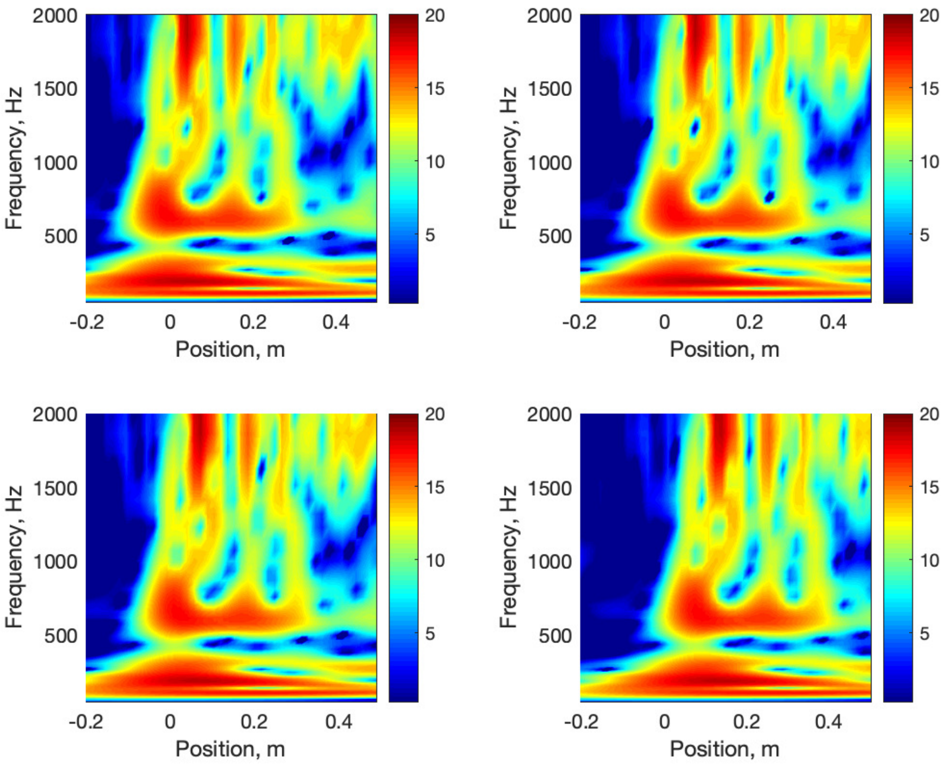
CHALMERS
UNIVERSITY OF TECHNOLOGY



Detection of Rail Squats from Axle Box Acceleration

Optimization of a Machine Learning Algorithm

Master's thesis in Sound and Vibration

FRIDA CARLVIK

Department of Architecture and Civil Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Master's thesis ACEX30

Detection of Rail Squats from Axle Box Acceleration

Optimization of a Machine Learning Algorithm

FRIDA CARLVIK

Department of Architecture and Civil Engineering
Division of Applied Acoustics
Chalmers University of Technology
Gothenburg, Sweden 2020

Detection of Rail Squats from Axle Box Acceleration
Optimization of a Machine Learning Algorithm

FRIDA CARLVIK

Supervisor: Astrid Pieringer (PhD), Department of Architecture and Civil Engineering
Examiner: Astrid Pieringer (PhD), Department of Architecture and Civil Engineering

Department of Architecture and Civil Engineering
Division of Applied Acoustics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Scalogram images of four axle-box acceleration measurements at a large squat

Department of Architecture and Civil Engineering
Gothenburg, Sweden 2020

Detection of Rail Squats from Axle Box Accelerations
Frida Carlvik
Department of Architecture and Civil Engineering
Division of Applied Acoustics
Chalmers University of Technology

Abstract

The detection of rail surface defects, such as squats, is vital to maintaining the structural health of railway tracks. Squats are a type of rail surface defect which results in large contact forces between the wheel and rail. Squats are self-sustaining; the increase in impact forces results in further rail deterioration. Squats are currently detected through various methods such as ultrasonic measurements, eddy currents, and human inspection. A promising alternative method of squat detection is the use of axle box acceleration (ABA) measurements. Even light squats are visible in the acceleration signal. The application of ABA data to automatically detect squats is an area of current research.

Using data and knowledge from a previous study of acoustic squat detection on the German railway, the thesis aimed to optimize a squat detection algorithm based on machine learning. Measured and simulated axle-box acceleration data were supplied from the previous research. The thesis investigated different methods of preprocessing acceleration data to improve the machine learning algorithm. Two algorithms were used- logistic regression and neural networks. The different methods of preprocessing data were spectrogram images, scalogram images, time-averaged wavelet power, and scale-averaged wavelet power.

To test the results, the data was divided into a training and a testing set. Furthermore, leave-one-out validation was conducted for the measured squats. Finally, the trained algorithm was tested on two 250 m test sequences of railway track. Issues were found with distinguishing insulated rail joints from squats. Furthermore, a higher success rate often led to a higher rate of false alarms. In these cases, the algorithm failed to generalize to new data. The final algorithm and method of preprocessing with scalogram images found 100% classification of medium to large squats and 87% classification of small squats. The algorithm found a total of 4 false alarms on the two test sequences, one of which was an insulated rail joint.

Although the final optimization did not find increased success in identifying small squats in comparison to the previous study, the use of ABA to identify squats was consolidated. Areas of further research are training the algorithm on tracks with varying track dynamics as well as testing other algorithms such as convolutional neural networks.

Keywords: rail surface defects, squats, axle box acceleration, machine learning, wavelet transform, scalogram, logistic regression, neural network

Acknowledgements

My greatest gratitude goes to the whole division of Technical Acoustics at Chalmers, where I not only found a field of engineering that I am passionate about but was given the motivation, tools, and opportunities to develop my knowledge and leadership skills.

I am grateful to DB Systemtechnik GmbH for their support for this thesis. Without the permission to use the data from their project AMONTRACK, this thesis would not have been possible.

For support during the thesis, I have some more thanks to give out. I am grateful to my examiner and supervisor Astrid Pieringer for support, brainstorming, and for giving me the freedom to be very creative with this thesis. I am also thankful to Jannik Theyssen, who wasn't my supervisor but still would discuss with me my ideas and troubles during his coffee breaks. And speaking of coffee - I was not the only morning person at the division, so thanks to Börje Wijk, I always knew that there was a fresh pot of coffee waiting for me in the kitchen. Congratulations on your retirement!

Frida Carlvik, Gothenburg, June 2020

Contents

List of Figures

List of Tables

1
Introduction

1.1 Background

Detection of surface defects is vital to maintaining the health of railway tracks and the safety of users of the railway. Squats are a type of surface defect which results in large contact forces between the wheel and rail. These impact forces may result in further rail deterioration [1]. Currently, some methods of detecting squats are through ultrasonic measurements, eddy currents, and using axle-box acceleration measurements [2]. Human inspection is a labor-intensive and possibly inefficient method of detection [3].

In a joint research project between Chalmers and DB Systemtechnik GmbH, AMON-TRACK, conducted in 2018/2019, an algorithm was developed to detect squats using axle-box acceleration [1]. The results were promising and this thesis aims to further develop and optimize the findings of the previous project.

Three central topics are investigated and act as pillars for the development of the thesis. First, railway noise and vibrations are studied. Second, wavelet analysis is studied, and a comparative study is made between wavelet analysis and other types of signal processing tools. Finally, machine learning is used for building the detection software.

1.2 Problem Statement and Purpose

The automatic detection of surface defects has been investigated in several parallel studies [1][3][4]. Input data has been both acoustic (axle-box acceleration, ABA) and visual (photograph and ultrasound). In the studies by Pieringer et. al [1] and Jamshidi et. al [4], machine learning was used to detect squats. By detecting surface defects through measurement cars with axle-box acceleration measurements, rail faults may be detected earlier compared to other methods such as ultrasound. Early detection results in higher safety and possibly lower maintenance costs.

A common method of automatic detection, ultrasound, can detect squats by finding cracks between 5 mm to 7 mm deep; such a squat is often too large to grind [2]. Furthermore, automatic detection minimizes the need for ocular inspection, resulting in lower labor costs and less risk due to personnel on the tracks. As even small squats lead to large forces on the wheel, ABA is a promising method of early detection.

The thesis aims to optimize a machine learning algorithm to detect squats using knowledge and resources from a previous study by A. Pieringer et. al [1]. The effect of different rail surfaces on the axle-box acceleration is studied. Different methods of preprocessing acoustic data are investigated. Furthermore, it is tested whether the machine learning algorithm itself can be modified to increase successful classification.

