perceptron, learning cector quantization and gaussian mixture models, are combined with nonstationary feature extraction techniques such as Short Time Fourier Transform, Continuous Wavelet Transform, Discrete Wavelet Transform and Wigner-Ville Distribution. Due to the presence of several different feature extraction and classification technqies, data fusion has been investigated. Data fusion in the current case has mainly been investigated on two levels, feature level and classifier level respectively. Fusion at the feature level demonstrated best results with an overall accuracy of 82% when compared to the human operator.

1 Introduction

Condition monitoring applications in the transportation have great importance in ensuring safe operations, since condition monitoring failure can cause serious results. Condition monitoring applications extensively deploy the use of non-destructive testing (NDT) procedures [1,2] to make key assessments and thereby classify the condition of the structure or material that is being inspected [3]. Due to the large scope of condition monitoring applications within the transportation domain, emphasis in this thesis has only been laid on condition monitoring applications involving wooden railway sleepers within the rail transportation domain. Condition monitoring of wooden railway sleeper has been investigated in the current work with the aim of automating the manual wooden railway sleeper inspection procedures.

Wooden railway sleeper inspections in Sweden are generally carried out manually. A human inspector incharge of the maintenance activities walks along the railway track visually examining each sleeper. Decisions concerning the condition of the sleeper are given out by the sleeper inspector and are largely based on intution (see section 1.2). Such a process of manually inspecting each sleeper is slow and time consuming. Human error together with maintaing an even quality standard are other serious issues. Hence it is desired to automate manual sleeper inspection procedures by deploying automatic procedures with an aim of achieving more reliable and robust results with increased speed and accuracy [4].

Automating wooden railway sleeper inspection procedures has already been researched [4]. Such work mainly investigates emulation of human behavior for achieving automation. The emulation process is achieved by selecting and evaluating two non-destructive testing methods. The first method (impact acoustic analysis) aims to build an automatic procedure to replace the usage of an axe for distinguishing sounds; which can be described qualitatively as a crisp sound in case of a good sleeper and a dull thud on their bad counterparts. The second method (vision analysis) is to develop an appropriate machine vision algorithm to replicate the visual examination. Data were collected for each of the above methods and appropriate features were extracted. A pattern recognition based approach was selected for classifying the condition of the sleeper into classes (good and bad in the current case). Results achieved by the work mentioned above demonstrate an overall accuracy of about 90% when compared to the human operator. Though good efficiency rates have been demonstrated through past work, certain short comings of the work have been identified and have been worked upon in the current thesis. A brief discussion concerning such shortcomings is as follows.

Previous works [4] demonstrate the efficient use of pattern recognition approach together with stationary (or frequency based) feature extraction techniques [3]. Frequency based feature extraction techniques produce an overall result detailing the frequencies in the entire signal, with no focus on where the frequencies occurred. Frequency extraction (FE), Mel-frequency cepstral coefficients (MFCCs), Homomorphic cepstral (HCCs) coefficients, Linear predictive coding (LPC) are the most popular frequency based techniques. In contrast, time-frequency based feature extraction techniques splits the input signal into discrete frames separated by time, thereby providing a chance of identifying frequencies that occur in a particular area of the signal. Short-time fourier transform (STFT), Discrete wavelet transform (DWT), Continuous wavelet transform (CWT) and Wigner-Ville distribution (WVD) are the most popular time-frequency based techniques. Hence in the current work, it is desired to investigate the usage of non stationary frequency extraction techniques with an aim of achieving more reliable and robust results. Pattern classifiers such as multi-layer perceptron, Gaussian mixture models, learning vector quantization are combined with non-stationary feature extraction techniques such as Short time fourier transform, Continuous wavelet transform, Discrete wavelet transform, Wigner-Ville distribution. Testing all possible combinations of four feature extraction techniques against different classifiers were performed to evaluate the techniques.

The rest of this thesis is organised as follows. Firstly a brief introduction to the wood inspection process is presented and the pattern recognition and classification system is described providing explanations concerning pattern classifiers. Secondly, data acquisiton method, details concerning preprocessing, feature extraction techniques and data reduction are given. Finally, results and discussion concerning the techniques are presented. The paper finally presents concluding remarks.

1.1 Wooden railway sleeper inspection

Condition monitoring applications involving wood are generally based on manual inspection procedure carried out by a human inspector. The inspector examines each structure visually and, if necessary, some deeper inspection may be performed such as using an axe to hit and judge the condition of the structure by listening to the sound produced. Trained personnel have the ability to intuitively classify the condition of the wood based on visual analysis and acoustic signal produced by striking the object that is being inspected. The visual intuition is that wooden structures in good condition do not bear wide cracks on them indicating good condition and when hit with an axe (acoustic examination) they produce a clear ‘‘crisp’’ sound, whereas rotten or bad structures emit a dull sound, indicating bad condition. Automating these intuitions is a hard task. However, such manual routines are challenged by several factors such as human error. There are mainly two strategies to automate the human inspection behaviour, developing an appropriate machine vision algorithm to compensate the visual examination and building an automatic system to distinguish sounds [5,3].

Since for visual analysis, the sleeper inspector identifies issues such as number of the crack, crack length and the width etc as key for determining the condition of the sleepers, the knowledge extracted from the sleeper inspector concerning visual analysis is very clear. On the contrary knowledge concerning acoustic examination is unclear, since it only explains one property about the sound, the sound being crisp in case of good sleeper and a dull sound in case of bad sleeper. Therefore, based on the knowledge that is largely driven by intuition a pattern recognition approach is suitable. On the automatic interpretation of NDT data using AI techniques, classification by neural networks have been the most popular, mainly because other techniques such as case based reasoning, fuzzy logic require demand the presence of knowledge in the form of cases or rules. The fact that knowledge concerning condition monitoring applications is largely intuitive, supports the choice of pattern recognition in the current problem [4].

2 Pattern recognition and classification

Pattern recognition and classification has been used in various applications such as face, speech, handwriting recognition with the aim of emulating intuitive human skills with increased speed and accuracy. The pattern recognition and classification is based on extracting patterns and distributing them into different groups.

In order to perform sound recognition, firstly impact acoustic signals are collected by using impact acoustics method involving striking the material with an impact source and recording the acoustic signal based on the sound. Then, the process of recognition has mainly two phases, feature extraction and classification respectively. Choosing the features and classifiers has a crucial influence on the recognition rates.

Feature extraction is where a signal is manipulated in order to produce a set of characteristic features for that signal. These features should be chosen in such a way that clear groups or classes of data can be identified. Different systems may distinctly vary in the features they use. In this work four techniques are used that are commonly preferred in nonstationary signal analysis, including Short Time Fourier Transform, Continuous Wavelet Transform, Discrete Wavelet Transform and Wigner-Ville Distribution. These techniques are tested for their ability to classify the condition of wooden railway sleepers.

Since using raw acoustic signals does not yield good results, raw acoustic signals are preprocessed to present better data and feature extraction techniques are applied to them to generate characteristic features.

Principal Component Analysis (PCA) was used after feature extraction to reduce the dimensionality of the resulting data. PCA is a method commonly used for data reduction purposes and it finds a new coordinate system with the axes, principal components, are ordered by the variance within the data to compress the data. Namely it decreases redundant information and input data with high dimension can be represented in a lower dimension space. After applying PCA to each feature set obtained from four feature extraction techniques, several normalization steps are followed to be able to present the data into the classifiers in a more efficient form. Normalization is to translate input values so that they can be exploitable efficiently. The first normalization is based on the eigen values generated through PCA of feature sets. In the current work, the combinations of features are tested. Therefore, when more than one feature set are used, they are simply concatenated to form a long feature vector. To make concatenation more reasonable, it is better for different feature sets to have the same scales in the resulting vector [6]. To achieve this, PCA is first applied to each feature set to compute the eigen values λ^j and eigen vectors U^j. Then each feature vector V^j is projected to eigen vectors and normalized by the sum of the eigen values as in the following [7].

$$\overline{V}^j = U^j V^j \Big/ \sqrt{\sum \lambda_i^j} \tag{1}$$

After this step, each feature set is scaled into the range of [0,1]. This normalization is good to improve training characteristics and in this way the effect of each input vector is of the same magnitude, making training faster generally. The distribution of the data is not changed only the magnitude is scaled into the range of [0,1]. So, the properties of the data is preserved after the normalization. And finally, simple normalization which rescales each input feature independently to have a mean of 0 and a variance of 1. This compensates for the differences in the means and variances of the input dimensions is applied to the feature sets.

Feature extraction is performed to obtain an efficient representation of the data, in this way unnecessary data are not sent to the classification step (reducing the computational burden) where the input is separated into suitable categories. The role of a classifier is to assign the input data represented by their features to a number of different categories. Classification is used to recognize the signal by cataloguing the features of existing signals in some way (training) and then comparing the test signal to the database of features (testing) [8]. The process followed in the system is illustrated in the Figure 1.

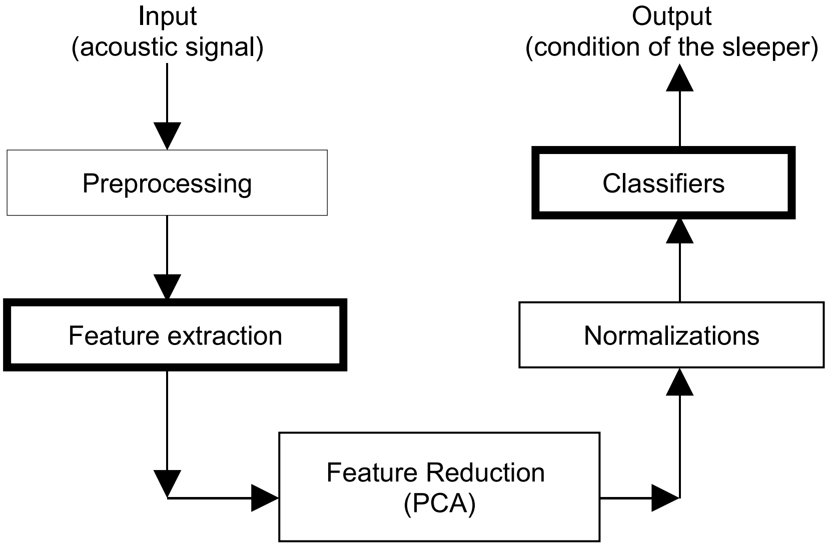


Figure 1. Classification process

Since feature extraction and classification are both required for a recognition task, each classification technique is tested against each feature extraction technique to determine the best combination. Three classification techniques, Multilayer Perceptron, Learning Vector Quantization and Gaussian Mixture Model, are tested in the work. The pattern classification was performed using LNKNET [9].

2.1 Multilayer Perceptron

An Artificial Neural Network (ANN) can be defined as a model of reasoning based on the human brain. An ANN is a system consisting of a lot of processing units, also called neurons, connected with each other. ANNs are developed based on the idea of how biological neurons function and produce results for similar cases learned according to samples provided. The neurons are connected by links, and each link has a numerical weight associated with it. Weights are the basic means of long-term memory in ANNs. A neural network "learns" through repeated adjustments of these weights. In the systems for recognition, Multilayer Perceptron (MLP) which could solve nonlinear problems are preferred. The structure of the MLP chosen in this work can be seen in the Figure 25. Information provided to the network through the Input layer reach the Output layer passing the Middle layer and the output value of the network is computed.

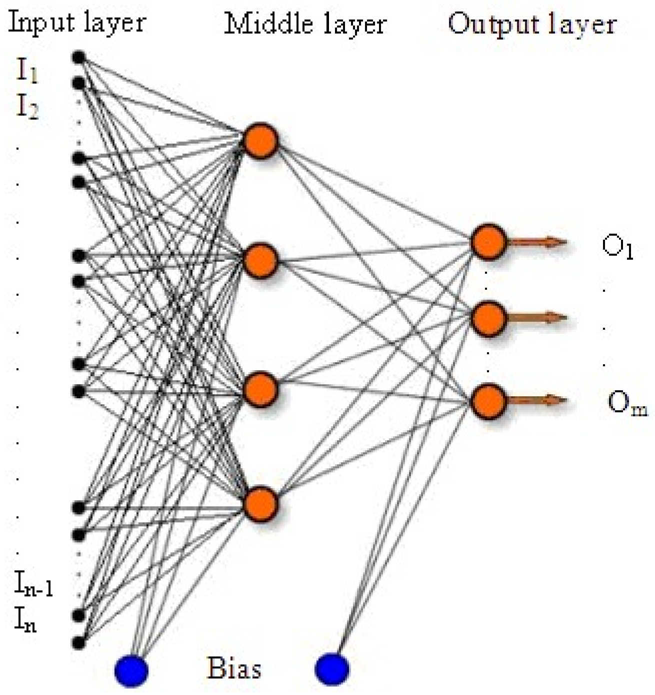


Figure 25. Architecure of MLP

In the training phase of an MLP, input values and the desired output values for these inputs are presented to the network together. When the training is completed, the network is able to generate correct and approximate values to the desired output values for similar inputs. The learning rule of the network has two parts: forward computation, backward computation. In the first one, the output of the network is computed. In the second one, the weights in the network are updated. The output of the network is compared with the desired output for the associated input and the difference between them is taken as the error value for the associated output neuron. Through the backward computation, the error value is decreased by updating the weights of the network.

$$w_{ji}(t) = w_{ji}(t\text{-}1) + \Delta w(t) \qquad (21)$$

If the total error of all output neurons is less than a specified threshold value, the training phase is completed. The output of the neurons in the network is computed by using an activation function. In this work, sigmoid function is used as the activation function which generates values in the range of [0-1]:

$$y=1/(1+e^{-x}) \qquad (22)$$

The backpropagation algorithm used in the training of the network can be summarized in the following steps [10]:

- Initialize all weights to random number between 0 and 1.
- Repeat until stopping condition (a given number of epochs are completed or a tolerable error value is reached) holds

  <u>Present a training sample to the network and compute the output:</u>

  (Forward computation)

  x: input vector, d: desired output vector; {(x(n),d(n)), n=1,2..N},

  $v_j^{(l)}$: the output of neuron j in layer *l*,

  $w_{ji}^{(l)}(n)$: weight of neuron j in layer l that is fed from neuron i in layer *l-1*, $y_i^{(l-1)}$: the output of neuron i in the previous layer *l-1* at iteration n.

  $$v_j^{(l)}(n)=\sum_{i=0}^{m_0} w^{(l)}{}_{ji}(n)y_i^{(l-1)}(n) \qquad (23)$$

  For i=0, we have $y_0^{(l-1)}(n)=1$ and $w_{j0}^{(l)}(n)=b_j^{(l)}(n)$ is the bias applied to neuron j in layer *l*.

  With the sigmoid function, the output of neuron j in layer l:

$$y_j^{(l)} = \vartheta_j(v_j(n)) \tag{24}$$

If neuron j is in the first hidden layer (*l=1*) then $y_j^{(0)}(n)=x_j(n)$

where $x_j(n)$ is the jth element in the input vector x(n). If neuron j is in the output layer (l=L:depth of the network)

$$y_j^{(L)}=o_j(n) \tag{25}$$

error: $e_j(n)=d_j(n)-o_j(n)$

where $o_j(n)$ is the jth element of the output vector

<u>Update each network weight:</u>

(Backward computation)

Compute local gradients of the network:

$$\delta_j(n) = \begin{cases} e_j^{(L)}(n)\vartheta_j'(v_j^{(L)}(n)) & \text{for neuron j in output layer L} \\ \vartheta_j'(v_j^{(l)}(n))\sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) & \text{for neuron j in hidden layer l} \end{cases} \tag{26}$$

$$\vartheta'_j(v_j(n)) = y_j(n)(1-y_j(n)) \tag{27}$$

and weights in layer *l* are adjusted according to the generalized delta rule:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha(w_{ji}^{(l)}(n-1)) + \eta\delta_j^{(l)}(n)y_i^{(l-1)}(n) \tag{28}$$

where η is the learning parameter and α is the momentum constant.

