TAMPERE UNIVERSITY OF TECHNOLOGY

ELIAS METSÄNTÄHTI

FREQUENCY DOMAIN METHODS IN DIAGNOSTICS OF ROTATING MACHINES

Master of Science Thesis

Examiners: professor Matti Vilkko,
research fellow Tomi Roinila
Examiners and topic approved by
the Faculty Council of the Faculty
of Automation Engineering on 4
December 2013

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Automaatiotekniikan koulutusohjelma
METSÄNTÄHTI, ELIAS: Taajuusvastemenetelmät Pyörivien Koneiden Diagnostiikassa
Diplomityö, 61 sivua, 20 liitesivua
Kesäkuu 2014
Pääaine: Prosessiautomaatio
Tarkastajat: professori Matti Vilkko, Tutkijatohtori Tomi Roinila
Avainsanat: Taajuustaso, Fourier transform, fast Fourier transform, short time Fourier transform, wavelet transform, continuous wavelet transform

Koneiden ja laitteiden kunnonseuranta on tärkeä osa tehtaiden ylläpitoa. Seuraamalla koneiden kuntoa reaaliajassa pystytään huollot suunnittelemaan paremmin ja välttämään turhia tuotannon menetyksiä ja koneiden rikkoutumisia. Taajuustasomenetelmät tarjoavat tehokkaan menetelmän reaaliaikaista seurantaa varten. Menetelmiä voidaan käyttää jaksollisten signaalien tunnistamiseen datasta, jolloin saadaan arvokasta informaatiota koneiden kunnosta ja mahdollisista virhelähteistä. Tämän lopputyön tarkoituksena on käydä läpi erilaisia taajuustasomenetelmiä, joita kirjallisuudessa on käytetty ja tutkia tämän jälkeen eri menetelmien suorituskykyä jaksollisten signaalien tunnistamisessa datasta.

Työssä on tarkasteltu viittä eri taajuusvastemenetelmää. Käytetyt menetelmät ovat autokorrelaatio, nopea Fourier-muunnos (fast Fourier transform (FFT)), ikkunoitu Fourier-muunnos (short time Fourier transform (STFT)) ja jatkuva wavelet-muunnos (continuous wavelet transform (CWT)). Näitä menetelmiä sovellettiin oikeaan dataan, jota kerättiin maanmurskauslaitokselta. Kahta erilaista tutkimusjärjestelyä sovellettiin eri menetelmien vertailemiseksi toistensa kanssa.

Tulokset näyttivät, että autokorrelaatio ja FFT ovat erittäin hyviä menetelmiä jaksollisten signaalien tunnistukseen datasta. Datan pituus ja kohina vaikuttivat FFT:hen enemmän kuin autokorrelaatioon, mutta tulokset olivat silti hyviä. Autokorrelaation ja FFT:n ongelmana olivat, että jaksollisen signaalin luonteesta ei säilynyt informaatiota muunnoksen jälkeen. FFT ei myöskään sisällä aikainformaatiota jaksollisesta signaalista.

STFT ja CWT antoivat erittäin hyvää informaatiota jaksollisen signaalin luonteesta. Tulokset osoittivat, että CWT oli hieman heikompi kuin autokorrelaatio ja FFT jaksollisen signaalin havaitsemisessa, mutta kohina ei vaikuttanut tuloksiin yhtä paljoa. STFT oli kaikissa tilanteissa heikoin menetelmä jaksollisten signaalien löytämisessä.

Tutkimukset osoittivat, että STFT:llä on erittäin hyvä taajuusresoluutio. CWT:n taajuusresoluutio oli huonompi ja vahvemmat taajuudet peittivät muut taajuudet, mutta kaikki taajuudet pystyttiin kuitenkin löytämään datasta.

Autokorrelaatio ja FFT ovat erittäin helppoja ja nopeita menetelmiä käyttää verrattuna STFT:hen ja CWT:hen, mutta STFT ja CWT antavat parempaa informaatiota datassa olevista jaksollisista signaaleista. Kaikilla menetelmillä on hyvät ja huonot puolensa. STFT on paras menetelmä, kun jaksolliset signaalit ovat jatkuvia, koska muunnos säilyttää aikatason informaation ja taajuusresoluutio on hyvä. CWT on paras menetelmä käyttää, kun jaksolliset signaalit ovat impulsiivisia, koska muunnos säilyttää

aikatason informaation ja impulsiiviset signaalit on helppo löytää muunnoksesta. Tämän lopputyön tuloksien perusteella voidaan valita sopiva taajuustasomenetelmä käytettäväksi laitteiden ja koneiden kunnon seurannassa ja analysoinnissa.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Automation Technology
METSÄNTÄHTI, ELIAS: Frequency Domain Methods in Diagnostics of Rotating Machines
Master of Science Thesis, 61 pages, 20 Appendix pages
June 2014
Major: Process automation
Examiners: Professor Matti Vilkko, research fellow Tomi Roinila
Keywords: Frequency domain, Fourier transform, fast Fourier transform, short time Fourier transform, wavelet transform, continuous wavelet transform

Condition monitoring of machines is very important in maintenance of factories. By monitoring the condition of machines in real time, maintenances can be planned better and unnecessary loss of production and break down of machines can be avoided. Frequency-domain methods provide efficient means for such monitoring. The methods can be used in recognizing periodic signals from data thus providing valuable information of the condition and possible error sources. The aim of the thesis is to review the frequency-domain methods used in literature, and then investigate their performance in recognizing periodic signals from data.

Five frequency-domain methods were chosen from the literature and applied in studies. The chosen methods were autocorrelation, fast Fourier transform (FFT), short time Fourier transform (STFT) and continuous wavelet transform (CWT). The methods were applied to real-world data obtained from an earth crushing facility. Two experiment setups were implemented to compare the different methods with each other.

The results showed that autocorrelation and FFT were very good at finding the periodic signal from the data. The noise and the length of the data affected FFT more than autocorrelation, but the results were still good. The problem with autocorrelation and FFT was that information on the nature of the periodic signal was lost. FFT also had no time information of the periodic signal.

STFT and CWT provided very good information on the nature of the periodic signal. The results showed that CWT was a bit weaker than autocorrelation and FFT at finding the periodic signal, but the noise did not affect the results so much. STFT was always the weakest of the methods at finding the periodic signal, but the results were still good.

The experiments showed that STFT has very good frequency resolution. The frequency resoultion of CWT was worse and the stronger frequencies drowned the other frequencies, but all the frequencies could still be found from the data.

Autocorrelation and FFT are very easy and fast methods to use compared to STFT and CWT, but STFT and CWT provide better information on the periodic content of the data. All the methods have pros and cons. STFT is the best method to use at finding continuous periodic signals from data because the time domain content is preserved in the transformation and the frequency resolution is good. CWT is the best method to use at finding impulsive periodic signals from data because the time domain content is preserved and the impulsive signals are clear and easy to find from the transform. The results of the thesis can be used to select a frequency-domain method for a specific application to perform frequency-domain analysis and condition monitoring.

PREFACE

This master's thesis was done independently and the idea for the contents of it was given by professor Matti Vilkko from Tampere university of technology. The idea came from discussions about fault diagnostics and work that had been done with Metso Oyj an earth crushing facility owned by Rudus Oy. Post doctoral researcher Tomi Roinila acted as supervisor and examiner with Matti Vilkko for this thesis and I would like to thank them for their time and all the help I have gotten from them. I would like to thank project researcher Teemu Väyrynen and doctoral studen Pekka Itävuo for providing me information and data from the earth crushing facility.

CONTENTS

TERMS AND THEIR DEFINITIONS

Ψ_H	Haar wavelet
T	Period of a function
τ	Time variable
ω	Frequency of a function
a_n	Fourier coefficient
b_n	Fourier coefficient
$F(\omega)$	Continuous Fourier transform
G_k	Discrete Fourier transform
$STFT_{f,w}(\tau,\omega)$	Short time Fourier transform
$w(t-\tau)$	Window function for short time Fourier transform
$X_W(a,b)$	Continuous wavelet transform
$h_{a,b}(t)$	Wavelet function for wavelet transform
b	Translation parameter for wavelet transform
a	Scaling parameter for wavelet transform
$\phi_{mo,n}(t)$	Discrete wavelet transform scaling function
$\psi_{m,n}(t)$	Dyatic wavelet
$DWT\,f(t)$	Discrete wavelet transform
d	Detail parameter for discrete wavelet transform
S_ϕ	Approximation coefficient for discrete wavelet transform
T_ψ	Detail coefficient for discrete wavelet transform
r_{xx}	Autocorrelation function
r_{xy}	Cross-correlation function
c_{xx}	Autocovariance
c_{xy}	Cross-covariance
σ_x	Standard deviation of signal x
σ_y	Standard deviation of signal y
A	Amplitude of the square wave
s	Simulated square wave
p	Simulated signal p = A• s
d	Measurement data
x	Data with the simulated signal x = d + p

y_1	Frequency domain method applied to d
y_2	Frequency domain method applied to p
y_3	Frequency domain method applied to x
y_4	y_1 removed from y_3
q_1	Location of peaks in y_2
q_2	Location of peaks in y_4
m_1	Mean amplitude of the peaks in y_2
m_2	Mean amplitude of the peaks in y_4
c_i	Proportion between m_1 and m_2
x_i	Fourier transform of measurement data
f_i	Frequency found from x_i
j_i	Inverse Fourier transform of f_i
r_i	Inverse Fourier transform removed from data
n_1	Frequency domain method applied to data
n_2	Frequency domain method applied to j_i
n_3	Frequency domain method applied to r_i
S_1	Amplitude of n_1 at frequency f_i
S_2	Amplitude of n_2 at frequency f_i
S_3	Amplitude of n_3 at frequency f_i
$e(t)$	Proportion between $[S_1(t) - S_3(t)]$ and $S_2(t)$
z_i	Maximum value of e(t)
FT	Fourier Transform
DFT	Discrete Fourier transform
FFT	Fast Fourier transform
STFT	Short time Fourier transform
WT	Wavelet transform
CWT	Continuous wavelet transform
DWT	Discrete wavelet transform
ZOH	Zero Order Hold

1 INTRODUCTION

Condition monitoring of machines is very important in factories. Manufacturers spend a lot of money on maintenance and machine repairs each year. Improving the maintenance system even a little can greatly reduce the factory's expenses. There are four kinds of maintenance strategies [1]. The first one is breakdown maintenance where machines are repaired only after they break and the machine cannot function anymore. The strategy is very expensive because the plant shuts down which causes loss of production and the breakdowns can cause other parts of the machines to break down too.

The second strategy is preventive maintenance where a time based maintenance schedule is followed and the machines are repaired at fixed intervals. This method prevents complete break downs, but causes unnecessary downtime when the machines do not require repair at the time of the scheduled maintenance and money is lost because of lost production from the duration of the maintenance.

The third strategy is predictive maintenance where the condition of the machinery is monitored in real time or in specific time intervals. This way faults can be noticed at an early stage before machine breakdown which makes it possible to plan the plant run downs better and avoid unnecessary production loss and remove the need for big spare part amounts in storage. Hence predictive maintenance is an important part of manufacturing processes today.

The fourth strategy is proactive maintenance which takes predictive maintenance a bit further; after a fault is found the fault is investigated to find where it originates from. After finding the early causes of the fault they are fixed to prevent the fault from appearing again. The problem is that proactive maintenance is difficult to use because it requires very experienced and knowledgeable employees.

1.1 Condition Monitoring of Rotating Machines

One method to monitor the condition of a machine is to analyse measurement data obtained from the machine. The vibrations caused by rotating parts of the machine cause periodic signals in the measurement data which can be found and analysed. There are numerous moving parts in a rotating machine which generate vibration and random noise. If some part of the machine breaks, it causes a change in the measurement data. By analysing the data, the change can be noticed and the fault recognized before the machine breaks down. Common faults in rotating machinery are unbalance, bent shaft, misalignment, looseness, gear defects and bearing defects. Each fault produces its own type of vibration. [1]

The simplest form of rotating machinery condition monitoring is listening to the sounds the machine makes. When there is a sound that is not normally heard during

unfaulty conditions, the user knows that the machine is breaking down and needs maintenance. This however does not work in loud conditions, and requires expert knowledge.

To find the faults at an early stage and in loud environments the vibrations need to be measured and recorded. The vibrations can be measured acoustically, with transducers or in other ways such as measuring variations in the power required by the motor rotating the machine. After acquiring measurement data it can be analyzed in the time domain, but because of all the background noise from other parts of the machine, it maybe difficult to find the faults. That is why the measurement data is usually converted into the frequency or time-frequency domain using Fourier Transform (FT) or similar methods. In the frequency domain it is easy to find spikes caused by faults separated from the background noise. Figure 1 (a) shows a sinusoidal signal with frequency 100 Hz and amplitude 10 and Figure 1 (b) shows its FT.

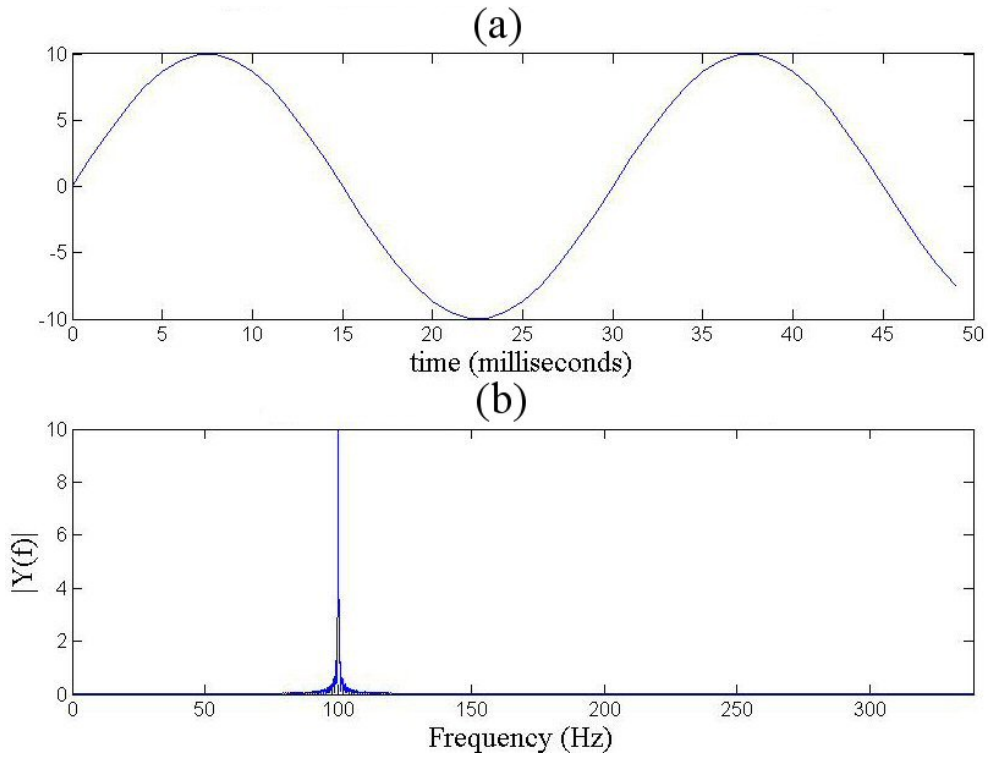

Figure 1: ***(a)*** *Sinusoidal Signal y(t)* ***(b)*** *FFT of y(t)*

As can be seen, Fourier Transform (FT) shows a spike at the sinusoidal wave's frequency with the same strength as the amplitude as the wave.

In practical environments the measurement data always has some random noise in the background disturbing the signal we are interested in. Figure 2 (a) shows a sinusoidal signal y(t) with a frequency of 100 Hz and amplitude 10 along with some randomly generated noise, and Figure 2 (b) shows the FT of the signal.

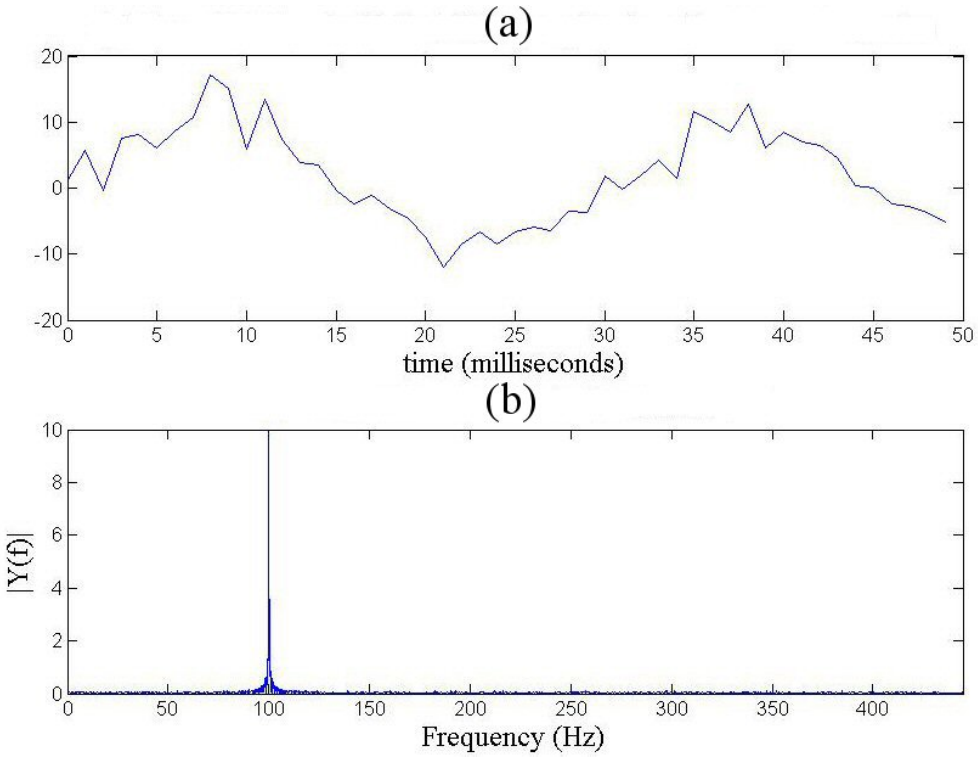

Figure 2: ***(a)*** *Sinusoidal signal with random noise y(t)* ***(b)*** *FFT of y(t)*

Figure 2 (b) shows that there is a spike at the sinusoidal signal's frequency 100 Hz that has the same amplitude as the sinusoidal wave, so the signal can be observed even with the random noise in the background.

1.2 Frequency Domain Analysis Methods

To analyse data in frequency domain for fault diagnostics, several methods have been proposed in scientific studies. Some of the proposed methods are discussed in the following chapters.

Wavelet Transform

Wavelet transform (WT) is similar to Fourier Transform (FT), but unlike FT, which uses sine and cosine functions as its basis, WT uses special functions with finite support, called wavelets. The most basic wavelet is Haar wavelet Ψ_H which is demonstrated in Figure 3.

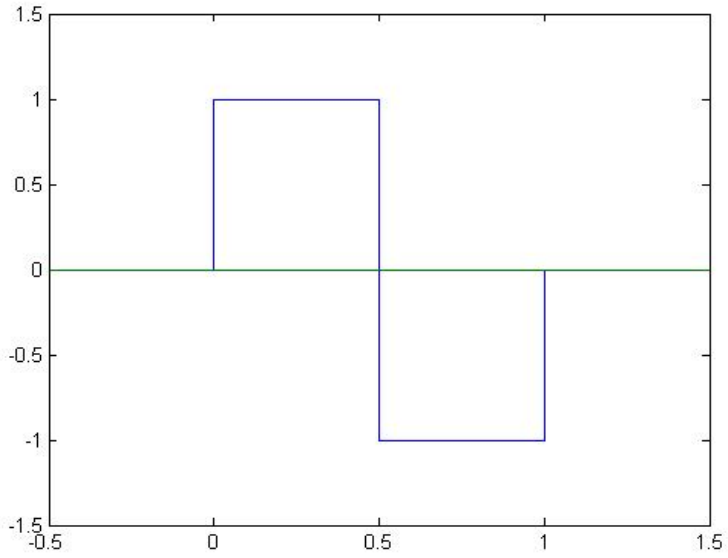

Figure 3: Haar wavelet

Haar wavelet gets values $\Psi_H = 1$ when $0 \leq t < 0{,}5$, $\Psi_H = -1$ when $0{,}5 \leq t < 1$ and $\Psi_H = 0$ at all other times. The advantages of WT over FT are its scalability and shifting operations which allow inspections of local properties of even nonperiodic signals with discontinuities instead of just global inspection of periodic signals because scalability generates series of wavelet functions with different window sizes. The different window sizes enable multi-resolution analysis which makes the analysis of non-stationary signals easier. Like FT, WT can be either discrete or continouus. [2]

A broken-bar fault of induction machines is diagnosed using discrete wavelet transform (DWT) in [3]. The goal of the paper is to avoid frequency leak estimations by applying the DWT to the machine's stator phase current and compute the energy associated to the rotor fault in the frequency bandwidth where the total rotor fault effect is localized. Energy computation is used to detect a rotor fault and to evaluate its severity without leakage estimation.

Harmonic wavelet packets method is used to analyze the rubbing vibration signal of turbine-generator units in [4]. In the paper a new method for using harmonic wavelet transform is researched to be used in signal analysis of rotating machines. Harmonic wavelet packets method gives a good frequency resolution in the high-frequency regions by analyzing only selected frequency bands of the signals.

Wavelet transform (WT) is used to design narrow filter banks used in fault diagnosis of machines running at different rotating speeds in [5]. The filter banks are obtained by matching the wavelet basis functions with the faulty signals.

Non-stationary vibration signal of rotating machinery is analyzed in frequency domain using Fast Fourier Transform (FFT) and in time-frequency domain using short-time Fourier transform (STFT) and a method based on a combination of continuous wavelet transform (CWT) and the wavelet packet transform in [6]. STFT is a modified FT where a suitable time window is used to window the signal before being Fourier transformed.

The discrete wavelet transform (DWT) and the discrete wavelet packet transform are used in bearing fault diagnosis to detect combustion failures in internal combustion engines in [7] and [8] respectively.

A machine-learning-based fault diagnostics method for condition monitoring of constant-speed rotating machines via vibration is proposed in [9]. The method consists of five steps: vibration signal measurement, discrete wavelet transform (DWT) based preprocessing, feature extraction, base-line encoding, and diagnosing the faults using fuzzy neural network.

Wavelet neural network is used to detect and locate bearing vibration of turbine-generators in [10]. When a fault appears in the turbine-generator the rotor can vibrate violently because of high rotating speeds, which damages the other major components and can cause serious accidents and economical loss. The transient signal measured from the turbine is decomposed into series of wavelet subspaces based on WT, each of which covers a specific frequency band in time-frequency domain. These feature vectors are used as input nodes to the wavelet neural network for fault pattern recognition, which operates on the feature vectors to produce recognition decisions based on previously accumulated knowledge.

A method for multi-concurrent fault diagnosis of aero engines based on integration of fractal exponent wavelet analysis and neural networks is presented in [11]. The WT is used to localize the characteristics of a signal both in the time-frequency domain. Using fractal theory with WT coefficients fractal exponents are obtained. The exponents are used as an input for radial basis function neural network to recognize faults in the aero engine.

An online fan fault diagnosis system using continuous wavelet transform (CWT) with Mallat algorithm and neural network is presented in [12]. A noise signal is measured from a fan and the recognition system utilizes power spectrum gravity center, sound level, and wavelet frequency segment power taken from the measured signal as feature vectors for back propagation neural network input to detect faults in the fan.

A method for the identification of characteristic components in frequency domain based on singularity analysis with WT is proposed in [13]. A continuous wavelet transform (CWT) is applied to vibration signal for singularity detection. Then, Lipcshitz

exponent function based on WT modulus maxima is defined through approximate estimation of Lipschitz exponent. In order to highlight the periodic components, correlation analysis of Lipschitz exponent function is performed and FT is employed to reveal fault related frequencies in the signal.

A wavelet neural network is constructed based on CWT in [14]. After collecting the vibration signal from rotating machinery and computing spectrum characteristic parameters in frequency domain, the samples are used to train the constructed back propagation wavelet neural network to detect faults.

The diagnosis sensitivities of short-time Fourier transform, wavelet analysis and the pseudo-Wigner-Ville distribution are investigated for condition diagnosis of rotating machinery in [15]. An extraction method for instantaneous feature spectrum is proposed using the Relative Crossing Information, by which the feature spectrum from time-frequency distribution can be automatically extracted by a computer in order to identify the condition of a machine. Based on the above studies, a fuzzy diagnosis method using sequential inference and possibility theory is also proposed. The methods are used to investigate the condition of rolling bearings.

Fourier Transform

Acoustic noise signal spectral analysis is used to predict the remaining useful life of a rotating machine comprised of several moving parts, including two rolling bearings, in [16]. Two methods are used for spectral analysis. First method is by extracting features with envelope analysis, which involves filtering the signal around the spectral region of interest. A band pass filter is used to focus on the particular region of interest to remove all other unwanted adjacent frequencies that are deemed noise. This band limited signal then passes through a stage of rectification or demodulation to produce an envelope signal. This envelope signal is then analyzed in the frequency domain by taking the short time Fourier transform (STFT). The second method is modulation spectral analysis.