To optimize the machine learning algorithm, the impact of different input parameters as well as different machine learning methods were investigated. This includes investigating different methods of preprocessing acceleration data. Previously, some deteriorated insulated rail joints were falsely classified as squats by the trained algorithm. As insulated rail joints in good condition are not a hazard, it is interesting to attempt to distinguish insulated rail joints from squats.

1.3 Limitations

The data used in the thesis, consisting of both measured ans simulated data, was gathered in a previous study [1]. Gathering and simulating data is outside the scope of the thesis. The few examples of measured squats may be limiting on the success of the algorithm, however with the simulated data, the data was sufficient to train the algorithms. The data gathered regards the railway on the German rail network and using a special monitoring car. The applicability of the findings in other countries is not investigated. However, the general methodology applies to all railway though the algorithm may need to be trained on new data relevant for the new case.

1.4 Placement in Research

The use of both axle-box acceleration and machine learning to detect algorithms has previously been a topic of research. The contribution of this thesis to research is the further investigation of different preprocessing methods to differentiate surface defects from healthy rails. Different machine learning algorithms are used as tools and their relevance for this type of pattern recognition were evaluated. Finally, by comparing the measured data with data from other papers on the subject, a difference in the spectral content of the measured ABA was found - perhaps this depends on the variation in track properties in different countries.

1.5 Ethical Aspects

The purpose of detecting surface defects through machine learning is to develop methods of early detection as well as to minimize the need for manual inspection. Squats lead to large forces on the wheel, cracks underneath the surface, and in the worst-case rail failure. As such, the detection and repair of squats is a question of ethics. Early detection may more be a question of economics as the repair of smaller squats is less invasive. Safety is central to the purpose of the thesis; the ethical substance of the research is clear.

2
Theory

The thesis covers three central topics - railway noise and vibration, signal processing, and machine learning. The goal of the theory section is to explain how the coupled system between axle-box, wheel, and rail is impacted by rail squats and other surface defects. Furthermore, it is of interest how this data can be processed and used for machine learning. A literature study of previous research on the topic of automatic detection of squats is presented in the section.

2.1 Railway Noise, Vibration and Defects

This section aims to investigate the correlation between surface defects of rails and the noise and vibration caused at the axle-box. To do so, knowledge of track geometry and physics is needed. The interaction between wheel and rail is investigated concerning the generation of noise and vibration. By correlating the wheel-rail contact forces with vibrations, squat detection through acceleration measurements can be investigated.

2.1.1 Railway Components and Definitions

The components of the railway can be organized as those regarding the track and those regarding the car. The track components, shown in Figure 2.1, are rail, fastening, sleepers, and ballast. The purpose of the track is to transfer the load of the passing vehicle uniformly to the ground. Relative to the noise of the railway, the rail pad is an important part of the track. The rail pad is located between the foot of the rail and the sleeper, serving the purpose of protecting the sleeper from high impact loads [5].

2.1.1.1 Contact Patch

The wheel-rail contact patch is the area of contact of the rolling wheel along the surface of the rail. The small contact patch between the hard surfaces of rail and wheel results in low losses, making rail transport an energy-efficient mode of transportation, according to R. Lewis and U. Olofsson in 'Wheel-Rail Interface Handbook' [7]. The small contact area leads to large contact forces and stresses on both the wheel and the rail. This may lead both to material yielding and fatigue. The contact

Figure 2.1: Main components of the rail track
source: adapted from [6]

patch is approximately 18 mm longitudinally and 11 mm laterally [7]; a damaged rail or wheel leads to a more circular patch.

The friction in the contact patch affects both energy consumption and wear of the rail and wheel. Friction coefficients vary largely between 0.08 and 0.5 [7]. A low friction coefficient may result in sliding wheels and wheel flats, whereas a large friction coefficient value may result in large energy consumption and excessive wear of the components.

Friction combined with the high rolling contact forces leads to an increase in the wear of the materials [7]. Furthermore, traction and braking may result in wheel sliding, which in turn can lead to rail burns and wheel flats. The result of these unfavorable consequences increases irregularities on the contact patch as well as worn profile geometries. This in turn leads to poor vehicle dynamics, an increase in contact forces, and an increase in noise and vibrations. In the worst-case scenario, fracture of either of the components or wheel flange climbing can lead to derailment.

As the contact patch determines the transfer of forces between wheel and rail, only the surface defects which are within the contact patch of the passing wheel will be detected by axle-box acceleration.

2.1.2 Noise and Vibration of the Wheel Rail Interface

Wheel-rail interface theory was pioneered by Heinrich Hertz in the 1880s, through studies of the elastic contacts which were applied to railway engineering [7]. Rolling noise is due to noise from the rail, the wheel, and the interaction of the two. An understanding of the excitation at the axle-box due to forces between the wheel and rail can increase the identification of squats and other rolling contact fatigue.

The track can be seen as an essentially infinite structure that acts as a waveguide. This allows for one or more structural waves to propagate along the rail. In an experiment presented by David Thompson [5], the frequency response of the rail was found by exciting the rail with an impact hammer and measuring with accelerometers. The result of this experiment found differences in the response depending on if the acceleration was measured on top of a sleeper or between sleepers. When measuring between sleepers, a peak was found around 1 kHz, with a dependency on the spacing of the sleepers. This mode is called the pinned-pinned mode. The resonance of the rail mass on the rail pad stiffness results in another peak in the frequency response function. A lower rail pad stiffness results in a lower resonance frequency; the rail pad has considerable influence on the overall rail frequency response function. Finally, in the frequency response functions, a resonance at 100 Hz corresponds to the oscillation between the total mass and the ballast stiffness.

Damping of the rail is due to losses in fastening systems and the transfer of energy into the sleepers and the ground. By altering the rail pad stiffness, the dampening is effected. Softer rail pads result in isolation of the sleepers, but the rail itself is less damped, allowing the rail to vibrate longer. Stiffer rail pads result in the vibration of the sleepers, but the rail is more damped. The damping of the rail results in an exponential decay of the vibration amplitude; the track decay rate is usually presented in dB/m [5]. The track decay rate is frequency-dependent; generally higher frequencies around 1 kHz-2 kHz have a slower decay rate.

Railway wheels typically have little damping and therefore the vibration is strongly defined by the resonances. For approximating mode shapes, the wheel can be compared to a flat disc. The mode shapes of a flat disc can be axial or radial. The axial modes are defined in number by the number of nodal circles and the number of nodal diameters. The wheel differs from a flat disc as the shape is not symmetric. This asymmetry is primarily due to the wheel flange, a part of the wheel which acts as a safety measure against derailment. Furthermore, the wheel web may be curved. The lack of symmetry results in the coupling of different modes.

The frequency response for a free wheel can be calculated with knowledge of the modes of the wheel and the damping. The mode shapes and amplitudes can be calculated with FE software, but the damping is either measured or estimated. A typical wheel is the UIC 920 mm wheel, for which the frequency response function of the wheelset was found by Thompson [5]. The radial mobility showed resonances around 300 Hz and 1150 Hz with an anti-resonance around 500 Hz. Above 1 kHz the wheelset had a higher modal density. The axial mobility showed more resonances. The half-power bandwidth of the wheel modes was found to be only a few Hz, which is related to the undamped characteristic of the wheel. The rotation of the wheel results in the eigenfrequencies splitting into pairs.

The primary source of rolling noise at the wheel-rail interface is roughness, which causes relative motion between the wheel and rail [5]. Small size micro-roughness is necessary for adhesion, leading to better traction and braking. However, the larger sized, macro-roughness contributes to noise. Unevenness on the rail or wheel results in noise, the frequency can be determined by:

$$f = \frac{V}{\lambda} \tag{2.1}$$

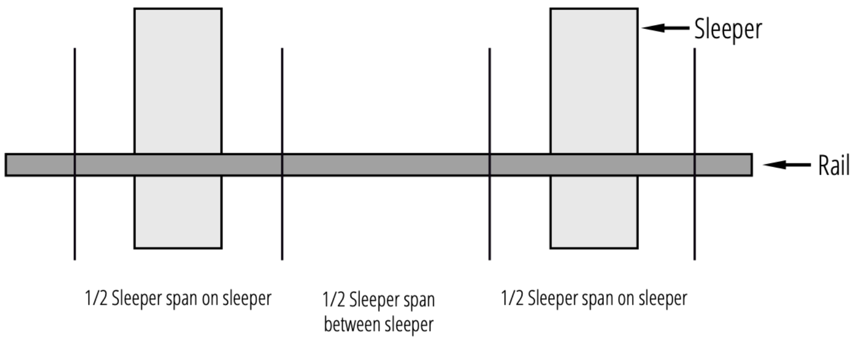


Figure 2.2: Definition of the two spans 'half sleeper span' centered on or centered between sleepers

where V is the velocity of the train and λ is the wavelength of the surface wave (unevenness) on the track or rail [5]. The wavelength can vary in size from a few millimeters to several meters. The longer wavelength can correspond to variations in the rail bed or straightness of the rail. Shorter, semi-periodic roughness with wavelength in order of magnitude of about 50 mm may be due to corrugation.