Forward and backward computations are iterated as explained until the stopping criterion is met.

2.2 Learning Vector Quantization

The LVQ network uses both supervised and unsupervised learning to form classifications. In the LVQ network, each neuron in the first layer is assigned to a class, with several neurons often assigned to same class. Each class is then assigned to one neuron in the second layer. The number of neurons in the first layer, S^1, will therefore be at least as large as the number of neurons in the second layer, S^2, and will usually be larger [11]. The architecture of the LVQ network can be seen in the Figure 26.

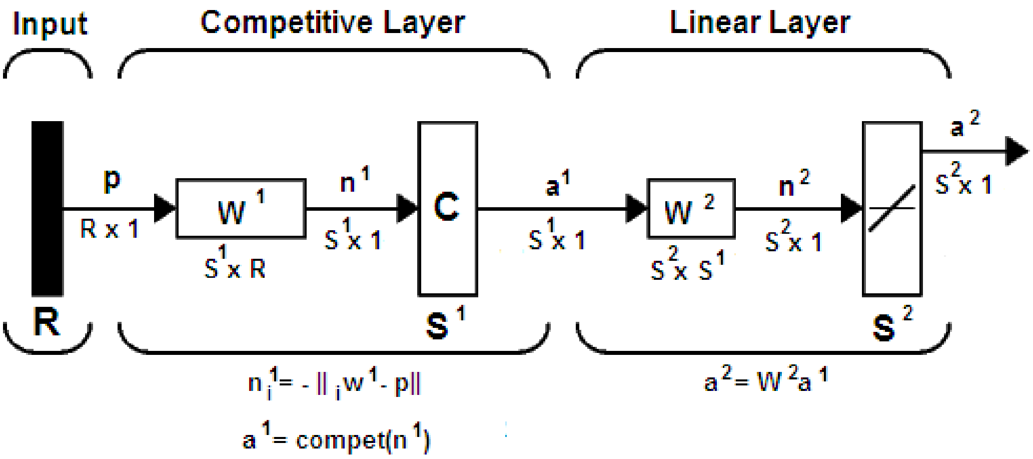


Figure 26. he architecture of the LVQ [11]

Aa with the competitive network, each neuron in the first layer of the LVQ network learns a prototype vector, which allows it to classify a region of the input space. The net input of the first layer of the LVQ is

$$n_i^1 = -\|{}_i w^1 - p\| \qquad (29)$$

or, in the vector form

$$n^1 = -\begin{bmatrix} \|{}_1w^1 - p\| \\ \|{}_2w^1 - p\| \\ . \\ . \\ . \\ \|{}_{s^1}w^1 - p\| \end{bmatrix} \tag{30}$$

and the output of the first layer of the LVQ is

$$a^1 = compet(n^1), \tag{31}$$

therefore the neuron whose weight vector is closest to the input vector will output a 1, and the other neurons will output 0. The winning neuron indicates a subclass rather than a class. There may be several different neurons (subclasses) that make up each class. The second layer of the LVQ network is used to combine subclasses into a single class. This is done with the W^2 matrix. The columns of W^2 represents subclasses, and the rows represent classes. W^2 has a single 1 in each column, with the other element set to zero. The row in which the 1 occurs indicates which class the appropriate subclass belongs to [11].

$$(w^2_{ki} = 1) \Rightarrow \text{subclass i is a part of class k} \tag{32}$$

The process of combining subclasses to form a class allows the LVQ network to create complex class boundaries. The learning in the LVQ network combines competitive learning with supervision. As with all supervised learning algorithms, it requires a set of examples of proper network behavior:

$$\{ p_1, t_1 \}, \{ p_2, t_2 \}, \ldots \{ p_Q, t_Q \}. \quad (33)$$

Each target vector must contain only zeros, except for a single 1. The row in which the 1 appears indciates the class to which the input vector belongs. Before learning can occur, each neuron in the first layer is assigned to an output neuron. This generates the matrix W^2. Typically, equal numbers of hidden neurons are connected to each output neuron, so that each class can be made up of the same number of convex regions. All elements of W^2 are set to zero, except for the following [11]:

If hidden neuron i is to be assigned to class k, then set $w_{ki}^2 = 1$.

Once W^2 is defined, it will never be altered. The hidden weights W^1 are trained with a variation of the Kohonen rule. The LVQ learning rule proceeds as follows. At each iteration, an input vector p is presented to the network, and the distance from p to each prototype vector is computed. The hidden neurons compete, neuron i^* wins the competition, and the i^*th element of a^1 is set to 1. Next, a^1 is multiplied by W^2 to get the final output a^2, which also has only one nonzero element, k^*, indicating that p is being assigned to class k^* [11].

The Kohonen rule is used to improve the hidden layer of the LVQ network in two ways. First, if p is classified correctly, then the weights ${}_{i^*}w^1$ of the winning hidden neuron are moved toward p [11].

$$ {}_{i^*}w^1(q) = {}_{i^*}w^1(q-1) + \alpha(p(q) - {}_{i^*}w^1(q-1)), \text{ if } a^2_{k^*} = t_{k^*} = 1 \tag{34}$$

Second, if p was classified incorrectly, then the wrong hidden neuron won the competition, and therefore its weights ${}_{i^*}w^1$ are moved away from p [11].

$$ {}_{i^*}w^1(q) = {}_{i^*}w^1(q-1) - \alpha(p(q) - {}_{i^*}w^1(q-1)), \text{ if } a^2_{k^*} = 1 \neq t_{k^*} = 0 \tag{35}$$

The result will be that each hidden neuron moves toward vectors that fall into the class for which it forms a subclass and away from vectors that fall into other classes [11].

2.3 Gaussian Mixture Model

In this model-based approach, certain models for clusters are used attempting to optimize the fit between the data and the model. Each cluster can be represented by a parametric distribution, Gaussian, so the data set is modelled by a mixture of these distributions, considering clusters as Gaussian distributions.

The Gaussian Mixture Model (GMM) classifier belongs to the unsupervised classifiers category [11, 13] meaning that the training samples of a classifier are not labelled to show their category membership [12]. During the training of the classifier, the underlying probablity density functions (pdf's) of the observations are estimated [14].

Gaussian mixture model (GMM) is a mixture of several Gaussian distributions and can therefore represent different subclasses inside one class. The probability density function is defined as a weighted sum of Gaussians * [15].

Given a set of m feature vectors $X = \{\bar{x}_1,...,\bar{x}_m\}$, $\bar{x}_i \in R^d$, assumed to be statistically independent and identically distributed, the likelihood that the set is produced by class C_1 is [16]

$$p(X = \{\bar{x}_1,...,\bar{x}_m\}|C_1) = \prod_{i=1,m} p(\bar{x}_i|C_1). \qquad (36)$$

Assuming that the likelihood of a vector can be expressed with a mixture of Gaussian distributions then,

$$p(\bar{x}_i|C_1) = \sum_{l=1}^{K} P(l|C_1)\, p(\bar{x}_i|l,C_1) \qquad (37)$$

where $p(\bar{x}_i|l,C_1) = \dfrac{exp\left(-\frac{1}{2}(\bar{x}_i - \mu_{l,1})^t \sum_{l,1}^{-1}(\bar{x}_i - \mu_{l,1})\right)}{\sqrt{(2\pi)^d \left|\sum_{l,1}\right|}}$. (38)

$P(l|C_1)$ is the prior probability of Gaussian *l* for class C_1 (a weight that changes with the class) , and $p(\bar{x}_i|l,C_1)$ is the likelihood of vector $\bar{x}_i$ produced by Gaussian *l* within class C_1. The parameters of the Gaussian distribution are the mean vector $\mu_{l,1}$ and the diagonal covariance matrix $\sum_{,1}$. To achieve the classification using feature vectors, the GMM needs training. In training phase, parameters of the Gaussian mixture, the weights, the mean vectors and the diagonal covariance matrices, are determined using the Expectation-Maximization (EM) algorithm which computes maximum likelihood estimates iteratively [17]. The initial Gaussian parameters (means, covariances, and prior probabilities) are generated by using the k-means method [18]. After finding Gaussian mixture parameters for each class, a test vector $\bar{x}$ is assigned to the class that maximizes $p(C_j|\bar{x})$, which is equivalent to maximizing $p(\bar{x}|C_j)p(C_j)$ using Bayes rule. When each class has equal a priori probability, the probability measure is $p(\bar{x}|C_j)$. Namely, the test vector $\bar{x}$ is classified into the class C_j that maximizes $p(\bar{x}|C_j)$ [16].

As mentioned, the parameters are estimated by the maximum likelihood criterion using the EM algorithm. In the EM, expectation and maximization steps that are seen in the following are repeated until GMM likelihood $\prod_{i=1,m} p(\bar{x}_i|C_j)$ of the set does not change appreciably or limit on number of iterations is reached [19, 20].

In Expectation step, responsibility p_{il} of each component for each data $\bar{x}_i$ is determined as

$$p_{il} = \frac{P(l|C_j)p(\bar{x}_i|l,C_j)}{\sum_{t=1}^{K} P(t|C_j)p(\bar{x}_i|t,C_j)} \tag{39}$$

In Maximization step, component pdfs and weights are re-estimated based on data and responsibilities:

$$\hat{P}(l, C_j) = \frac{1}{m}\sum_{i=1}^{m} p_{il} \tag{40}$$

$$\hat{\mu}_{lj} = \frac{\sum_{i=1}^{m} p_{il}\bar{x}_i}{\sum_{i=1}^{m} p_{il}} \tag{41}$$

$$\hat{\Sigma}_{lj} = \frac{\sum_{i=1}^{m} p_{il}(\bar{x}_i - \hat{\mu}_{lj})(\bar{x}_i - \hat{\mu}_{lj})^T}{\sum_{i=1}^{m} p_{il}} \tag{42}$$

The task of training a classifier only needs a good enough approximation of the distribution of each class. The number of components, K, is a parameter defining the complexity of the approximating distribution. Too small K prevents the classifier from learning the sample distributions well enough and too large K may lead to an overfitted classifier. More importantly, too large K will definitely lead to singularities when the amount of training data becomes insufficient [15].

The mixture model covers the data well and dominant patterns in the data are captured by component distributions. However, because of its greedy nature, the EM algorithm has some defects; it is sensitive to the initial cofiguration and usually gets stuck at local maxima; for mixtures, it cannot choose the component number automatically and sometimes converges to the boundary of the parameter space [21].

GMM classifiers have been used in many fields from image pattern recognition [22] to text-independent speaker recognition [23, 14].

3 Data preparation

In this section, with the aim of classifying condition of wooden railway sleepers, how data is prepared in order to be able to present them to the mentioned classifiers in the best possible way. The data acquisiton, preprocessing, feature extraction and data reduction steps are explained below.

3.1 Data Acquisition

After significant experimentation concerning the data collection methodology, a metal hammer weighing 1.5 kg dropped from a height of 50 cm was used as the impact source. Best acoustic emissions have resulted on using such an impact source. The resulting impact acoustic emissions have been recorded using a high directional microphone and a 16 bit Analog/Digital card with 44,100 Hz sampling rate. All the acoustic signals were saved on a computer in a WAV format. The WAV format for digital audio is simply the left and the right stereo signal samples. Such an impact system generates a sample input with suitable characteristics for further signal interpretation [4].

Based on the methodology above, data collection was carried out on 200 sleepers of which 144 were in good condition and 56 in bad condition. Data was collected by making impact acoustic tests on both (left and right) ends of the railway sleeper. As a result 400 acoustic signals were acquired. Though 200 sleepers is not a huge number, the limited number of sleepers tested is due to the operational constraints in the rail transportation domain. Since collecting impact acoustic signals of railway sleepers demands re-routing or even cancellation of traffic operations it is an expensive procedure. Moreover, the difference in the number of sleepers in each class (good and bad) was due to the fact that only a limited stretch of railway track could be allocated for closure, which has minimized the scope for handpicking the number of sleepers in each class [4].

3.2 Preprocessing

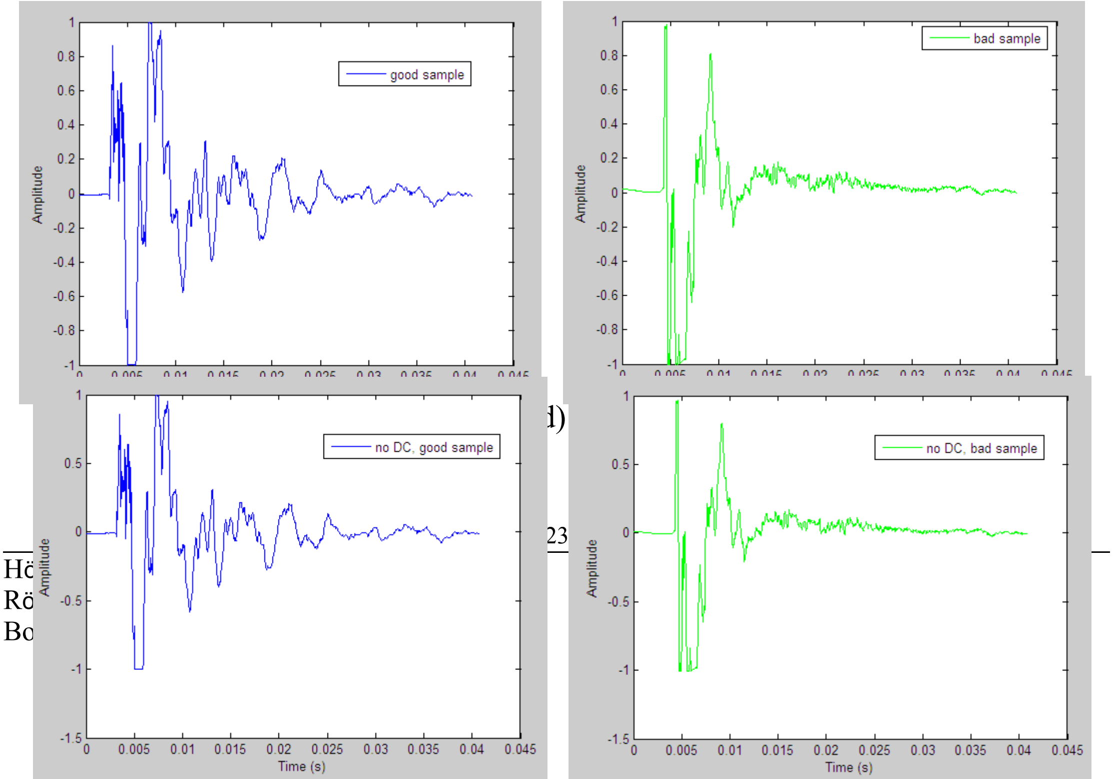
Firstly, the raw signal data in time domain was normalized to a peak value of 1. Then, the mean has been subtracted to remove any direct current (DC) component. The removal of DC component was performed in time domain for two reasons. The first reason is to prevent the effect of smearing from the frequency bin f=0 into neighbourig low-frequency bins. And the other is that presence of the DC component in the signal results in a high peak at frequency bin f=0 with a high spectral peak value. Since useful data have low spectral peak values in comparison with spectral peak of the DC, removing the DC component facilitates easy interpretation of useful peaks in the frequency domain [4].