A digital signal processing based measurement system dedicated to vibration analysis of rotating machines was designed in [17]. An intelligent FFT-analyzer for rotating machine monitoring and diagnosis that consists of a pattern matching procedure, which detects and isolates the fault by comparing the actual device model with unfaulty and faulty ones. A number of tests were carried out on small-size three-phase asynchronous motors.

A diagnosis method for detecting bearing faults is carried out based on frequency response analysis is designed in [18] and in [19]. The frequency response is obtained by using two measured signals from the input and output of the system. The calculation of the frequency response is carried out by applying the Welch-method on the two signals which involves sectioning the data, using short time Fourier transform (STFT) on a section, and calculating the frequency response by using periodograms. The system is

excited by pseudo random binary test signals and the deviation between the frequency response obtained during the commissioning of the plant and the curve measured under fault condition serve as indicators for damage.

A fault detection and diagnosis method for induction motor bearings is proposed in [20]. Artificial Neural Network by using Genetic Algorithm is used to identify the faults. Genetic algorithm is utilized to find the optimum weights and biases for Elman Network. During preprocessing stage, vibration signals are converted from time domain into frequency domain through fast Fourier transform (FFT). Then enveloping analysis is used to eliminate the high frequency components from vibration signal.

A fault detection methods based on grey relation degree is proposed in [21]. The sampled data is transformed by fast Fourier transform (FFT) into frequency domain and then the interesting frequency band is selected by harmonic window decomposition. The energy distribution of each frequency band is calculated. Different fault types correspond with different energy distributions. The grey relation between the symptom set and standard fault set is calculated as the identification evidence for fault diagnosis.

Quality of a turbo generator was diagnosed using leakage field analysis in [22]. The leakage field was measured using a specially designed search coil. The search coil's voltage was analyzed in time domain and in frequency domain using FFT.

An approach for multiple-faults diagnosis for rotor-bearing system based on Multiple Frequency Energy Spectrum and Dempster-Shafter evidence theory was presented in [23]. First, the original acceleration signals are processed by FFTation from the time domain to frequency domain. According to the analysis of the frequency information, the multiple frequency energy spectrum is put forward to extract features from vibration under normal and faulty conditions of rotational mechanical systems. Secondly, these features were given as inputs for training and testing the model of Radial Basis Function neural network. Finally, all the neural network results of multi-sensors are fused to diagnose the condition of the machine.

Fault diagnosis of bearings of rotating machinery is done by using multiple frequency energy spectrum and Bayesian network interface in [24]. Original acceleration signals are preprocessed by FFT and energy spectrum is used to extract features from the signal in frequency domain. These features are given as inputs for training and testing the neural network model.

1.3 Problems of Frequency Domain Methods

The biggest problem with FT is that it does not contain any time domain information of the signal. Different frequencies can be found from the signal, but there is no information on when the periodic signal appeared in the data or how long it lasted. Spectral leakage is also a problem when researching non-periodic measurement data.

The good side of FT is that the frequency information is accurate and easy to understand. FT is also very easy and fast to use.

Short time Fourier transform (STFT) uses a windowing function that is moved in the time domain and FT is used on each windowed piece at a time. This way the time domain information can be kept after the FT and the periodic signals can be located in both the time and frequency domains. The problem with using STFT is that selecting the right windowing function for the measured data can be difficult and the windowing function is the same for the whole duration of the analysis. Because of this choosing the right size for the window is difficult; short window means good time resolution, but bad frequency resolution and long window means good frequency resolution, but bad time resolution.

WT solves the STFT's problem by using a basis function that changes size over time so the time and frequency resolutions are better. The problem with WT is that the selection of the right basis function is very difficult and requires a lot of experience and knowledge of the system.

1.4 Goals of the Thesis

The goal of this thesis is to research different methods for analysing signals in the frequency domain and to find the best method for finding periodic signals from data. The researched methods are implemented on data obtained previously from an earth moving facility. The data is measured from the power lines of the motors rotating the conveyor belts in the facility and from it can be seen how much power the motors use.

In the first experiment a simulated square wave signal with different amplitudes is added to the data and then the ability of different methods to find the simulated signal is compared. In the second experiment the data is analysed without the simulated signal in it and different frequency domain analysis methods are compared.

1.5 Contents of the Thesis

In chapter 2 the theories for different frequency domain methods are introduced. FT is discussed in chapter 2.1, WT is discussed in chapter 2.2 and autocorrelation is discussed in chapter 2.3.

In chapter 3 the measurement data preprocessing methods and experiment setups are introduced. In chapter 3.1 the earth moving facility and data obtained from it are introduced. In chapter 3.2 the implementation methods for different frequency domain methods are introduced. In chapter 3.3 the generated simulated signal is introduced. In chapter 3.4 and 3.5 the two different experiment setups are introduced. In chapter 4 the results of the experiments introduced are shown and the conclusions are made in chapter 5.

2 FREQUENCY DOMAIN METHODS

In Chapter 2 the theory of three different frequency domain methods is introduced. Two different frequency domain methods were found from literature in Chapter 1: Fourier transform and wavelet transform. In Chapter 2.1 four different forms of Fourier transform are introduced: Fourier series, continuous Fourier transform, discrete Fourier transform and short time Fourier transform. In addition a method for fast calculation of Fourier transform called fast Fourier transform is introduced. In Chapter 2.2 two different forms of wavelet transform are introduced: continuous wavelet transform and discrete wavelet transform. In Chapter 2.3 autocorrelation is introduced.

2.1 Fourier Transform

Methods based on FT are used a lot to analyze signals and systems in communications theory, fault diagnostics and speech and image processing. In this chapter four different Fourier transforms are explained: Fourier series, continuous Fourier transform, discrete Fourier transform (DFT) and short time Fourier transform. In addition an algorithm for fast calculation of DFT called fast Fourier transform (FFT) is introduced.

Fourier Series

Functions can be represented in the form of series comprised of base set components like power functions x^n in the power series of the form (1). [27]

$$f(x)=\sum_{n=0}^{\infty} a_n x^n \tag{1}$$

The advantages of expanding a function in series is that integration and differentiation can become easier and function approximations can be made from the first few terms of the series.

With Fourier series a periodic function with period T can be expanded into base sets of sine and cosine waves in frequency domain according to (2)

$$f(t)=\frac{1}{2}a_0+\sum_{n=0}^{\infty} a_n \cos(n\omega t)+\sum_{n=0}^{\infty} b_n \sin(n\omega t) \tag{2}$$

where ω is the function's frequency $1/T$ or circular frequency $2\pi/T$. Fourier coefficients a_n and b_n can be obtained with Euler's formulae

$$a_n=\frac{2}{T}\int_d^{d+T} f(t)\cos(n\omega t), \qquad (n=0,1,2,\ldots) \tag{3}$$

$$b_n = \frac{2}{T}\int_d^{d+T} f(t)\sin(n\omega t), \qquad (n=0,1,2,\ldots) \tag{4}$$

Fourier series can be also presented in complex or exponential form. If $\sin(n\omega t) = 1/2\mathrm{j}(e^{jn\omega t} - e^{-jn\omega t})$ and $\cos(n\omega t) = 1/2(e^{jn\omega t} + e^{-jn\omega t})$ are inserted into (2) we get (5) for periodic functions of period T

$$f(t) = \sum_{n=0}^{\infty} c_n e^{jn\omega t} \tag{5}$$

where

$$c_n = \frac{1}{T}\int_d^{d+T} f(t)e^{-jn\omega t}dt, \qquad (n=0,\pm 1,\pm 2,\ldots) \ . \tag{6}$$

Continuous Fourier Transform

Fourier series is used to create an alternative frequency domain representation to a periodic function's time domain waveform. A different approach is needed for non-periodic functions. [27]

One way to do this is is to select a portion called window from the non-periodic function f(*t*) with length of *T* and then consider what happens as *T* gets larger. The function inside the window is periodic and could be defined by

$$g(t) = \begin{cases} f(t), & |t| < \frac{1}{2}T \\ f(t-nT), & \frac{1}{2}(2n-1)T < |t| < \frac{1}{2}(2n+1)T \end{cases}$$

So g(*t*) = f(*t*) between -½*T* and ½*T*. The Fourier series of function g(*t*) is

$$g(t) = \sum_{n=0}^{\infty} G_n e^{jn\omega_0 t} \tag{7}$$

with

$$G_n = \frac{1}{T} \int_{\frac{-T}{2}}^{\frac{T}{2}} g(t) e^{-jn\omega_0 t} dt \tag{8}$$

where $\omega_0 = 2\pi/T$.

The frequency of the general term is $2\pi n/T = n\omega_0 = \omega_n$ and so the difference in frequency between successive terms is

$$\Delta\omega = \frac{2\pi}{T}[(n+1)-n] = \frac{2\pi}{T} = \omega_0 \tag{9}$$

Because of (9) it can be shown [27] that (7) and (8) can be expressed as

$$g(t) = \frac{1}{2\pi} \sum_{-\infty}^{\infty} G(\omega_n) e^{-j\omega_n t} \Delta\omega \tag{10}$$

and

$$G(\omega) = \int_{\frac{-T}{2}}^{\frac{T}{2}} g(\tau) e^{-j\omega t} d\tau \tag{11}$$

respectively. As T increases and nears infinity, the window becomes bigger and g(t) = f(t) everywhere so $\Delta\omega$ nears zero. When we rename the variables we can write (10) and (11) as

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{\omega t} d\omega \tag{12}$$

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt = \int_{-\infty}^{\infty} f(t)[\cos(\omega t) - i\sin(\omega t)] dt \tag{13}$$

respectively. F(ω) as defined by equation (13) is called the continuous Fourier transform of f(t) and it converts the signal from time domain into frequency domain. Equation (12) is called the inverse Fourier transform and it transforms F(ω) back to time domain.

Fourier transforms are generally complex-valued functions and show the complex frequency spectrum of f(t). Writing F(ω) in its exponential form we get

$$F(\omega) = |F(\omega)| \, e^{j \arg F(\omega)} \tag{14}$$

where

$$|F(\omega)| = \sqrt{Re(F(\omega))^2 + Im(F(\omega))^2} \tag{15}$$

is the amplitude spectra of the signal f(t) and

$$arg\ F(\omega) = \tan^{-1}\left(\frac{Im[F(\omega)]}{Re[F(\omega)]}\right) \tag{16}$$

is the phase spectra of the signal f(t). By plotting the amplitude spectra the frequency content of the signal can be investigated. An example of the amplitude spectra can be seen in Figure 1 (b).

Discrete Fourier Transform

Discrete Fourier transform (DFT) converts a signal with finite length to the frequency domain. It is used to make mathematical calculations easier for the computer programs. To form DFT let's first create a FT for sequence of numbers $\{g_k\}$ using (5) and (6) by writing $\theta = \omega T$ where T is the time between signal samples. By using $\{g_k\}$ to define continuous function G(θ) we get (17)

$$G(\theta) = \sum_{n=-\infty}^{\infty} g_n e^{-jn\theta} \tag{17}$$

and its inverse transformation (18).

$$g_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} G(e^{j\theta}) e^{jk\theta}\, d\theta \tag{18}$$

G(θ) is the FT of $\{g_k\}$ and it is a function of a continuous variable θ and because $e^{j\theta}$, G(θ) is periodic even if the sequence $\{g_k\}$ is not periodic.

Taking a sequence $\{g_k\}$ from a continuous signal g(t) at intervals of T we get a sequence

$$\{g_k\} = \{g(kT)\} \begin{matrix} k=0 \\ N-1 \end{matrix}$$

Using (17) and inserting $\theta = \omega T$ we get (19)

$$G(\omega T) = \sum_{n=-\infty}^{\infty} g_n e^{-jn\omega T} \tag{19}$$

By sampling G(θ) at $\Delta\omega$ intervals to create N equally spaced samples over the interval $0 \le \theta \le 2\pi$ we get $N\,\Delta\theta = 2\pi$ where $\Delta\theta$ is the normalized frequency spacing. Because T is constant we can write $\Delta\theta = T\,\Delta\omega$ and

$$\Delta\omega = \frac{2\pi}{NT} \tag{20}$$

Sampling (19) at the intervals $\Delta\omega$ we get

$$G_k = \sum_{n=0}^{N-1} g_n \mathrm{e}^{-jnk\,\Delta\omega T} \tag{21}$$

The inverse FT for G_k is

$$g_n = \frac{1}{N} \sum_{k=0}^{N-1} G_k \mathrm{e}^{jnk\,\Delta\omega T} \tag{22}$$

(21) and (22) define the discrete Fourier transform pair. [27]

Fast Fourier Transform

FFT was introduced in 1965 by Cooley and Tukey to reduce the computational complexity of the DFT to make it a practical tool for engineering applications. [27] The FFT approach comprises of three steps:

1. matrix formulation
2. matrix factorization
3. rearranging.

Let's consider a situation where the number of samples N is $N = 2^{\gamma} = 2^2 = 4$. So we are using integer value of $\gamma = 2$, but the method can be extended for other values of γ. First let's define

$$W = \mathrm{e}^{-j\Delta\omega T} = \mathrm{e}^{-j2\pi/N} = \mathrm{e}^{-j\pi/2}$$

Then using (21) we get

$$G_k = \sum_{n=0}^{N-1} g_n \mathrm{e}^{-jnk\,\Delta\omega T} = \sum_{n=0}^{N-1} g_n \mathrm{W}^{nk} = \sum_{n=0}^{3} g_n \mathrm{W}^{nk} \qquad (k = 0,1,2,3) \tag{23}$$

Writing out the terms of the transformed sequence it can be shown [27] that (23) can be written in matrix form as

$$\begin{bmatrix} G_0 \\ G_1 \\ G_2 \\ G_3 \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W^1 & W^2 & W^3 \\ 1 & W^2 & W^0 & W^2 \\ 1 & W^3 & W^2 & W^1 \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = W^{nk} g_n \tag{24}$$

which is the result of step 1 matrix formulation.

For step 2 we factorize and interchange parts of (24) so that it can be written as

$$\begin{bmatrix} G_0 \\ G_2 \\ G_1 \\ G_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W^2 & W^0 & W^2 \\ 1 & W^1 & W^2 & W^3 \\ 1 & W^3 & W^2 & W^1 \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & W^0 & 0 \\ 0 & 1 & 0 & W^0 \\ 1 & 0 & W^2 & 0 \\ 0 & 1 & 0 & W^2 \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} \tag{25}$$

$$= \begin{bmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{bmatrix} \begin{bmatrix} g_0' \\ g_1' \\ g_2' \\ g_3' \end{bmatrix} = \begin{bmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{bmatrix} g' = G'$$

Writing out (24) and (25) in equation form we get

$$\begin{aligned} G_0 &= g_0 W^0 + g_1 W^0 + g_2 W^0 + g_3 W^0 \\ G_1 &= g_0 W^0 + g_1 W^1 + g_2 W^2 + g_3 W^3 \\ G_2 &= g_0 W^0 + g_1 W^2 + g_2 W^4 + g_3 W^6 \\ G_3 &= g_0 W^0 + g_1 W^3 + g_2 W^6 + g_3 W^9 \end{aligned} \tag{26}$$

and

$$\begin{aligned} G_0 &= g_0' + W^0 g_1' \\ G_2 &= g_0' + W^2 g_1' \\ G_1 &= g_2' + W^1 g_3' \\ G_3 &= g_2' + W^3 g_3' = g_2' - W^1 g_3' \end{aligned} \tag{27}$$

respectively. Comparing (26) and (27) we can see that (27) requires less operations to be solved compared to (26). The total number of operations needed to resolve equation (27) is four complex multiplications and eight complex additions compared to 16 complex multiplications and 12 complex additions of equation (26). Generally, when N

$= 2^{\gamma}$, equation (27) will require $\frac{1}{2}N\gamma = N \log_2 N$ complex multiplications and $N\gamma$ complex additions. The number of complex multiplications of (26) can be calculated as N^2. This concludes the second step.

In the third step the order of the transform vector is changed back to what it was before step 2. We do this by using binary notation and applying bit reversal process. In bit reversal process a binary number is written with its digits in reverse order. In the case of N=4 samples equation (25) becomes $G' = \begin{bmatrix} G_0 & G_2 & G_1 & G_3 \end{bmatrix}^T = \begin{bmatrix} G_{00} & G_{10} & G_{01} & G_{11} \end{bmatrix}^T$. Using bit reversal process it is easy to see that by reversing the digits we get the natural order $G = \begin{bmatrix} G_0 & G_1 & G_2 & G_3 \end{bmatrix}^T = \begin{bmatrix} G_{00} & G_{01} & G_{10} & G_{11} \end{bmatrix}^T$. [27]

Short Time Fourier Transform

The amplitude of the FT as represented in (13) shows the power of the oscillating component at a certain frequency as shown in Figure 1, but nothing is told about the component in the time domain. This information can be obtained by applying a windowing function on the signal which then Fourier transforms only a portion of the signal contained in an interval moving over time. The windowing function also removes any discontinuities of the signal which makes FT work better. The only problem is that the FT is taken also of the windowing function which may distort information from the original signal. This FT is called Short Time Fourier Transform (STFT).

The equation for STFT is

$$STFT_{f,w}(\tau, \omega) = \int_{-\infty}^{\infty} f(t) w(t-\tau) e^{-j\omega t} dt \qquad (28)$$

At each time instant t, we get a spectral decomposition obtained by applying the FT to the portion of signal f(t) viewed through the window w(t-τ). All the signal parts outside the window are neglected.

If a signal has most of its energy in a given time interval [-T,T] and frequency interval [-Ω,Ω], STFT will be localized in the region [-T,T] x [-Ω,Ω] and close to zero in time and frequency intervals where the signal has little energy. In normal FT the high energy peaks will be spread over the whole frequency domain. The limitation of STFT is that because only single window is used for all frequencies, the resolution of the analysis is the same everywhere in the time-frequency plane. [28]

2.2 Wavelet Transform

Wavelet transform is used a lot in various fields of signal analysis such as in image analysis and fault diagnostics. Instead of using sinusoidal base functions like FT, WT

uses special wavelet base functions. By using wavelet base functions WT can capture frequencies in transient states. Figure 4 shows some wavelet base functions.

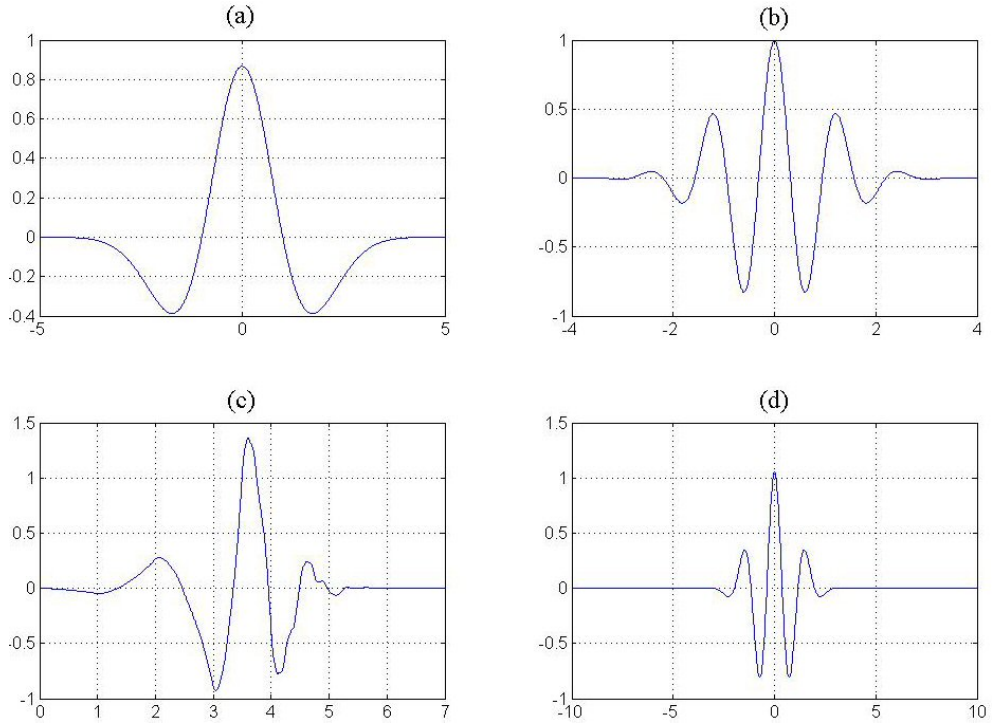

Figure 4: (a) Mexican Hat Wavelet (b) Morlet Wavelet (c) Daubechies Wavelet db4 (d) Gaussian Wavelet

In this chapter different WT methods are introduced. First, CWT is introduced for continuous signals. Then, wavelet series is introduced and expanded to DWT.

Continuous Wavelet Transform

The CWT of a continuous signal f(t) is defined as

$$X_W(a,b) = \int_{-\infty}^{\infty} f(t)\, h_{a,b}(t)\, dt \tag{29}$$

where $h_{a,b}(t)$ is the basis function of WT. Basis functions are obtained from a mother wavelet h(t) by translation and scaling by using equation (30).

$$h_{a,b}(t) = \frac{1}{\sqrt{a}} h\left(\frac{t-b}{a}\right) \tag{30}$$

where b is translation parameter and a is a scaling parameter. For CWT parameters b and a are continuous.

Scale is proportional to the duration of the wavelet functions. Large scaling parameter makes the basis function a low frequency stretched version of the mother wavelet with long duration. Large scaling parameters are used to capture the long-term behaviors. If scaling parameter is small, the basis function is a contracted version of the mother wavelet with short duration and high frequency. Small scaling parameters are used to capture short-term behavior of the signal. Because of the scaling of wavelet functions, WT captures both long- and short-term trends in a signal unlike FT, which captures only long-term behavior because all the basis functions have infinite duration. The factor $1/\sqrt{(a)}$ guarantees that all the wavelets have the same energy. Figure 5 shows the scaling effect of scaling parameter *a* and translation parameter *b* on Morlet (blue line) wavelet together with a sinusoidal wave (red line).

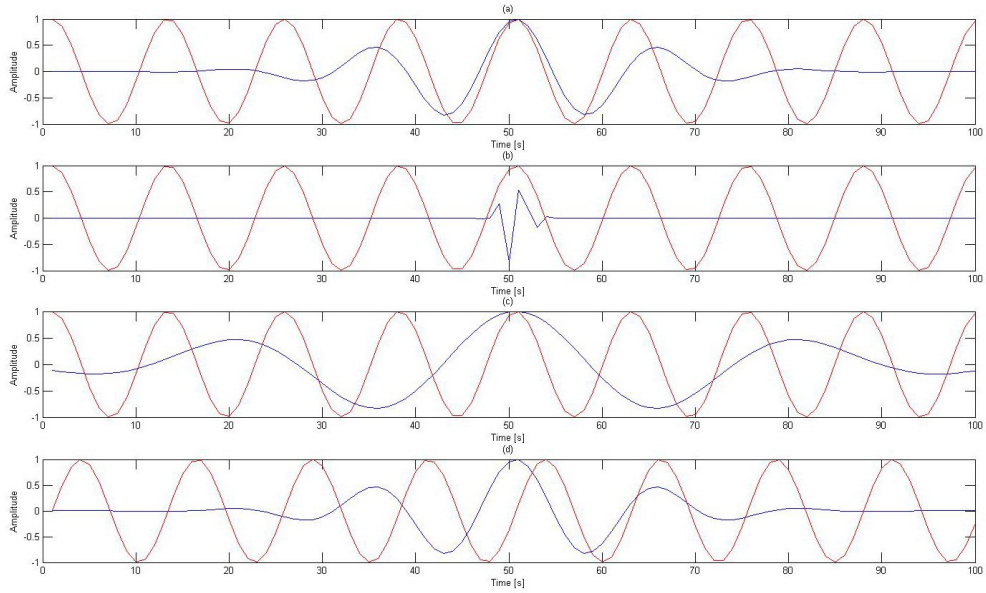

Figure 5: (a) Mother wavelet a = 1 (b) Basis function a = 0.1 (c) Basis function a = 2 (d) Basis function b = 1.25

CWT calculates a coefficient similar to correlation between the wavelet and the signal at each time point for each scale. The coefficient is the product between the signal and the wavelet and then takes the integral of the product. The larger the correlation between the signal and the wavelet, the larger the coefficient is. If the signal and the wavelet correlate perfectly, that is they are the same at each point, then the coefficient returned is 1. The coefficients taken from the signals shown in Figure 5 are (a) 6.8, (b) 0.0001, (c) 0.18 and (d) 0. Figure 5 (a) has a scale and location that corresponds the most with the sinusoidal signal, so it has the largest coefficient. Figure 5 (b) has a very small scaling parameter, so the coefficient is close to 0 even though the location is right. Figure 5 (c) has the right location so the coefficient is a bit above 0. Figure 5 (d) the wavelet has the different translation parameter, so the coefficient is 0.