2.1.3 Rolling Contact Fatigue and Squats

The primary focus of the thesis is the detection of a type of rolling contact fatigue of the rail profile, squats. Squats are characterized visually as a local visible surface deformation and characterized in the wheel-rail interaction by an increase in dynamic force. The force itself may effect the continued growth of the deformation [2]. Squats are a class of rolling contact fatigue defects that develop from small surface irregularities. If the surface defect grows beyond a critical size the irregularity may risk growing into a squat. On Dutch railways, this critical size has been assessed as between 6 mm to 8 mm both for rolling and traverse directions [3]. If detected early, squats can be easily treated. However, as squats develop in size crack growth beneath the surface may occur. Squats were identified as a distinct type of failure in the 1970s, and in 2009 there was still limited research in the subject [2].

Squat occurrence is characterized by large local plastic deformation. The increase in dynamic wheel-rail contact force upon an existing squat causes rapid local track deterioration [2]. The occurrence of squats is often isolated. As no consensus in the cause of squats and how they therefore can be localized preemptively has been found, the occurrence of squats is often seemingly random. Predicting and preventing squats is therefore a difficult practice and an area of active research.

Research in correlating occurrence of squats with their track parameters has been conducted to increase the possibilities of preventative actions through prediction. On the Dutch railway, it was found that approximately 74% of the squats were found on the ½ of the rails centered on the sleepers (Fig 2.2), which implies that stiffness and damping characteristics of the track may play a role in the emergence of squats [2]. Furthermore, short pitch corrugation, i.e. variation of rail surface with a short wavelength [8], was found near 72% of the squats. These corrugations had a wavelength between 2 cm and 6 cm.

The squats themselves shared characteristics with the short pitch corrugation, in terms of the wave pattern. Furthermore, a study in the UK found that 75% of squats were related to either corrugation, welds, or periodic indentations in the rail running surface [2]. Other correlations found with the occurrence of squats were welds (10-15%), indentations from hard alien objects into the wheel or between the wheels and rail, vertical misalignment of rail, non-uniform wear, and non-uniform plastic deformation, skidding and sliding during traction and braking. Wheel burns, caused by braking, can also be seen as an initiation source of squats.

Counter-measures to squats are presented by Z. Li in the 'Wheel-Rail Interface Handbook' [2]. Small squats can often be removed by grinding, whereas larger squats require replacement of the affected rail. Replacing the rail results in two new welds. Welds themselves may be sensitive to squat formation due to the heat-affected zone. To conclude, there is a clear advantage in detecting squats early.

2.1.3.1 Impact Noise Due to Insulated Rail Joints

Rail joints result in impact noise due to a discontinuity for the passing wheel. The rail joint can be defined geometrically by a gap width, a step height, and a dip angle [5]. The dip angle and the step height were found to be determining in the impact noise, and the gap width is negligible.

2.2 Signal Processing Methods

Signal processing and analysis serves the purpose of identifying relevant information from a signal. This can be done with transformations between different domains, such as time- and frequency domain. For the sake of preparing measurement data for teaching a machine learning algorithm, the information should represent the relevant properties of the signal. For efficiency in the machine learning algorithm, a second goal in signal processing is to reduce the size of the data.

The Fourier transform is a method of obtaining the frequency content of a signal from the time-domain data. The concept behind the Fourier transform is the idea that any continuous signal can be interpreted as the sum of infinite oscillations [9], or sine waves, with varying amplitude and frequency. As such, two domains are evaluated - the time domain and the frequency domain. In the frequency domain, there is no localization in time, and vice-versa. The Fourier transform can be expressed as:

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{j\omega t}d\omega \tag{2.2a}$$

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt \tag{2.2b}$$

where $F(\omega)$ is the signal in the frequency domain, $f(t)$ is the signal in time domain, and $e^{-j\omega t}$ represents harmonic oscillation given an angular frequency ω and time t.

2.2.1 Representation of a Signal Simultaneously in Time and Frequency Domain

For transient signals, it is interesting to know information of how the frequency content of a signal is located in time. The Fourier transform proves insufficient in such a case. Furthermore, the simultaneous representation of time and frequency is limited by the Heisenberg uncertainty principle, which states that a particle cannot be described by its position and momentum with simultaneous, arbitrarily small resolution [10]. Although the uncertainty principle refers to quantum mechanics, the principle remains true for discrete signals; the increased resolution in the time domain is limited by the decreased resolution in the frequency domain.

A method of representing information in both time and frequency domain is the short-time Fourier transform, STFT. The localization in time is found by isolating portions of a signal with a window [9]. Performing a short-time Fourier transform results in a spectrogram which shows the magnitude as a function of frequency and window location.

The short-time Fourier transform can be expressed as:

$$F(\omega, t) = \int_{-\infty}^{\infty} f(t)w(t-u)e^{-j\omega t}dt \tag{2.3}$$

where the function $f(t)$ is the signal in the time domain, w is the window function which is shifted in time by u and $e^{-j\omega t}$ is a harmonic oscillation as a function of time t and frequency ω. A smaller window results in better time localization. However, the width of the spectral peak broadens, resulting in a lower frequency localization.

Windowing a signal before performing the Fourier transform results in a main lobe around the frequency of interest and side lobes [11]. The window determines how each segment of the signal is cut; a smoother window leads to a wider main lobe and therefore losses in spectral resolution, whereas a very sharp window results in a higher noise floor of the side lobes but a sharper main lobe. The window can be shaped in many different ways which varies the shape of both the main and side lobes.

If the shape of the window is decaying toward the edges of the window, information at the edges of each segment is under-represented. Overlapping the segments of the signal assures that information at the edges of each segment not being lost. However, the overlapping also leads to a smoothing of the analysis process, which may decrease resolution. In implementing a short-time Fourier transform, the parameters that must be chosen are the window shape, the window size, and the overlap.

Wavelet decomposition is a method of analyzing a signal which, compared to Fourier decomposition, presents some localization in time with the price of a less defined frequency representation [12]. Rather than assuming that the signal is composed of infinitely many sine waves with different frequencies and phases, each wavelet belongs mainly to a frequency band and has a localization in time. The wavelet decomposition is defined by a mother wavelet, which determines the shape of the function. The mother wavelet is scaled and shifted into several basis functions. Each basis function is convolved with the signal, resulting in time and frequency-specific magnitudes.

The continuous wavelet transform, CWT, is given by Equation 2.4 [13].

$$W_x(s,\tau) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} x(t)\psi^*\Big(\frac{t-\tau}{s}\Big)dt \tag{2.4}$$

where the signal $x(t)$ is multiplied by the complex conjugate of the mother wavelet $\psi(t)$ which is scaled by $s > 0$ and shifted in time by the continuous variable τ. The result is a continuous variable $W_x(s,\tau)$ which are the wavelet coefficients. The function $\psi(t)$ has some localization in time and frequency, and has the total energy equal to zero, as shown in Equation 2.5.

$$\int \psi(t)dt = 0 \tag{2.5}$$

Compared to the short-time Fourier transform, where the window size is kept constant and the number of oscillations within the window increases to capture higher frequencies, the wavelet has a constant number of oscillations regardless of frequency. Instead, the width of the wavelet is skewed, resulting in a broader or more narrow window [10]. The result is a varying resolution depending on the scale of the wavelet.