Finally the data were tapered with a Hamming window using the standard hamming window equation

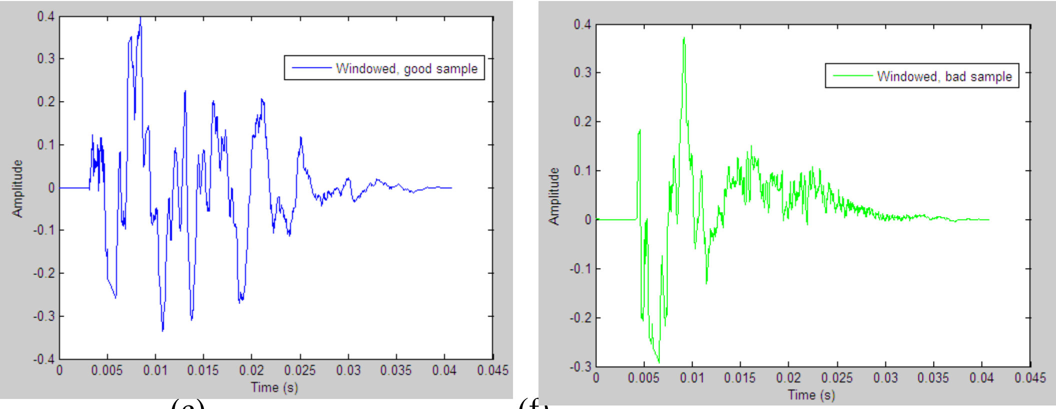
$$h(k) = 0.54 - 0.46\cos\left(2\pi k \big/ _{N-1}\right) \quad k = 1, \ldots, N \,. \tag{2}$$

A hamming window was selected since it gave good side-lobe suppression. Suppression of side-lobes is considered essential for avoiding ambiguity in detecting the important peaks during frequency analysis [4].

Preprocessing is illustrated on two sample acoustic signals of sleeper in good and bad conditions (Figure 2). Figure 2.a shows the raw signal from a good sleeper. Figure 2.b shows the effect of removing the DC component of that signal. Figure 2.c shows the result of applying a hamming window. Figure 2.d, e, f show the same steps for the signal from the bad sleeper. As seen, in time domain the waveform of the signals demonstrate the differences of the signals from a bad and a good sleeper.



(b) (e)



(c) (f)

Figure 2. Preprocessing steps on signals emitted from a good and a bad sleeper

3.3 Feature Extraction

Feature extraction is used to obtain the most relevant information from the original data to be able to represent the data compactly and efficiently. The goal of the feature extractor is to characterize an object to be recognized by measurements whose values are very similar for objects in the same category, and very different for objects in different categories. This leads to the idea of seeking distinguishing features that are invariant to irrelevant transformations of the input [24]. Feature extraction techniques are determined based on the nature of the data to be classified.

Feature extraction can be split into two types: stationary (frequency-based) feature extraction and non-stationary (time-frequency based) feature extraction. Stationary feature extraction produces an overall result detailing the frequencies contained in the entire signal. With stationary feature extraction, no distinction is made on where these frequencies occurred in the signal. In contrast, non-stationary feature extraction splits the signal up into discrete time units. This allows frequency to be identified as occurring in a particular area of the signal, aiding understanding of the signal [8].

The raw signals in practice, are time-domain signals. When we plot time-domain signals, we obtain a time-amplitude representation of the signal. This representation is not always the best representation of the signal for signal processing. In general, the frequency content of the signal has significant information. In other words, the information that cannot easily be seen in the time-domain can be seen in the frequency domain. The frequency spectrum of a signal shows what frequency components (spectral components) exist in the signal. The frequency content of a signal is obtained by applying Fourier Transform (FT). The FT does not provide when in time the frequency components exist, no time information is available in the Fourier transformed signal. This information is not required when the signal is stationary. Signals whose frequency content do not change in time are called stationary signals, meaning that all frequency components exist at all times therefore, there is no need to know at what times frequency components exist [25].

Accurate signal analysis and processing require meaningful representations of the signals involved. Since most real signals are typically nonstationary and their statistical characteristics are changing with time, time varying frequency content, traditional time or frequency analysis is not sufficient. The spectral nature of such signals can not be illustrated by a function, which depends only on one argument, frequency. Instead, the non-stationary signal has to be illustrated by a joint time-frequency representation [26]. So, a combined time-frequency representation is needed due to the inadequacy of either time domain or frequency domain analysis to describe the nature of non-stationary signals completely. Time-frequency representations are of great interest when analyzing and classifying acoustic signals. A time-frequency representation of a signal provides information about how the spectral content of the signal evolves with time to interpret non-stationary signals. One dimensional signal in the time domain is mapped into a two dimensional time-frequency representation of the signal by using four different techniques, Short Time Fourier Transform, Continuous Wavelet Transform, Discrete Wavelet Transform, Wigner-Ville Distribution. These techniques have been successfully used in speech recognition and music instrument recognition.

3.3.1 Frequency Analysis

In order to point out the properties of signals obtained from bad and good sleepers, frequency analysis is examined in this subsection.

The Fourier theory is based on the idea that any function can be composed of sines and cosines of different frequencies. In other words, any space or time varying data can be transformed into a different domain called the frequency space. For many signals, Fourier analysis is very useful because the frequency content of the signal is of great importance. Fourier transform (FT) of a function is a summation of sine and cosine terms of different frequency [27]. FT decomposes a signal to complex exponential functions of different frequencies, defined by the following [25]: (considering that exponential term can also be written as $cos(2\pi ft)+j\,sin(2\pi ft)$)

$$X(f)=\int_{-\infty}^{\infty} x(t).e^{-2j\pi ft}dt \tag{3}$$

where t stands for time, f stands for frequency, and x denotes the signal in time domain and the X denotes the signal in frequency domain. $X(f)$ is the Fourier transform of $x(t)$. So, $X(f)$' s are the data in the frequency space and its magnitude is called Fourier spectrum of $x(t)$. The Fourier spectrum is often plotted against values of f.

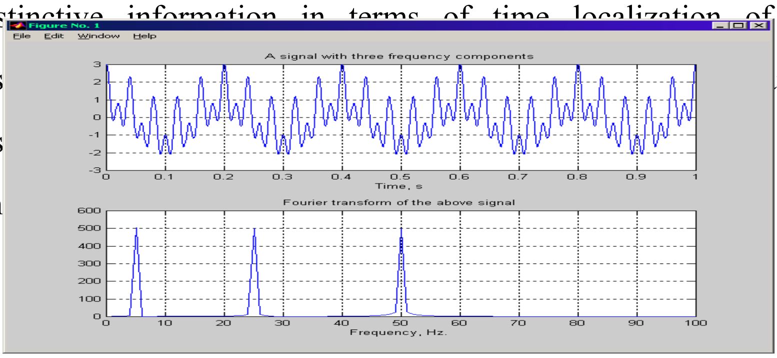
FT shows how much of each frequency exists in the signal (the spectral content), providing the frequency-amplitude representation of that signal. But time information is lost, it does not provide information about where in time the spectral components appear. Therefore, FT is not a suitable technique for nonstationary signal in terms of time localization of the spectral components. To illustrate this situation, two example signals, $x_4(t)$ (stationary) and $x_5(t)$ (nonstationary) are analyzed by FT below.

$$x_1(t) = cos(2\pi \cdot 5 \cdot t) \qquad x_2(t) = cos(2\pi \cdot 25 \cdot t) \qquad x_3(t) = cos(2\pi \cdot 50 \cdot t)$$

$$x_4(t) = cos(2\pi \cdot 5 \cdot t) + cos(2\pi \cdot 25 \cdot t) + cos(2\pi \cdot 50 \cdot t) \qquad x_5(t) = [x_1 \oplus x_2 \oplus x_3]$$

$\oplus$ Concatenation

The Figure 3 and 4 show the signals in time and the FT of the signals. As seen, they constitute of the same frequency components but for the nonstationary signal they occur at different times. FT results are nearly the same, the major peaks correspond to the same frequencies (the other peaks for the nonstationary signal appear because of the sudden change between the frequencies). But there is no information about where these frequencies are located in time, hence the FT can not provide distinctive information in terms of time localization of the spectral components me frequency components is not suitable for analyzin



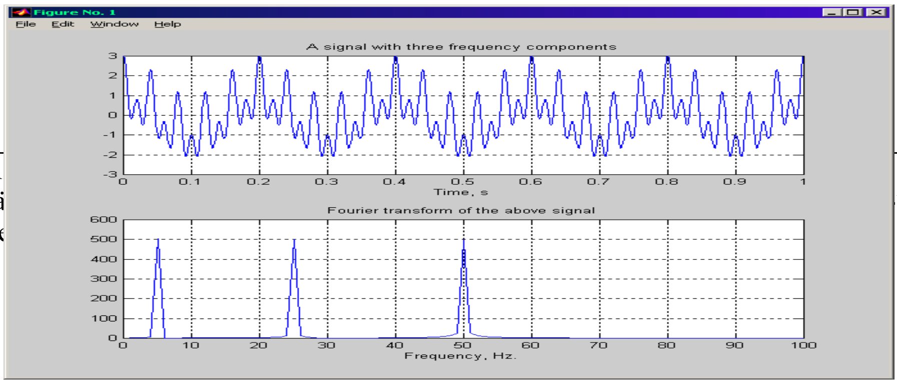


Figure 3. Signal $x_4(t)$ and its FT

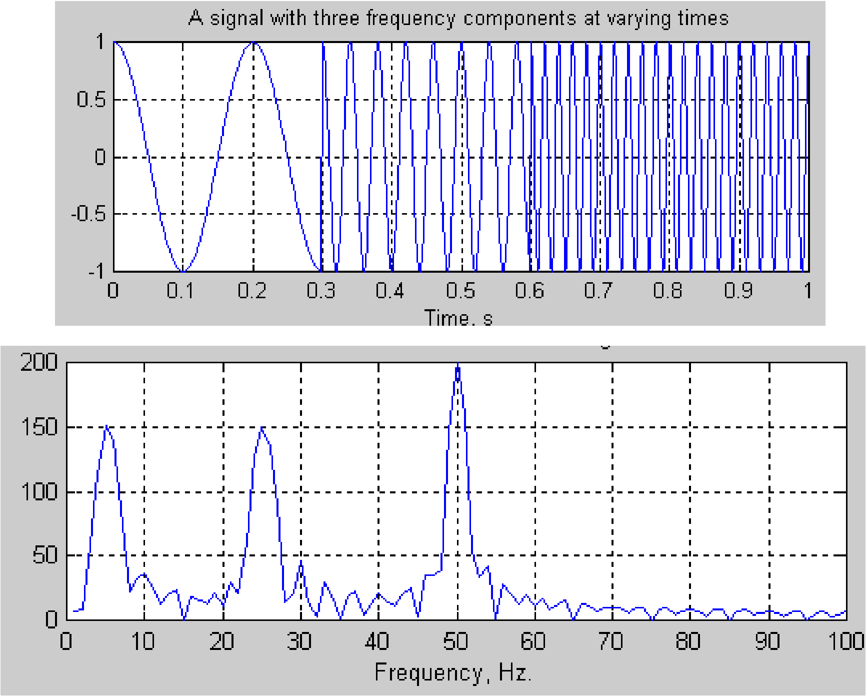


Figure 4. Signal $x_5(t)$ and its FT [28]

The FTs of two example signals from a bad and a good sleeper are given in the Figure 5 (Figure 6 shows the results by zooming into a small range of frequency). As seen, a resonance behavior in spectrum is available in case of the good sleeper.

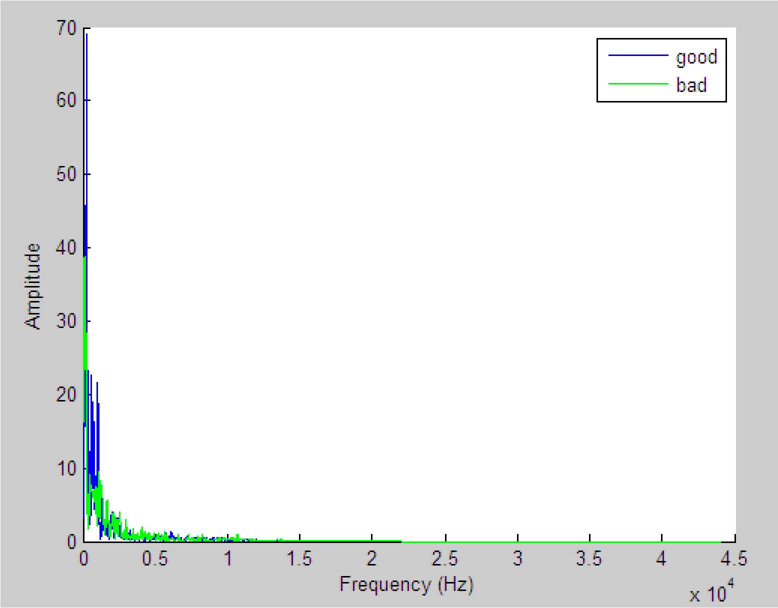


Figure 5. Frequency spectrum from good and bad wooden sleepers

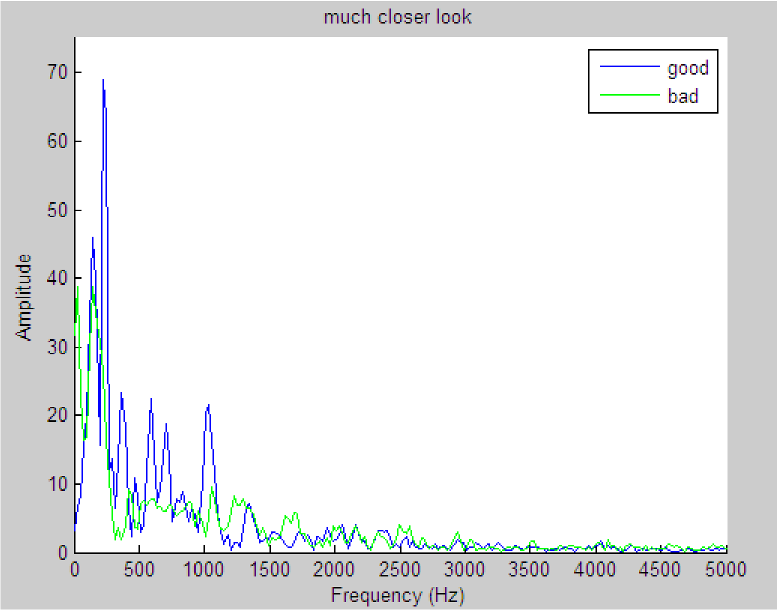


Figure 6. the results by zooming into a small range of frequency

3.3.2 Short Time Fourier Transform

To overcome the deficiency of FT which was mentioned, the Fourier transform is adapted to analyze only a small section of the signal at a time, *windowing* the signal. This adaptation, the *Short-Time Fourier Transform* (STFT), maps a signal into a two-dimensional function of time and frequency. It provides some information about both when and at what frequencies a signal event occurs [29].

In STFT, the signal is divided into small enough segments. These segments of the signal can be assumed to be stationary. Therefore, a window function is used. The width of this window must be equal to the segment of the signal where its stationarity is valid, windows should be narrow enough that the portion of the signal seen from these windows are indeed stationary. The window function is first located to the beginning of the signal. The window function and the signal are then multiplied. By doing this, only a part of the signal is being chosen, with the appropriate weighting of the window. Then this product is assumed to be just another signal, FT of this product is taken. After that, the window is shifted to a new location, multiplying with the signal, and taking the FT of the product. This procedure is followed, until the end of the signal is reached by shifting the window [25]. Because the window function has a short time duration, the FT of the product result reflects the signal’s local frequency properties. Finally a rough idea of how the signal’s frequency contents evolve over time is obtained. The following definition of the STFT summarizes the above explanations:

$$STFT_x^W(\Gamma,\omega)=\int_t [x(t).W(t-\Gamma)].e^{-j\omega t}dt \tag{4}$$

where x(t) is the signal to be processed, w(t) is the window function. As seen from the equation, the STFT of the signal is the FT of the signal multiplied by a window function, windowed version of the FT, resulting in a two-dimensional representation of the signal.