Figure 6 shows the CWT of a 1 Hz sine wave with amplitude 0.1.

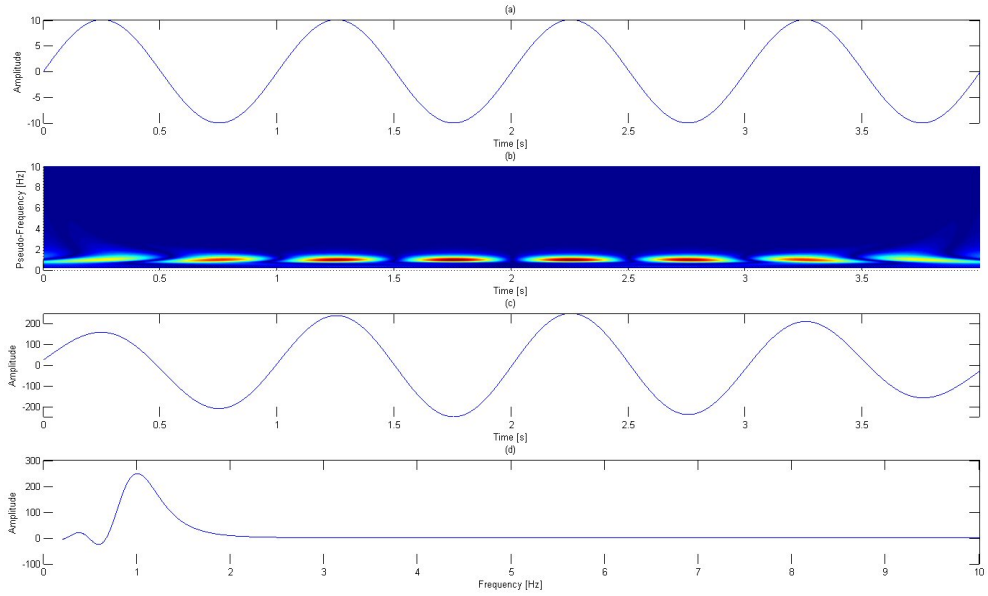

Figure 6: CWT of a sine wave

CWT returns a matrix with time on the columns and frequency on rows. In Figure 6 (c) a scale row corresponding to 1 Hz frequency is plotted. In Figure 6 (d) the coefficients of the CWT are plotted at a certain time to see a peak at the 1 Hz frequency. [29]

The CWT is one of the best methods for the detection of singularities, such as impact faults, in signal. When the signal changes abruptly, it results in the wavelet maxima. The singularity is detected using the local maxima lines in the CWT domain by finding the abscissa (x-coordinate) where the wavelet modulus maxima converge at fine scales. [13]

Discrete Wavelet Transform

Using discrete scale and translation parameters in (30) signal *f(t)* can be expanded into wavelet series which uses summation rather than integral. This makes the WT computationally faster [29]. The wavelet series of *f(t)* is

$$f(t)=\sum_{n} a(m_0,n)\phi_{m_0,n}(t)+\sum_{m=m_0}^{\infty}\sum_{n=-\infty}^{\infty} d(m,n)\psi_{m,n}(t) \tag{31}$$

f(t) does not need to be periodic. In (31) $\phi_{m_0,n}(t)$ is a basis function with a fixed scale j_0 and the first summation is over all possible translation values *k*. Functions $\phi_{m_0,n}(t)$ are called scaling function and they are obtained by scaling and translating a prototype function

$$\phi_{m,n}(t)=2^{m/2}\phi(2^m t-n) \tag{32}$$

which has the property $\int_{-\infty}^{\infty} \phi_{0,0}(t)\,dt = 1$. $\phi_{0,0}$ is sometimes called the father wavelet. $\psi_{m,n}(t)$ are called the dyadic wavelets, which are expressed as

$$\psi_{m,n}(t) = 2^{m/2}\psi(2^m t - n) \tag{33}$$

The wavelet series parameters in (31) can be defined as

$$\begin{aligned} a(m_0, n) &= \langle f, \phi_{m_0,n} \rangle = 2^{m/2} \int_{-\infty}^{\infty} f(t)\phi_{m_0,n}(t)\,dt \\ d(m, n) &= \langle f, \psi_{m,n} \rangle = 2^{m/2} \int_{-\infty} f(t)\psi_{m,n}(t)\,dt \end{aligned} \tag{34}$$

Similar to scaling and wavelet functions, the coefficients *a* are called the scaling parameters and the *d* parameters are called the detail parameters.

If the signal's scaling functions and the wavelets are discrete in time, then (31) is called the discrete wavelet transform. DWT consists of two series expansions, one for approximation and the other for details of the sequence. The formal definition of DWT of a sequence $x[k], 0 \le k \le N-1$ is

$$DWT\, f(t) = S_\phi(m_0, n) + T_\psi(m, n) \tag{35}$$

where

$$\begin{aligned} S_\phi(m_0, n) &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x[k]\phi_{m_0,n}[k] \\ T_\psi(m, n) &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x[k]\psi_{m,n}[k], m \ge m_0 \end{aligned} \tag{36}$$

The sequence x[*k*] can be recovered from the DWT coefficients S_ϕ and T_ψ with

$$x[k] = \frac{1}{\sqrt{N}} \sum_k S_\phi(m_0, n)\phi_{m_0,k}[k] + \frac{1}{\sqrt{N}} \sum_{m=-\infty}^{m_0} \sum_k T_\psi(m, n)\psi_{m,n}[k] \tag{37}$$

2.3 Autocorrelation Analysis

In time-domain correlation analysis we examine signal's linear correlation with the signal itself or with another signal. In the analysis a correlation function is defined, where *x* and *y* are variables and time difference is *τ*. Function is called autocorrelation

function if the variables are the same and cross-correlation function if they are different. Figure 7 shows the autocorrelation of a sine wave with frequency of 3 Hz.

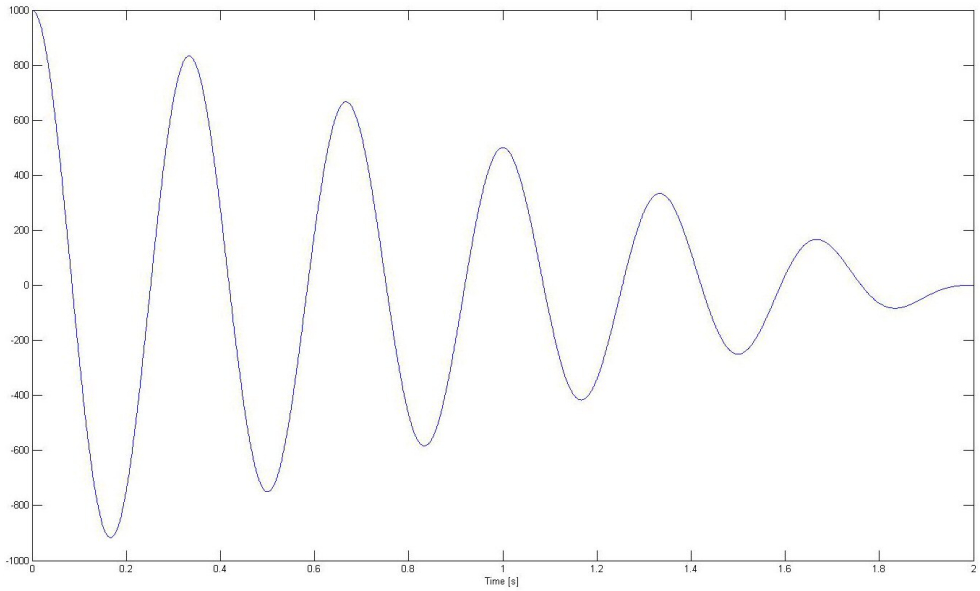

Figure 7: Autocorrelation of a 3 Hz sine wave with sampling frequency of 1000 Hz

Figure 7 shows that there are three peaks inside 1 second so the period of the sine wave is 0.33 seconds and the frequency is 3 Hz. Using autocorrelation analysis on signals with random noise allows us to find periodic components from them and determine their period and frequency.

Signal x(t) autocorrelation function $r_{xx}(\tau)$ is

$$r_{xx}(\tau)=\sum_{n=-\infty}^{\infty}[x(n)x(n-\tau)] \tag{38}$$

and the cross-correlation between x(t) and y(t) is

$$r_{xy}(\tau)=\sum_{n=-\infty}^{\infty}[x(n)y(n-\tau)] \tag{39}$$

where τ gains values of $\tau=\ldots,-2,-1,0,1,2,\ldots$. The value of correlation grows the more the signals correlate with each other. In Figure 7 we can see that when x(n) and x(n-τ) both have negative values, the autocorrelation has a positive value and there is a high point in the graph. Same happens when x(n) and x(n-τ) both have positive values. When x(n) is positive and x(n-τ) is negative or the other way around, the correlation gains negative values and there is a pit in the graph.

After removing the mean from (38) and (39) we get autocovariance and cross-covariance functions

$$c_{xx}(\tau)=r_{xx}(\tau)-\mu_x^2 \tag{40}$$

$$c_{xy}(\tau)=r_{xy}(\tau)-\mu_x \mu_y \tag{41}$$

respectively.

Covariance functions represent the common correlation analysis' correlation coefficient scaled by the variables' standard deviation σ_x and σ_y. Perfect correlation gives scaled values that correspond to +1 or -1. Using autocovariance functions we can for example study the period of vibrations in the signal. With cross-covariance we can determine the characteristic delay between point *x* to point *y* because the functions maximum is located at this time value.

3 EXPERIMENTS WITH DATA

Data used for experiments in this masters thesis is obtained from an earth crushing facility. The setup of the facility is shown in Figure 8.

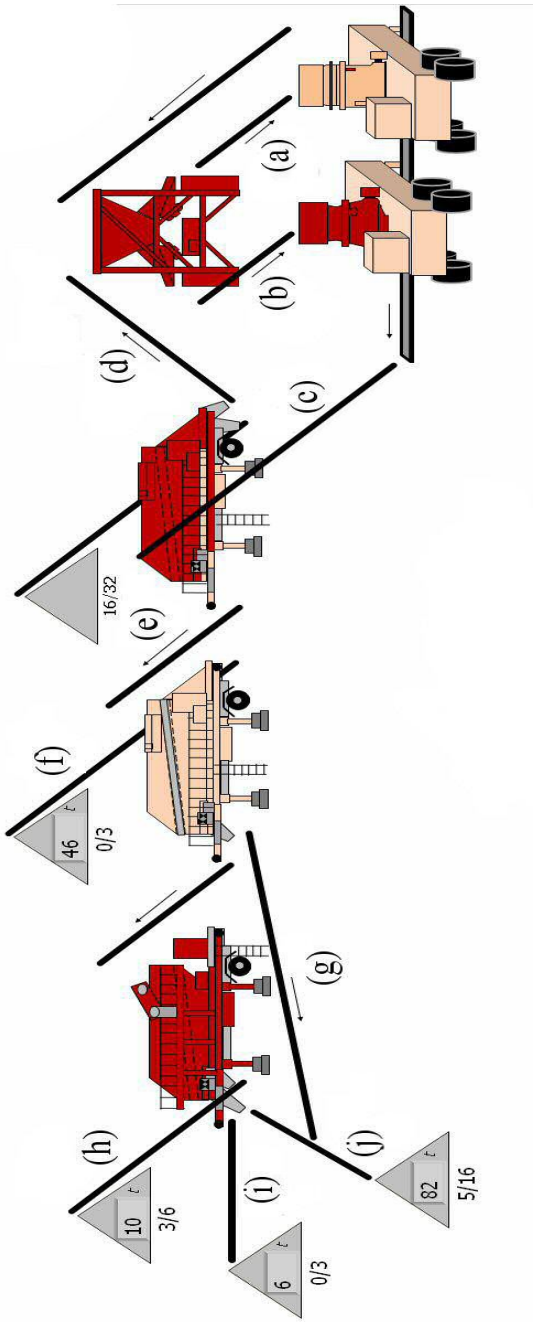

Figure 8: Earth crushing facility

In the facility rocks are crushed into gravel of different sizes which are sorted into different piles. Gravel is moved around to different parts of the facility using conveyor belts. Conveyor belts (a) and (b) feed earth into two grinders called Proto and Sequor respectively. From there conveyor belt (c) takes the ground stone to the first separating station where gravel with circumference of 16/32 is separated from the crushed earth and taken to its own pile and gravel bigger than 16/32 is taken back into the grinding machine with conveyor belt (d). Conveyor belt (e) takes the smaller gravel forward to the next separating station where gravel with the sizes 0/3 and 5/16 are separated from the gravel. Then the gravel is taken to the final sorting station where the gravel is sorted into piles of 3/6, 0/3 and 5/16 with conveyor belts (h), (i) and (j) respectively. Power required by the conveyor belts is measured from ten different conveyor belts marked from (a) to (j). The required power depends on the amount of gravel on the conveyor belt. The level of the two grinders after conveyor belts (a) and (b) is also measured with ultrasonics.

3.1 Measuring and Preprocessing the Data

The data has been obtained with three different measuring frequencies: 25000 Hz 1630 Hz and 1 Hz using measurement setup shown in Figure 9.

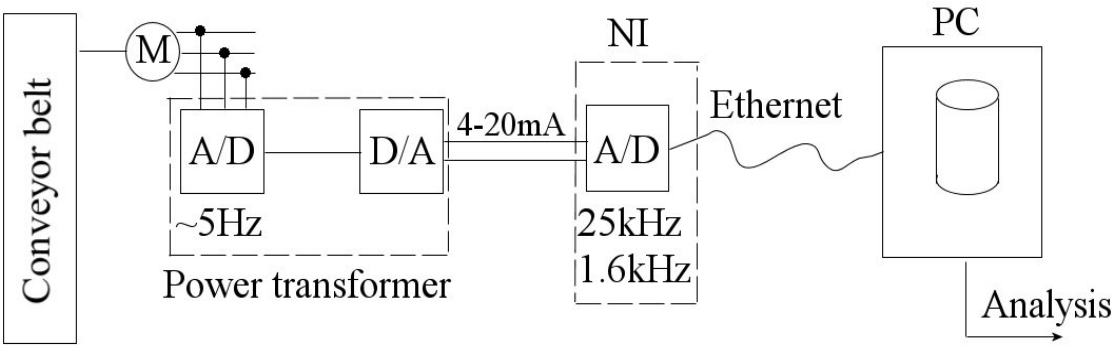

Figure 9: Measurement setup

The power required by the motor rotating the conveyor belt is first transformed by power transformer into electric current. The minimum current measured is 4 mA and the maximum current is 20 mA and the measurement data can be scaled into power values using Table I.

Table I: Scaling range of different measurement channels

Channel	Scaling range
Level Sequor	0-300 mm
Level Proto	0-300 mm
Power Proto Feed	0-20 kW
Power Sq Feed	0-20 kW
Power Screen3 5-16	0-10 kW
Power Screen3 3-6	0-6 kW
Power Sequor	0-20 kW
Power Circulating	0-20 kW
Temperature	-50-200 C
Power Screen2 0-3	0-6 kW
Power Screen3 0-3	0-6 kW
Power Screen2 11-16	0-6 kW
Power Screen2 Feed	0-20 kW

A zero order hold (ZOH) model is used to hold each sample value for ~0.2 seconds so the sampling frequency after power transformer is ~5 Hz.

Figure 10 shows an example of unsampled data using measuring frequency of 1 Hz.

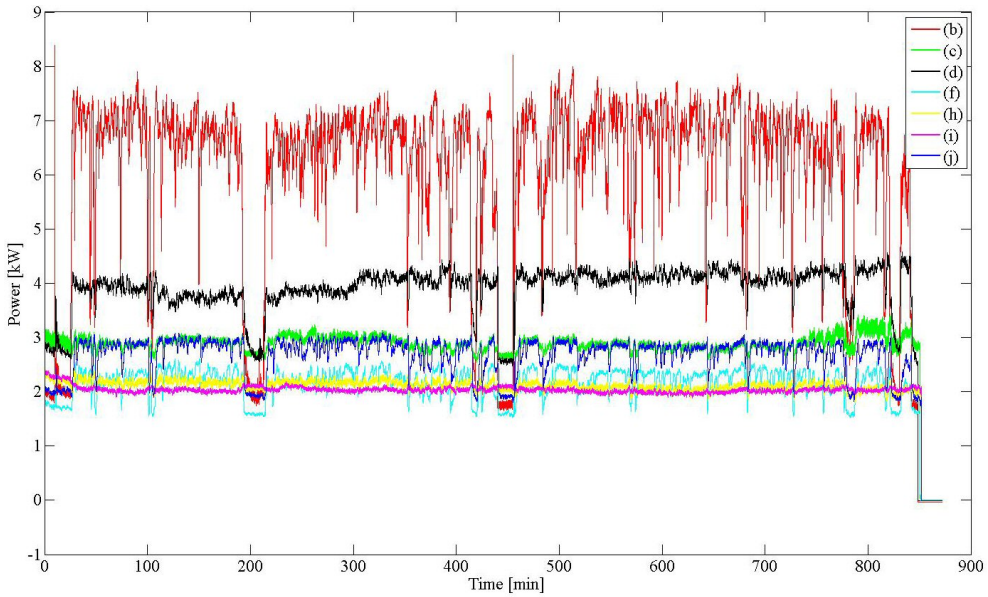

Figure 10: Power required by conveyor belts with 1 Hz measuring frequency

Figure 11 shows an example of unsampled data using measuring frequency of 1630 Hz.

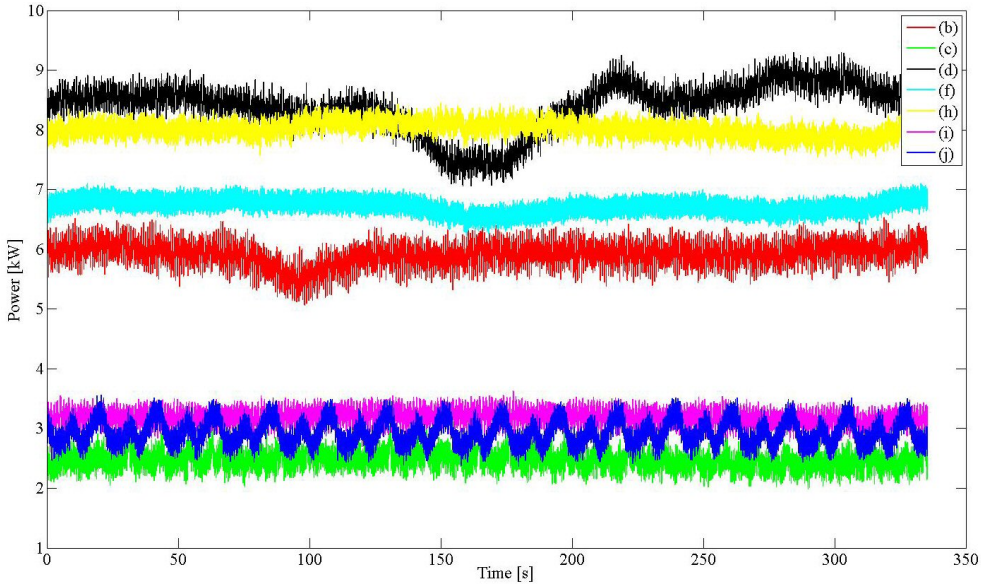

Figure 11: Power required by conevyor belts with 1630 Hz measuring frequency

Figure 12 shows unsampled data obtained using measuring frequency 25000 Hz.

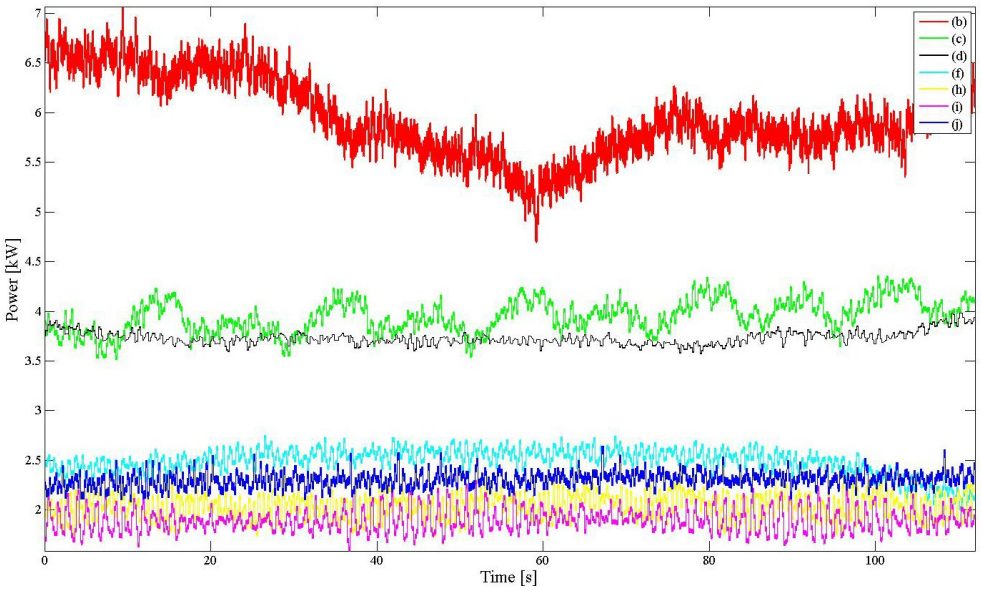

Figure 12: Power required by conevyor belts with 25000 Hz measuring frequency

Figure 13 shows a short segment of data obtained from conveyor belt (c).

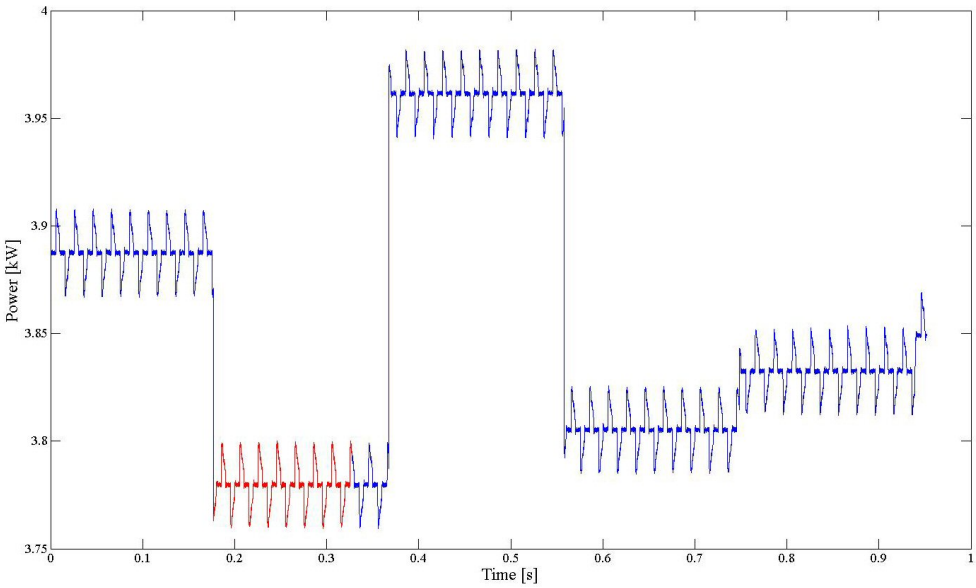

Figure 13: A segment of data obtained from conveyor belt (c)

Figure 13 shows that the measurement is held for ~0.2 seconds and then the signal jumps to the next measurement point. To analyze the data it needs to be sampled first to obtain information from the system itself and to remove noise from the data. The sampling is done by taking the average of each ZOH interval and using the average as the new measurement point. Using high measurement frequency the obtained average will be better and the noise from the measuring system can be removed almost completely. After sampling the measuring frequency is ~5Hz. For example in Figure 13 the red signal is used to calculate the average of that piece of the data. Figure 14 shows a sampled segment of data obtained from conveyor belt (c).