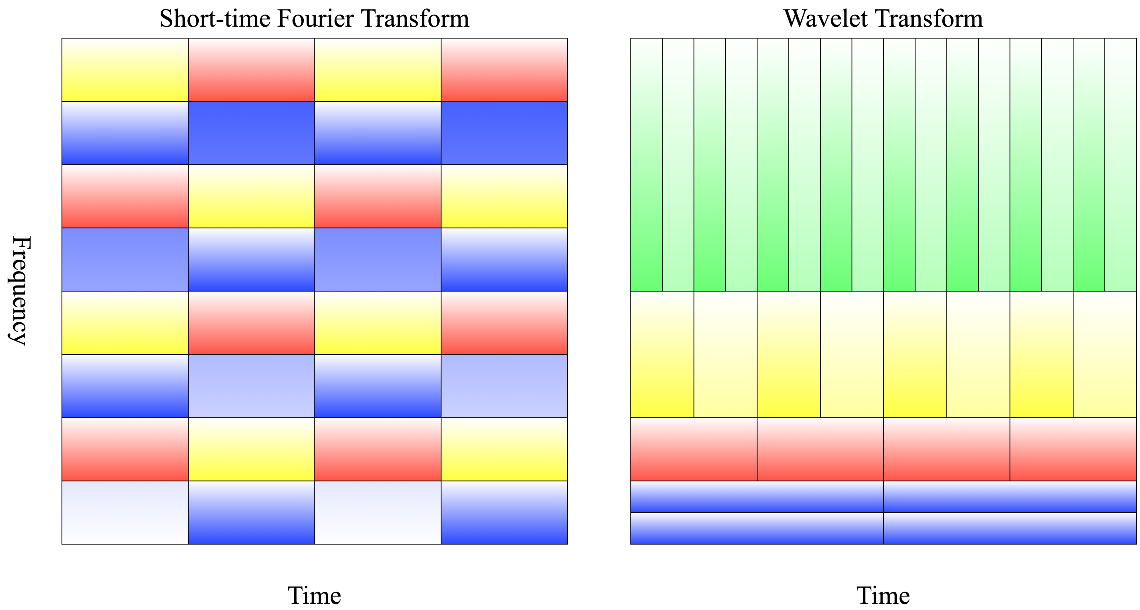


Figure 2.3: Example of resolution in time and frequency domain of the short-time Fourier transform (left) and the continuous wavelet transform (right)

Two common mother wavelets are the Haar Wavelet and the sinc wavelet. The former results in very fine time localization by accepting poor frequency resolution and the latter gives up time localization to achieve a finer frequency resolution [14].

The Morse wavelet is an analytical wavelet which is available in many libraries on Matlab [15].

An example of the resolution in time and frequency domain of the two transforms can be seen in Figure 2.3. The resolution for the short-time Fourier transform is consistent throughout both time and frequency. For the wavelet transform, however, the frequency resolution decreases as the frequency increases. Instead, the time resolution increases at higher frequencies. For both transforms, if the resolution in the frequency domain is increased, the resolution in time must be proportionately decreased.

2.2.1.1 Time Averaged Wavelet Power

A method of minimizing the number of samples representing the data is to average the wavelet transform for each scale. This results in average levels at each scale over a given segment of time. The justification behind averaging over time is that the signals analyzed are segments that are just long enough to represent passing a squat. Although the ABA when passing a squat is highly transient, the short segment shows clear prominence in certain frequencies.

The time averaged wavelet spectrum can be calculated according to equation 2.6 [16].

$$\bar{W_n}^2(s) = \frac{1}{n_a} \sum_{n=n_1}^{n_2} |W_n(s)|^2 \tag{2.6}$$

where n_a is the number of points averaged over, defined by $n_a = n_2 - n_1 + 1$ and $W_n(s)$ is the wavelet transform at a given sample n and scale s.

2.2.1.2 Scale Averaged Wavelet Power

The average energy over a given number of scales (frequency ranges) in a signal can be found through the scale averaged wavelet power. This is summarized by equation 2.7 [17].

$$SAP^2(n) = \frac{1}{M} \sum_{i=1}^{M} |W_s(n)|^2 \tag{2.7}$$

given the scales chosen are $i = 1, 2, ..., M$. The function $W_s(n)$ is the continuous wavelet transform. The averaging may be performed across all scales of the wavelet transform or of specific scales of interest.

2.3 Machine Learning and Neural Networks

Machine learning is a method of teaching a program, or a computer, to recognize patterns and solve tasks. Machine learning can be either supervised, where a set of features is given along with a correct identification of the features, or unsupervised where only the set of features is given. In unsupervised learning, the machine performs other tasks such as clustering data into different groups. In the case of detecting rail squats and other irregularities, classification problems are of use. The classification problem asks the question if a certain case, or set of variables, belongs to a class. In the case of rail squats, the question could be "is there a squat on this piece of rail track?". With knowledge of squat locations, supervised learning is possible.

A classification problem can either have single or multiple classes. In the case of a single class, there is a negative class represented by a 0 and a positive class represented by a 1. If there are three or more possible outcomes, multiclass classification is used, where each outcome is either 1 if true for that outcome or 0 if false.

2.3.1 Logistic Regression

Regression is a technique for solving many types of machine learning problems, among them classification problems. The regression model can either be linear or logistic. Logistic regression aims to predict the probability that a set of features belong to a certain class. To do so, the relationship between the input parameters and the given outcome must be expressed. This can be done as following [18]:

$$y = \Theta_0 + \Theta_1 x_1 + \Theta_2 x_2 + ... + \Theta_{n-1} x_{n-1} + \Theta_n x_n \tag{2.8}$$

where y is the outcome to be predicted, $\mathbf{x}$ is the set of input features related to the output y, and $\mathbf{\Theta}$ is a coefficient relating the contribution of x_i to y. However, as the values of x_i and Θ_i vary, the output y can take on any value. This is useful for linear regression. For classification problems the question is whether $y = 1$ or $y = 0$. The predicted outcome should be expressed as the probability of the input features belonging to $y = 0$ or $y = 1$. This can be expressed with the hypothesis defined as $h_\Theta = g(\mathbf{\Theta}^\mathbf{T}\mathbf{x})$ given a function g that satisfies $0 \leq h_\mathbf{\Theta} \leq 1$.

One function which satisfies the previous equality is the sigmoid function, given by Equation 2.9:

$$g(z) = \frac{1}{1 + e^{-z}} \tag{2.9}$$

To determine whether an input belongs to a certain class, a threshold classifier is used. For each possible classification, Equation 2.10 is evaluated:

$$y = \begin{cases} 1 & \text{if } h_\mathbf{\Theta}(\mathbf{x}) \geq 0.5 \\ 0 & \text{if } h_\mathbf{\Theta}(\mathbf{x}) < 0.5 \end{cases} \tag{2.10}$$

By adjusting $\mathbf{\Theta}$ to our data, we have the classification hypothesis:

$$h_\mathbf{\Theta}(\mathbf{x}) = p(y = 1 | \mathbf{x}; \mathbf{\Theta}) \tag{2.11}$$

Equation 2.11 can be read as the probability that $y = 1$ given $\mathbf{x}$ parameterized by $\mathbf{\Theta}$. For multi-class classification, this is repeated for all classes, with coefficients $\mathbf{\Theta}$ belonging to that class. The classification is given by the classifier which results in the maximum value of $h^i_{\mathbf{\Theta}}(\mathbf{x})$ as this is the classification with the highest probability. This method is called one vs. all classification [19].

2.3.1.1 Cost Function

Given a set of values of $\mathbf{\Theta}$, it is of interest to estimate how well Equation 2.11 classifies a set of data. This is determined by the cost function. The cost function finds the difference between the estimated value of $h_{\mathbf{\Theta}}(\mathbf{x})$ and the true value of y for all sets of data. For logistic regression, the cost can be expressed by Equation 2.12

$$Cost(h(\mathbf{x}), y) = \begin{cases} -\log(h_{\mathbf{\Theta}}(\mathbf{x})) & \text{if } y = 1 \\ -\log(1 - h_{\mathbf{\Theta}}(\mathbf{x})) & \text{if } y = 0 \end{cases} \tag{2.12}$$

As the true value of y only can take on the values 0 or 1, Equation 2.12 can be simplified by:

$$Cost(h(\mathbf{x}), y) = -y \log(h_{\mathbf{\Theta}}(\mathbf{x})) - (1 - y) \log(1 - h_{\mathbf{\Theta}}(\mathbf{x})) \tag{2.13}$$

The total cost, $J(\mathbf{\Theta})$, for all data sets is given by:

$$J(\mathbf{\Theta}) = -\frac{1}{m} \left[\sum_{i=1}^{m} y^{(i)} \log \left(h_{\mathbf{\Theta}}(\mathbf{x}^{(i)}) \right) + (1 - y^{(i)}) \log \left(1 - h_{\mathbf{\Theta}}(\mathbf{x}^{(i)}) \right) \right] \tag{2.14}$$

where m is the number of training examples in the data and i is the index of each example in the data. The goal of training the machine learning algorithm is to minimize the cost, as this means that the difference between the guessed value and the true value is small. This is done by fitting the parameters, $\mathbf{\Theta}$. One method of minimizing the cost function is through gradient descent, that is, finding the slope of the cost function and following the path of descent iteratively. The gradient descent function is given in Equation 2.15.

$$\Theta_j := \Theta_j - \alpha \frac{\delta}{\delta \Theta_j} J(\mathbf{\Theta}) \tag{2.15}$$

which is updated for each iteration j, and α is called the step-size, which determines the rate of change of the $\mathbf{\Theta}$ values after each iteration. As the step-size is a coefficient belonging to the derivative, the rate of change naturally decreases as the slope decreases.