In order to obtain the stationarity, a short enough window is used, in which the signal is stationary. The width of the windowing function relates to how the signal is represented. It determines whether there is good frequency resolution (frequency components close together can be separated) or good time resolution (the time at which frequencies change). A wide window gives better frequency resolution but poor time resolution and wide windows may violate the condition of stationarity. A narrower window gives good time resolution and better the assumption of stationarity, but poor frequency resolution [25]. Too short window may miss lower frequencies while too long window may miss any frequency changes in time. If the frequency components are well separated from each other in the original signal, than we may sacrifice some frequency resolution and go for good time resolution [25]. Therefore, the information obtained through STFT is with limited precision which is determined by the size of the window [29]. Empirical testing showed that a window size of the sample frequency scaled by 100 produced the most accurate results [8]. The drawback of this approach is that the window is the same for all frequencies, the same window is used in the entire analysis. Many signals require a more flexible approach varying the window size to determine more accurately either time or frequency [29]. The trade-off between time and frequency resolution in the STFT has motivated a number of other time-frequency methods.

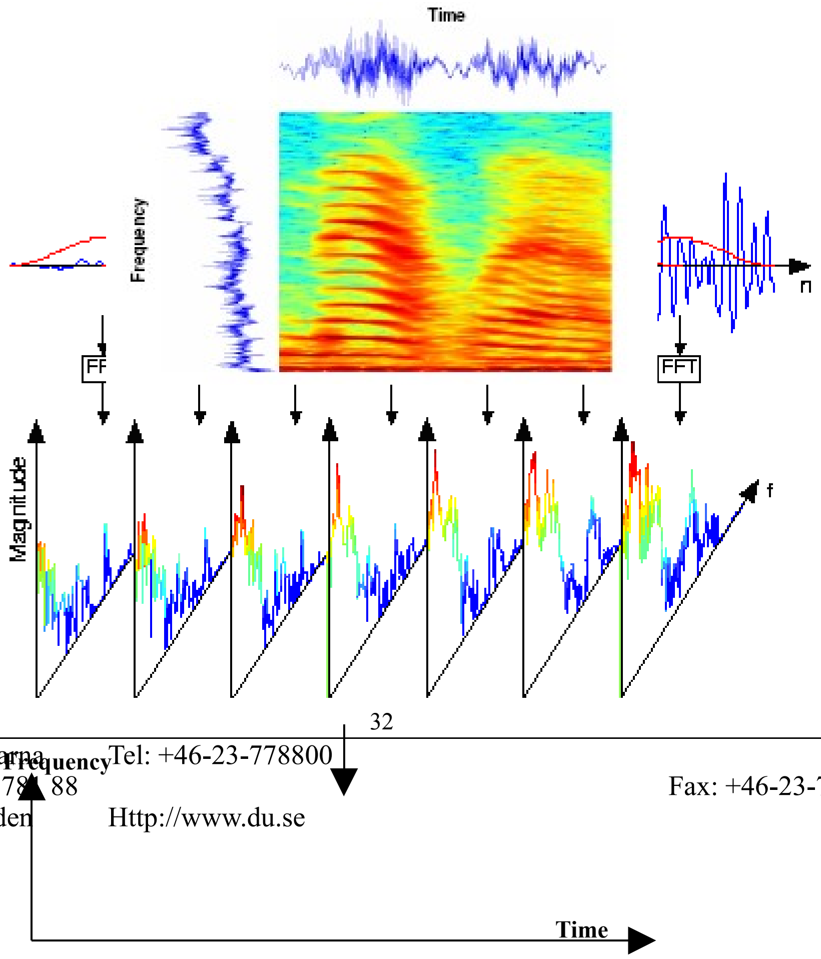
When discussing the joint time-frequency resolution, a time-frequency resolution rectangle, defined as $\Delta f \Delta t$ is widely used. The resolution rectangle satisfies the Heisenberg uncertainty principle [30]

$$\Delta f \Delta t \geq {}^{1}\!/_{4\pi} \tag{5}$$

which means that an increase in time resolution results in a decrease in frequency resolution, and vice versa [31] (Time resolution Δt : How well two spikes in time can be separated from each other in the transform domain; Frequency resolution Δf : How well two spectral components can be separated from each other in the transform domain) . Therefore, good resolutions both in the time and frequency domains cannot be achieved at the same time. We cannot precisely know at what time instance a frequency component is located. We can only know what interval of frequencies are present in which time intervals.

To sum up, this method yields which frequencies are present over the span of time defined by the window, computed by FFTs of overlapping windowed signal segments.

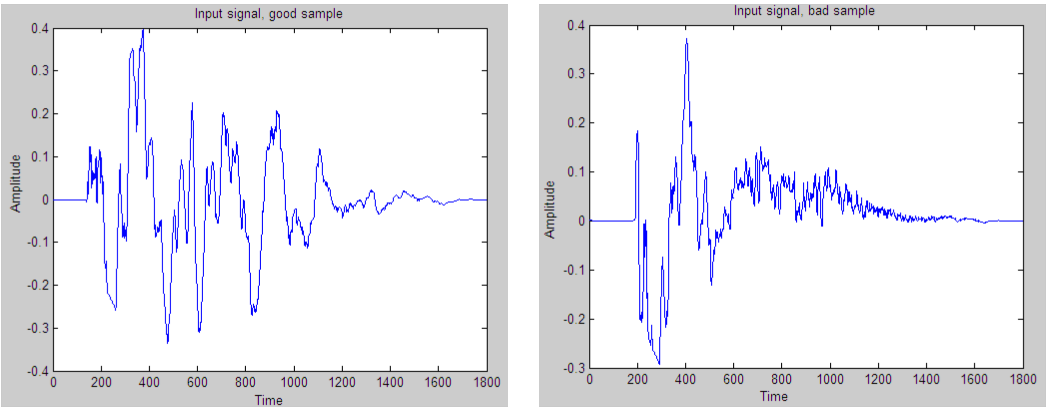
The magnitude of the Short Time Fourier Transform is called the spectrogram. 2 dimensional plots of the spectrogram with time on the horizontal axis, frequency on the vertical axis and amplitude given by a color can be made (an example in Figure 7). Alternately 3 dimensional plots where we plot amplitude on the third axis can be made. Often, the spectrogram uses dB as unit on the coloring.



Figı ned into overlap 33]. The spectra nto a 2 dimens

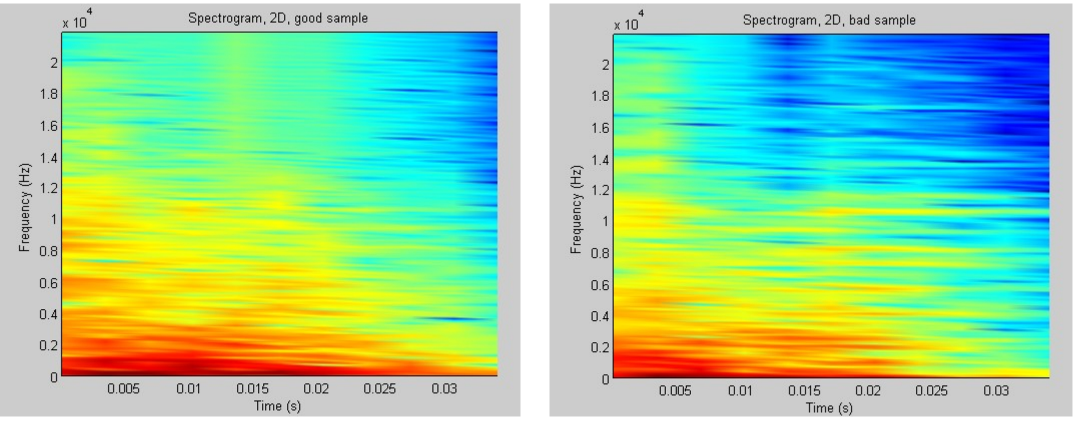
Figure 8 The way to compute STFT [33,34]

Figure 10 and 11 depicts the results of computing the STFT of preprocessed signals from a good and a bad sleeper in Figure 9 that are available in the data set used in the work. Results are displayed in spectrograms with frequency extending vertically, window time location running horizontally, and spectral magnitude color-coded. Frames were 300 samples long and a Hamming window was applied with a half-frame overlap.



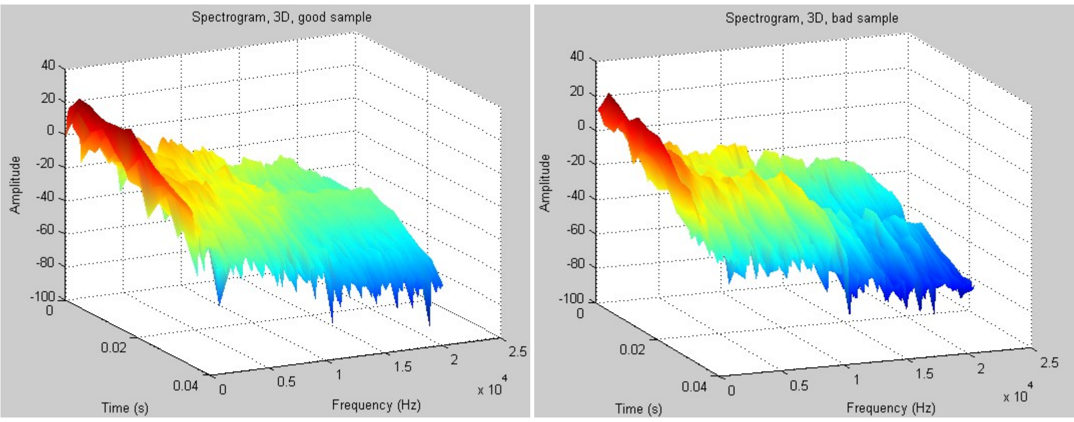
(a) (b)

Figure 9. Preprocessed signals from a good and a bad sleeper



(a) (b)

Figure 10. Spectrogram in 2D of signals from good and bad sleepers



(a) (b)

Figure 11. Spectrogram in 3D of signals from good and bad sleepers

3.3.3 Continuous Wavelet Transform

The continuous wavelet transform (CWT) is an alternative approach to STFT to overcome the resolution problem. The wavelet analysis is similar to the STFT analysis, in the way that the signal is multiplied with a function (the wavelet) like the window function in the STFT, and the transform is computed for different segments of the time domain signal[25].

As seen in the following equation, the continuous wavelet transform is defined as the sum over all time of the signal multiplied by scaled, shifted versions of the wavelet function. The results of the CWT are many wavelet coefficients C, which are functions of position and scale, τ and **s**. Multiplying each coefficient by the appropriately scaled and shifted wavelet yields the constituent wavelets of the original signal[29]. $\psi(t)$ is the transforming function called the mother wavelet.

$$CWT_x^{\psi}(\tau,s)=\frac{1}{\sqrt{|s|}}\int x(t)\psi^*\left(\frac{t-\tau}{s}\right)dt \tag{6}$$

The result is multiplied by the constant number $\frac{1}{\sqrt{s}}$ for energy normalization purpose so that the transformed signal will have the same energy at every scale. The position is related to the location of the window, corresponding to time information, as the window is shifted through the signal. The term wavelet means a small wave, implying that this (window) function is of finite length, the wave implies that this function is oscillatory and the term mother implies that other window functions are derived from one main function, the mother wavelet [25].

A wavelet is a waveform of effectively limited duration that has an average value of zero. Comparing wavelets with sine waves, which are the basis of Fourier analysis, sinusoids do not have limited duration, they extend from minus to plus infinity. And where sinusoids are smooth and predictable, wavelets tend to be irregular and asymmetric [29].

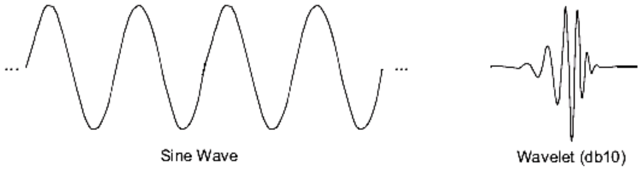


Figure 12 example wavelet [29]

Wavelet analysis is the breaking up of a signal into shifted and scaled versions of the original (or mother) wavelet. Intuitively, it is seen from the above figure (Figure 12) that signals with sharp changes might be better analyzed with an irregular wavelet than with a smooth sinusoid, and also local features can be described better with wavelets that have local extent[29].

All the windows that are used are the dilated (or compressed) and shifted versions of the mother wavelet. Scaling a wavelet indicates stretching or compressing it. Larger scales correspond to stretched wavelet and small scales correspond to compressed wavelet. The more stretched the wavelet, the longer the portion of the signal with which it is being compared, and thus the coarser the signal features being measured by the wavelet coefficients (slowly changing coarse features, low frequency). Smaller scales correspond to higher frequencies, frequency decreases as scale increases. So, the scale is related to the frequency of the signal [25,29].

Once the mother wavelet is chosen the computation starts with first scale and the continuous wavelet transform is computed for all values of **s**. Generally the signals are band-limited therefore, computation of the transform for a limited interval of scales is usually adequate. The procedure starts from the smallest scale **(**high frequency**)** and continues for the increasing values of **s (**low frequencies**)**. This first value of **s** will correspond to the most compressed wavelet. As the value of **s** is increased, the wavelet will dilate. By shifting the wavelet in time, the signal is localized in time, and by changing the value of **s** , the signal is localized in scale (frequency) [25].

The wavelet compared to a section at the start of the original signal. A coefficient is calculated representing how closely correlated the wavelet is with the present section of the signal. Higher coefficient means more similarity. Then the wavelet is shifted to the right and another coefficient is calculated, this is repeated until the whole signal is covered. Finally the wavelet is scaled (stretched) and all the steps are repeated for the new scale. Scaling and shifting is illustrated in Figure 13. When calculations are performed for all scales, coefficients produced at different scales by different sections of the signal are obtained [29].

The definition of the CWT shows that the wavelet analysis is a measure of similarity between the basis functions (wavelets) and the signal itself. The similarity is based on frequency content. The calculated CWT coefficients refer to the closeness of the signal to the wavelet at the current scale. If the signal has a major component of the frequency corresponding to the current scale, then the wavelet at the current scale will be similar to the signal at the particular location where this frequency component occurs. Therefore, the CWT coefficient computed at this point in the time-scale plane will be a relatively large number. As the window width increases, the transform starts picking up the lower frequency components [25].

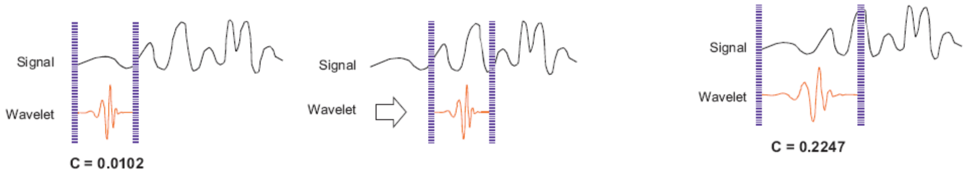


Figure 13 (a). the beginning, (b). first shift, (c). the beginning with the second scale [29]

When the process is completed for all desired values of **s**, the CWT of the signal has been calculated. As a result, for every scale and for every time (interval), one point of the time-scale plane is computed. The computations at one scale construct the rows of the time-scale plane, and the computations at different scales construct the columns of the time-scale plane[25].

Unlike the STFT which has a constant resolution at all times and frequencies, the WT has a good time and poor frequency resolution at high frequencies, and good frequency and poor time resolution at low frequencies. The illustration in Figure 14 is commonly used to explain how time and frequency resolutions should be interpreted. Every box in the figure corresponds to a value of the wavelet transform in the time-frequency plane. All the points in the time-frequency plane that falls into a box is represented by one value of the WT [25].