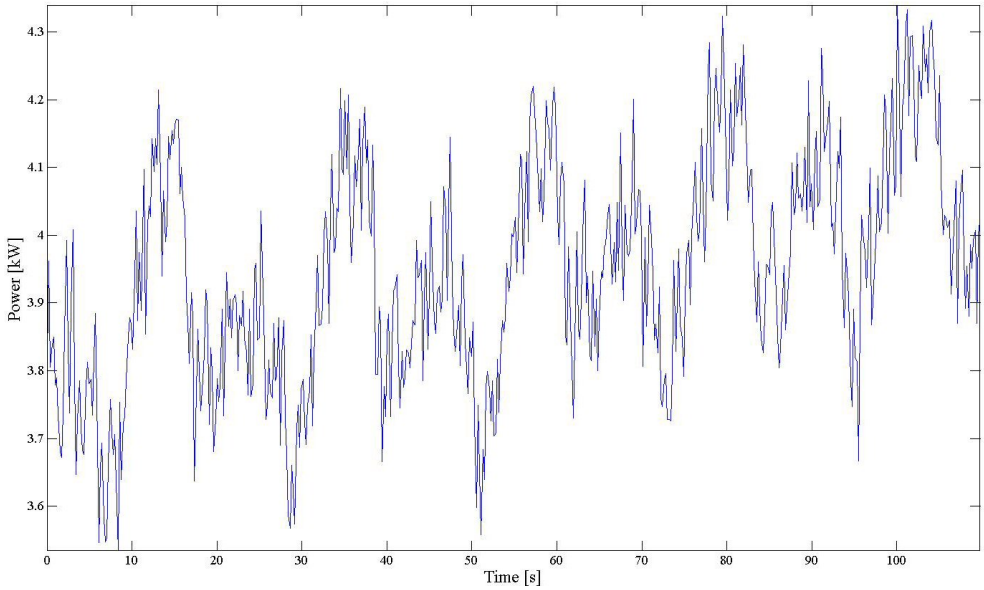

Figure 14: A sampled segment of data obtained from conveyor belt (c)

The data obtained from the facility is cyclic in nature because of the rotating conveyor belts. That is why periodic events like faults appear periodically in the data and the periodic events can be found using frequency domain methods. Possible sources of periodic signals are a crooked axle which is present at the gearbox of each conveyor belt motor, rolls underneath the conveyor lines which makes the loose belt flap and vibrate and belts are crooked which can cause vibrations. The conveyor belts are 14,5 meters long and they rotate at the constant speed of 1,38 m/s so a full cycle is 29 meters long and takes 21 seconds.

3.2 Implementation of Frequency Domain Analysis Methods

Four different methods that were presented in chapter 2 are used in two different experiments to compare their abilities at finding periodic signals from the measured data sets. The four methods used are autocorrelation, FFT, STFT and CWT. Data sets from different operation points (during idle time and during workload time) are used in the experiments.

STFT is done using spectrogram command in Matlab. By trying different window functions on the data, Hann window was found to be the best window at detecting periodic signals from the data. Figure 15 shows the Hann window with window length 30.

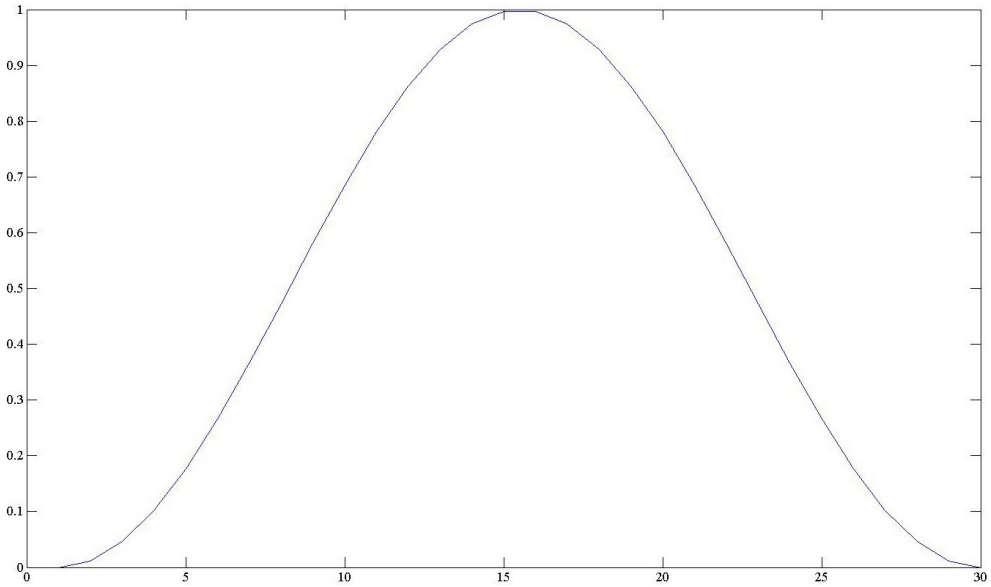

Figure 15: Hann window

In the first experiment window length of 30 and overlapping of 50% was used to obtain a good time resolution. In the second experiment window length of 200 and overlapping 50% was used to obtain good frequency resolution.

CWT is done by using cwt and scal2freq commands in Matlab. Different wavelets were tried out and the morlet wavelet, shown in Figure 6, was found to be the best at finding periodic signals from the data. 2^10 scales were used which represent the scaling parameters of CWT. The cwt function of Matlab begins with scale one and then changes to translation parameter to go through the whole signal and then moves on to scale two and repeats until scale 2^10 is reached.

3.3 Simulated Signal

In the first experiment a simulated square wave signal $p = a \cdot d$ is added into the signal where a is the amplitude of the square wave and d is the square wave with amplitude 1. The purpose of the experiment is to see how well the different methods can find the simulated signal from the data. Figure 16 shows the square wave used in the simulated data experiment.

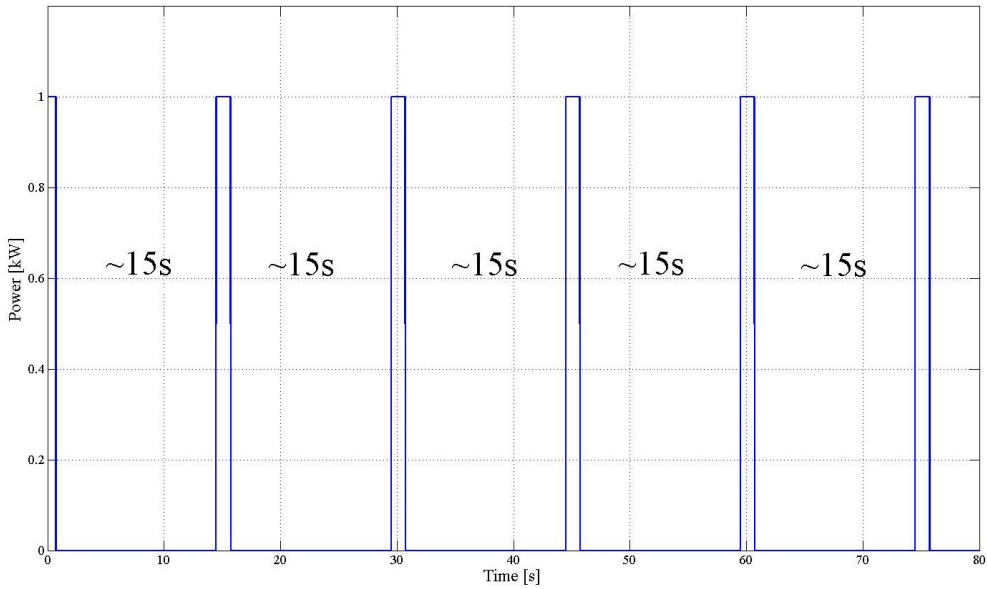

Figure 16: Simulated periodic square wave

In the experiment the period of the simulated signal stays constant and the square wave that lasts for a second repeats every ~15 seconds. The signal could for example be caused by a broken spot on the conveyor belt which causes impulsive periodic peaks once per rotation in the power requirements of the motor running the conveyor belt. Figure 17 shows the sampled version of the simulated signal.

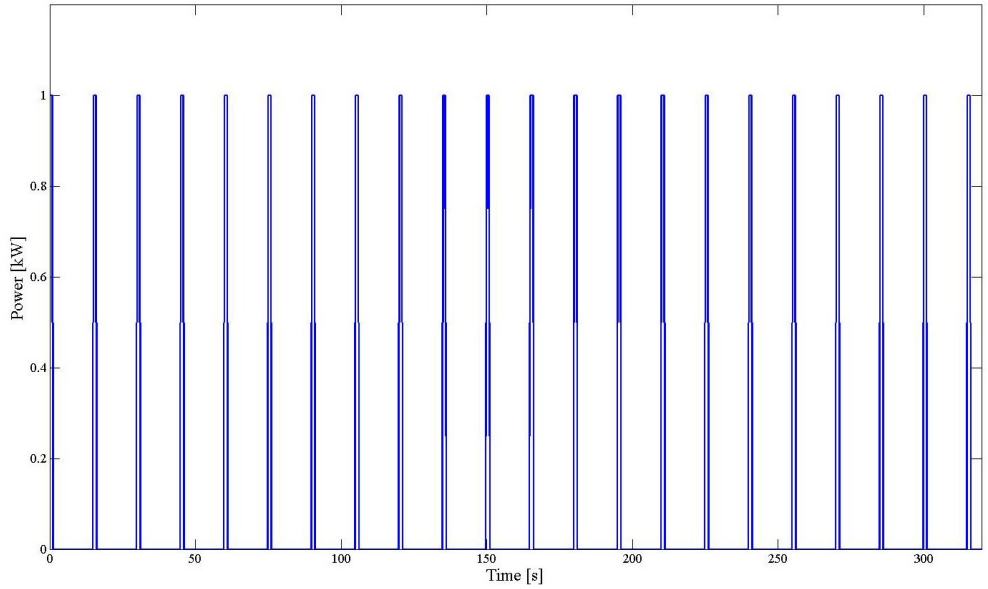

Figure 17: Simulated periodic square wave sampled

Figure 18 shows the results of different methods applied to the simulated signal to see what kind of results can be expected from the methods.

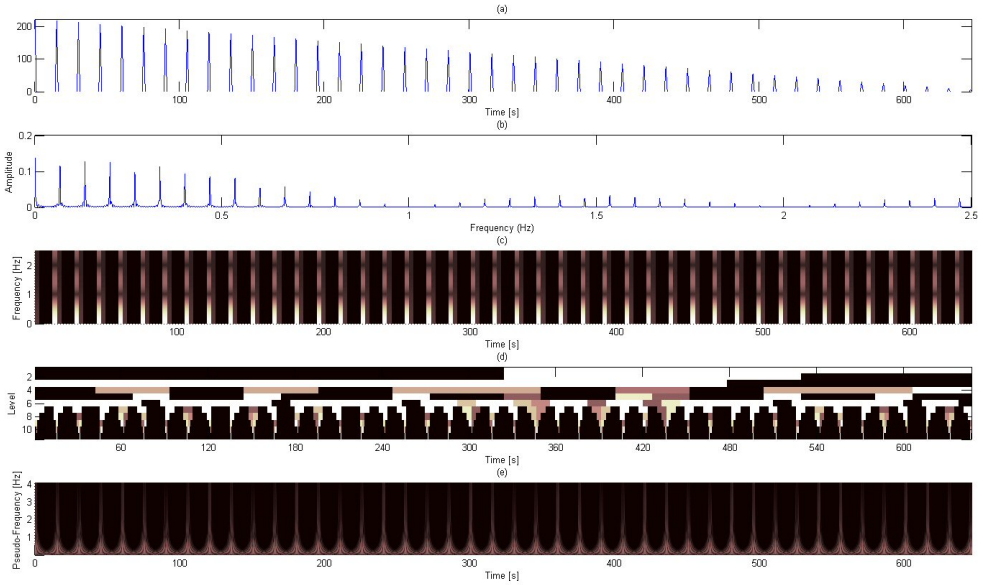

Figure 18: Different methods applied to the simulated signal (a) Autocorrelation (b) FFT (c) STFT (d) DWT (e) CWT

Figure 18 shows that in autocorrelation the square wave creates a clear spike at the location of the square wave, in FFT the peaks are spread on different frequencies in half sphere like form, in STFT the square wave is a square block at the location of the square signal and in CWT there are spikes at the location of the square signal.

3.4 Experiment 1 With the Data and the Simulated Signal

In the first experiment the ability of four methods (autocorrelation, FFT, STFT and CWT) in finding the simulated square wave signal from the data is analyzed and compared using the following steps.

1. Choose an analysis method (autocorrelation, FFT, STFT, CWT)
2. Obtain measurement data d.
3. Choose amplitude a
4. Create a simulated square wave signal $p = A \cdot s$ where a is the amplitude of the signal
5. Add simulated signal to measurement data $x = d + p$
6. Apply the chosen method to the data $y_1 = \text{method}(d)$
7. Apply the chosen method to p $y_2 = \text{method}(p)$
8. Apply the chosen method to x $y_3 = \text{method}(x)$
9. Remove y_1 from y_3 $y_4 = y_3 - y_1$
10. Find the location q_1 of the square waves in y_2

11. Find the location q_2 of the square waves in y_4
12. Obtain the mean value of the amplitude of the square waves in the signal y_2
    $m_1 = \text{mean}[y_2(q_1)]$
13. Obtain the mean value of the amplitude of the square waves in the signal y_4
    $m_2 = \text{mean}[y_4(q_2)]$
14. Find how much the noise from the data affects the chosen method's ability to find the simulated signal from x by comparing m_2 with m_1

    $$c_i = \frac{m_2}{m_1}, \qquad i = 1 \ldots k$$

    where k is the a number of amplitudes used (k = 5).
15. Repeat from step 3 with a different amplitude A (A = 1, 0.6, 0.3, 0.15, 0.1)
16. Plot c_i
17. Repeat from step 1 with a different method

The reduction in step 9 is done because the noise from the data raises the level of the signal slightly and makes it higher than the level in the plain simulated signal p. If the reduction was not done, the division in step 14 would give wrong results, because m_2 would be larger than m_1. The differences between the signals are very small, so even tiny changes have a big impact.

In step 14 the division is done between the mean obtained from the data with the simulated signal in it and the mean obtained from the plain simulated signal. This is done because different methods give different magnitudes of results. For example autocorrelation can give results of magnitude 10 and CWT gives results of magnitude 0.01. So to compare the strengths of the signal on equal standing, the results are first compared to transformation done on the plain simulated signal without noise using the same method used on the data with the simulated signal. This comparison gives proportion on how much the results weaken because of the noise from the data. The proportion depends on the qualities of the different methods and vary with different amplitudes and lengths of the simulated signal and the noise caused by the data. By comparing these proportions the methods can be compared and the best method at detecting the simulated signal from data can be found. The results of the experiments are given in chapter 4.1. Figure 19 shows a simplified version of the key points in the first experiments set up.

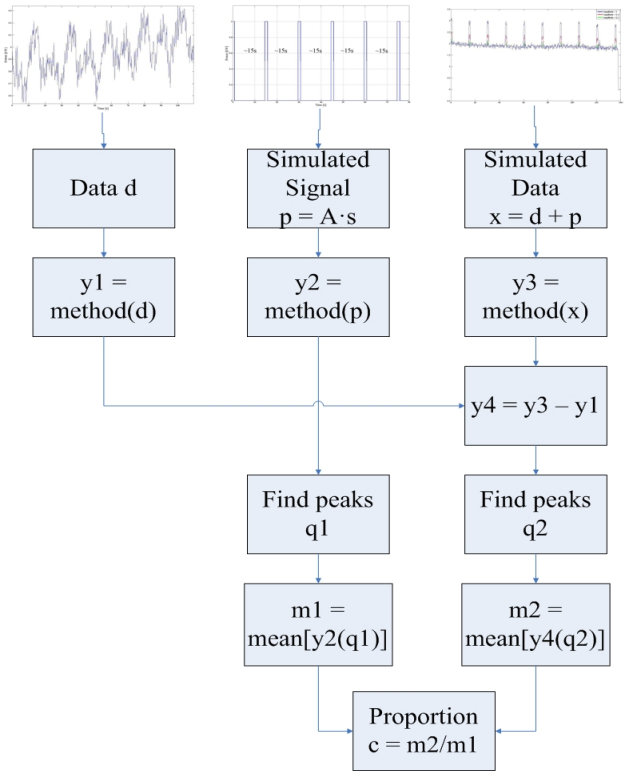

Figure 19: Experiment 1

Figure 19 shows the steps 2 and 4-14 of the experiment 1.

3.5 Experiment 2 With the Data

In the second experiment the ability of two methods (STFT and CWT) at finding the periodic signals from the data is analyzed and compared using the following steps.

1. Obtain measurement data sets d_1 and d_2 from same location at different operation points
2. Apply FFT to d_1 and d_2 $x_1 = \text{fft}(d_1)$, $x_2 = \text{fft}(d_2)$
3. Search for frequencies f_i (i = 1, 2,..., k) from x_1 and x_2 that appear in both of them
4. Choose a data set d_k (k = 1, 2)
5. Choose an analysis method (STFT, CTW)
6. Choose a found frequency f_i found from x_k and set all other frequencies to 0 to obtain FT x_i
7. Perform inverse FT on x_i $j_i = \text{ifft}(x_i)$
8. Remove the frequency from the chosen data set $r_i = d_k - j_i$
9. Apply the chosen analysis method on d_i $n_1 = \text{method}(d_i)$
10. Apply the chosen analysis method on y_i $n_2 = \text{method}(y_i)$
11. Apply the chosen analysis method on y_2 $n_3 = \text{method}(r_i)$
12. Find the chosen frequency f_i from n_1 $S_1 = n_1(f_i)$
13. Find the chosen frequency f_i from n_2 $S_2 = n_2(f_i)$

14. Find the chosen frequency f_i from n_3 $S_3 = n_3(f_i)$
15. Find how strong the amplitude of the frequency is in S_1 in proportion to S_2 at different times by removing S_3 from S_1 and comparing the remainder to S_2
$$e(t) = \frac{S_1(t) - S_3(t)}{S_2(t)}, \qquad t = 1,2,\ldots,r$$
16. Find the maximum value of q(t) $z_i = \max[e(t)]$
17. Repeat from step 6 with the next frequency
18. Plot z_i
19. Repeat from step 5 with another method
20. Repeat from 4 with another dataset

The experiments with the data obtained from the earth moving facility are done in a similar way as done with the simulated data. The results of the experiment are given in chapter 4.2. Figure 20 shows a simplified version of the key points in the second experiment set up.

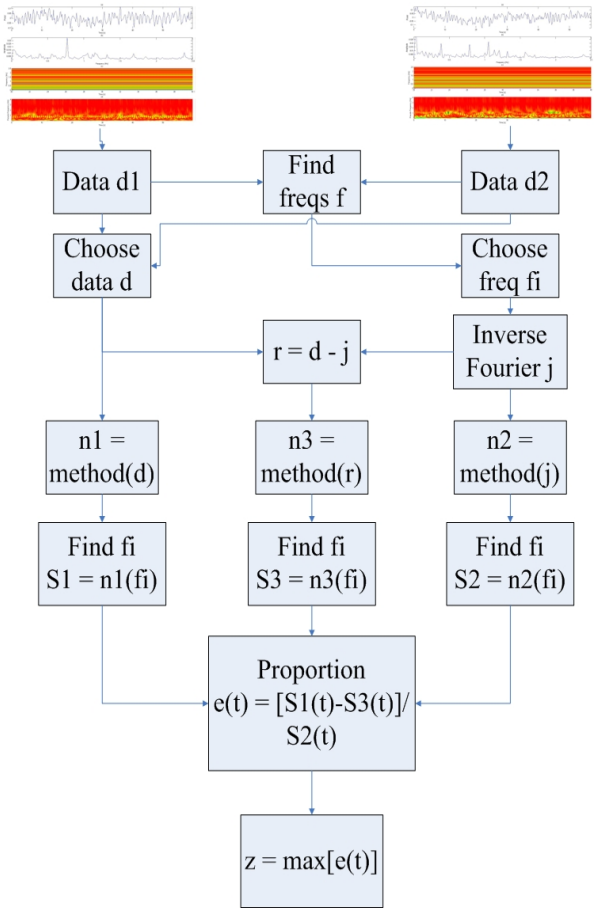

Figure 20: Second experiment

Figure 20 shows steps 2-16 of experiment 2.

4 EXPERIMENT RESULTS

In the first experiment the simulated signal is inserted into the sampled data obtained from conveyor belt (j) during idle time when there was no earth on the conveyor belt and during load time when the machines are transferring earth. The measurements were done with measuring frequency 25 000 Hz and after sampling the frequency is 5 Hz. The combination of the data and the simulated signal with different amplitudes during idle operation is shown in Figure 21.

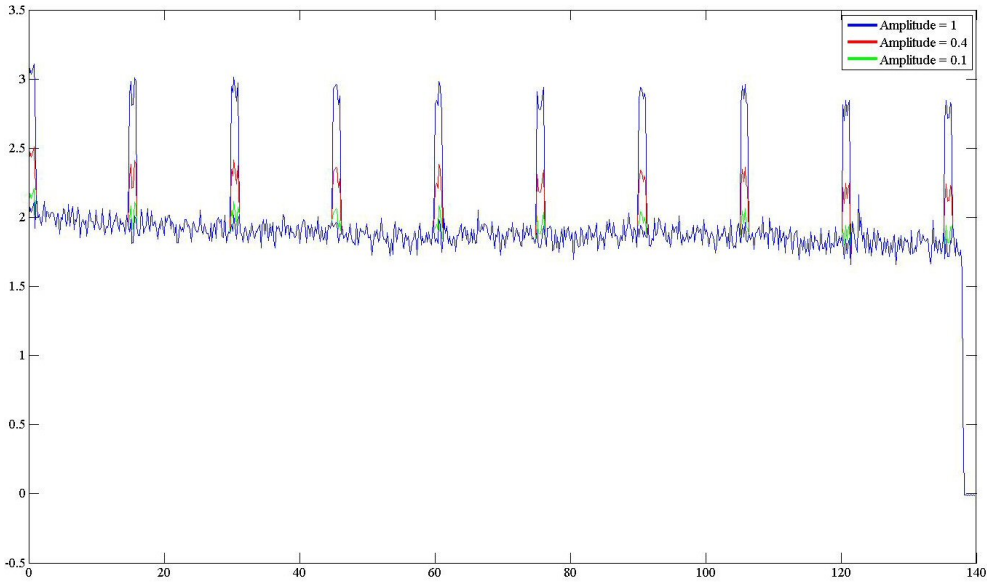

Figure 21: Simulated signal with different amplitudes and sampled 25 kHz data from conveyor belt (j)

Figure 21 shows that the simulated signal is easy to see from the data when the amplitude of the square wave is high. When the amplitude is lowered it becomes harder to see the square wave in the data.

The combination of the data and the simulated signal with different amplitudes during workload operation is shown in Figure 22.

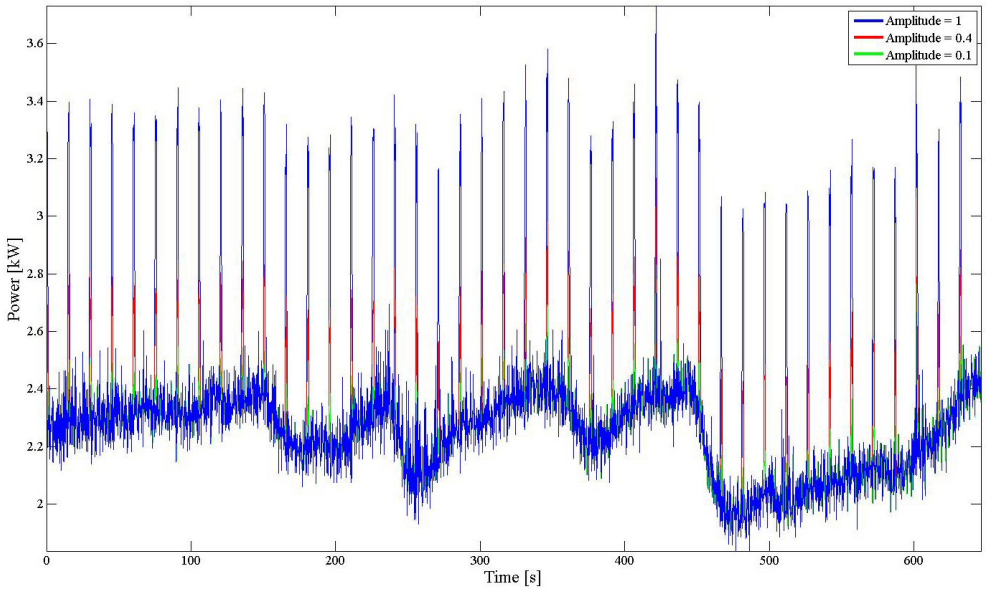

Figure 22: Simulated signal with different amplitudes and sampled 25 kHz data from conveyor belt (j)

Figure 22 shows similar results to Figure 21. Because the noise from the data is higher during workload operation, the lower amplitude square waves become even harder to see.

4.1 Results of Experiment 1

The experiments introduced in chapter 3.4 are applied on the data seen in Figures 21 and 22 . Figures 23 and 24 show an example of steps 6 and 8. Figure 23 shows the different methods are applied on the data obtained from conveyor belt (j) without the simulated signal in it.