Although the gradient descent is easily implemented into mathematical programming software such as Matlab, there are many advanced optimization algorithms which are often faster than gradient descent. Another advantage of using a library available algorithm is that it usually cuts out the need to choose an appropriate value for α.

2.3.2 Neural Networks

Neural networks are a method of machine learning which has similarity with the network of neurons in the brain. A network of parameters is organized into layers where each parameter from one layer is connected to every parameter in a neighboring layer with some amount of weighting. The first layer is called the input layer, the final layer named the output layer, and each layer in between the two is considered a hidden layer.

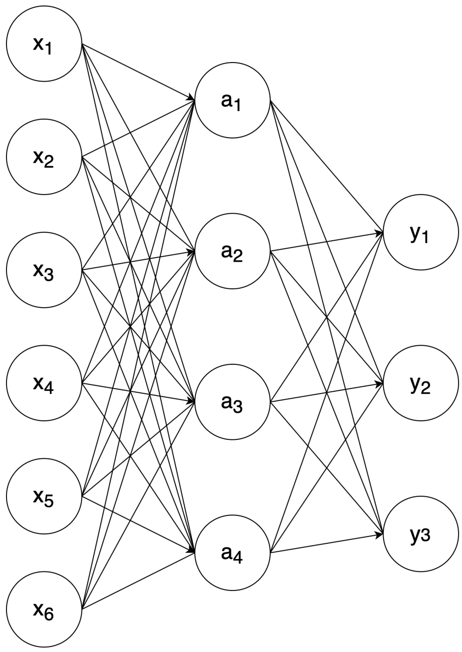


Figure 2.4: Visual representation of a neural network with 6 features, a hidden layer with size 4 and 3 possible outcomes

For neural networks, the same task of minimizing a cost function is primary. The cost of each layer is found by first calculating the error in the output layer and propagating backward to find the error of each preceding layer.

The method of creating and training a neural network can be summarized as follows, according to lecture notes from an online Coursera course in Machine Learning from Stanford University [19]. The first step is choosing a suitable architecture, which is the number of input features, number of classes, and the sizes and number of hidden layers. The number of hidden layers increases computation; as a rule of thumb one hidden layer is used. The second step is to randomly initialize weights between a given range, e.g. $-0.5 \leq \Theta_1 \leq 0.5$. Forward propagation is implemented to compute the cost function $J(\mathbf{\Theta})$ and backpropagation is used to compute the partial derivatives $\delta J/\delta\Theta_j$. To check that the algorithm is functioning correctly, gradient checking is done for a few numerical estimates. Finally, gradient descent or another optimization algorithm is used to minimize $J(\mathbf{\Theta})$ until a satisfactory error level is found.

2.3.3 Troubleshooting Machine Learning Issues

The machine learning theory presented is based on curve fitting and statistics; the assumption made is that the cost function can be minimized and that the minimum value can be found. However, this is not always true. For example, local minima in the cost function may result in the algorithm failing to update to better values.

The number of features, which directly corresponds to the number of coefficients $\mathbf{\Theta}$, needs to be sufficient to properly describe the correct classification. If too few features are available to describe the problem, there is a problem of underfitting and the algorithm has a high bias. The converse, too many features, may instead lead to high variance. Although many features may lead to very little cost, the algorithm may fail when generalizing new data. To survey overfitting, data may be divided into a training set and a test set. The training set is used to train the algorithm and the test set is used to assess the ability of the algorithm to generalize to new examples.

The problem of overfitting has some solutions. One method is simply to reduce the number of features. This can be done manually or by a model selection algorithm. Another method is regularization which keeps all features but reduces their magnitude. This is a valuable method if the data contains many important parameters that contribute to the classification in a small way. The regularization parameter is represented by λ. The cost function $J(\mathbf{\Theta})$ is calculated in the same way as in equation 2.14 but with an added penalty of $\frac{\lambda}{2m}\sum_{j=1}^{n}\Theta_j^2$ where n is the number of features.

2.4 Literature Study - Automatic Squat Detection

The automatic detection of surface defects on rail tracks has been investigated in several previous studies [1][3][4]. In the papers by A. Pieringer et. al [1] and M. Molodova et. al [3], axle-box acceleration was used as measured data. In the paper by A. Jamshidi et. al [4], image data and ultrasonic measurements are used. The two studies which used ABA differed through processing method and detection algorithm.

2.4.1 Surface Defect Detection Using Wavelets

In 2013, a Dutch study was published where surface defects were detected using wavelet analysis of axle-box acceleration (ABA) measurements [3]. The research investigated a method of early detection of squats. Furthermore, the power spectrum in frequency ranges related to squats could indicate the severity of the squat. The investigated squats were on rail tracks in the Netherlands. The study developed an algorithm that had a detection rate of over 78% for light squats and 100% for severe squats.

Axle-box acceleration methods are suggested in the report due to their low cost and easy maintenance. By installing accelerometers on the axle-box of specialized measuring trains, the vertical and longitudinal acceleration was measured. It was early identified that the acceleration data in the time domain was not sufficient for detecting squats.

A measuring train runs over the entire Dutch network twice annually, collecting geometrical irregularity data. Information about the position of squats, as well as e.g. joints, switches, and level crossings are found in the database storing this measurement data. To keep up to date for the research, short track irregularities were monitored regularly on the sections where ABA was measured.

The data was preprocessed to improve the signal-to-noise ratio. The processing included low-pass filtering, averaging several measurements, and finding the scale-averaged wavelet power. The method of using wavelets was chosen due to the benefits found when investigating transient signals with varying frequency composition. The chosen mother wavelet was the Morlet wavelet. The scale-averaged wavelet power, SAWP, was calculated in the frequency band related to squats, which varied depending on the stage of the squats.

The prediction of a squat was made if the value of the calculated SAWP exceeded a certain threshold. The threshold was determined statistically and depending on whether light squats or severe squats were investigated. Finally, the severity was assessed either by the frequency content or by the relation of the power spectrum at 300 Hz and the area of the squat. The predictions were validated by visual inspection of the track.

The frequency response of the ABA was observed up to 2.5 kHz. The maximum of the wavelet power spectrum for light squats was found around 300 Hz. For moderate to severe squats, the power spectrum showed a strong response in two ranges: below 600 Hz and between 600 Hz to 2000 Hz. The maximum was between 250 Hz and 350 Hz, as well as between 1000 Hz and 1300 Hz.

One possible issue with the measurement method is hunting, meaning a lateral oscillation of the wheel [7]. This could result in the wheel not passing all squats at each pass by. The probability of hunting leading to a squat remaining undetected increase if the squat is smaller. A method of mitigating this error was passing each track several times and finding the average acceleration.

The results of the prediction method showed a 78% hit rate and 15% false alarms for light squats and a 100% hit rate for severe squats. Insulated joints were detected as squats when detecting severe squats, as the ABA at these locations is strong, and the frequency response partially overlaps that of the severe squat. The report also suggests the possibility of expanding acceleration measurements on in-service trains to cover more of the railway infrastructure. Furthermore, the management of the large amounts of accumulated data was pinpointed as a focus of further research.

2.4.2 Squat Detection Using Camera Images

A study from 2017 assesses the possibility of using video camera images and ultrasonic measurement for risk assessment of railway [4]. The image data was used to detect rail defects and the ultrasonic measurements were used for crack growth analysis. The focus of the study was to investigate handling large amounts of monitoring data as well as developing a method for determining the risk of failure for the railway.

For detecting squats a deep convolutional neural network, DCNN, was used. Convolutional neural networks consist of several layers; in each layer, features are extracted from the previous layer. The study used squats as an example of a surface defect to be monitored. The training time for the DCNN was about 40 hours. The trained DCNN had an accuracy of about 96.9% in predicting squats. Furthermore, the study produced a correlation between the appearance of a squat, crack growth due to the use of the rail, and the risk of failure. The produced results were a failure probability at each assessed squat.

2.4.3 Squat Detection Using Short-Time Fourier Transform

In 2019, a joint project between DB Systemtechnik GmbH and Chalmers University of Technology was published [1]. The research and the included measurement and simulation data are the basis of this thesis. In the study, a measurement car "Schallmesswagen" (SMW) is used which is equipped both with a microphone above the measurement bogie, as well as accelerometers on all four axle boxes of the bogie. The advantages of using acoustic data are presented as two-fold. Firstly, light squats are more easily classified by acoustic data than by photographs or ultrasound measurements. Furthermore, the severity of the squat can be judged since the measured quantity - acceleration - can be directly related to wheel/rail contact forces. The report also identifies the issue that defects located outside the running band cannot be detected by ABA measurements.