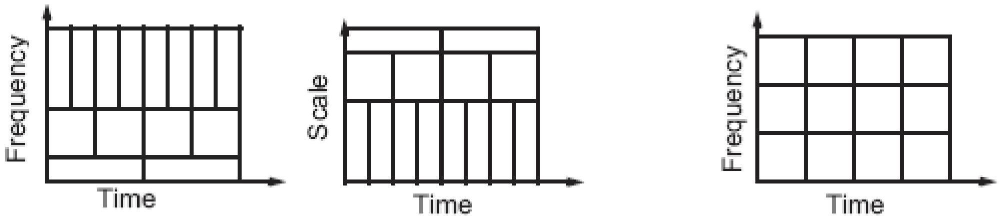


Figure 14. Resolution of WT and STFT [29,25]

Although the widths and heights of the boxes change, the area is constant (determined by Heisenberg's inequality, all areas are lower bounded by 1/4π). So, each box represents an equal portion of the time-frequency plane, but giving different proportions to time and frequency. At low frequencies, the height of the boxes are shorter (which corresponds to better frequency resolutions, since there is less ambiguity regarding the value of the exact frequency), but their widths are longer (which correspond to poor time resolution, since there is more ambiguity regarding the value of the exact time). At higher frequencies the width of the boxes decreases, the time resolution gets better, and the heights of the boxes increase, the frequency resolution gets poorer. In STFT the time and frequency resolutions are determined by the width of the analysis window, which is selected once for the entire analysis, both time and frequency resolutions are constant. Therefore the time-frequency plane consists of squares in the STFT [25].

What distinguishes CWT from the discrete wavelet transform is the set of scales and positions at which it operates. Unlike the discrete wavelet transform, the CWT can operate at every scale, from that of the original signal up to some maximum scale that is determined by trading the need for detailed analysis with available computational power. The CWT is also continuous in terms of shifting, the analyzing wavelet is shifted over the full domain of the analyzed function [29].

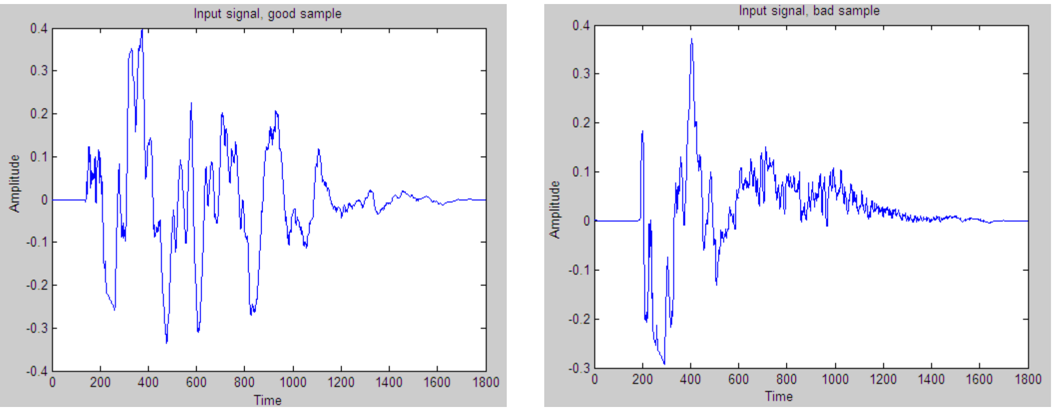
For the CWT, the discretized CWT algorithm from Matlab’s toolbox was used and Morlet mother wavelet [35] was chosen to be used in the work and it is defined as

$$\psi(t) = e^{jat} e^{\frac{-t^{٢}}{٢s}} \tag{7}$$

where a is a modulation parameter and s again represents scale. This mother wavelet has been used for recognition tasks and produced acceptable results [36, 37].

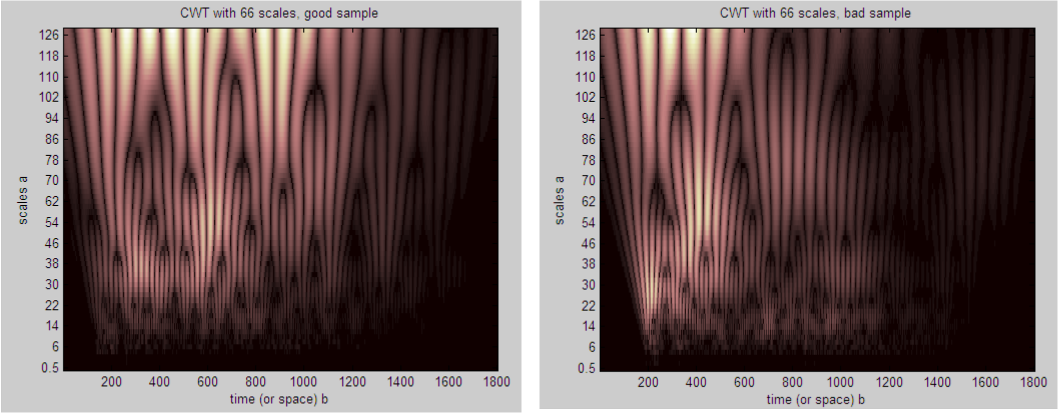
On the plot providing the time-scale view of the signal, *x*-axis represents position along the signal (time), the *y*-axis represents scale, and the color at each *x-y* point represents the magnitude of the wavelet coefficient C. These coefficient plots were generated by the graphical tools in Matlab. Inspection of the CWT coefficients plot for this signal reveals patterns among scales [29].

Below, results of applying CWT with 66 scales ([0.5,1,2,...,128]) to preprocessed signals from a good and a bad sleeper in Figure 15 are seen indicating the difference characteristics at different frequencies and times (Figure 16).



(a) (b)

Figure 15. Preprocessed signals from a good and a bad sleeper



(a) (b)

Figure 16. CWT with 66 scales of signals from good and bad sleepers

3.3.4 Discrete Wavelet Transform

By using scales and positions based on powers of two, dyadic scales and positions, the wavelet analysis become more efficient and just as accurate rather than calculating wavelet coefficients at every possible scale and also the amount of data and work are reduced in this way. Such an analysis is obtained from the discrete wavelet transform (DWT). An efficient way to implement this scheme using filters was developed in 1988 by Mallat [37] by passing the signal through a series of low-pass and high-pass filter pairs.

In the filtering process, the original signal, S, passes through two complementary filters and emerges as two signals (Figure 17)[29]. So, DWT enables decomposition of the input signal into two signals - Approximation A (The high-scale, low-frequency components of the signal) and Details D (low-scale, high-frequency component). Details are obtained when the signal is passed through the half band high-pass filter, (impulse response represents the wavelet function) Approximation is obtained if the signal is passed through the half band low-pass filter.

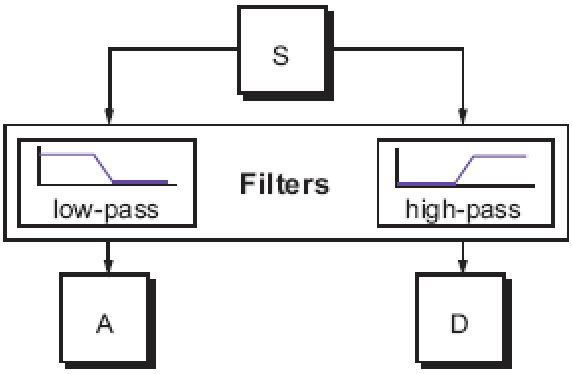


Figure 17. Filtering in wavelet analysis [29]

The main idea is the same as it is in the CWT. A time-scale representation of a digital signal is obtained using digital filtering techniques. The CWT is a correlation between a wavelet at different scales and the signal with the scale (or the frequency) used as a measure of similarity. The continuous wavelet transform was computed by changing the scale of the analysis window, shifting the window in time, multiplying by the signal, and integrating over all times. In the discrete case, filters of different cutoff frequencies are used to analyze the signal at different scales. The signal is passed through a series of high pass filters to analyze the high frequencies, and it is passed through a series of low pass filters to analyze the low frequencies[25].

The procedure starts with passing the signal through a half band digital lowpass filter with impulse response h[n]. Filtering a signal corresponds to the mathematical operation of convolution of the signal with the impulse response of the filter. The convolution operation in discrete time is defined as follows [25]:

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k].h[n-k] \tag{8}$$

A half band lowpass filter removes all frequencies that are above half of the highest frequency in the signal. After passing the signal through a half band lowpass filter, half of the samples can be eliminated according to the Nyquist’s rule (subsampling the signal by two, since half of the number of samples are redundant). The scale of the signal is then doubled (because of subsampling). Resolution is related to the amount of information in the signal, it is halved after the filtering operation.

The subsampling (downsampling) operation after filtering does not affect this resolution, since removing half of the spectral components from the signal makes half the number of samples redundant. Half the samples can be discarded without any loss of information. This procedure can mathematically be expressed as [25]

$$y[n] = \sum_{k=-\infty}^{\infty} h[k].x[2n-k] \tag{9}$$

The resolution of the signal, which is a measure of the amount of detail information in the signal, is changed by the filtering operations, and the scale is changed by subsampling operation. Subsampling a signal corresponds to reducing the sampling rate, or removing some of the samples of the signal. For example, subsampling by two refers to dropping every other sample of the signal. Subsampling by a factor *n* reduces the number of samples in the signal *n* times.

The DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a coarse approximation and detail information. The decomposition of the signal into different frequency bands is simply obtained by successive highpass and lowpass filtering of the time domain signal. The original signal x[n] is first passed through a halfband highpass filter g[n] and a lowpass filter h[n]. After the filtering, half of the samples can be eliminated according to the Nyquist's rule, since the highest frequency of the signal is halved. The signal can therefore be subsampled by 2, simply by discarding every other sample. This constitutes one level of decomposition and can mathematically be expressed as follows:

$$y_{high}[k] = \sum_{n} x[n].g[2k-n] \tag{10}$$

$$y_{low}[k] = \sum_{n} x[n].h[2k-n] \tag{11}$$

where $y_{high}[k]$ and $y_{low}[k]$ are the outputs of the highpass and lowpass filters, respectively, after subsampling by 2.

The process produces DWT coefficients, two sequences called cA and cD in Figure 18 [29].

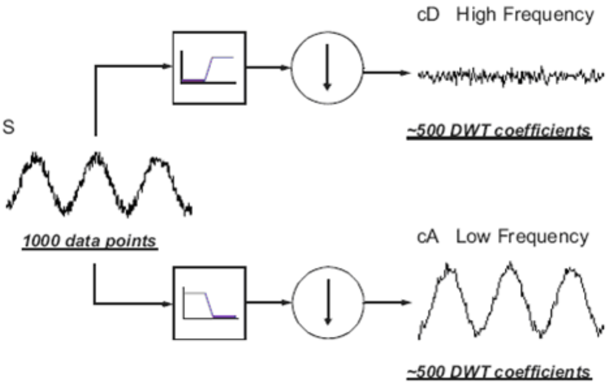


Figure 18. DWT process [29]

This decomposition halves the time resolution since only half the number of samples now characterizes the entire signal and doubles the frequency resolution, since the frequency band of the signal now spans only half the previous frequency band, effectively reducing the uncertainty in the frequency by half. The procedure can be repeated for further decomposition (the lowpass filter output is filtered again) [25]. At each step of the decomposition process, the frequency resolution is doubled through filtering and the time resolution is halved through subsampling [38]. So one signal is broken down into many lower resolution components [29]. This is called the wavelet decomposition tree. Figure 19 illustrates this procedure.

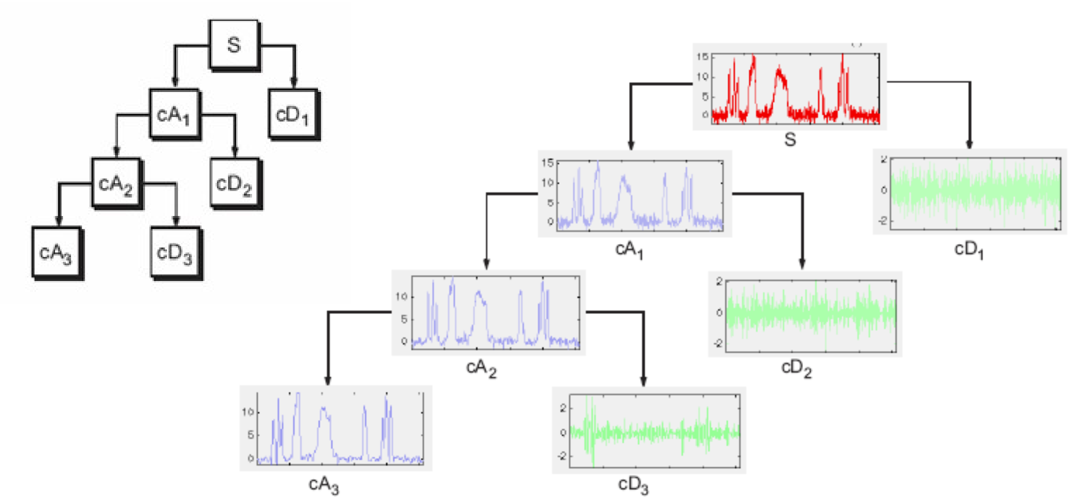


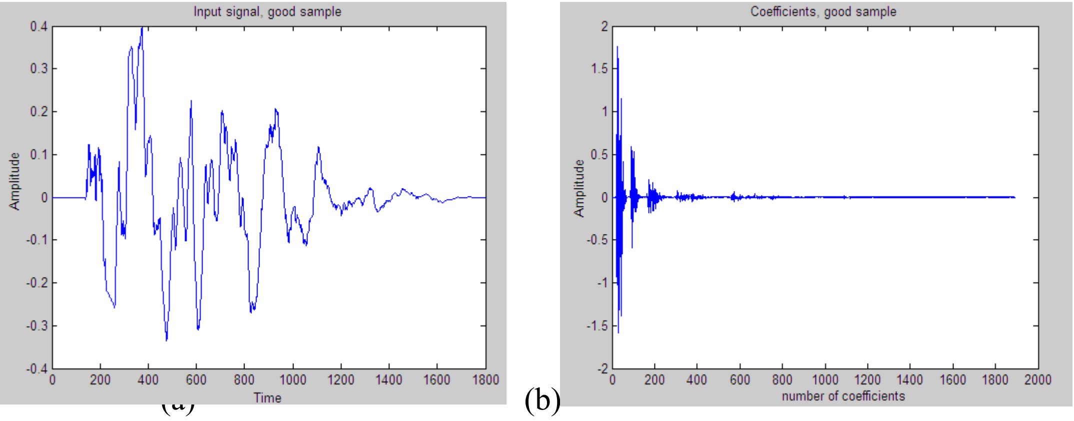
Figure 19. Wavelet decomposition [29]

The detail coefficients cD consist mainly of a high-frequency noise, while the approximation coefficients cA contain much less noise than does the original signal [29]. The DWT of the original signal is obtained by concatenating all coefficients starting from the last level of decomposition [25].