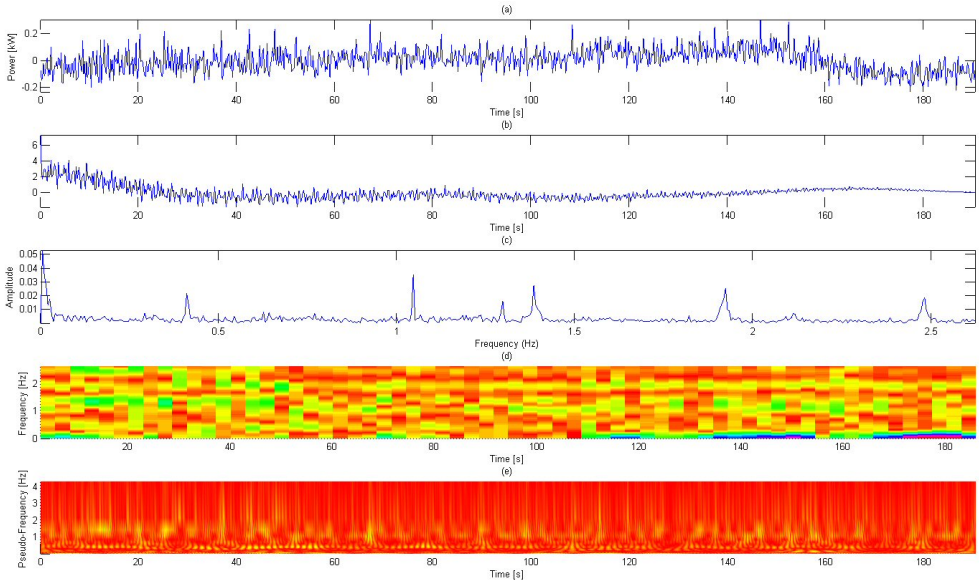

Figure 23: Experiment 1 step 5 methods on data (a) Data (b) Autocorrelation (c) FFT (d) STFT (e) CWT

Figure 23 shows that there are some frequencies present in the data and how they appear in the different transforms. The frequencies can't be found using autocorrelation and the results are very noisy, the frequencies are clear and easy to see using FFT, in STFT the frequencies can be found, but the frequency resolution is bad because of the small window used and in CWT the strongest frequency at 1.1 Hz can be seen in the transform.

An example of the experiments done on the simulated signal is shown in Figure 18. Figure 24 shows the methods are applied on data with the simulated signal with amplitude 0.6 added in it.

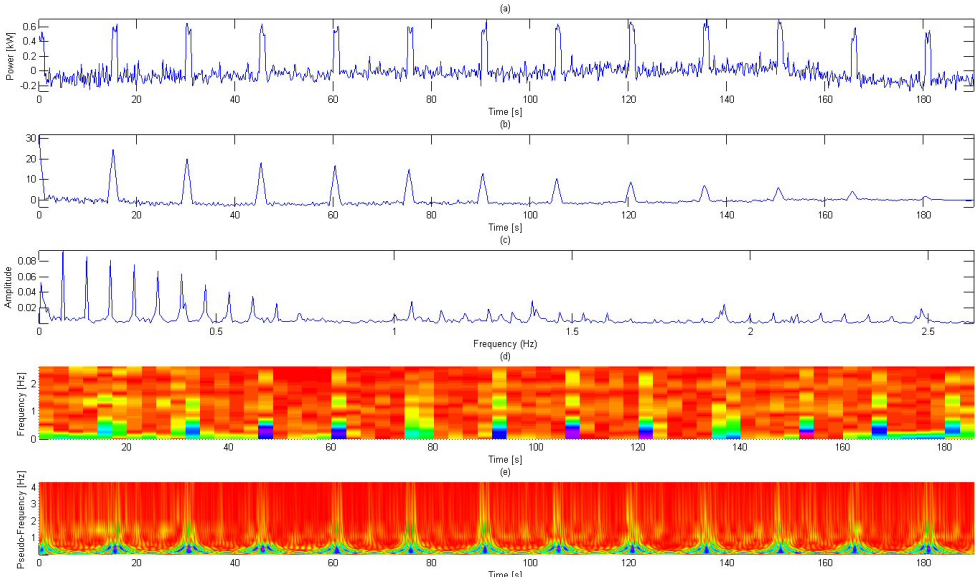

Figure 24: Experiment 1 step 7 methods on data with the simulated signal (a) Data (b) Autocorrelation (c) FFT (d) STFT (e) CWT

Figure 24 shows that the simulated square wave appears as different kinds of peaks in the transforms with the signals weakening because of the noise caused by the data. Similiar to Figure 18 in autocorrelation the square wave creates a clear spike at the location of the square wave, in FFT the peaks are spread on different frequencies in half sphere like form, in STFT the square wave is a square block at the location of the square signal and in CWT there are spikes at the location of the square signal.

Next in step 9 the results shown in Figure 23 are removed from the results shown in Figure 24. Figures 25 and 26 show an example how step 9 works with autocorrelation and simulated signal with various amplitudes.

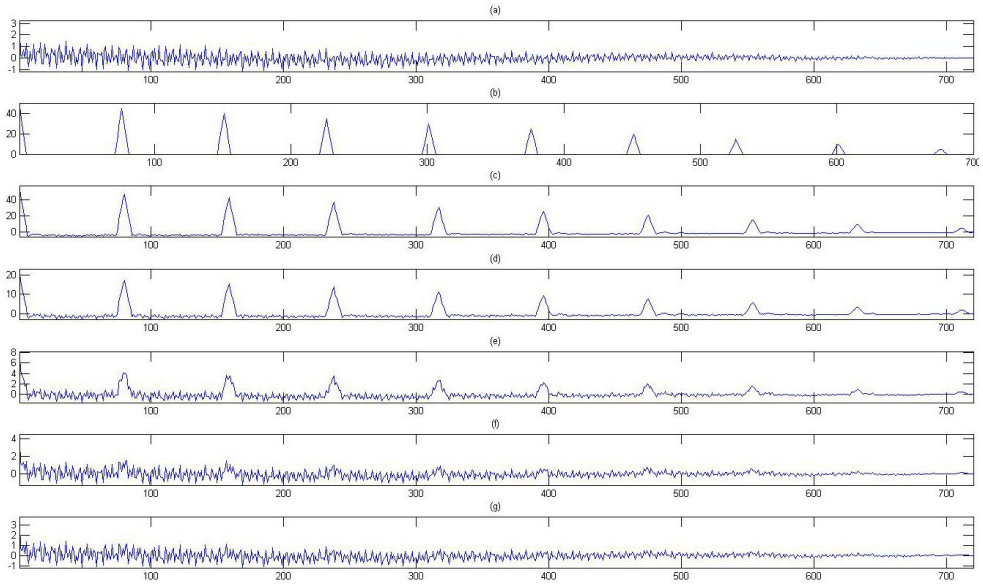

Figure 25: Experiment 1 step 8 Autocorrelation of y3 (a) Data (b) Simulated signal Amplitude 1 (c) Data and Simulated signal Amplitude 1 (d) Amplitude 0.6 (e) Amplitude 0.3 (f) Amplitude 0.15 (g) Amplitude 0.1

Figure 25 shows the autocorrelation done on d. Next the autocorrelation of d is removed from the autocorrelations seen in Figure 25.

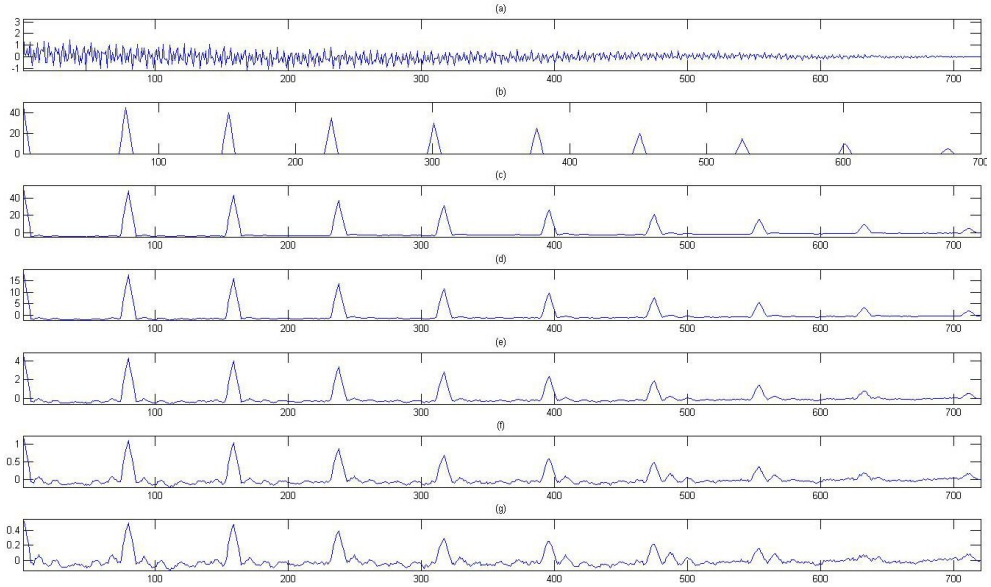

Figure 26: Experiment 1 step 8 Autocorrelation of y4 (a) Data (b) Simulated signal Amplitude 1 (c) Data and Simulated signal Amplitude 1 (d) Amplitude 0.6 (e) Amplitude 0.3 (f) Amplitude 0.15 (g) Amplitude 0.1

Figures 25 and 26 show that after removing the data from the simulated signal, the peaks are easier to find, but some noise still remains from the data. The strength of the peaks is lower compared to the strength of the peaks in the plain simulated signal.

For steps 10 and 11 peaks are found from the results of the methods. Figures 18 (a) and (b) show what kind of peaks the simulated signal creates in the autocorrelation and FFT of the data. Figure 27 shows an example of peaks found with STFT at a certain frequency. STFT returns a matrix where each row represents a different frequency and each column represents a different time location. By plotting a row vector it can be seen how the amplitude varies over time at a certain frequency.

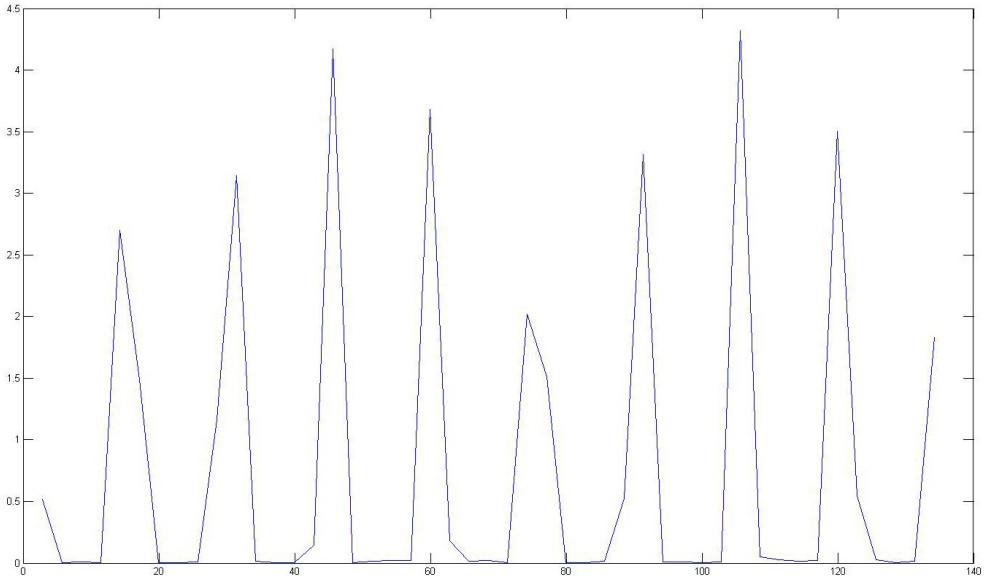

Figure 27: Experiment 1 step 9 STFT peaks

Figure 27 shows that there is a peak every 15 seconds corresponding to the simulated square wave. Calculating the mean from the amplitudes of those peaks the result for steps 12 and 13 can be obtained.

Figure 28 shows an example of peaks found with CWT at one scale. CWT returns a matrix similar to the matrix returned by STFT with time represented by the columns and frequency represented by rows. By plotting a row vector it ca be seen how the amplitude of the signal varies over time.

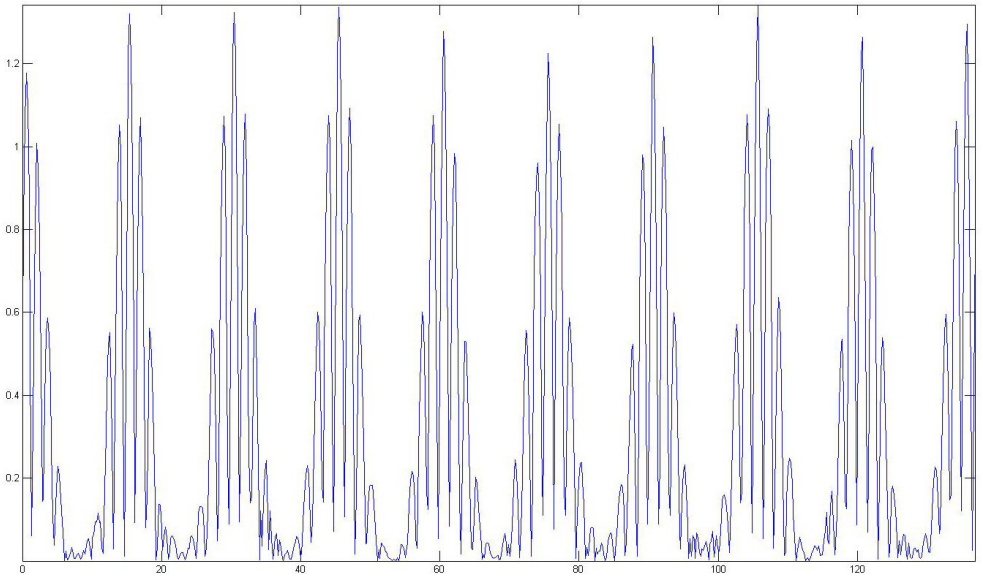

Figure 28: Experiment 1 step 9 CWT peaks

Figure 28 shows that same as with STFT, there is a peak every 15 seconds

corresponding to the simulated square wave.

For steps 12-14 the mean amplitude of the peaks found with the different results are calculated and then the mean value m_2 of the peaks found from y_4 is divided with the mean value m_1 of the peaks found from y_2. By repeating these steps with simulated signals with different amplitudes, the results shown in Figures 29, 30 and 31 are obtained. Figure 29 shows the results of experiment 1 done on data obtained from conveyor belt (j) during idle operation time.

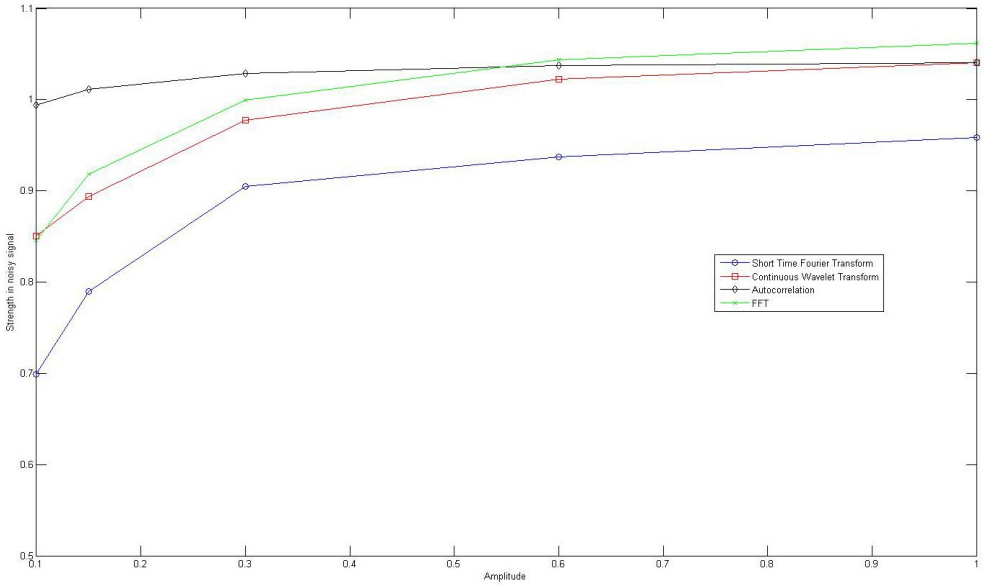

Figure 29: Experiment 1 step 15: Proportion n during idle operation. Data segment length 700.

Figure 29 shows that autocorrelation gives strong results even when the amplitude of the signal is weak. STFT has the weakest results compared to the other methods.

Figure 30 shows the results of experiment 1 done on data obtained during full workload using a portion of the data that is the same size as the idle data set. Because of the full workload the data is noisier than during idle operation.

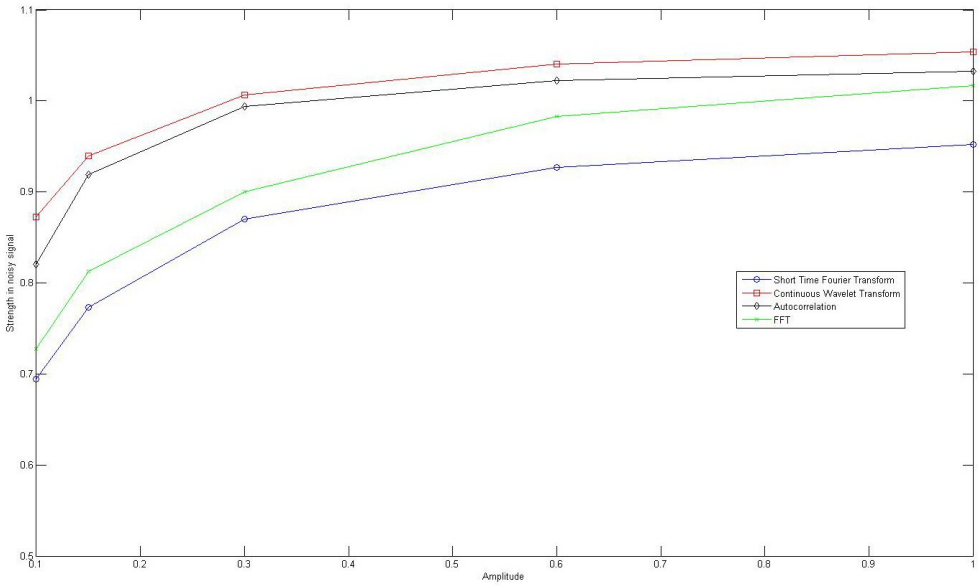

Figure 30: Experiment 1 step 15: Proportion n during workload operation. Data segment length 700.

Figure 30 shows that when the noise level of the data becomes stronger, the strength of the methods used goes down. CWT is the strongest method compared to the other methods on all the different amplitudes.

Figure 31 shows the results of experiment 1 done on same data as in Figure 30, but the full length of the data is used.

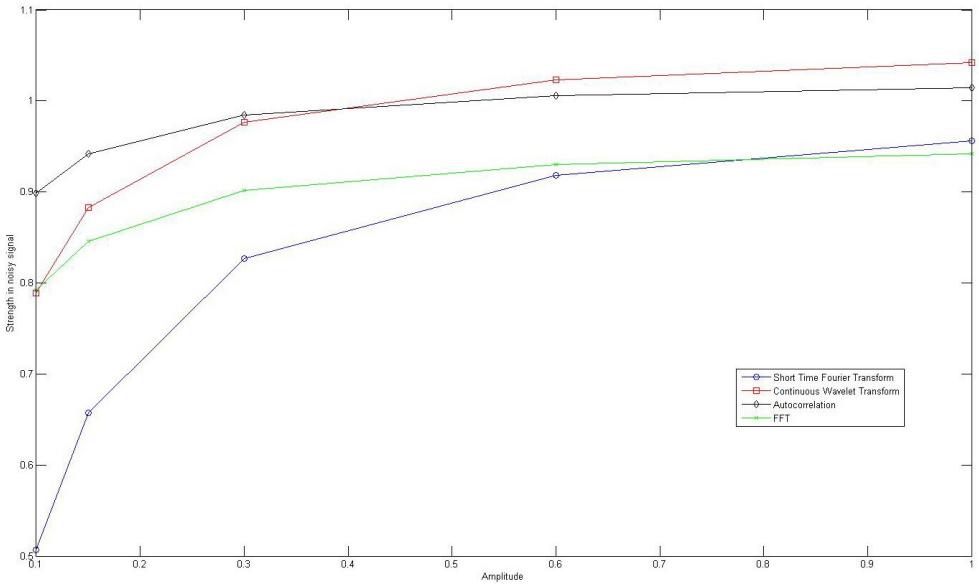

Figure 31: Experiment 1 step 15: Proportion n during workload operation. Data segment length 3000.

Figure 29 shows that when data segment is short and noise from data is low, autocorrelation and FFT are the best methods to use to find impulsive periodic signals from the data. In Figure 30, when the noise caused by the data becomes stronger, the performance of autocorrelation and FFT goes down and CWT becomes the best method to use. In Figure 31, when the data segment becomes longer, we can see that the performance of FFT is influenced a lot and its ability to find the simulated signal goes down drastically. The performance of CWT and STFT stay almost constant in all of the experiments. More conclusions are made from the results in chapter 5.1.

4.2 Results of Experiment 2

The experiments introduced in chapter 3.5 are first done with data obtained from conveyor belt (d). In Figures 32 and 33 the different methods are applied on two data sets obtained first during idle operation and then during full workload operation. The FFT of the signals are shown in Figure 32 (b) and 33 (b).

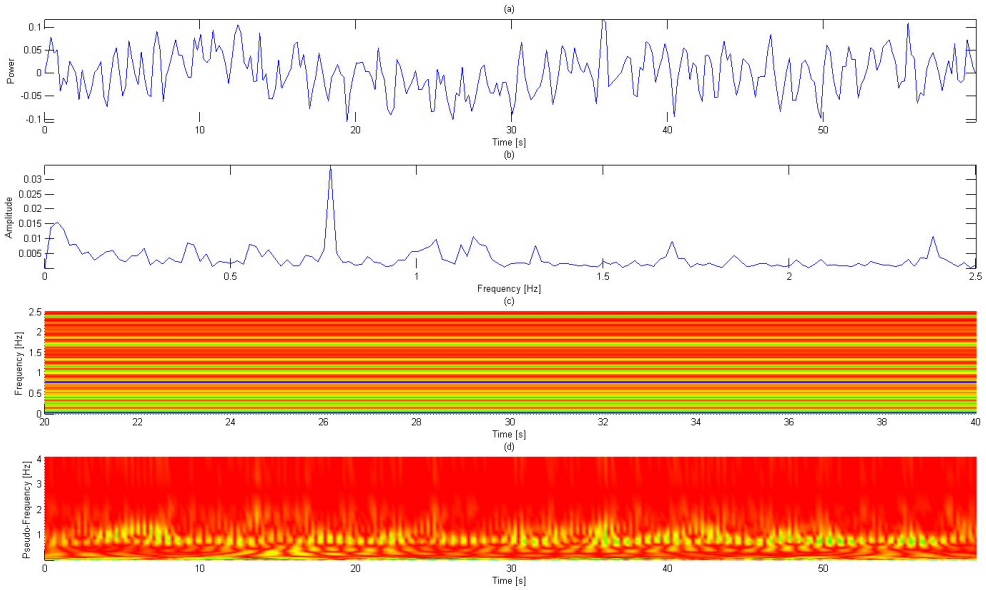

Figure 32: Experiment 2 step 2 idle operation (a) Data (b) FFT (c) STFT (d) CWT

Figure 32 shows the different methods applied to data obtained during idle operation.

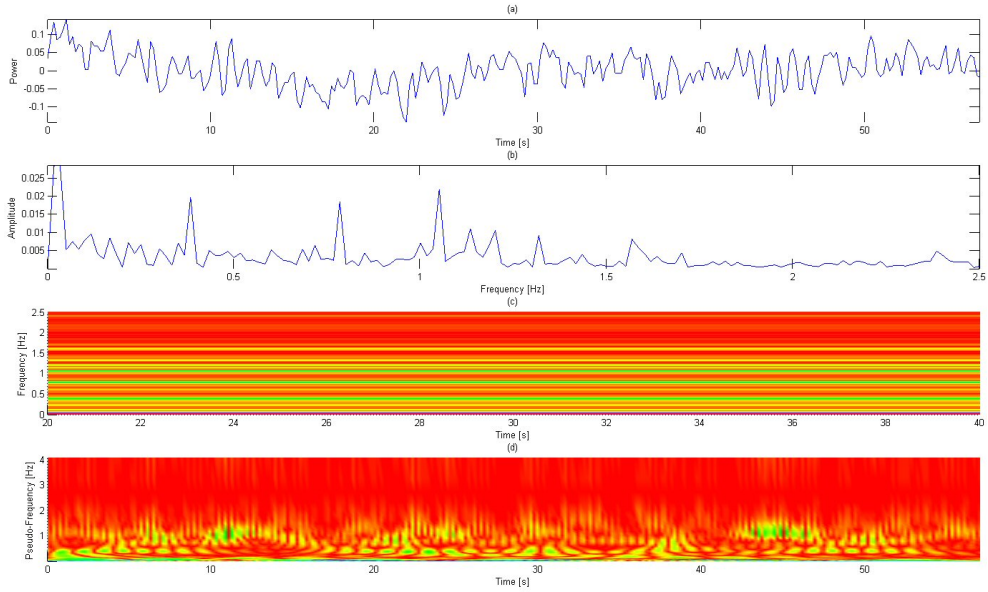

Figure 33: Experiment 2 step 2 workload operation (a) Data (b) FFT (c) STFT (d) CWT

Figure 33 shows the different methods applied to data obtained during load operation.