Beyond measurement data, simulated axle-box acceleration data is collected using a model called WERAN (WhEel/RAil Noise). The method is a computationally efficient method based on Kalker's variational method. The model can simulate both vertical and lateral wheel/rail interaction, but only the vertical interaction was included. The squats were designed for the simulations with varying lengths, width, and depth. Healthy rail was simulated with varying roughness. The depths are between 0.1 mm and 1.4 mm and the diameters vary between 10 mm to 70 mm. In general, the diameter of the squat was more influential to the maximum ABA as well as the shape of the squat compared to the depth. This was not true for very shallow squats.

The characteristic of the axle-box acceleration when passing a squat was identified in the research. First, as the wheel enters the squat the acceleration decreases due to unloading in the contact force. This decrease in acceleration is followed by a strong peak in acceleration due to the subsequent impact. A larger diameter led to a larger peak in magnitude as well as a wider peak in time. The frequency response of the simulated ABA showed similarities in frequency response above 700 Hz for different size squats and larger variation in frequency response below 700 Hz. Furthermore, the simulation investigated the influence of changing the rail pad stiffness and found that a decreased stiffness resulted in a higher and narrower first peak of the ABA.

The ABA derived from the simulations was compared to measured acoustic signatures of squats. This was done by using the measurement car SMW, on a 5 km length of track in Munich-North in March 2019. The track was investigated visually to localize any squats and other surface irregularities. The measurement conducted using the SMW was repeated four times in the primary traveling direction and four times in the opposite direction, with a target velocity of 80 km/h. During the measurement campaign, 8 squats were noted along with 5 insulated rail joints and two switches. The simulated data consisted of 68 squats with varying sizes. The simulated and measured data showed similarities both in time-domain as well as the frequency domain. However, some peaks were shifted in frequency - the measured peak at 1.4 kHz appeared in the simulated data at 1.2 kHz for a given squat.

A pattern recognition algorithm was developed using a logistic regression classifier. The input data was the spectrogram of the ABA for 93 by 29 acceleration levels. The produced spectrogram was found through a spatial resolution of 1 mm, the frequency resolution of 22 Hz, and a 95% overlap. For severe squats, 100% detection was achieved, whereas 87% detection was found for light squats. The validation was made only on measured squats, and the squat which was to be tested was removed from the training data for each validation. Furthermore, the algorithm was tested on two test sequences of the measured passbys. False-positive classification occurred in the appearance of insulated rail joints and at a few locations of measured rail which were possibly false alarms.

2.4.4 Comparison of the Literature

The three papers investigated the possibility of automatically detecting squats by equipping train-cars with some type of measurement device. The methods found promising results through different data types, processing methods and algorithm complexity.

The method of detecting squats with ultrasonic images, as tested by A. Jamshidi et. al [4], requires crack growth to detect squats. This is disadvantageous compared to acceleration measurements in regards to detecting less severe squats. However, the successful identification was much higher. The high accuracy could also be due to the complex algorithm, which required much more data and training time compared to both the other methods.

The scale-averaged wavelet power method by M. Molodova et. al [3] did not use any machine learning, decreasing the complexity. The detection of small squats was lower compared to the paper by Pieringer et. al [1]. Once again, it is unclear whether this is due to the processing method or the detection method. The frequency response of the squats varied between the two papers, which is likely due to the difference in track parameters as well as properties of the train car between the two countries.

3 Building a Machine Learning Algorithm

Two different types of machine learning algorithms were designed and tested using Matlab. The algorithms chosen were logistic regression and a neural network with a single hidden layer. The mathematics explained in Section 2.3 were used to design the cost function and the hypothesis function used was the sigmoid function. Two different descent functions were used depending on the machine learning method. In the case of the logistic regression the built in function `fminunc.m` was used; in the case of neural networks, `fmincg.m`, written by C.E. Rasmussen, was used. The reasoning behind this was that `fmincg.m` better handles large numbers of features [19]. Although the same number of features were used for both algorithms, the hidden layer of the neural network added complexity and processing time; the `fmincg.m` algorithm was selected.

3.1 Data Retrieval

A summary of the gathering of data and the findings of the report by A. Pieringer et. al [1] is given in section 2.4.3. Both measured and simulated acceleration data was retrieved, and the simulated data was validated in this study. The data available for this thesis was thus both the acceleration data from both measured and simulated track passbys, as well as the resulting images created from using short-time Fourier transforms on the acceleration data. The given acceleration data was then processed using several methods. The preprocessing methods, presented and evaluated in Chapter 4, result in a varying number of features for each training set.

The measurements were made using a specialized noise measurement car, SMW, on a track section called "specially monitored track" [20]. The noise measurement car is equipped with a microphone mounted in a semi-anechoic chamber above the measurement bogie as well as accelerometers on the wheel-sets of the measurement bogie. The accelerometers measure vertical axle-box acceleration (ABA) on all four of the axle boxes at a sampling rate, $f_s = 20$ kHz. Only measurements which satisfied the target velocity, 80 km/h, within ± 5 km/h were included. An earlier measurement where the measurement car had a target velocity of 100 km/h was also available.

The investigation of the track found 8 squats and 5 insulated rail joints which could be used as measurement data. The naming of the squats and insulated rail joints with the prefix 'Sf' was adopted from the previous research. Furthermore, rail switches and measurements of portions of rail were used as measured input data. The portions of rail used as examples of healthy rail were visually inspected and deemed free of squats. For training a machine learning algorithm, many training examples are important to sufficiently distinguish different classes. As such, simulation data of ABAs was acquired using a wheel-rail interaction model. The simulated track contained several squats, as well as lengths of healthy rail. The squats had varying geometries with a maximum length of 30 mm and a maximum depth of 1.5 mm. The track parameters and the wheel and rail roughness varied for three different simulations to obtain a more robust set of training examples. The frequency resolution of the simulated acceleration data was about $f_s = 22.2$ kHz.

3.1.1 Geometric Information of the Squats

The variation of surface geometry along the track at the eight identified squats is shown in Figure 3.1 [1]. It can be seen that the squats vary in depth, the width of the deepest trough, and the overall length of the squat. From the squat geometry, the squats can be divided roughly into three groups. The large squats are identified as Sf8 and Sf20, the medium squats identified as Sf13 and Sf29, and the small squats identified are Sf10, Sf12, Sf27, and Sf28. Although the overall depth of the corrugation around Sf10 is quite deep, the notch at the center of the squat is shallow relative to the squats identified as large.

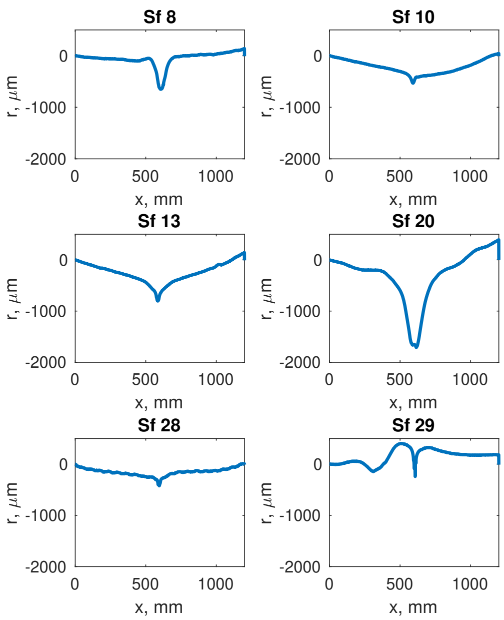


Figure 3.1: Geometry of squats along track length (x-axis) and depth (y-axis) for 8 squats

When investigating the acceleration data of the squats in the time domain, it was difficult to identify squat Sf27 sufficiently. As such, Sf27 was excluded from the data. Squat Sf28 and Sf29 were located at the same location along the track but on opposite rails. As Sf28 was much smaller than Sf29, in some measurements the acceleration of the axle-box (on the side of squat Sf28) due to Sf29 was more apparent on the acceleration data of the wheel passing Sf28. Although the algorithm should be able to identify squats in such a case, it is understandable that the hit rate may vary depending on the timing of the wheels and the direction of the train.

3.1.2 Collecting and Classifying Simulated Data

The available simulation data included acceleration data, location data along the simulated track, as well as location information about the simulated squats. As such, the data could be processed into examples of healthy rail and examples of squats. This was done systematically from one end of the track to the other. To verify that the code was succeeding in locating the squats, the short-time Fourier transform was plotted- if a squat was in the image a ring showed up on the plot. If the squat was localized sufficiently in the frame, the ring was green and the data was classified as a squat. If the squat was only slightly in the frame, the ring was blue and the data was classified accordingly. After visually verifying that the localization was performing well, the plot was turned off and the script ran through the whole data.