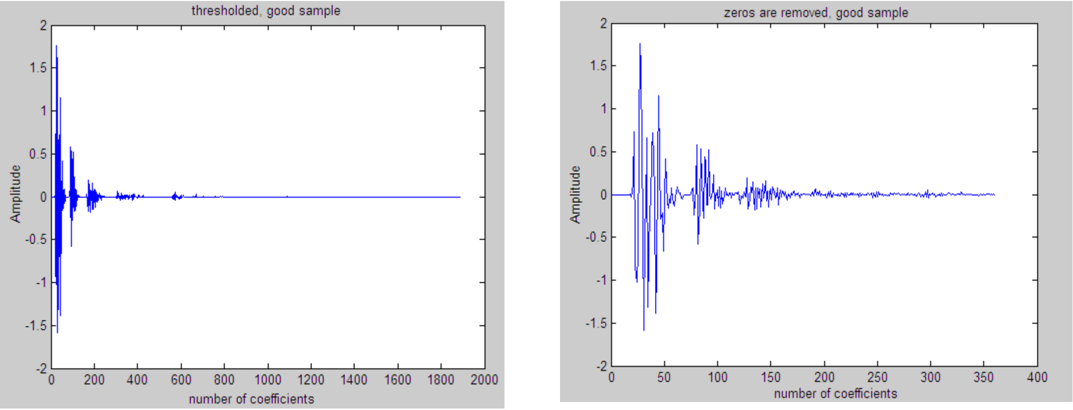
Carrying out the decomposition not only on the lowpass side but on both sides can also be applied, namely zooming into both low and high frequency bands of the signal separately, known as the wavelet packages which can be visualized as having both sides of the tree structure. For many signals, the low-frequency content is the most important part, while the high-frequency content imparts flavor or nuance, so zooming into low frequency bands is generally enough.

The frequencies that are most prominent in the original signal will appear as high amplitudes in that region of the DWT signal that includes those particular frequencies. the time localization of these frequencies will not be lost. However, the time localization will have a resolution that depends on which level they appear. If the main information of the signal lies in the high frequencies, the time localization of these frequencies will be more precise, since they are characterized by more number of samples. If the main information lies only at very low frequencies, the time localization will not be very precise, since few samples are used to express signal at these frequencies. This procedure offers a good time resolution at high frequencies, and good frequency resolution at low frequencies[25].

The frequency bands that are not very prominent in the original signal will have very low amplitudes, and that part of the DWT signal can be discarded without any major loss of information, allowing data reduction. Figure 20 and 21 illustrate examples of how how data reduction is provided. Figure 20.a shows a preprocessed signal from a good sleeper. The horizontal axis is the number of samples, whereas the vertical axis is the amplitude. Figure 20.b shows the 5 level DWT of the signal (with db10) in. The last samples in this signal correspond to the highest frequency band in the signal first coeffcieints carry relevant information and the rest has virtually no information. Therefore, thresholding can be applied (Fig 20.c) and small values can be discarded without any loss of information. In Fig 20.d, zero values after thresholding are removed, in this result, even the last values which are relatively small than the first ones can be discarded. The same process is seen for a preprocessed signal from a bad sleeper in Figure 21. from This is how DWT provides a very effective data reduction scheme [25].

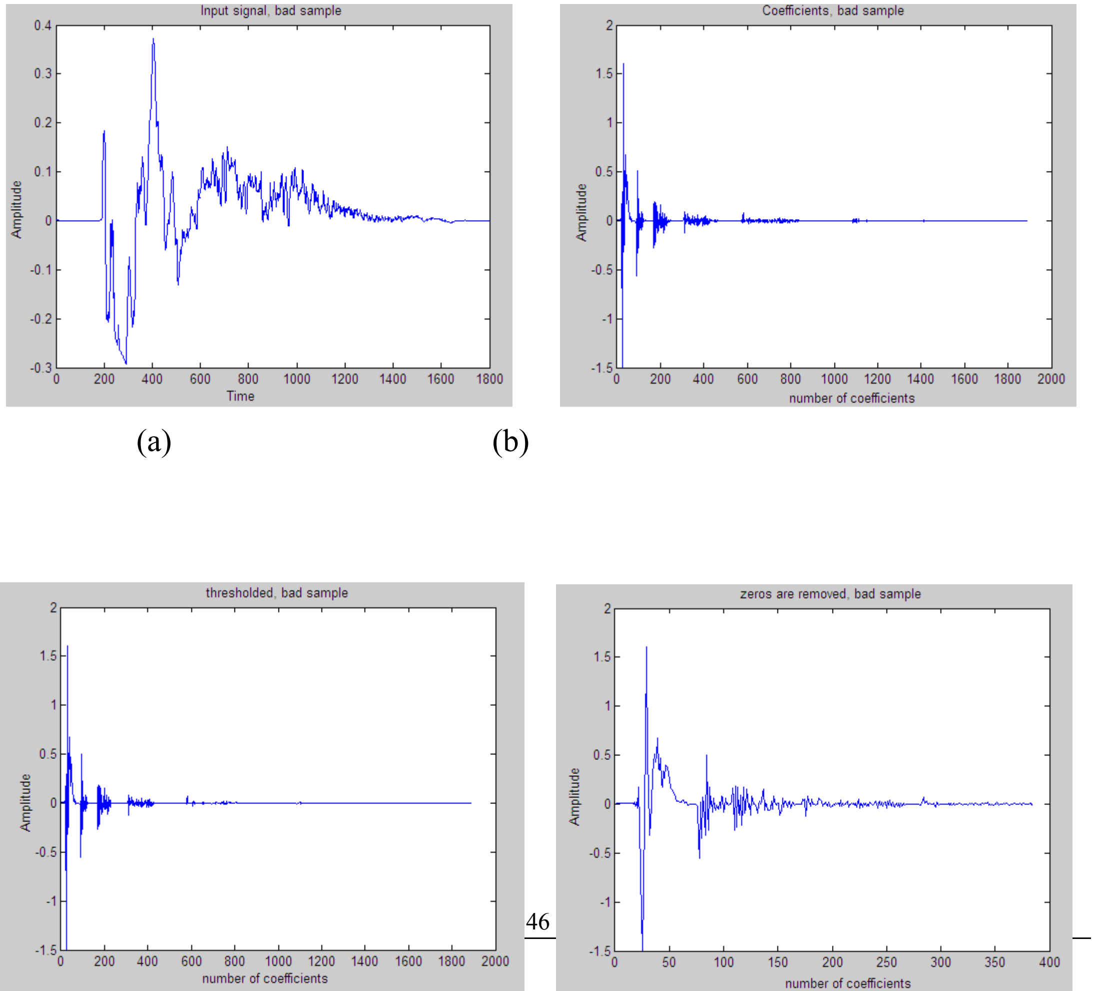


(a) (b)



(c) (d)

Figure 20. DWT of a signal from a good sleeper



(a) (b)

(c) (d)

Figure 21. DWT of a signal from a bad sleeper

The type of mother wavelet chosen for the analysis and the number of the levels of decomposition are the parameters to be defined. In the work the decomposition was based on db10 (Daubechies 10) wavelet and 5 levels of analysis. As filters (h and g) the Daubechies [39] filters were applied to the signal. Among the several wavelet functions that were mentioned in the literature, the Daubechies family of wavelets are the most widely used. The family of Daubechies wavelets was chosen because it consists of biorthogonal, compactly supported wavelets, satisfactorily regular and not symmetrical. These attributes were considered very important for the analysis of transient signals [40]. Daubechies filters allow for the perfect reconstruction of a signal from the DWT [8]. The number of vanishing moments of a wavelet indicates the smoothness of the wavelet function as well as the flatness of the frequency response of the wavelet filters (filters used to compute the DWT) [41]. A vanishing moment variable can be set for the filters; however the value of this coefficient seemed to make little difference to the classification rate [8]. A suitable criterion used by [42] for selecting optimal wavelets, is the energy retained in the first N/2 coefficients. Based on this criterion alone the Daubechies 10 wavelet preserves perceptual information well [43]. Additionally, the db10 is a very good compromise of smooth function, without sharp edges and not too difficult to create numerically [40].

The decomposition can only proceed until the individual details consist of a single sample. The length of the signal determines the number of levels that the signal can be decomposed to. However there is little or no advantage gained in decomposing a signal beyond a certain level (less is insufficient and more is redundant). Choosing a decomposition level for the DWT usually depends on the type of signal being analysed or some other suitable criterion such as entropy [43] to determine if a decomposition is sufficient or more levels are needed. For each node is calculated entropy, based on the values of coefficients belong to that node. If the entropy of the originating node is less than the sum of entropies of successor nodes, decomposition of that node is not performed. By pruning the tree with respect to the entropy criteria, the best tree is obtained [44, 45]. Based on this calculation on the data set, on average 5 levels was found suitable for the decomposition. The number of levels within the decomposition depends on both the size of the data and the resolutions of interest. Increasing the number of levels to a large number has an effect of creating very low frequency DC-like waveforms in the highest scales and do not tell anything useful. On the other hand having a very low number of levels in the decomposition, would not give the decomposition sufficient frequency resolution [46]. The fact that the decomposition up to scale 5 is adequate [42], with no further advantage gained in processing beyond scale 5 is mentioned in many sources for signal processing [40, 43, 46, 47].

3.3.5 Wigner-Ville Distribution

As one of the methods enabling simultaneous signal analysis in time and frequency domain, the Wigner-Ville distribution has been drawing a lot of attention lately. This distribution was first introduced by E. Wigner in the context of quantum mechanics [48], and later independently developed by J. Ville who applied the same transformation to signal processing and spectral analysis [49, 50]. A Wigner-Ville function is derived from computing the Fourier transform of a correlation function [51]. Wigner Ville Distribution (W) of signal x (*t*) is given by the following

$$W(t,\omega) = \int_{-\infty}^{+\infty} x\left(t+\frac{\tau}{2}\right) x^{*}\left(t-\frac{\tau}{2}\right) e^{-j\omega\tau}\, d\tau \tag{12}$$

where t,ω,τrepresents the time, the angular frequency, the time delay and $\left[x\left(t+\frac{\tau}{٢}\right)x^{*}\left(t-\frac{\tau}{٢}\right)\right]$is the instantaneous autocorrelation function. From the above equation, the Wigner-Ville distribution is regarded as Fourier transform for the time delay τ of $\left[x\left(t+\frac{\tau}{2}\right)x^{*}\left(t-\frac{\tau}{2}\right)\right]$which is a time-dependent autocorrelation function, and represents the distribution of power on the time-frequency plane. In order to calculate the Wigner-Ville distribution from a signal data of limited record length, an approximate discrete value of the equation must be calculated [52]. The complex conjugate (indicated by *) is introduced to generalise the analysis to complex signals. The result is a function of both frequency and time [53]. The Wigner–Ville distribution provides a high resolution of instantaneous energy density both in time and frequency domains [54].

J. Ville [49] proposed the use of the analytic signal in time-frequency representations of a real signal. An analytic signal is a complex signal which contains both real and imaginary components. The advantage of using the analytic signal is that in the frequency domain the amplitude of negative frequency components are zero. This satisfies mathematical completeness of the problem by accounting for all frequencies, yet does not limit the practical application since only positive frequency components have a practical interpretation. The imaginary part is obtained by Hilbert transform [55]. If x(t) is a real signal, the analytic signal is defined as follows:

$$\bar{x} = x(t) + jx_{ht}(t) \qquad (13)$$

where $x_{ht}(t)$ is the Hilbert transform of $x(t)$, which is shown as [56]

$$x_{ht}(t) = \frac{1}{\pi}\int_{-\infty}^{+\infty} x(\tau)\frac{1}{t-\tau}d\tau \tag{14}$$

The Wigner-Ville distribution has high time-frequency resolution. A major drawback of the Wigner-Ville distribution, reported by Cohen [57], is that this distribution propagates noise. It has been shown that if there is noise present in a small section of the signal, it will appear again within the distribution. This effect is a general property of the Wigner-Ville distribution, and is related to the interference caused by cross-terms which appear when the cross-correlation of the two signals is non-zero. In this case, part of the data of one shift is repeated in the following one, causing redundant information. To reduce this problem, windows can be applied in the time and frequency domains, and it has then been known as the 'pseudo-Wigner-Ville' distribution (PWVD)[57,58,59].

An analytic signal is a complex signal that contains only positive frequencies. It is associated with a real signal by the removal of the real signals negative frequencies and doubling the value of its positive frequencies [60]. Since the spectral domain is divided by two, the number of interference terms decreases (If the real signal is used then both the positive and negative spectral terms produce them). The interferences are caused by the fact that the Wigner-Ville Distribution is quadratic in *x*, so if *x* is a sum *(a + b)*, the Wigner-Ville distribution of *x* contains an interference term *2ab* in addition to the desired value *(a2 + b2).* Normally, if we have two points on the diagram, we will receive interferences in the middle of the distance between them. To avoid this inconvenience a suitable choice of smoothing factors can be applied which can dramatically reduce the interference terms by smoothing in time and frequency [61, 62, 63].

Additionally, for real-time computations or for long time series, the computation time for WVD is an important practical problem [51,64].

Below, on preprocessed signals from a good and a bad sleeper in Figure 22, Wigner-Ville distribution results can be seen with real (Figure 23.a, 23.c) and analytic signal (Figure 23.b, 23.d).

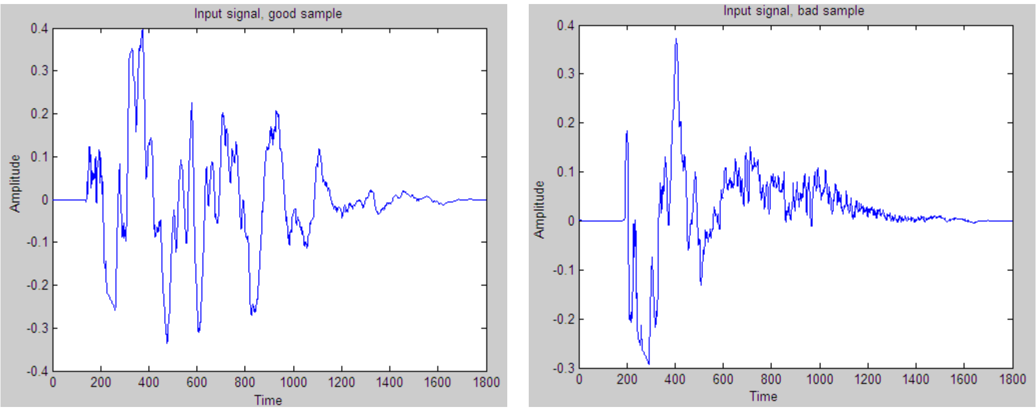
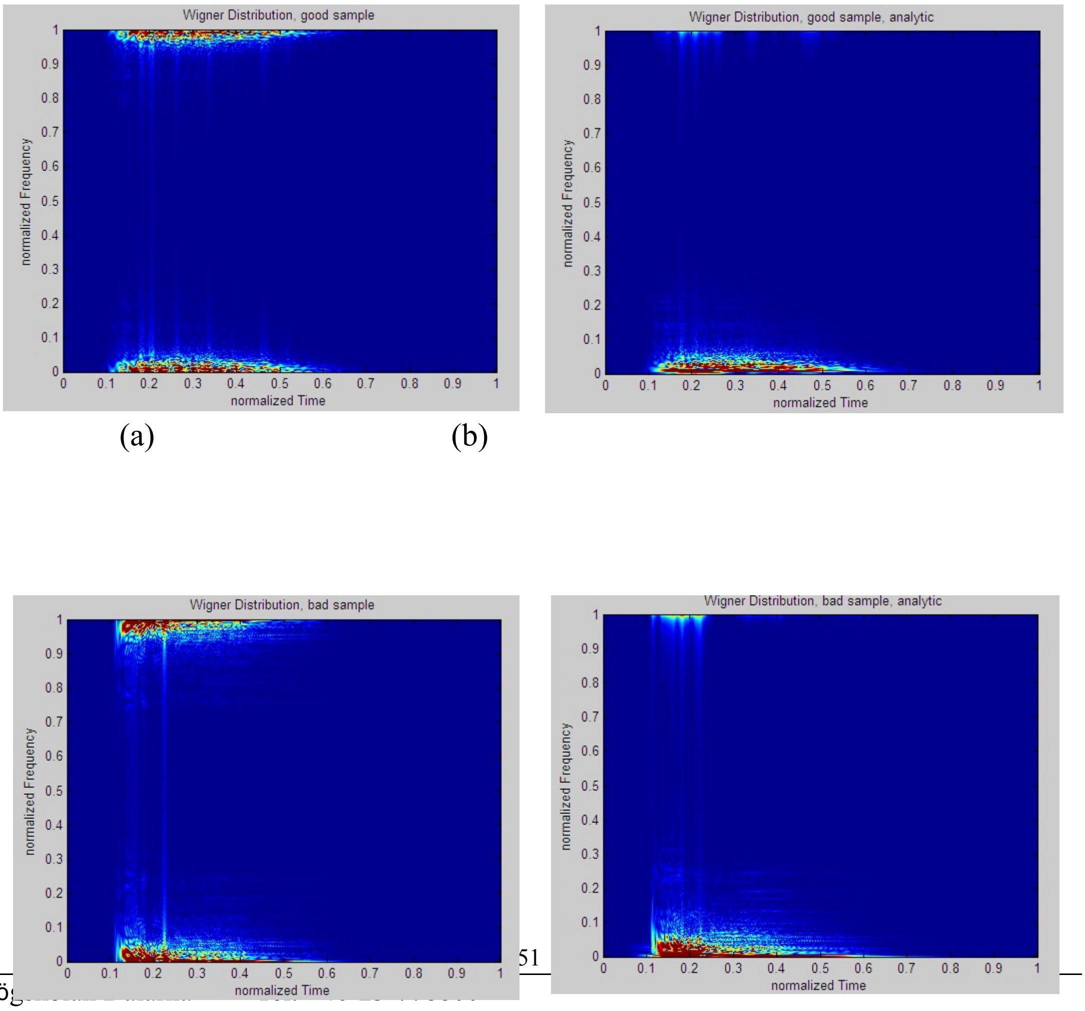