In step 3 frequencies that appear both in Figure 32 (b) and 33 (b) are found. Figure 34 shows the frequencies found from conveyor belt (d).

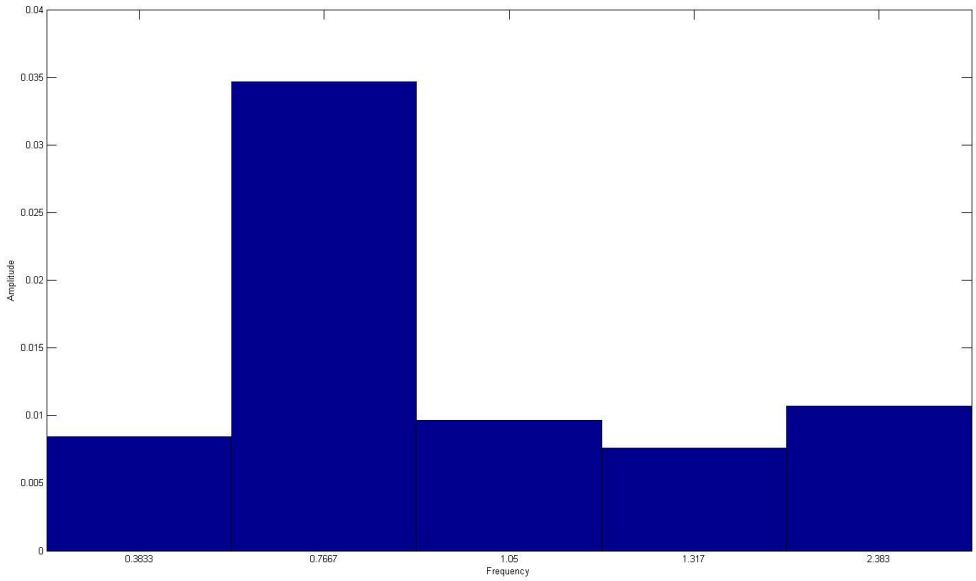

Figure 34: Experiment 2 step 3 found frequencies

In step 6 after finding the frequencies from the data, one of the frequencies is chosen and the rest are set to 0 as shown in Figure 35.

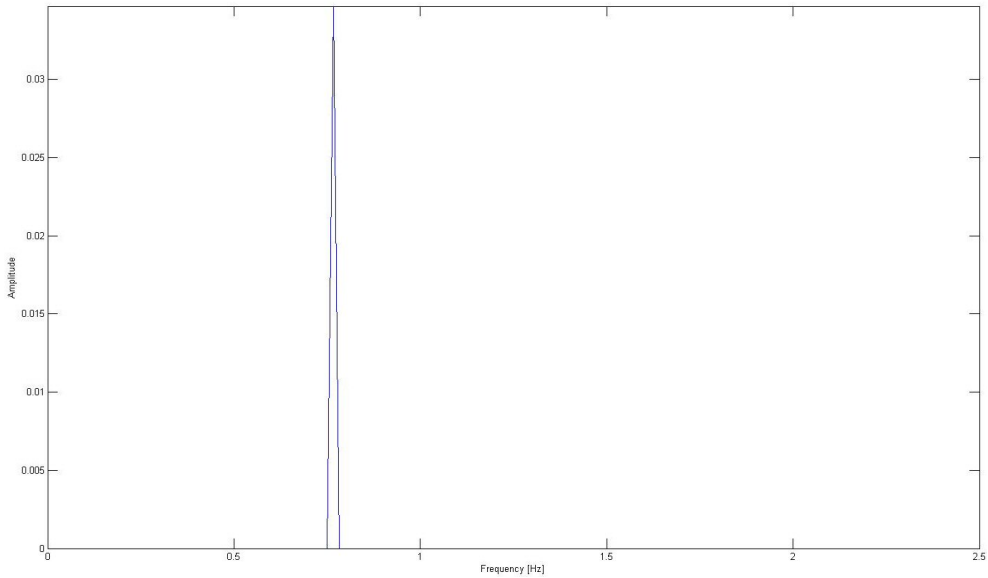

Figure 35: Experiment 2 step 6 FFT with only the chosen frequency

By applying inverse FT on FT shown in Figure 35, we get a signal shown in Figure 36 (a). Figures 36 (b), (c) and (d) show different methods used on the inverse transformed signal.

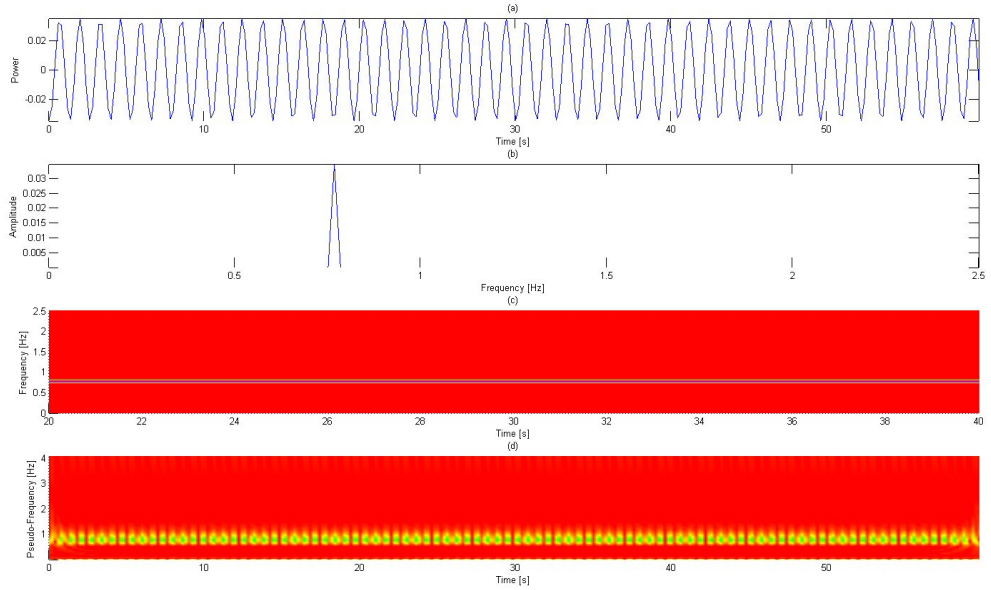

Figure 36: Experiment 2 steps 8 and 10: analysis of the chosen frequency (a) Periodic signal (b) FFT (c) STFT (d) CWT

In Figure 37 the signal shown in 36 (a) is removed from the data shown in Figure 32.

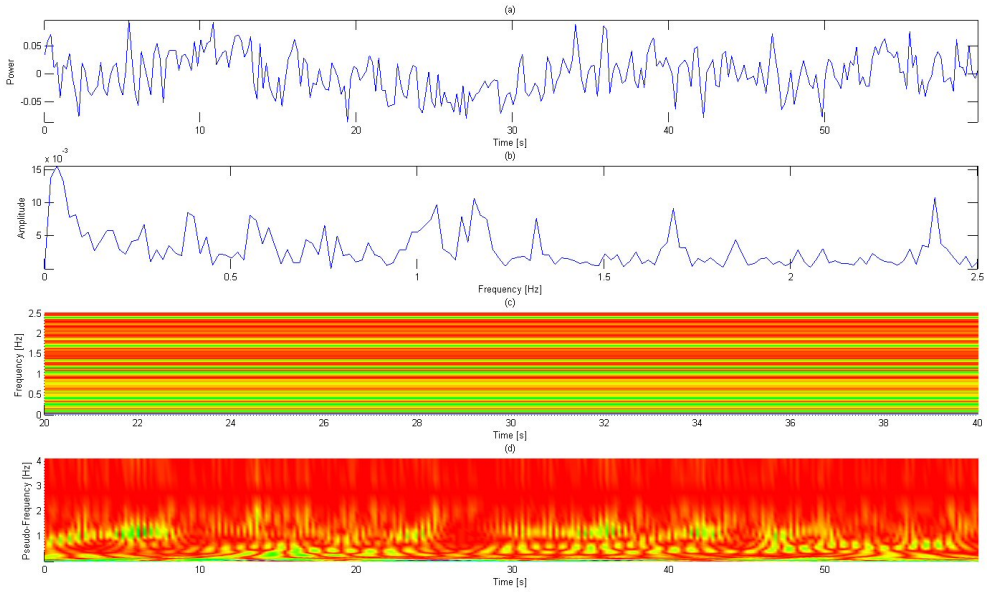

Figure 37: Experiment 2 step 11: analysis with the frequency removed

Figure 37 (b) shows that the frequency at 0.76 Hz is gone from the data.

In steps 12-14 the amplitude at the chosen frequency is found from Figures 32, 36 and 37. Figure 38 shows the amplitude of each frequency at certain time using STFT. As said before, STFT returns a matrix with rows representing frequency and columns representing time. By plotting a column, the amplitude of the signal can be seen at each frequency at one time location.

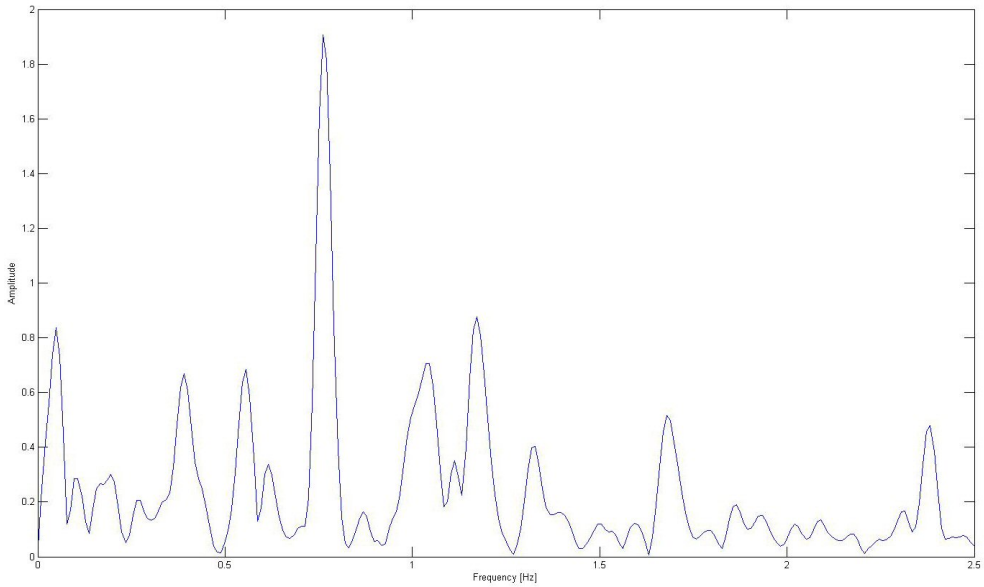

Figure 38: Experiment 2 step 12: Find frequency from STFT

The chosen frequency 0.76 Hz can be seen as the tallest peak in Figure 38.

Similar to the STFT, in Figure 39 shows a column from the CTW.

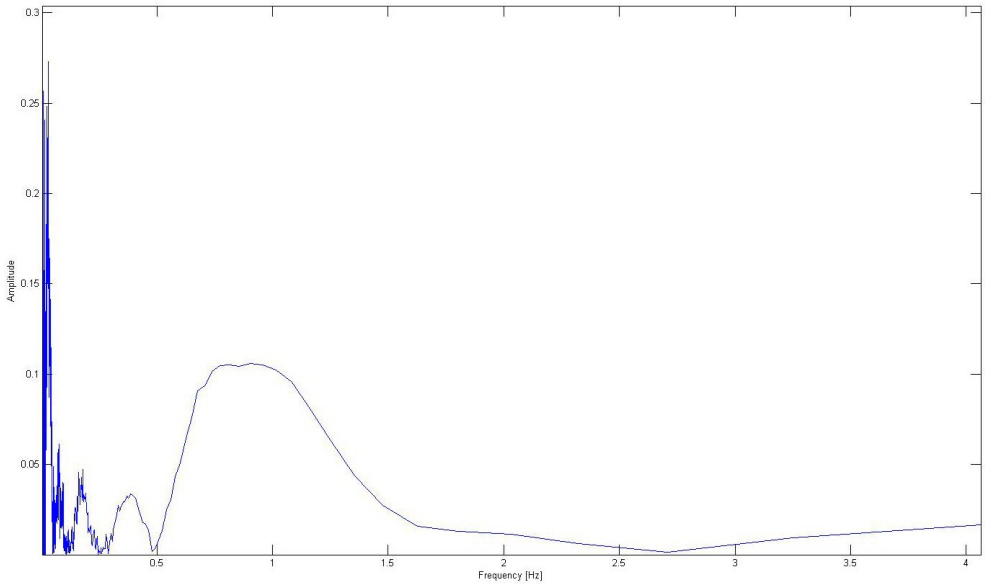

Figure 39: Experiment 2 step 12: Find frequency from CWT

Figure 39 shows that CWT's frequency resolution is not as good as STFT's. Because the frequency 0.76 Hz is so strong, it covers the other frequencies under it. Each frequency raises the amplitude curve a little and the amount raised can be figured out by experiment 2 by removing other frequencies from the data in step 15, so all the frequencies can be detected by CWT.

In step 15 the signal's strength in the data is compared in proportion to the signal's strength without the noise of the data. Figure 40 shows the proportion at different times using STFT.

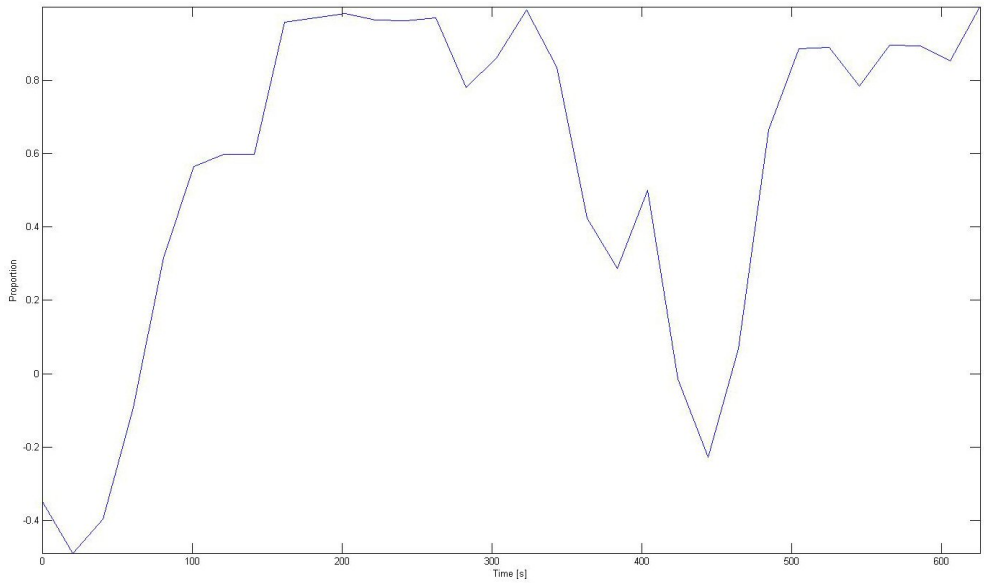

Figure 40: Experiment 2 step 15: STFT proportion q(t)

Figure 40 shows that the frequency is the strongest at around 200 seconds in the data and from there we can find the maximum of the proportion in step 16.

Figure 41 shows the proportion using CWT.

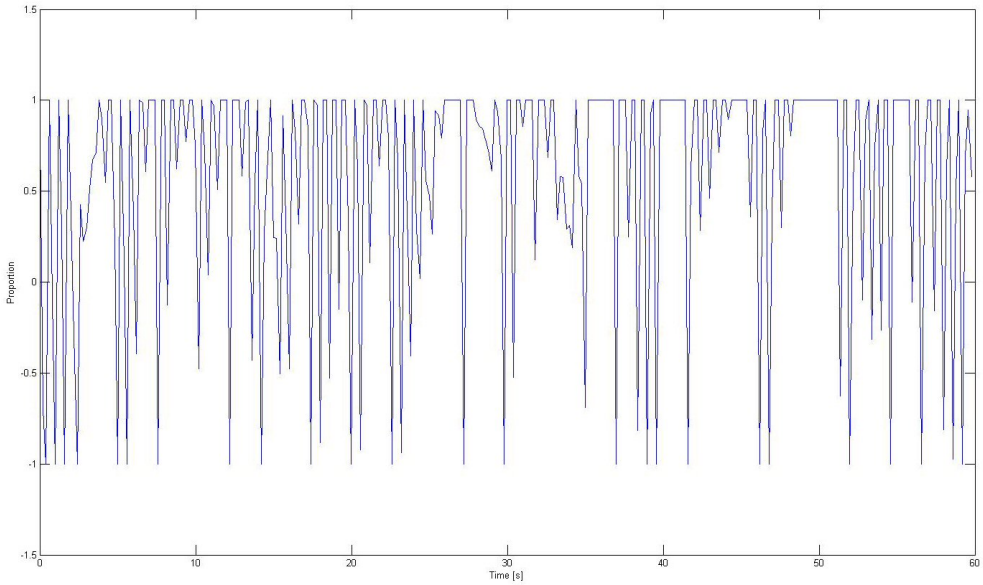

Figure 41: Experiment 2 step 15: CWT Proportion q(t)

Figure 41 shows that the proportion varies between 1 and -1. As can be seen in Figure 6, the value of the transformation varies like a sinusoidal wave corresponding to the periodic signal in the data. The proportion is -1 at the low point of the signal and +1 at the high point of the signal. From this it can be seen that CWT is very good at representing the signal in time-frequency domain. From Figure 41 can be found the maximum value of the proportion for step 16.

In steps 17 and 18 the tests are done again for the next frequency found from the data. Figure 42 shows the results for different frequencies using STFT and CWT.

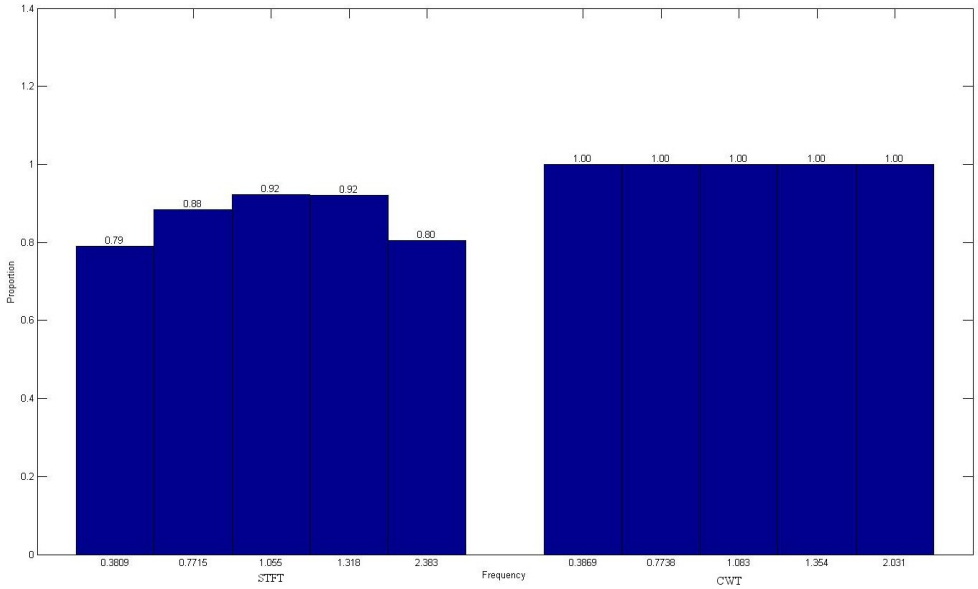

Figure 42: Experiment 2 step 18: Proportions with STFT and CWT idle operation. Data segment length 400.

Figure 42 shows that CWT shows a stronger signal compared to STFT in proportion to a signal with no noise. Figure 43 shows the same proportion together with the original frequencies found in step 3. The frequencies are different after the transformations.

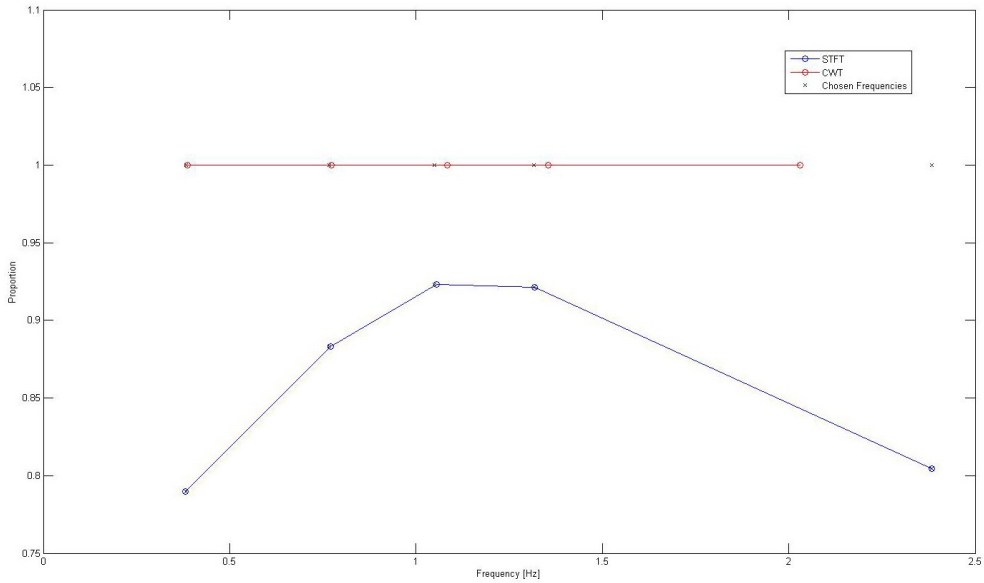

Figure 43: Experiment 2 step 18: Compare frequency accuracy conveyor belt (d) idle operation

Figure 43 shows that STFT has better frequency resolution than CWT even if the strength of the signal is lower.

Figures 44 and 45 show results after repeating the same experiments on the data set obtained from conveyor belt (d) during workload operation using same amount of data as before.

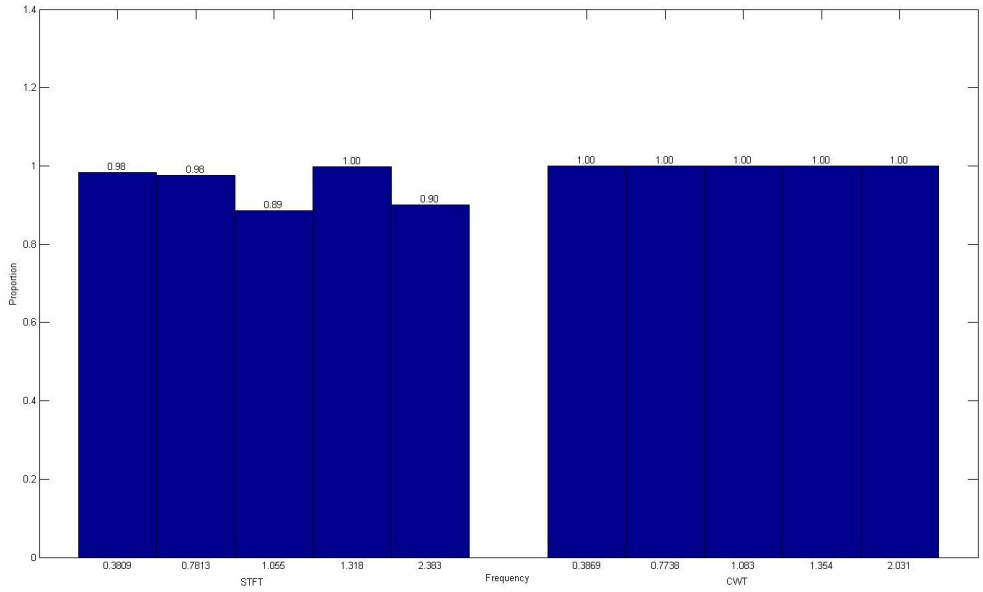

Figure 44: Experiment 2 step 18: Proportions with STFT and CWT conveyor belt (d) workload operation. Data segment length 400.

Figure 44 shows that when using a small amount of data the results of STFT are weaker than CWT.