3.2 Variable Parameters

The machine learning algorithms inherently contain variable parameters that must be selected wisely to optimize the algorithm. For both logistic regression and neural networks, the number of iterations and the regularization parameter are variables. For the neural network, the size of the hidden layer must be chosen as well. The proportion of the data used as training and testing sets is also a variable. Furthermore, the method of preprocessing the data affects both what information is used for training the algorithm as well as the number of features used. Finally, the choice of data could be manipulated. For example, it could be unnecessary to train the algorithm with examples of insulated joints and switches as knowledge about the location of these track variations is already available and could practically be excluded from the measured data. Using too many examples of healthy rail may lead the algorithm to skew results that favor predicting a healthy rail.

By increasing the number of iterations, the machine learning algorithm can minimize the cost of the training set. If the algorithm is overall performing well, increasing the number of iterations should increase successful identification. However, if the data has large variations within one class, minimizing the error to the training data may lead to the algorithm finding difficulty in classifying new data i.e. overfitting. This may be an issue when the number of features is large. For example, there may be some variation in the exact location of the squat in a spectrogram image; training the algorithm too rigidly to one image may lead to the algorithm failing to identify the same squat shifted only slightly in the time domain.

The regularization parameter, λ, can be seen as a penalty added to the cost of the algorithm. This is useful if there are many features which all contribute slightly to the identification. This is true for the cases of using spectrogram images and scalogram images as input data as the data is two-dimensional. The regularization parameter can be varied heavily, but a rule of thumb used was that the parameter should be the same order of magnitude or smaller than the input data.

The size of the hidden layer, used in the neural network, increases the complexity and flexibility of the algorithm. Adding a feature to the hidden layer results in one more node which connects each feature of the input layer to each feature in the classification. However, this also increases the time needed for training the algorithm, as the complexity increases exponentially. Furthermore, if the algorithm is already overfitting, increasing the hidden layer size may not improve the performance of the algorithm.

3.3 Classification

The measured and simulated data was divided into four different classifications: no squat, squat completely in the frame, squat slightly in the frame and insulated rail joint. These classifications were altered for different tests of the machine learning algorithm; the effect of different classification methods on the performance of the algorithm was investigated. The squat data-sets were always given the observation value of $y = 1$. The different observation values assigned to each track behavior are given in Table 3.1.

Table 3.1: Observation values of different observed track behaviours for the first test of training the machine learning algorithm

	No Squat	Squat	Squat slightly in frame	Joint Isolation
Test 1	$y = 0$	$y = 1$	$y = 0$	$y = 0$
Test 2	$y = 0$	$y = 1$	$y = 0$	$y = 2$
Test 3	$y = 0$	$y = 1$	$y = 2$	$y = 3$
Test 4	$y = 0$	$y = 1$	$y = 2$	$y = 0$

3.4 Preparing Data for Machine Learning

When the data had been preprocessed the data from simulations and measurements were combined into a single `.mat`-file. The matrix was ordered such that the measured data preceded the simulated data, and was ordered by group, i.e. squats were first, followed by insulated rail joints, switches, and any rail.

The combined data was shuffled before being used as data for the machine learning algorithm. Furthermore, to test how the algorithm analyzed new data, one squat or insulated rail joint was left out of the training data. To do so, a function `leaveOneOut.m` was created. First, the squat or insulated rail joint which was to be left out was selected. Furthermore, the classification setup according to Table 3.1 was chosen. The datasets and their corresponding observation value were shuffled. Finally, the function divided the shuffled data so that a portion of the data was used as training data and the rest of the data was used as test data. The most common division of training and testing data was 90% and 10% respectively. As squat Sf27 was removed from the data, 7 squats and 5 insulated joints resulted in 12 training- and test runs for each preprocessing and classification method. Finally, two test sequences were tested; when training the algorithm for these two sequences, no measured data was left out.

3.5 Training and Validating Data

Once the algorithm design has been chosen and developed, the algorithm is tested and validated by using a portion of the prepared data which was not used to train the machine learning algorithm. Important results in this step are the successful classification of squats, undamaged rail, and insulated joints, as well as the percentage of false alarms (classification any other examples as squats). As training and updating of the algorithm can be done "offline" the priority of the results is set such that correct classification is the most important result, and the speed of training the algorithm is secondary.

3.5.1 Test Run on Measured Data

To investigate and present how the algorithm could work in practice, axle-box-acceleration data for two 250 m lengths of the measured rail was used as test sequences. The acceleration data was taken from the ABA of wheel number 2 for the first passby in the forward direction. The acceleration data was preprocessed in the same way as the training data for each attempt. After training the algorithm, the machine learning algorithm was used on the test sequence. As can be seen in the acceleration data in Figure 3.2, the first test sequence covered a rougher portion of the track. The first test sequence also passed squat Sf20 and insulated joint Sf21, at approximately distances 200 m and 220 m. The second test sequence covered a smoother portion of the track but passed by squat Sf13 after a distance 180 m.

As can be seen from the acceleration data in 3.2, a squat cannot be identified simply from acceleration; the overall acceleration on test sequence 1 is almost always larger than the acceleration at Sf13. Relative to the local acceleration, however, both squats and insulated joints lead to a large increase in acceleration.

Figure 3.3 shows the scalogram image of two random portions of test sequence 1 and 2. In the wavelet data, it can be seen that around 450 Hz the system is damped; this range has low levels even on the rougher test sequence 1. Both test sequences show higher levels at the lower frequency range around 200 Hz, even for smoother track.

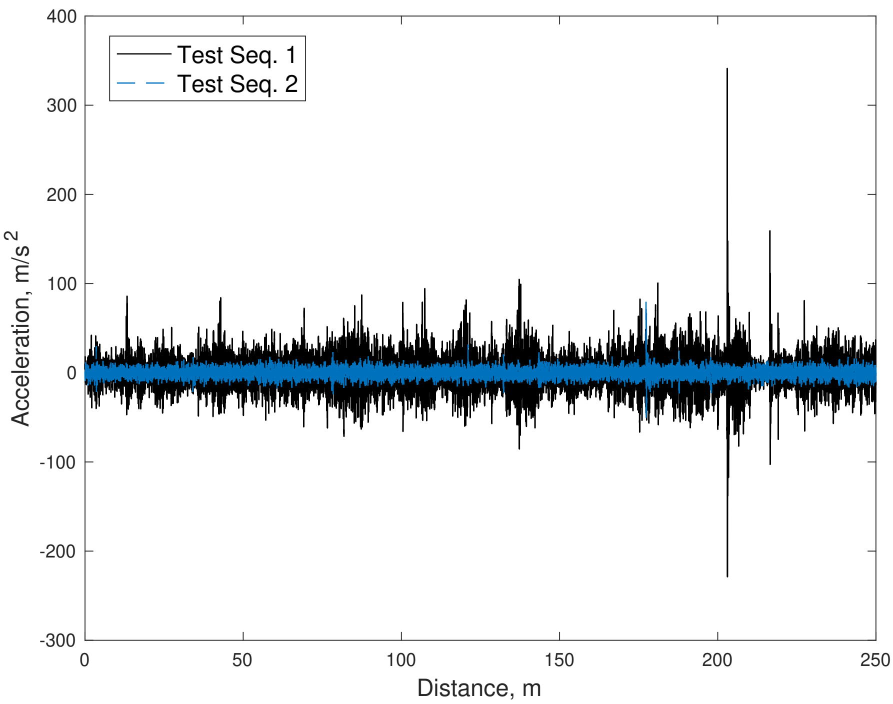
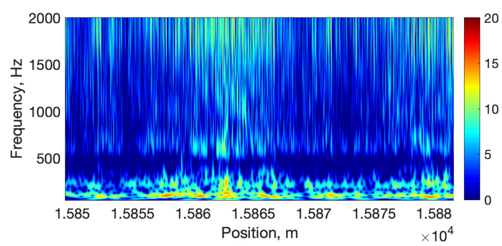
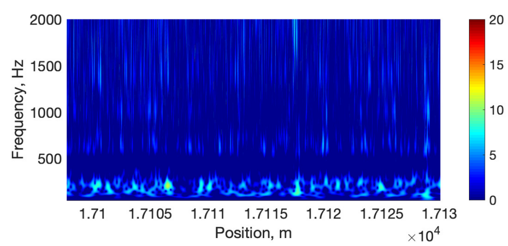


Figure 3.2: Acceleration measurements for two test sequences along the measured track



(a) test sequence 1



(b) test sequence 2

Figure 3.3: Scalogram image of random segments of the two test sequences using the Morse wavelet and a spatial resolution of 1.6 cm

4

Preprocessing and Feature Identification

Optimizing the machine learning algorithm was approached as an iterative process, where either change was made to the controlled variables of the algorithm, or to how the data was preprocessed. By processing the data in several different ways, key features relative to squats of the acceleration data were found for the different cases. As one of the problem statements was to optimize for clearer differentiation between squats and insulated rail joints, these two phenomenon were observed critically to identify differences in time and frequency domain.