Figure 22. Preprocessed signals from a good and a bad sleeper



(a) (b)

(c) (d)

Figure 23. WVD of signals from a good an a bad sleeper

3.4 Dimension Reduction using Principal Component Analysis

PCA [65, 66, 67, 68] provides an orthogonal projection basis leading to dimensionality reduction and feature selection. PCA involves the calculation of the eigenvalue decomposition of a data covariance matrix, usually after mean centring the data for each attribute. The data to which PCA is to be applied contains the extracted features. The data matrix is constructed by placing feature vectors into the columns. Namely, if there are M samples of size N, then the matrix X has the size of N*M,

$$X = [\, x_1\, x_2 \ldots x_M],\ x_i = [\, d_1\, d_2 \ldots d_N]^T,\ i=1,2,\ldots,\ M. \qquad (15)$$

The data, *Q,* is prepared for the covariance matrix calculation by subtracting the mean,

$$avr = \frac{1}{M} * \sum_{i=1}^{M} x_i\ ,\quad q_i = x_i - avr. \qquad (16)$$

The data, *Q,* is to be transformed by using the eigenvectors of the covariance matrix of the data. Since the size of the covariance matrix*, $Cov(Q) = QQ^T$* , is N*N and the number of eigenvectors of this matrix is N, generating these eigenvectors, *(u_i)*, has a high computational burden

$$Cov(Q) = \frac{1}{M} * \sum_{i=1}^{M} q_i q_i^{\,T} = QQ^T,\ (N{*}N). \qquad (17)$$

If the number of samples is less than the dimesion of the data (M<N), there will be only M-1 meaningful eigenvectors. The remaining eigenvectors will have associated eigenvalues of zero [66]. Therefore, the eigenvectors are derived from the eigenvectors of the matrix Q^TQ,

$Q^TQ\ v_i = \mu_i\ v_i$, where v_i is the eigenvectors and μ_i is the eigenvalues of Q^TQ.

The size of Q^TQ is M*M and it has M eigenvectors with M components. When both sides of the equation are multiplied by Q,

$$Q\ Q^TQ\ v_i = \mu_i\ Qv_i \tag{18}$$

is obtained where $u_i = Qv_i$ is the eigenvector of QQ^T (Qv_i is normalized in order to have $||\ Qv_i\ ||= 1$) and μ_i is the eigenvalue. The data is to be expressed in terms of these eigenvectors. Eigenvectors are ordered by eigenvalue from highest to lowest to get the components in order of significance. $x<<M<<N$ eigenvectors are chosen and the data is transformed by using them,

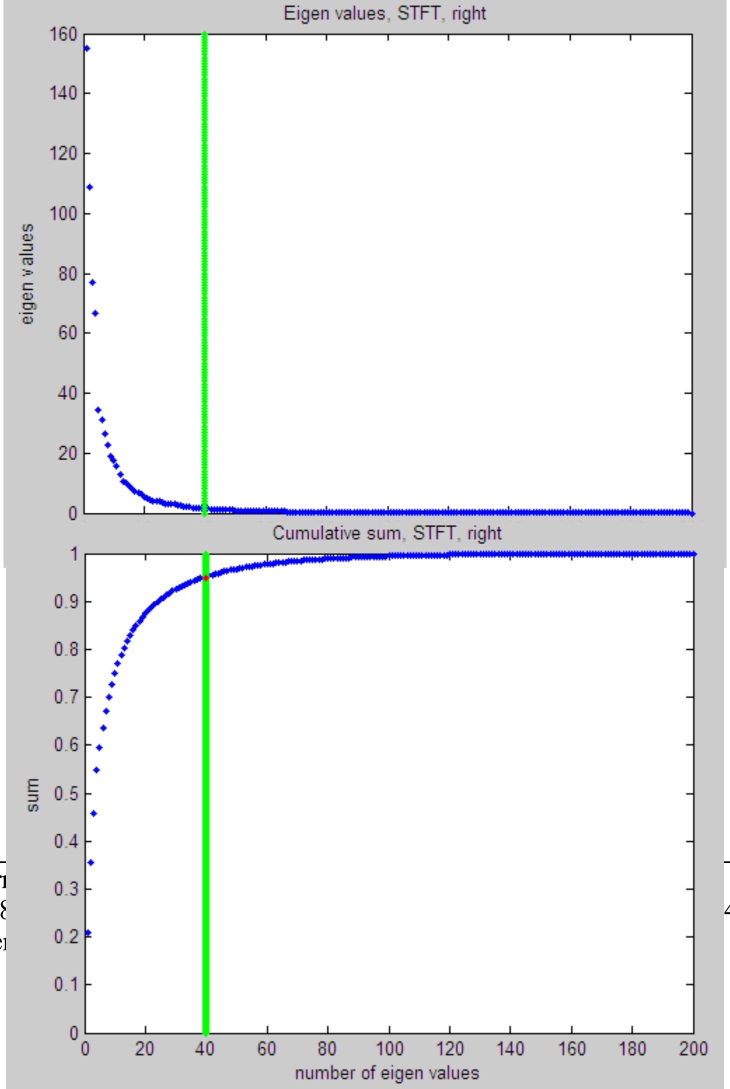
$$y_i = u_i^Tq_j\ ,\ i=1,\ \ldots\ ,\ x;\ \ j=1,\ldots,\ M,\ Y_j=[y_1\ y_2\ \ldots\ y_x]^T\ ,Y = [Y_1,\ Y_2,\ \ldots Y_M]. \tag{19}$$

y_i is the i th component of the transformed data Y_j. Small eigenvalues and their associated eigenvectors are ignored using a threshold value,

$$(\sum_{i=1}^{\chi}\lambda_i\ /\sum_{i=1}^{M}\lambda_i\ )>0.95. \tag{20}$$

The lower dimensional vector u captures the most expressive features of the original data. In this way, the data is compressed by reducing the number of dimensions without much loss of information.

In the following Figure 24, it is seen that 40 eigenvectors are chosen to transform the data obtained from STFT of signals from bad and good sleepers, since their eigenvalues were big enough according to the above criterion.



(b)

Figure 24. Eigen values obtained to reduce STFT results

4 Results and Discussion

In this work, a comparison of multilayer perceptron, learning vector quantization and gaussian mixture model classification techniques combined with nonstationary feature extractions for condition monitoring of wooden railway sleepers is presented.

Each classification technique is tested against different combinations of feature extraction techniques to determine the best combination of these two techniques. The implementation details of the classifiers used in the tests are as follows: A three layer backpropagation network (MLP) with 35 hiden nodes and 2 output nodes, using step size 0.7 was used for classification. LVQ using k-means clustering with k=17 was used. And a GMM trained with EM algorithm was used. The parameters were initialized by k-means clustering algorithm with k=33 and a diagonal matrix for all the mixture components.

The data set were split into training and test sets for classifiers. In the work, impact acoustic signals were collected by making experiments on 200 sleepers [4]. The data obtained from these signals were divided into training (75%) and test (25%)

sets as seen in the following table (Table 1).

Table 1. Partition of data into training and test sets [4]

Clas s	Training (75%)	Testing (25%)	Tota l
Goo d	108	36	144
Bad	42	14	56
Tota l	150	50	200

In feature extraction process, in STFT for each signal 1650 features were generated since 11 frame with 300 length were used without symmetrical FFT results, in CWT 66 scales were used so for each signal 66*1800 features were generated, in DWT in decomposition process 1892 coefficients were generated for each signal and in WVD on 129 points calculations were done ommitting half of the FFT results and finally having 129*900 features for each signal. WVD was demanding great deal of computational resources hence it has not been investigated on all points which would yield a 1800*1800 features for each signal. But still WVD provided very satisfactory results. As mentioned previously, dimension reduction was applied after feature extraction. And the final number of features after this process are given below (Table 2):

Table 2. The number of features before and after reduction

	RIGHT		LEFT		FLF	
	direct	with PCA	direct	with PCA	direct	with PCA
STFT	1650	40	1650	40	1650	80
CWT	66*1800	24	66*1800	27	66*1800	51
DWT	1892	51	1892	53	1892	104
WVD	129*900	73	129*900	59	129*900	132

The results from different classifiers can be combined by a rule such as majority voting between the classifiers to achieve better results. It is possible that more classifiers would increase performance of majority voting.

The data are collected from both ends of the inspected sleepers to get more reliable results. The tests were performed on the data from each end seperately and also by using the obtained data from both ends. To be able to use the data from both ends features which are obtained from both ends need to be combined in an appropriate way. Therefore, feature fusion was applied in two levels, feature level fusion and classifier level fusion. In feature level fusion extracted features from signals obtained from both ends were concatenated and again different combinations of resulting features were tested against classifiers. In classifier level fusion classification results obtained for extracted features from left and right ends were combined by an “and” operation.

Several tests were done and results are demonstrated in tables. The best results were obtained when feature level fusion was used. Below, the classification rates for all combined features against each classifier are seen (Table 3, 4), the majority voting (MV) approach unites the decisions and gives the best rate as 82% classifying 41 of 50 test samples correctly. In feature level fusion, firsly all extracted features from both ends of the same sleepers are combined in the same vectors.

Table 3. Best classification rates with combined features

	ALLfeatures
MLP	82
GMM	80
LVQ	80
MV	82

Table 4. The number of correctly classified samples with MV

	MV on ALLfeatures
correct bad	6
correct good	35
Overall	41

Below, the number of incorectly classified samples in classification based on feature level fusion are seen (Table 5).

Table 5. The number of incorrectly classified samples

		MLP	GMM	LVQ
	Patterns	No of Errors	No of Errors	No of Errors
Bad	14	7	10	5
Good	36	2	0	5
Overall	50	9	10	10

When features are compared alone WVD gives the best classification rate with the features of the different ends of the same sleepers combined before classification (feature-level fusion) as seen from the following table (Table 6).

Table 6. Classification rates on separeate features

	STFT	DWT	CWT	WVD
MLP	74	72	72	80
GMM	74	72	70	80
LVQ	58	40	56	66
MV	74	72	72	82

The other combinations of features used in feature level fusion tests are seen below (Table 7, 8). Different characteristics can be seen from the results such as when WVD is used the rates seemed to increase and GMM clasifier provide better results than the others on these combinations of features.

Table 7. Combinations with 2 features used in feature level-fusion tests

	STFT,DWT	STFT,CWT	STFT,WVD	DWT,CWT	DWT,WVD	CWT,WVD
MLP	72	72	66	66	70	62
GMM	72	74	80	70	78	80
LVQ	68	68	72	52	62	68
MV	72	74	78	66	74	70

Table 8. Combinations with 3 features used in feature level-fusion tests

	STFT,CWT DWT	STFT,CWT WVD	CWT,DWT WVD	STFT,DWT WVD
MLP	74	78	68	72
GMM	72	80	80	80
LVQ	60	72	64	78

MV	74	76	76	74

These recognition rates encourages the use of fusion for better results. When the classifier-level fusion was applied by combining classfication results obtained for the data set from different ends of the sleepers, the results were not as good as feature-level fuison as seen below (Table 9, 10, 11). And it did not improve much the results before this kind of fusion. (The results by using other combinations of features used in tests for classifier-level fusion are in appendix.) The reason for this is for feature level fusion, the classifiers were fed with complete data for sleepers and the classification became more efficient.

Table 9. classification rates of classifier-level fusion

	ALL features
MLP	74
GMM	64
LVQ	66
ALL	74

Table 10. the number of correctly classified samples with MV

	MV on ALLfeatures
correct bad	7
correct good	30
Overall	37

Table 11. the number of incorrectly classified samples

		MLP	GMM	LVQ
	Patterns	No of Errors	No of Errors	No of Errors
Bad	14	7	11	4
Good	36	6	7	13
Overall	50	13	18	17

The other tests on the data obtained from the left and right ends of the sleepers seperately can be found in appendix. To sum up, feature-level fusion provided the best recognition rates. The results can be improved by using more classifiers.

Machine vision techniques can be applied to improve the system when fed with suitable visual input. In the proposed approach there are many parameters to be tuned such as number of features, window size for STFT, number of scales for CWT to be able to get better results and it is technology-driven since the only characteristic of being a good sleeper is that it produces a “crisp” sound as suggested by the inspector. So to get good results heavily relies on the feature extraction techniques providing good results. The obtained signals are evaluated based on a very detailed processing and testing procedure by tuning many parameters to be able to reach a good judgement of the inputs since the extracted features need to provide rich characteristic properties of the data which can not be defined by the inspector. However, in a machine vision approach a possible solution would be knowledge driven, since the visual input data for classifiers can be identified easily for being classified as good or bad and to have a good classification results does not heavily depend on feature extraction process. Therefore, machine vision techniques can be expected to provide better results.

5 Conclusion

For automating the process of condition monitoring of wooden sleeper inspection problem, a classification based approach was examined in the work. The use of nonstationary feature extraction techniques as a means of classifying condition monitoring of wooden railway sleepers were discussed. Nonstationary feature extraction techniques were considered suitable since the characteristics of impact acoustic signals are changing with time, time varying frequency content, traditional time or frequency analysis is not sufficient. The spectral nature of such signals can not be illustrated by considering one argument, frequency, to interpret these signals efficiently. Instead for this kind of nonstationary signals a combined time-frequency representation is needed due to the inadequacy of either time domain or frequency domain analysis to describe the nature of non-stationary signals completely. Hence, time-frequency representation of the signals were generated by using four different techniques, Short Time Fourier Transform, Continuous Wavelet Transform, Discrete Wavelet Transform, Wigner-Ville Distribution. The techniques ara analyzed and results are presented for testing them in combination with several classification techniques, Gaussian Mixture Model, Learning Vector Quantization and Multilayer Perceptron, including majority voting evaluation and feature fusion experiments as well. As expected the nonstationary techniques provided better results than stationary feature extraction techniques. Experimental results on an example dataset demonstrate the validity of the approach and relevant results were presented. In experiments, feature fusion were applied in two ways, feature level fusion and classifier level fusion. Feature level fusion provided the best recognition rates, although the computational complexity for classifiers were increased since the total number of features used in this case were increased. The reason why the feature level fusion provided better results than classifier level fusion can be explained in a way that in the presence of feature level fusion the classifiers were fed with more complete and supporting descriptions of the sleepers thanks to the combined extracted features. The system can be improved by including more classifiers and supporting with machine vision techniques.