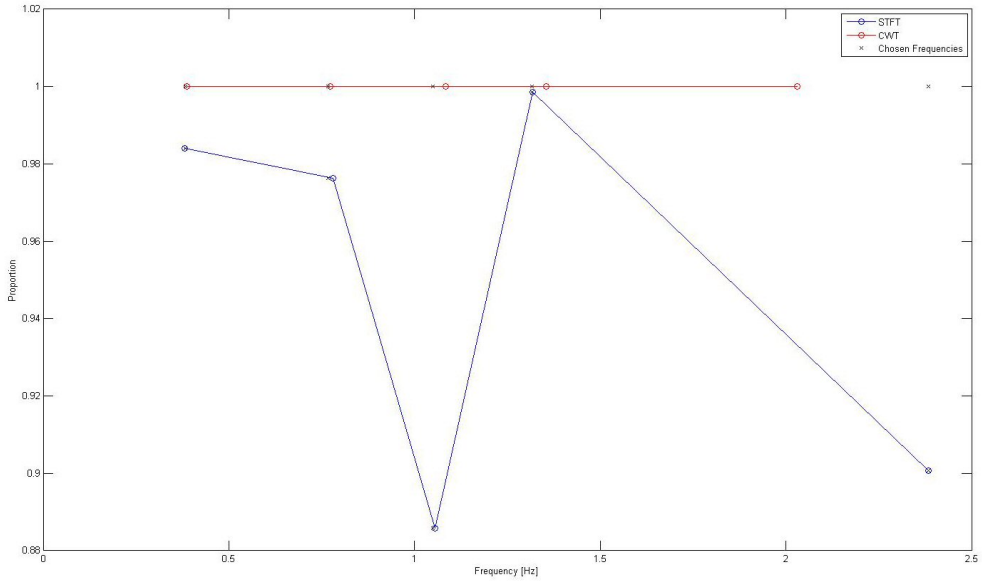

Figure 45: Experiment 2 step 18: Compare frequency accuracy conveyor belt (d) workload operation. Data segment length 400.

Figures 44 and 45 show similar results to Figures 42 and 43. The signal strength of STFT is weaker than when using CTW, but the frequency resolution is better when using STFT.

In the next experiment a longer piece data is taken from the workload data set and the result is shown in Figure 46.

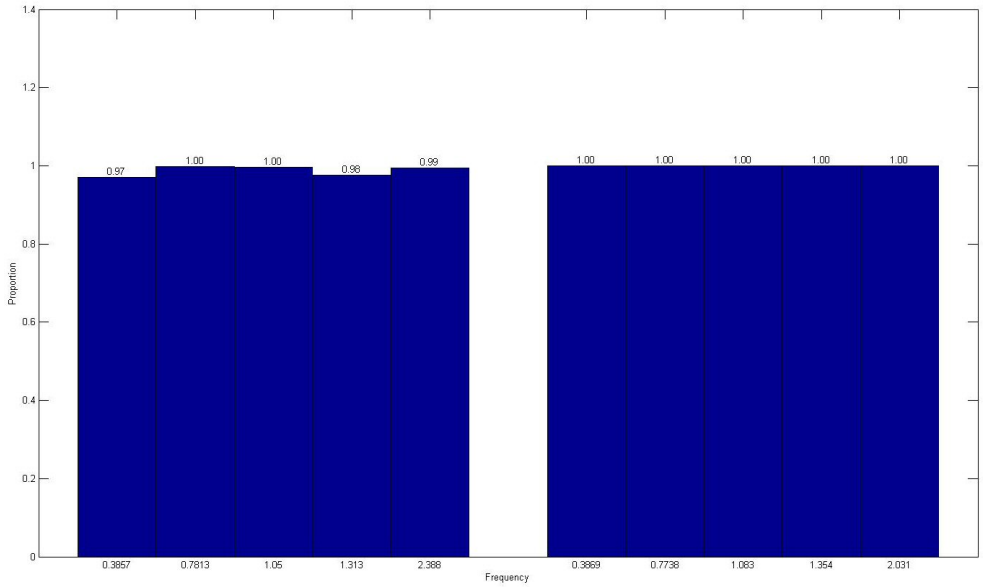

Figure 46: Experiment 2 step 18: Proportions with STFT and CWT conveyor belt (d) workload operation. Data segment length 1000.

Figure 46 shows that by using a longer piece of data, a better result can be obtained from the experiment.

Finally, the whole workload data set is used in the experiment and the result is shown in Figure 47.

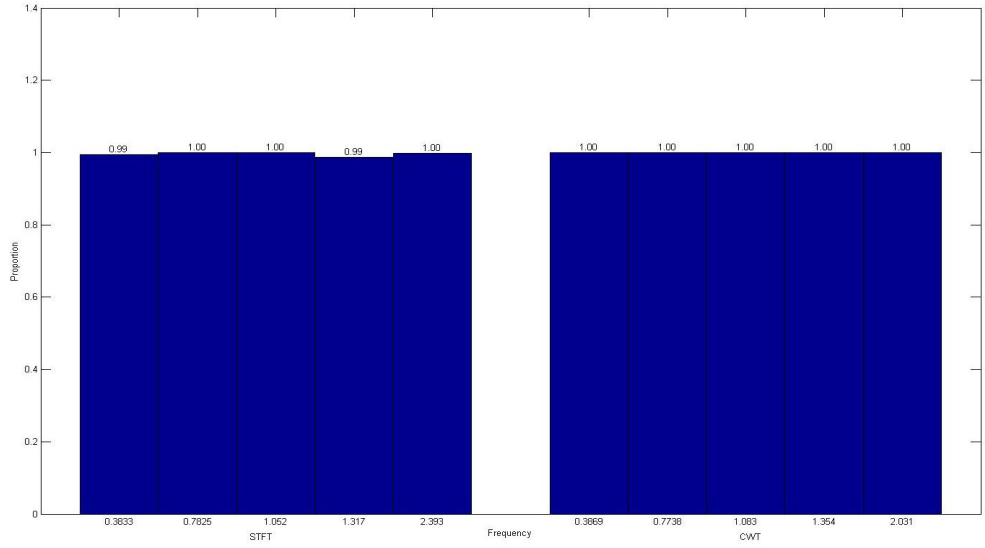

Figure 47: Experiment 2 step 18: Proportions with STFT and CWT conveyor belt (d) workload operation. Data segment length 4000.

Figure 47 shows that the results with STFT get even better using the full data set and there is not much different between STFT and CWT.

From results of experiment 2 it can be seen that both STFT and CWT can find the continuous periodic signals from the data. STFT is affected by the length of the data, but

CWT gives solid results regardless of the length of the data. STFT's frequency resolution is a lot better than CWT. Continuous periodic signals with frequencies close together are hard to discern using CWT and it is hard to see the amplitude of the frequencies. More detailed conclusions of the second experiment are drawn in chapter 5.2.

5 CONCLUSIONS

In Chapter 5 conclusions from the results of Chapter 4 are drawn. In Chapter 5.1 the conclusions from experiment 1 are introduced. In Chapter 5.2 the conclusions from experiment 2 are introduced. In Chapter 5.3 the final conclusion of the thesis is made.

5.1 Conclusions on Experiment 1

The results of experiment 1 from Figures 32, 33 and 34 are gathered together in Table II.

Table II: Results from experiment 1

Tested data	Method → / Amplitude ↓	Auto-correlation	FFT	STFT	CWT
Idle	1	2nd	1st	4th	2nd
	0.6	2nd	1st	4th	3rd
	0.3	1st	2nd	4th	3rd
	0.15	1st	2nd	4th	3rd
	0.1	1st	2nd	4th	2nd
Workload short	1	2nd	3rd	4th	1st
	0.6	2nd	3rd	4th	1st
	0.3	2nd	3rd	4th	1st
	0.15	2nd	3rd	4th	1st
	0.1	2nd	3rd	4th	1st
Workload long	1	2nd	4th	3rd	1st
	0.6	2nd	3rd	4th	1st
	0.3	1st	3rd	4th	2nd
	0.15	1st	3rd	4th	2nd
	0.1	1st	2nd	4th	2nd

From Table II the best method for finding the simulated signal from the data can be determined. The pros and cons of the different methods are discussed next.

Autocorrelation

Table II shows that autocorrelation is overall the best at finding the simulated signal from the data. Autocorrelation is always the best or second best method at finding the

signal and neither the amplitude of the simulated signal, noise caused by the data or the length of the signal affects the method much. Autocorrelation keeps the time information of the simulated signal so the time location of the simulated signal can be seen from autocorrelation. Autocorrelation is easy to use and the computation times are very fast even for long data lengths.

The problems with autocorrelation are that the nature of the periodic signal is hard to interpret. Different kinds of signals have similar autocorrelations. Also if the period between the square waves in the simulated signal is not constant, figuring out the time location of the signals becomes harder when using autocorrelaiton. Autocorrelation does not give information about the amplitude of the simulated signal and the frequency of periodic signals has to be calculated separately. If multiple periodic signals are present in the data to be analyzed, separating them from each other using autocorrelation is difficult. Low amplitude signals are impossible to see from the autocorrelation with naked eye, even though they can be found by the experiment 1.

Pros of autocorrelation:

+ Overall the best at finding the simulated signal from data
+ Has time information
+ Fast and easy to use

Cons of autocorrelation:

- Difficult to interpret the nature of the periodic signal
- No straight information on the amplitude or the frequency of the periodic signal
- Hard to discern between periodic signals if multiple present in the data
- Can not see low amplitude signals with naked eye

Fast Fourier Transformation

Table II shows that FFT is best or second best at finding the simulated signal from the data when there is not much noise and the data segment is short. When the data becomes noisier and longer, the performance of FFT goes down to third best and the results are influenced by the changes a lot. The amplitude of the simulated signal does not affect the method much. FFT gives very good information on what frequencies can be found from the data and what their amplitudes are. FFT is very easy to use and the computation times are short.

The problems with FFT are that the time information is lost from the signal, so it is not possible to see when and where the periodic signals happen. It is also difficult to

interpret the nature of the simulated signal without prior information especially if the period between the square waves is not constant. The frequency information is lost in the noise easily. Like with autocorrelation, low amplitude signals are impossible to see from the transformation with naked eye.

Pros of FFT:

+ Good at finding the simulated signal when noise level low and segment short
+ Gives accurate information on frequencies and their amplitudes that can be found from data
+ Very easy to use
+ Fast and easy to use

Cons of FFT:

- Can be difficult to interpret the nature of the periodic signal
- Time information lost
- Information easily lost in noise
- Can not see low amplitude signals with naked eye

Short Time Fourier Transformation

Table II shows that STFT is the worst at finding the simulated signal from the data according to experiment 1. At large amplitudes STFT can find the simulated signal easily, but at lower amplitudes, STFT's performance goes down fast. The noise and the length of the data lower STFT's performance strongly also. STFT keeps the time information of the signal and it is easy to interpret the nature of the simulated signal. The time resolution depends on the size and type of the window used, but compared to CWT, the time resolution is not as good. STFT is easier to use and understand than CWT and the computation times are shorter, but the choice of window and window size makes STFT harder and slower to use than autocorrelation and FFT. STFT has information on the amplitude of the periodic signal, but the window used affects the results

Pros of STFT:

+ Good at finding the simulated signal at high amplitudes

+ Keeps time information from the transformed data
+ Easy to interpret the nature of the periodic signal visually
+ Faster and easier to use than CWT

Cons of STFT:

- Performance goes down at lower amplitude signals
- Performance goes down with noisy and long data segments
- Window related decisions can be difficult
- Chosen window affects results

Continuous Wavelet Transform

Table II shows that CWT is consistently one of the best methods at finding the simulated signal from the data. Neither the amplitude of the signal, the noise from the data or the length of the data segment does not affect the method much. The time information of the signal is conserved, the time resolution is good and the nature of the simulated signal is easy to interpret. CWT is the hardest method to use and to understand. The decision of the mother wavelet is difficult. The computation times are long and with long data segments the method becomes very hard to use.

Good points of CWT:

+ Very good at finding the simulated signal at every situation
+ Keeps time information from the transformed data and has good time resolution
+ Easy to interpret the nature of the periodic signal visually

Negative points of CWT:

- Hard to use
- Long computation times

Conclusions on Experiment 1

CWT is the best method at finding the simulated signal from the data. The simulated signal is a periodic square wave which simulate impulsive fault signals in the data.

According to experiment 1 CWT is very good at finding and understanding impulsive signals in the data. STFT does not give as strong results as CWT does, but the impulsive signals are easy to see and understand. Autocorrelation and FFT are good at finding the impulsive signal from the data, but the results are harder to understand without prior knowledge of the impulsive signal.

5.2 Conclusions on Experiment 2

Figures 42, 44, 46 and 47 show that STFT is worse than CWT at finding periodic signals from short data data segments. When the data segment becomes longer the methods become equal in strength according to experiment 2. This is partly because of the long window length used in the experiments. With short data segment and long window, finding the maximum values from the transformation becomes difficult. When the data segments become longer, more examples can be obtained and the maximum value found.

Long windows are used in STFT to obtain good frequency resolution. Because the data is noisy with many frequencies close together, good frequency resolution is needed to separate the frequencies from each other. Compared to CTW, STFT has a lot better frequency resolution as can be seen in Figure 47. Because of the long window, the time resolution of STFT becomes bad and finding impulsive signals from the data becomes impossible. With CWT both the continuous periodic signals and the impulsive signals can be found.

CWT has great signal strength regardless of the length of the data. Because the frequency resolution is very low and because there are so many frequencies present in the data, finding the different frequencies visually is difficult. The low frequency resolution is partly because of low sampling frequency of only 5 Hz, but CWT has trouble finding single frequencies from noisy data. The stronger frequencies cover up the weaker ones near it as can be seen in Figure 41. Impulsive signals are easy to find with CWT because the time resolution is good. Figure 48 shows how the sampling frequency affects the CWT's ability to detect different frequencies close together. In it CWT is done on a signal with two sinusoidal waves of frequencies 1 Hz and 1.5 Hz.

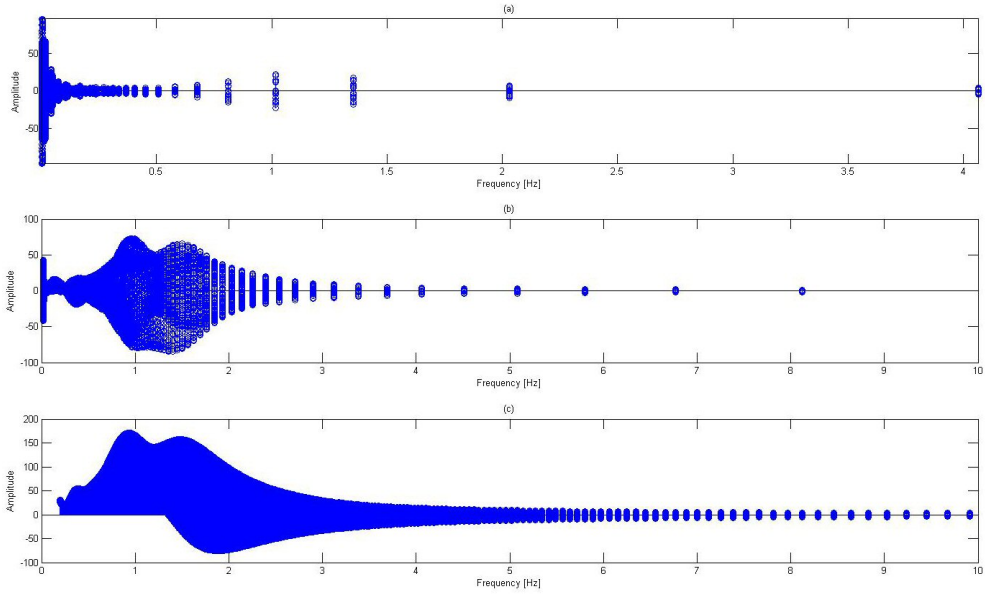

Figure 48: CWT Coefficients of a signal with two sine waves at different times using different sampling frequencies (a) 5 Hz (b) 50 Hz (c) 500 Hz

Figure 48 shows that with a slightly larger sampling frequency, detecting different frequencies would become easier.

Pros of STFT:

+ Good frequency resolution
+ Can find all frequencies from the data
+ Very easy to see the periodic signals visually

Cons of STFT:

- Bad time resolution
- Can not find impulsive signals from data
- Requires longer data segments

Pros of CWT:

+ Can find all periodic signals from data regardless of data segment lengths
+ Good time resolution
+ Easy to see impulsive signals visually

Cons of CWT:

- Bad frequency resolution
- Possibly requires larger sampling frequency for more definite analysis

5.3 Final Conclusions

In this thesis five different frequency domain methods were researched: autocorrelation, FFT, STFT, DWT and CWT. The researched methods were implemented on data obtained from an earth crushing facility conveyor belts. In the first experiments a simulated impulsive square wave signal was added into the data obtained from the facility and then the ability to find the simulated signal from the data was compared between four of the methods: autocorrelation, FFT, STFT and CWT. In the second experiment periodic signals were found from the obtained data using FFT and then the ability to find the found frequency from the data was compared between STFT and CWT.

From the first experiment it was found that autocorrelation and FFT are very good at finding the simulated impulsive signal from the data, but because they lack either the time information or the frequency information, it is difficult to figure out the nature of the periodic signal in the data without prior knowledge. STFT and CWT both give good information on the type of periodic signal found in the data in time-frequency domain. CWT was found to be the stronger method at finding impulsive signals from the data.

From the second experiment it was found that CWT gives a stronger signal for different frequencies in the data, but its time frequency is bad so it is difficult to recognize different frequencies from the transformation visually. STFT requires a bit longer data segments to be equally strong with CWT, but STFT's frequency resolution is very good in all situations so the periodic signals are easy to visually find from the data. Because of the high frequency resolution, STFT's time resolution is bad so it can not recognize impulsive signals in the data. On the other hand CWT has a very good time resolution so it is easy to find impulsive signals from the data.

As a conclusion, different methods have different strengths and using all of them together to fully understand the periodic content of the data is recommended. STFT is the best method of finding when and where continuous periodic signals appear in data and CWT is best at finding impulsive signals from the data.

REFERENCES

[1] Paresh Girdhar & C. Scheffer. 2004. P*ractical Machinery Vibration Analysis and Predictive Maintenance*

[2] Mark A. Haidekker. 2011. *Advanced Biomedical Image Analysis*

[3] Shahin Hedayati Kia, Humberto Henao & Gérard-André Capolino. 2009. *Diagnosis of Broken-Bar Fault in Induction Machines Using Discrete Wavelet Transform Without Slip Estimation*. IEEE Transactions on Industry Applicastions, vol. 45, no. 4, July/August 2009

[4] Guiji Tang, Jinsheng Ye, Rongpei Zhang & Aijun Hu. 2009. *Harmonic Wavelet Packets Method and Its Application to Signal Analysis of Rotating Machinery.* Image Analysis and Signal Processing, 2009. IASP 2009. International Converence on

[5] Tommy W. S. Chow & Shi Hai. 2004. *Induction Machine Fault Diagnostic Analysis With Wavelet Technique*. IEEE Transactions on Industrial Electronics, vol. 51, no. 3, june 2004.

[6] Fadi Al-Badour, L. Cheded M. Sunar. 2010. *Non-Stationary Vibration Signal Analysis of Rotating Machinery via Time-Frequency and Wavelet Techniques*. 10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010)

[7] M. Deriche. 2005. *Bearing Fault Diagnosis Using Wavelet Analysis.* Computers, Communications, & Signal Processing with Special Track on Biomedical Engineering

[8] F. A. Shirazi M. J. Mahjoob. 2007. *Application of Discrete Wavelet Transform (DWT) in Compustion Failure Detection of IC Engines.* Image and Signal Processing and Analysis, 2007. ISPA 2007. 5th International Symposium on

[9] Chih-Chung Wang, Chien-Wei Lee & Chen-Sen Ouyang. 2010. *A Machine-Learning-Based Fault Diagnosis Approach For Intelligen Condition Monitoring.* Proceedings of the Ninght International Conference on Machine Learning and Cybernetics, Qingdao, 11-14 July 2010

[10] Kang Shanlin, Zhang Huanzhen & Kang Yuzhe. 2009. *Detection and Localization of Turbine-generator Bearing Vibration Using Wavelet Neural Network.* Proceedings of the 2009 IEEE International Conference on Mechatronics and Automation August 9-12, Changchun, China

[11] Liao Wei & Han Pu. 2008. *Wavelet Neural Network Aided On-line Detection and Diagnosis of Rotating Machine Fault.*

[12] Guang-Zhong Cao, Xiao-Yu Lei & Chang-Geng Luo. 2009. *Research of a Fan Fault Diagnosis System Based on Wavelet and Neural Network.* 2009 3rd International Conference on Power Electronic Systems and Applications

[13] Qiang Miao, Dong Wang & Hong-Zhong Huang. 2010. *Identification of Characteristic Components in Frequency Domain From Signal Singularities.* Review of Scientific Instruments 81

[14] Xu Yangwen. 2009. *Study on Fault Diagnosis of Rotating Machinery Based on Wavelet Neural Network.* 2009 International Conference on Information Technology and Computer Science

[15] Huaqing Wang & Peng Chen. 2011. *Fuzzy Diagnosis Method for Rotating Machinery in Variable Rotating Speed.* IEEE Sensors Journal, vol. 11, No. 1, January 2011.

[16] Patricia Scanlon, Darren F. Kavanagh & Francis M. Boland. 2013. *Residual Life Prediction of Rotating Machines Using Acoustic Noise Signals.* IEEE Transactions on Instrumentation and Measurement, Vol. 62, No. 1, January 2013

[17] Giovanni Betta, Consolatina Liguori, Alfredo Paolillo & Antonio Pietrosanto. 2002. *A DSP-Based FFT-Analyzer for the Fault Diagnosis of Rotating Machine Based on Vibration Analysis.* IEEE Transactions on Instrumentation and Measurement, Vol. 51, No. 6, December 2002.

[18] Sebastian Villwock & Henning Zoubek. 2007. *Rolling Bearing Condition Monitoring Based on Frequency Response Analysis.*

[19] Mario Pacas, Sebastian Villwock & Ralf Dietrich. 2009. *Bearing Damage Detection in Permanent Magnet Synchronous Machines.*

[20] Abd Kadir Mahamad & Takashi Hiyama. 2009. *Improving Elman Network Using Genetic Algorithm for Bearing Failure Diagnosis of Induction Motor.*

[21] Wenbin Zhang, Yanping Su, Jie Min, Ruijing Teng & Yanjie Zhou. 2012. *Application of Grey Relation Degree in Rotor's Fault Identification.* 2012 International Symposium on Instrumentation & Measurement, Sensort Network and Automation (IMSNA)

[22] Alan Miletic, Dragan Turk & Josip Polak. 2007. *Turbogenerator Rotor Interturn Fault Detection Using the Leakage Field Analysis – Case Study.*

[23] Jiang Fan, Li Wei, Wang Zhongqiu, Wang Zewen & Cao Baoyu. 2012. *Fault Diagnosis of Rotating Machinery Based on MFES and D-S Evidence Theory.*

[24] Wenqiang Guo, Qiang Zhou, Yongyan Hou, Zoe Zhu, Jingjing Yang & Baorong Zhang. 2013. *Early Classification for Bearing Faults of Rotating Machinery Based on MFES and Bayesian Network.*

[25] Manuela Pineda-Sanchez, M. Riera-Guasp, Jose A. Antonino-Daviu, J. Roger-Folch, J. Perez-Cruz & R. Puche-Panadero. 2010. *Diagnosis of Induction Motor Faults in the Fractional Fourier Domain.* IEEE Transactions on Instrumentation and Measurement, Vol. 59, No. 8, August 2010

[26] Keijo Ruohonen, Lasse Vehmanen. 2006. *Fourier Fourier Methods Compiled from: Advanced Modern Engineering Mathematics Third Edition by Glyn James*. Pearson Custom Publishing

[27] S. Cerutti & C. Marchesi. 2011. *Advanced Methods of Biomedical Signal Processing*. The institute of Electrical and Electronics Engineers, Inc

[28] K.S. Thyagarajan. 2011. *Still Image and Video Compression with MATLAB*. John Wiley & Sons, Inc.