6 Appendix

6.1 Results by using the data from right ends of the sleepers

Table 12. Classification rates on separeate features

	STFT	DWT	CWT	WVD
MLP	72	72	72	74
GMM	74	72	68	78
LVQ	64	46	42	62
ALL	74	72	70	74

Table 13. 2-features combinations against classifiers

	STFT,DWT	STFT,CWT	STFT,WVD	DWT,CWT	DWT,WVD	CWT,WVD
MLP	72	72	66	68	72	72
GMM	72	72	76	70	74	76
LVQ	60	60	74	46	66	62
ALL	74	72	74	68	74	72

Table 14. 3-features combinations against classifiers

	STFT,CWT DWT	STFT,CWT WVD	CWT,DWT WVD	STFT,DWT WVD
MLP	74	72	70	72
GMM	72	76	72	76
LVQ	52	72	56	78

ALL	74	74	70	74

Table 15. Classification rates with all features

	ALL features
MLP	74
GMM	72
LVQ	74
ALL	78

Table 16. the number of incorrectly classified samples

		MLP	GMM	LVQ
	Patterns	No of Errors	No of Errors	No of Errors
Bad	14	7	12	6
Good	36	6	2	7
Overall	50	13	14	13

Table 17. the number of correctly classified samples with MV

	MV on ALL features
correct bad	7
correct good	32
overall	39

6.2 Results by using the data from left ends of the sleepers

Table 18 Classification rates on separeate features

	STFT	DWT	CWT	WVD
MLP	72	72	28	72
GMM	72	68	70	60
LVQ	62	64	46	70
ALL	72	70	44	72

Table 19. 2-features combinations against classifiers

	STFT,DWT	STFT,CWT	STFT,WVD	DWT,CWT	DWT,WVD	CWT,WVD
MLP	66	68	72	72	72	64

GMM	62	68	64	70	62	64
LVQ	70	60	58	68	68	66
ALL	68	68	66	72	64	66

Table 20. 3-features combinations against classifiers

	STFT,CWT DWT	STFT,CWT WVD	CWT,DWT WVD	STFT,DWT WVD
MLP	54	72	72	72
GMM	64	62	62	62
LVQ	72	66	74	70
ALL	70	68	70	72

Table 21. Classification rates with all features

	ALL features
MLP	76
GMM	64
LVQ	66
ALL	72

Table 22. The number of incorrectly classified samples

		MLP	GMM	LVQ
	Patterns	No of Errors	No of Errors	No of Errors
Bad	14	12	13	10
Good	36	0	5	7
Overall	50	12	18	17

Table 23. The number of correctly classified samples with MV

	MV on ALL features
correct bad	2
correct good	34
Overall	36

6.3 Results by classifier-level feature fusion

Table 24. Classification rates on separeate features

	STFT	DWT	CWT	WVD
MLP	72	72	28	74

GMM	74	68	66	66
LVQ	54	44	38	58
ALL	74	70	44	74

Table 25. 2-features combinations against classifiers

	STFT,DWT	STFT,CWT	STFT,WVD	DWT,CWT	DWT,WVD	CWT,WVD
MLP	66	68	66	68	72	64
GMM	62	68	66	68	62	68
LVQ	56	52	58	44	62	58
ALL	70	70	68	68	66	68

Table 26. 3-features combinations against classifiers

	STFT,CWT DWT	STFT,CWT WVD	CWT,DWT WVD	STFT,DWT WVD
MLP	54	72	70	72
GMM	64	66	62	64
LVQ	42	62	56	78
ALL	70	72	68	74

7 References

[1] Vary, A., 1972. “Investigation of an electronic image enhancer for radiographs”, Materials Evaluation, ASNT, Columbus, OH, USA., 30(12), pp. 259–267.

[2] Golis, M. J., 1991. “An introduction to nondestructive testing, Appendix B: Nondestructive testing approaches”, The American Society for Nondestructive Testing, Inc.

[3] Siril, Y., Gupta, N. K., and Dougherty, M. S., 2007. “Comparison of pattern recognition techniques for the classification of impact acoustic emissions”, Science Direct

[4] Siril Y., “Pattern recognition for automating condition monitoring of wooden railway sleepers”, Phd Thesis, Napier University, 2008

[5] Siril, Y., Gupta, N. K., Dougherty, M. S., 2006. “Pattern recognition approach for the automatic classification of data from impact acoustics”, Proceedings of IASTED International Conference on Artificial Intelligence and Soft Computing, Palma De Mallorca, Spain.

[6] Wang, X., and Tang, X., 2004. “Using random subspace to combine multiple features for face recognition”, The Sixth IEEE International Conference on Automatic Face and Gesture Reocgnition.

[7] Y., Xu, C., Zhang, and J., Yang, “Semi-supervised classification of musical genre using multi-view features”,

[8]Cowling, M., and Sitte, R., 2003. “Comparison of techniques for environmental sound recognition”, Pattern Recognition Letters, 24, pp. 2895–2907.

[9] LNKNET, http://www.ll.mit.edu/mission/communications/ist/lnknet/index.html

[10] Simon, H., 1999. “Neural networks a comprehensive foundation”, Prentice-Hall, Inc.

[11] Hagan, M., Demuth, H., and Beale, M., 1996. “Neural network design”, PWS publishing.

[12] Duda, R. O., and Hart, P. E., 1973. “Pattern classification and scene analysis”, John Wiley and Sons, Inc.

[13] Moore, A. W., 2004. “Clustering with Gaussian mixtures”, URL: http://www.autonlab.org/tutorials/gmm.html.

[14] Scherrer, B., “Gaussian mixture model classifiers”, URL: http://www.music.mcgill.ca/~scherrer/MUMT611/a03/Scherrer07GMM.pdf, February 5, 2007.

[15] Paalanen, P., 2004. “Bayesian classification using Gaussian mixture model and EM estimation: Implementations and comparisons”, Information Technology Project, http://www.it.lut.fi/project/gmmbayes/downloads/doc/report04.pdf.

[16] Marques, J., and Moreno, P. J., 1999. “A study of musical instrument classification using Gaussian mixture models and support vector machines”, (Technical Report), Cambridge Research Laboratory, Technical Report Series.

[17] Dempster, P., Laird, N. M., and Rubin, D. B., 1977. “Maximum likelihood from incomplete data using the EM algorithm,” Journal of the Royal Society of Statistics, 39(1), pp. 1-38.

[18] Duda, R. O., and Hart, P. E., 1973. “Pattern classification and scene analysis”, John Wiley & Sons, New York.

[19] Schwardt, L., 2003. “Gaussian mixture models”, URL: http://www.dsp.sun.ac.za /pr813/lectures/lecture06/pr813_lecture06.pdf.

[20] Goldberger, J., “Gaussian mixture model”, URL: http://www.wisdom.weizmann.ac.il/~irenak/ml02-03/gmm.pdf.

[21] Sanderson, C., and Paliwal, K., 2002. “Likelihood normalization for face authentication in variable recording conditions”, 2002 International Conference on Image Processing, 1, pp. I–301–I–304.

[22] Zhang, B., Zhang, C., and Yi, X., 2004. “Competitive EM algorithm for finite mixture models”, Pattern Recognition, 37(1), pp. 131-144.

[23] Reynolds, D., and Rose, R., 1995. “Robust text-independent speaker identification using gaussian mixture speaker models”, IEEE Transactions on Speech and Audio Processing, 3(1), pp. 72–83.

[24] Duda, R. O., Hart, P. E., and Stork, D. G., 2001. "Pattern classification", Wiley.

[25] Polikar R., "The wavelet tutorial", URL: http:// users.rowan.edu/ %7Epolikar/ WAVELETS/WTtutorial.html.

[26] Akay, M., Ed., 1998. "Time frequency and wavelets in biomedical signal processing", IEEE Press.

[27] Yoo, Y., "Tutorial on Fourier theory", http://www.cs.otago.ac.nz/cosc453 /student_tutorials/fourier_analysis.pdf, March 2001.

[28] Polikar, R., "The story of wavelets", URL: engineering.rowan.edu/~polikar /RESEARCH/PUBLICATIONS/Story_of_wavelets.ppt.

[29] "User's guide", Wavelet Toolbox™ 4, URL: http://www.mathworks.com/.

[30] Papoulis, A., 1977. "Signal analysis", McGraw-Hill, New York .

[31] Zhang, Y., Guo, Z., Wang, W., He, S., Lee, T., and Loew, M., 2003. "A comparison of the wavelet and short-time Fourier transforms for Doppler spectral analysis", Elsevier.

[32] "Joint time-frequency analysis lecture note"s, URL: http://www.control.auc.dk /~alc/Lectures/JTFA/Joint_Time-Frequency_Analysis_mm_1__4_per_page_.pdf.

[33] "Spectrograms", URL: http://cnx.org/content/m0505/latest/.

[34] Qian, S., and Chen, D., "Joint time-frequency analysis methods and applications", Prentice Hall.

[35] Daubechies, I., 1992. "Ten lectures on wavelets", Regional Conference Series in Applied Mathematics. Society forIndustrial and Applied Mathematics, Vermont, USA.

[36] Orr, M., Pham, D., Lithgow, B., and Mahony, R., 2001. "Speech perception based algorithm for the separation of overlapping speech signal", Proceedings of The Seventh Australian and New Zealand Intelligent Information Systems Conference, Perth, Western Australia, pp. 341– 344.

[37] Mallat, S. 1989, "A theory for multiresolution signal decomposition: the wavelet representation", IEEE Pattern Anal. and Machine Intell., 11(7), pp. 674–693.

[38] Ocak, H., 2008,"Automatic detection of epileptic seizures in EEG using discrete wavelet transform and approximate entropy", Expert Systems with Applications.

[39] Daubechies, I., 1992. “Ten lectures on wavelets”, Regional Conference Series in Applied Mathematics Society for Industrial and Applied Mathematics, Vermont, USA.

[40] Loutas, T. H., Sotiriades, G., and Kostopoulos, V., 2004. “On the application of wavelet transform of AE signals from composite materials”, Department of Mechanical and Engineering and Aeronautics, University of Patras, Greece and Institute of Chemical Engineering and High Temperature Processes.

[41] Viswanathan, V., Anderson, W., Rowlands, J., Ali, M., and Tewfik, A., 1994. “Real-time implementation of a wavelet-based audio coder on the T1 TMS320C31 DSP chip”, 5th International Conference on Signal Processing Applications & Technology (ICSPAT), Dallas, TX.

[42] Agbinya, J. I., 1996. “Discrete wavelet transform techniques in speech processing”, IEEE Tencon Digital Signal Processing Applications Proceedings, IEEE, New York, pp. 514-519.

[43] Rao, N., 2001. “Speech compression using wavelets”, (Thesis Project), School of Information Technology and Electrical Engineering.

[44] Coifman, R., and Wickerhauser, M. V., 1992. “Entropy-based algorithms for best basis selection”, IEEE Trans. on Inf. Theory, 38(2), pp. 713–718.

[45] Ilic, S., 2007. “Comparison of compression ratios for ECG signals by using three time-frequency transformations”, SER.: ELEC. ENERG., 20(2), pp. 223-232.

[46] Hapuarachchi, P., 2006. “Feature selection and artifact removal in sleep stage classification”, (Master Thesis), Applied Science in Electrical and Computer Engineering, The University of Waterloo ,Waterloo, Ontario, Canada.

[47] Zhu, T. X., Tso, S. K., and Lo, K. L. , 2004. “Wavelet-based fuzzy reasoning approach to power-quality disturbance recognition,” IEEE Transactions on Power Delivery, 19(4), pp. 1928–1935.

[48] Wigner, E., 1932. “On the quantum correction for thermodynamic equilibrium”, Physical Review, 40, pp. 749.

[49] Ville, J., 1948. “Theorie et applications de la notion de signal analytique”, Cables et Transmission, 2a(1), pp.61-74.

[50] Bradford, S. C., Yang, J., and Heaton, T., 2006. "Variations in the dynamic properties of structures: The Wigner-Ville distribution", Proceedings of the 8th U.S. National Conference on Earthquake Engineering, San Francisco, California, USA.

[51] Newland, D., 1997. "Practical signal analysis: Do wavelets make any difference?", 1997 ASME Design Engineering Technical Conferences, Sacramento, California.

[52] Kim, Y. B., Kim, S. J., Dongchung, H., Park, Y. W., and Park, J. H., 2003. "A study on technique to estimate impact location of loose part using Wigner-Ville distribution", Progress in Nuclear Energy, 43(1-4), pp. 261-266.

[53] "Time frequency analysis", URL: http : // www . medinfo . dist . unige . it / didattica % 5 Ceds % 5C05 _ Time – frequency _ analysis.pdf.

[54] Wu J. D., and Chiang, P. H., 2008. "Application of Wigner–Ville distribution and probability neural network for scooter engine fault diagnosis", Elsevier.

[55] Jeon, J. J., and Shin, Y. S., 1993. "Pseudo Wigner-Ville distribution", Computer Program And Its Applications To Time-Frequency Domain Problems, Naval Postgraduate School, Monterey, California.

[56] Wu J. D., Chiang, P. H., 2008. "Application of Wigner–Ville distribution and probability neural network for scooter engine fault diagnosis", Expert Systems with Applications.

[57] Cohen, L., 1989. "Time-frequency distributions-a review", Proceedings of the IEEE 77, pp. 941-981.

[58] Moss, J. C., and Hammond, J. K., 1994. "A comparison between the modfied spectrogram and the pseudo-Wigner-Ville distribution with and without modfication", Mechanical Systems and Signal Processing, 8, pp. 243-258.

[59] Ribeiro, M. P., Ewins, D. J., and Robb, D. A., 2003. "Non-stationary analysis and noise filtering using a technique extended from the original prony method", Mechanical Systems and Signal Processing, 17(3), pp. 533-549.

[60] Lukasiak, J., and Buvnett, IS., 2000. "Exploring the characteristics of analytic decomposition of speech signals", IEEE.

[61] Gillich, G. R., 2006. “Machine dynamics – vibrations”, Editura AGIR, Bucuresti.

[62] Isar, A., and Nafornița, I., 1998. “Time-frequency representations”, Editura Politehnica, Timişoara.

[63] Gillich, N., Potoceanu, N., Gillich, G. R., Raduca, M., and Chioncel, C. P., 2008. “Intelligent system for the control of ambient parameters in hi-tech workplaces”, 6th International DAAAM Baltic Conference INDUSTRIAL ENGINEERING 2008, Tallinn, Estonia.

[64] Cowling, M., Sitte, R., 2003. ”Comparison of techniques for environmental sound recognition”, Elsevier.

[65] “A tutorial on PCA”, URL: http://www.snl.salk.edu/~shlens/pub/notes/pca.pdf, 24 April 2007.

[66] Turk, M., and Pentland, A., 1991. “Eigenfaces for recognition”, Journal of Cognitive Neurosci, 3(1), pp. 71-86.

[67] Turk, M., 2001. “A random walk through eigenspace”, IEICE Trans. Inf. & Syst., E84-D, 12, pp. 1586-1595.

[68] Turk, M., and Pentland, A., 1991. “Face recognition using eigenfaces”, Proc. of the IEEE Conf. On Computer Vision and Pattern Recognition, pp. 586-591.