[29] Paul S. Addison. 2002. *The Illustrated Wavelet Transform Handbook*. IOP Publishing Ltd

APPENDIX 1: SAMPLING DATA MATLAB CODE

```
% Sampling data from earth crushing facility

% Measurement frequency

fs = 25000;

% Read measurement time from workspace

t = time;

% Data channel to sample

y = Power_Screen2_11_16;

% Plot a piece of data

figure(1)
plot(y(1:25000))

% Search for beginning of zero order hold from Figure 1

start = 1259;

% Search for end of ZOH from Figure 1

finish = 6014;

% Duration of ZOH

duration = finish-start;

% Search for average number of samples between level change from
Figure 2

change = 10;

% Take some data away from the end of ZOH to compensate for variation
in "change" by checking Figure 2

compensation = 300;

% Obtain the length of signal by trial and error by running the for
loop with a large ii once

ii = 566;

sampled = [];

for i = 1:ii
```

```
%     figure(2)
%     plot(y(start:start+duration-compensation))
%     axis tight

    sampled(i) = mean(y(start:start+duration-compensation));

    % Move the starting point to the beginning of next level change

    start = start+duration+change;

%     pause
end

% Rename the sampled data to the name of the channel

Power_Screen2_11_16 = sampled;
y = sampled;

% Calculate the duration of the sampled data

time_end = t(end)-t(start);

% Calculate new measurement frequency

Fs = 1/(time_end/length(y));

% Calculate new time vector for sampled data

t2 = 0:1/Fs:length(y)/Fs;
t2 = t2(1:end-1);

% Plot sampled data

figure(3)
plot(t2,Power_Screen2_11_16);
```

APPENDIX 2: SIMULATED SQUARE WAVE MATLAB CODE

```
% Square wave for idle data settings

% Measurement frequency after sampling

fs = 5;

% Start of square wave

start = 1;

% Duration of square wave

duration = fs*5000;

% Length of the idle data channel

pituus = 3500000;

% Measurement frequency of the idle data before sampling

fs2 = 25000;

%Signal evaluation time

t = 0:1:pituus;

% Delay between square waves

t_d = 15;

D = duration/2:(t_d*fs2/duration)*duration:pituus;

% Duration of each pulse = 1 s

w = duration;

%Create square wave

yp = pulstran(t,D',@rectpuls,w);

% Create new time vector

t2 = 0:1/fs2:pituus/fs2;
% t2 = t2(1:end-1);

% Plot the simulated square wave

h = plot(t2,yp);
```

```
ylim([0 1.2])
% xlim([0 80])
xlabel('Time [s]')
ylabel('Power [kW]')
set(h,'linewidth',2);
grid on
```

APPENDIX 3: INSERTING SIMULATED SQUARE IN DATA MATLAB CODE

```
% For more information on the variables, see Appendix 1

% Data channel to add wave to

y = Power_Screen2_11_16;

% Amplitude of the square wave

amplitude = 1;

% Compensation for variance in "change". See Appendix 1.

compensation = 3000;

% Obtain the length of signal by trial and error by running the for
loop with a large iii once. See Appendix 1.

iii = 3175;


for i = 1:iii


    if ii == 1 % Add square wave to the beginning of the signal

        for iv = 1:6 % Duration of the square wave 1 second = 6
                     %  samples

            sampled(ii) = mean(y(start:start+duration-compensation)...
                        +amplitude);

            ii = ii+1;

            start = start+duration+change;

        end

    elseif mod(ii,79) == 0 % Add square wave to the signal every ~15
                           seconds. 79 samples = ~15 seconds

        for iv = 1:6 % Duration of the square wave 1 second = 6
                     samples

            sampled(ii) = mean(y(start:start+duration-compensation)...
                        +amplitude);

            ii = ii +1;
```

```
            start = start+duration+change;
        end
    else
        sampled(ii) = mean(y(start:start+duration-compensation));
        start = start+duration+change;
        ii = ii+1;
    end
end
```

APPENDIX 4: EXPERIMENT 1 MATLAB CODE

Autocorrelation experiments

```
% Load data workspace
load('3516.mat')
% Remove trends, make zero mean
y = detrend(Power_Screen3_5_16(1:720));
% Load time vector
t = time2;
% Calculate autocorrelation
kory1 = xcorr(y,y);
% Load square wave workspace
load('squareidle.mat')
% Square wave's amplitude
A = 1;
y = A*squarewave;
t = time2;
% Calculate square wave's autocorrelation
kory2 = xcorr(y,y);
% Load workspace with data and simulated signal amplitude 1
load('idle3516amp1.mat')
% Remove trends and make the data same length as data and square wave
y = detrend(Power_Screen3_5_16(1:720));
t = time2;
kory3 = xcorr(y,y);
```

```
%Remove the autocorrelation of data from the autocorrelation of data
%and simulated wave

kory3 = kory3-kory1;

% Find peaks from the autocorrelations

q_1 = 76:75:601;
q_2 = 80:79:633;

% Take only half of the autocorrelation

squarep = kory2(length(kory2)/2:end);

% The value of the autocorrelation at the location of the peaks in the
% simulated square wave at amplitude 1

squarepeaks = squarep(q_1);

% Take only half of the autocorrelation

datapamp1 = kory3(length(kory3)/2:end);

% The value of the autocorrelation at the location of the peaks in the
% data with the simulated square wave at amplitude 1

datapeaksamp1 = datapamp1(q_2);

% Calculate the proportion between the values of the the peaks from
the square wave and the data with the square wave at amplitude 1

autocoramp1 = mean(datapeaksamp1./squarepeaks);
```

Fast Fourier transform experiments

```
load('3516.mat')

y = detrend(Power_Screen3_5_16(1:720));

t = time2;

% Sampling frequency from the workspace

fs = fs2;

% Calculate the FFT of the data

L1 = length(y);

NFFT1 = 2^nextpow2(L1); % Next power of 2 from length of y
Y1 = fft(y,NFFT1)/(L1);
f1 = fs/2*linspace(0,1,NFFT1/2+1);
```

```
% Data and simulated wave with amplitude 1

load('idle3516amp1.mat')

y = detrend(Power_Screen3_5_16(1:720));

t = time2;

fs = fs2;

L1 = length(y);

NFFT2 = 2^nextpow2(L1); % Next power of 2 from length of y
Y2 = fft(y,NFFT1)/(L1);
f2 = fs/2*linspace(0,1,NFFT1/2+1);

% Square wave of amplitude 1

load('squareidle.mat')

fs = fs2;

y = squarewave;

% t = time2;

L1 = length(y);

NFFT7 = 2^nextpow2(L1); % Next power of 2 from length of y
Y7 = fft(y,NFFT1)/(L1);
f7 = fs/2*linspace(0,1,NFFT1/2+1);


% Plot the FFT of the transformations

figure(1)
subplot(3,1,1)
plot(2*abs(Y1(1:NFFT1/2+1)))
ylabel('Amplitude')
xlabel('Frequency (Hz)')
subplot(3,1,2)
plot(2*abs(Y7(1:NFFT7/2+1)))
ylabel('Amplitude')
xlabel('Frequency (Hz)')
subplot(3,1,3)
plot(2*abs(Y2(1:NFFT2/2+1)))
ylabel('Amplitude')
xlabel('Frequency (Hz)')

% Remove the Fourier transform of the data from the FFT of the data
with simulated signal

Y2 = 2*abs(Y2(1:NFFT1/2+1)) - mean(2*abs(Y1(1:NFFT1/2+1)));

% Find peaks from Figure 1

% Square

q_1 = [15 28 42 56 69 83 97 110 124 137];
```

```
% Data

q_2 = [14 27 40 53 66 79 92 105 118 131];

% Amplitude of the peaks from the square wave

% Amplitude 1

square1 = 2*abs(Y7(1:NFFT7/2+1));

squarepeaks1 = square1(q_1);

% Amplitude from the peaks in the data with the simulated signal

datapamp1 = 2*abs(Y2(1:NFFT2/2+1));

datapeaksamp1 = Y2(q_2);

% Proportion between the simulated data and the square wave

fftamp1 = mean(datapeaksamp1./squarepeaks1);
```

Experimenst with STFT

```
NFFT = 2^13; %NFFT > length(3232)

load('3516.mat')

y1 = detrend(Power_Screen3_5_16(1:720));

t = time2;
fs = fs2;

n = 30;
w = hann(n);

[S1,F1,T1,P1] = spectrogram(y1,w,n/2,NFFT,fs);

load('idle3516amp1.mat')

y = detrend(Power_Screen3_5_16(1:720));

[S32,F3,T3,P3] = spectrogram(y,w,n/2,NFFT,fs);

load('squareidle.mat')

y = squarewave;

[S2,F2,T2,P2] = spectrogram(y,w,n/2,NFFT,fs);

% Remove the STFT of the data from the STFT of the simulated data

S1 = abs(S1);
S3 = abs(S32);
```

```
for i = 1:length(S3(:,1))

    S3(i,:) = S3(i,:)-mean(S1(i,:));

end

% Plot a frequency from the simulated data

figure(1)
plot(T3,abs(S32(605,:)))

% Find the locations of all peaks with amplitude higher than treshold
% found from figure 1

q = find((S3(577,:))>2.6);

% Select the found peaks from data

peaksamp1 = (S3(577,q));

% Plot a frequency from the simulated data

peaks_square = (abs(S2(577,:)));
figure(2)
plot(peaks_square)

% Find the locations of all peaks with amplitude higher than treshould
% found from figure 2

q2 = find(peaks_square>3.5);

% Select the found peaks from the square wave

peaks_square2 = mean1(q2);

% Calculate the proportion

suhdeamp1 = (peaksamp1)./mean(peaks_square2);
```

Experiment with CWT

```
% Choose wavelet

wave = 'morl';

% Choose scales

scales = 1:200;

load('3516.mat')

y = detrend(Power_Screen3_5_16(1:720));

t = time2;
fs = fs2;

ccfs1=cwt(y,scales,wave);
```

```
freq1 = scal2frq(scales,wave,1/fs);

load('idle3516amp1.mat')

y2 = detrend(Power_Screen3_5_16(1:720));

ccfs22=cwt(y2,scales,wave);
freq2 = scal2frq(scales,wave,1/fs);
load('squareidle.mat')

fs = fs2;

%Amp 1 square

y7 = (squarewave);

ccfs7=cwt(y7,scales,wave);
freq7 = scal2frq(scales,wave,1/fs);


% Remove data from the simulated data

ccfs1 = abs(ccfs1)

ccfs2 = abs(ccfs22);

for i = 1:length(ccfs2(:,1))

    ccfs2(i,:) = ccfs2(i,:)- ccfs1(i,:);

end

% Plot a scale from the CWT of the simualted data

figure(1)
plot(abs(ccfs2(13,:)))


% Find peaks with higher amplitude than the treshold from figure 1

q = find(abs(ccfs2(13,:))>1.1);

peaksamp1 = abs(ccfs2(13,q));
square_peaks = (abs(ccfs7(13,:)));

figure(2)
plot(square_peaks)

% Find peaks with higher amplitude than the treshold from figure 2

q2 = find(square_peaks>1.15);

square_peaks2 = square_peaks(q2);

% Calculate proportion

suhdeamp1 = mean(peaksamp1)./mean(square_peaks2);
```

APPENDIX 5: EXPERIMENT 2 MATLAB CODE

```
load('sampledload25khz.mat')

%Beginning of the data used

alku = 1;

%End of the data used

% pituus = 1000;

y = detrend(Power_Circulating(alku:end));

t = time2(1:length(y));

%Sampling frequency

fs = 5;

L1 = length(y);
NFFT1 = 2^nextpow2(L1); % Next power of 2 from length of y

%Methods #1

%Autocorrelation

kory = xcorr(y,y);

%FFT

Y1 = fft(y)/L1;
f1 = fs/2*linspace(0,1,L1/2+1);

%STFT

n = 200;
w = gausswin(n);

[S,F,T,P] = spectrogram(y,w,n/2,NFFT1,fs);

%CWT

wave = 'morl';

scales = 1:0.25:2^10;

ccfs1=cwt(y,scales,wave);
freq = scal2frq(scales,wave,1/fs);
```

```
%Find frequencies with FFT

figure
plot(f1,2*abs(Y1(1:L1/2+1)))
ylabel('Amplitude')
xlabel('Frequency [Hz]')
axis tight

freq_idle = [0.385 1.05 1.317 2.383];
amp_idle = [0.008445 0.009651 0.007586 0.01072];

freq_load = [0.3835 0.7832 1.05 1.317 2.39];
amp_load = [0.01735 0.01384 0.01799 0.008028 0.003821];

%Remove other frequencies

f1 = round(f1*10000)/10000;

Y12 = (Y1);
keepInd = f1 == freq_load(1);
Y12(~keepInd) = 0;
Y12(length(f1):end)=0;

figure
plot(keepInd)

%Inverse Fourier transform

y2 = (L1)*(ifft((Y12),'symmetric'));

L2 = length(y2);
NFFT2 = 2^nextpow2(L2); % Next power of 2 from length of y

%Methods #2

%Autocorrelation

kory2 = xcorr(y2,y2);

%FFT

Y2 = fft(y2)/L2;
f2 = fs/2*linspace(0,1,L2/2+1);

%STFT

n = 200;
w = gausswin(n);

[S2,F2,T2,P2] = spectrogram(y2,w,n/2,NFFT2,fs);

%CWT

wave = 'morl';

scales = 1:0.25:2^10;

ccfs2=cwt(y2,scales,wave);
freq2 = scal2frq(scales,wave,1/fs);
```

```
%Remove the frequency from data

y3 = y-y2;

L3 = length(y3);
NFFT3 = 2^nextpow2(L3); % Next power of 2 from length of y

%Methods #3

%Autocorrelation

koryy = xcorr(y3,y3);

%FFT

Y3 = fft(y3)/L3;
f3 = fs/2*linspace(0,1,L3/2+1);

%STFT

n = 200;
w = gausswin(n);

[S3,F3,T3,P3] = spectrogram(y3,w,n/2,NFFT3,fs);

%CWT

wave = 'morl';

scales = 1:0.25:2^10;

ccfs3=cwt(y3,scales,wave);
freq3 = scal2frq(scales,wave,1/fs);

%Plot figures if data segment not too long


% %Data
%
% figure('name','Power Circulating','numbertitle','off')
% subplot(4,1,1)
% plot(t,y)
% axis tight
% title('(a)')
% ylabel('Power')
% xlabel('Time [s]')
%
% subplot(4,1,2)
% plot(f1,2*abs(Y1(1:L1/2+1)))
% ylabel('Amplitude')
% xlabel('Frequency [Hz]')
% axis tight
% title('(b)')
%
% subplot(4,1,3)
% surf(T,F,abs(S),'edgecolor','none'); axis tight; colormap hsv
% view(0,90);
% xlabel('Time [s]'); ylabel('Frequency [Hz]');
```

```
% title('(c)')
%
% subplot(4,1,4)
% surf(t,freq,abs(ccfs1));shading('interp');
% axis tight; xlabel('Time [s]'); ylabel('Pseudo-Frequency [Hz]');
% zlabel('Amplitude')
% view([0 90])
% title('(d)')
%
% %Found frequency
%
% figure('name','Found Frequency','numbertitle','off')
% subplot(4,1,1)
% plot(t,y2)
% axis tight
% title('(a)')
% ylabel('Power')
% xlabel('Time [s]')
%
% subplot(4,1,2)
% plot(f2,2*abs(Y2(1:L2/2+1)))
% ylabel('Amplitude')
% xlabel('Frequency [Hz]')
% axis tight
% title('(b)')
%
% subplot(4,1,3)
% surf(T2,F2,abs(S2),'edgecolor','none'); axis tight; colormap hsv
% view(0,90);
% xlabel('Time [s]'); ylabel('Frequency [Hz]');
% title('(c)')
%
% subplot(4,1,4)
% surf(t,freq2,abs(ccfs2));shading('interp');
% axis tight; xlabel('Time [s]'); ylabel('Pseudo-Frequency [Hz]');
% zlabel('Amplitude')
% view([0 90])
% title('(d)')
%
% %Data with the frequency removed
%
% figure('name','Power Circulating wtih frequency
removed','numbertitle','off')
% subplot(4,1,1)
% plot(t,y3)
% axis tight
% title('(a)')
% ylabel('Power')
% xlabel('Time [s]')
%
% subplot(4,1,2)
% plot(f3,2*abs(Y3(1:L3/2+1)))
% ylabel('Amplitude')
% xlabel('Frequency [Hz]')
% axis tight
% title('(b)')
%
% subplot(4,1,3)
% surf(T3,F3,abs(S3),'edgecolor','none'); axis tight; colormap hsv
```

```
% view(0,90);
% xlabel('Time [s]'); ylabel('Frequency [Hz]');
% title('(c)')
%
% subplot(4,1,4)
% surf(t,freq3,abs(ccfs3));shading('interp');
% axis tight; xlabel('Time [s]'); ylabel('Pseudo-Frequency [Hz]');
% zlabel('Amplitude')
% view([0 90])
% title('(d)')

%Find frequency from STFT

%Plot the amplitude of the frequencies at certain time in data
figure
plot(F,abs(S(:,2)))

%Plot the amplitude of the frequencies at certain time in removed
%frequency

figure
plot(F2,abs(S2(:,2)))

F23 = round(F2*10000)/10000;

% Found frequency
f_valittu = 0.3833;

q3 = find(F23==f_valittu);

r = [];

%Calculate the proportion in all time locations

for i = 1:32

    r(i) = (abs(S(q3,i))-abs(S3(q3,i)))/abs(S2(q3,i));

end

%Find the maximum proportion

stft_suhde1 = max(r);


%Find frequency from CWT

figure
plot(freq,abs(ccfs1(:,109)))

figure
plot(freq2,abs(ccfs2(:,109)))

freq12 = round(freq*10000)/10000;

f_valittu2 = 0.3869;

q2 = find(freq12==f_valittu2);
u = [];
```

```
for i = 1:length(ccfs1(1,:))
    u(i) = abs(ccfs1(q2,i));
    u(i) = (abs(ccfs1(q2,i))-abs(ccfs3(q2,i)))/abs(ccfs2(q2,i));
end
q3 = find(u==max(u));
cwt_suhde1 = (abs(ccfs1(q2,q3))-abs(ccfs3(q2,q3)))/abs(ccfs2(q2,q3));
```

APPENDIX 6: RESULTS OF EXPERIMENT 2 FROM ANOTHER DATA CHANNEL

In Figure 49 the experiments were done on conveyor belt (f) during idle operation.

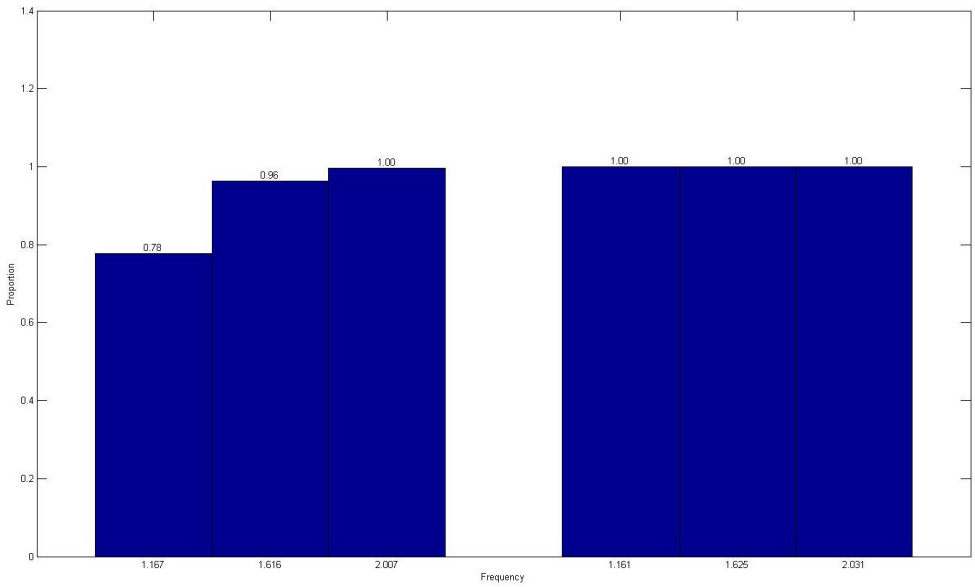

Figure 49: Experiment 2 step 18: Proportions with STFT and CWT (conveyor belt (f) Idle)

In Figure 50 the frequency accuracy is compared between STFT and CWT.

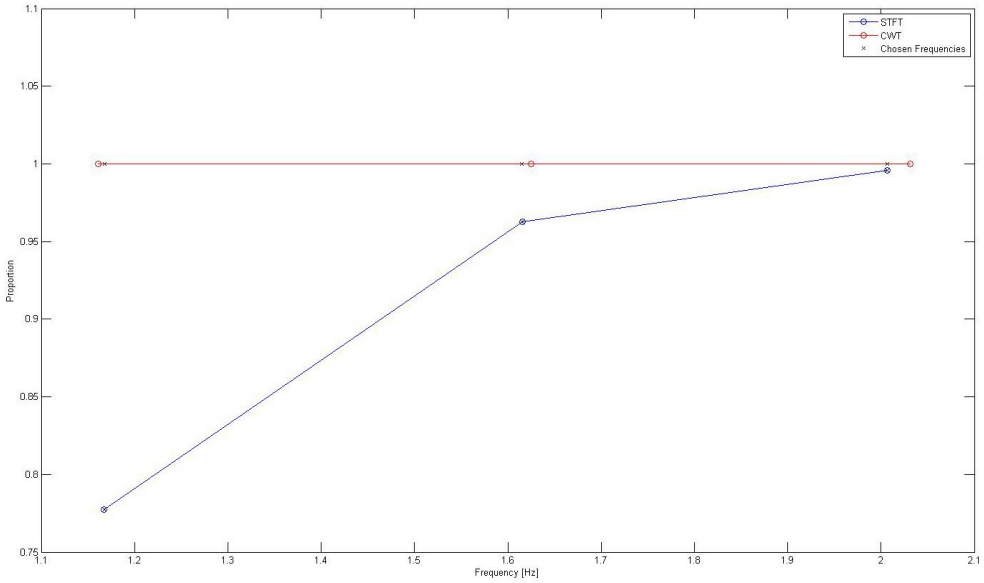

Figure 50: Experiment 2 step 18: Compare Frequency Accuracy (conveyor belt (f) Idle)

In Figure 51 the experiments were done on data from conveyor belt (f) during workload operation for same length data as during idle operation.

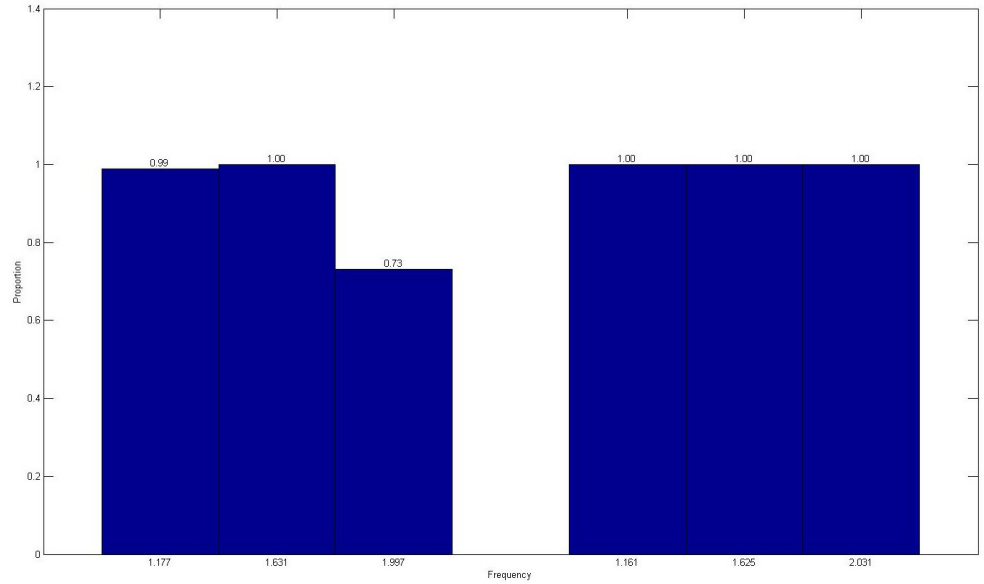

Figure 51: Experiment 2 step 18: Proportions with STFT and CWT (conveyor belt (f) Load Short 700)

In Figure 52 the full data set from the load operation was used in the experiments.

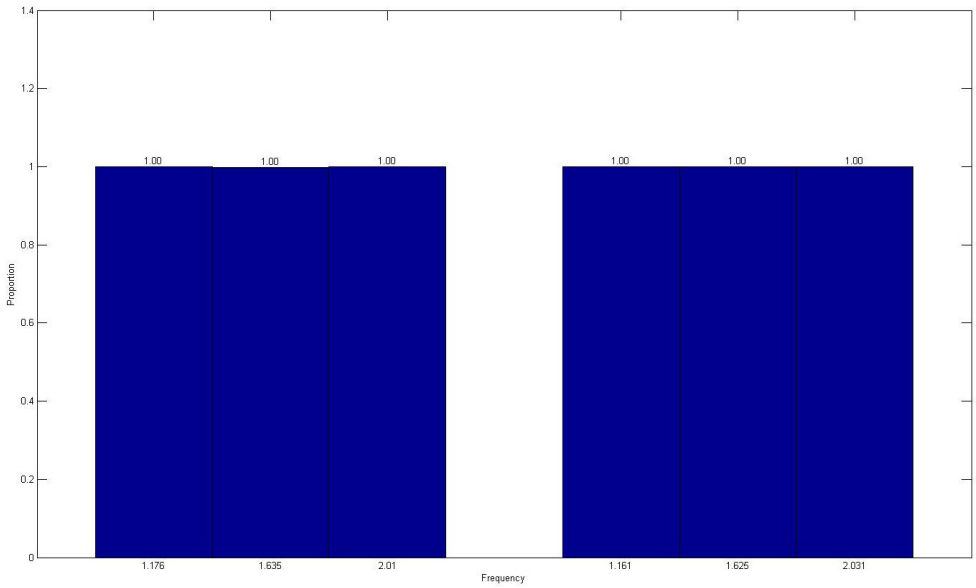

Figure 52: Experiment 2 step 18: Proportions with STFT and CWT (conveyor belt (f) Load Full Data 4000)

Figures 49-52 show that CWT gives a strong signal from every frequency at all data lengths. STFT gives weaker signals with short data lengths, but the strength becomes equal to CWT when the data segments get longer. STFT has better frequency resolution than CWT in all situations.