Another interesting observation was whether a squat could be visually identified. For example, the roughness at a squat may lead to oscillation at a specific frequency related to the roughness of the surface defect. Although this would not feasibly replace the time saved by automatic handling of large amounts of acceleration data, it could lead to a secondary evaluation of the locations at which the algorithm found a squat.

In the previous study by Pieringer et. al [1], simulations were validated by comparing the acceleration data of a squat with similar geometry as a measured squat. It was found that the simulation matched the magnitude and width of the first acceleration peak. However, the simulation was too highly damped, leading to a smoother late response. This may impact the ability to use data in the time domain as input data. Furthermore, some of the peaks in the frequency domain were shifted. The simulated squats were found to be sufficiently similar to the measured data when comparing time-frequency and STFT; a short investigation of the similarity of the wavelet transform for measured and simulated squats is presented in the following section.

4.1 Features in the Time- and Frequency Domains

Vertical acceleration data from three axle-boxes in the measurement bogie were obtained. Along with knowledge of the location of several squats and other track anomalies, the axle-box acceleration at surface defects could be localized in the data. The measured axle-box acceleration was assessed in both time- and frequency domain. The accelerations at a squat showed peaks with large variation between

40 m/s^2 and 400 m/s^2. In the frequency domain, clear peaks at approximately 200 Hz, 600 Hz, 750 Hz, 1000 Hz, 1400 Hz, and 1500 Hz were identified.

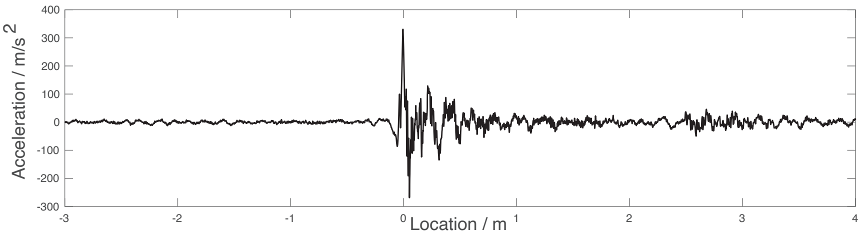


Figure 4.1: Acceleration measurements of squat Sf8 for a passby in forward direction. The x-axis is localized such that 0 m is the location of the squat.

The acceleration at the squat Sf8 is shown in Figure 4.1. As can be seen in Figure 4.1, a squat in the time domain can be signified by a dip in acceleration as the wheel starts to pass the squat due to the indentation of the rail surface. The squat impacts the wheel with a large force, which results in a peak in acceleration following the dip. The wheel continues to oscillate with some damping. For squat Sf8, the acceleration amplitude decays primarily the first four meters after passing the squat. However, it can be seen both before and after the squat that there is some amount of low amplitude oscillation at the axle-box. For squats located on rougher track, the decay was harder to distinguish in the time domain although the initial peak was clear.

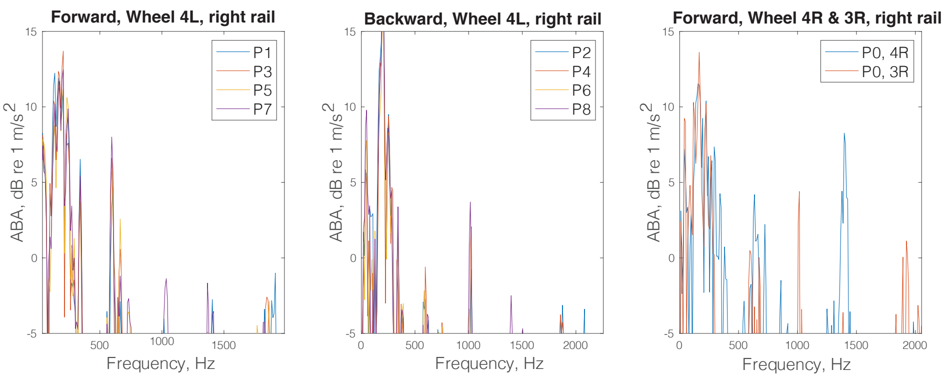


Figure 4.2: Passbys of squat Sf13 in frequency domain for 9 different passbys. For passby P4 the velocity of the train was below 80 km/h ± 5 km/h and disregarded. For passby P0, the velocity was around 97 km/h

In Figure 4.2, the frequency content of the ABA around squat Sf13 can be seen for 10 different measurements. Measurements labeled by passby 'P1', 'P3', 'P5', 'P7' and 'P0' pass the squat in the opposite direction compared to 'P2', 'P4', 'P6' and 'P8' which may explain the difference in the frequency composition. Furthermore the train was turned around for measurement 'P0'. In general, there is a large similarity between the measurements with regards to the frequency of the peaks; however, the level of the peaks varies. Furthermore, when the velocity of the train is increased to almost 100 km/h, a peak around 720 Hz is found as well as the peak around 600 Hz. Using equation 2.1, it can be seen that velocity of 100 km/h, or 27.8 m/s, and a frequency of 720 Hz relates to a wavelength $\lambda = 38.6$ mm. A velocity of 80 km/h, or 22.2 m/s, and a frequency of 600 Hz relates to a wavelength of 37 mm. This similarity in wavelength corresponds well to a hypothesis that short-pitch corrugation around the squat is impacting the measured ABA.

The frequency composition seems to vary somewhat depending on the direction of impact on the squat, as can be seen by comparing the two left plots in Figure 4.2. This is reasonable, as squats are not perfectly symmetric. The same peaks can be distinguished regardless of the direction of the train; the spectral balance of the peaks varies.

4.1.1 Spectrogram Images

The spectrogram images produced used the same method as the research by Pieringer, A. et. al. However, rather than centering the squat in the image, the time axis was adjusted so that one-quarter of the time axis was before the squat and three quarters was after the squat. This was chosen as the acceleration of the wheel after passing the squat contained more information than before passing the squat.

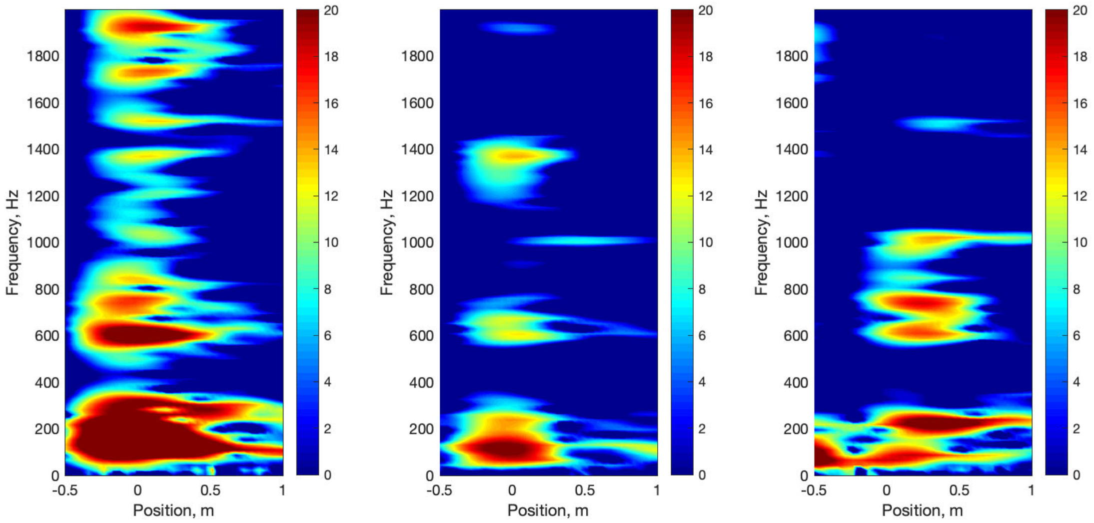


Figure 4.3: Spectrogram image of measured ABA with f_s=22.2 kHz at three different squats, with a spatial resolution of 5 cm. From left, Sf8, Sf12 and Sf28

The spectrogram was found using a short-time Fourier transform, where the window size was 1024 samples, and a Hanning window was used. The transform was performed by moving the window 50 samples between each transform, resulting in a 95% overlap. After performing the STFT, the data was cut 7 samples before the squat and 21 samples after the squat, resulting in 29 samples in the time/location domain. In the frequency domain, the data was cut so that the first 93 samples were included, corresponding to a frequency range between 0 Hz and 2 kHz.

Figure 4.3 shows spectrogram images from axle box acceleration measurements at three squats. In Figure 4.4 two insulated rail joints and one switch is represented

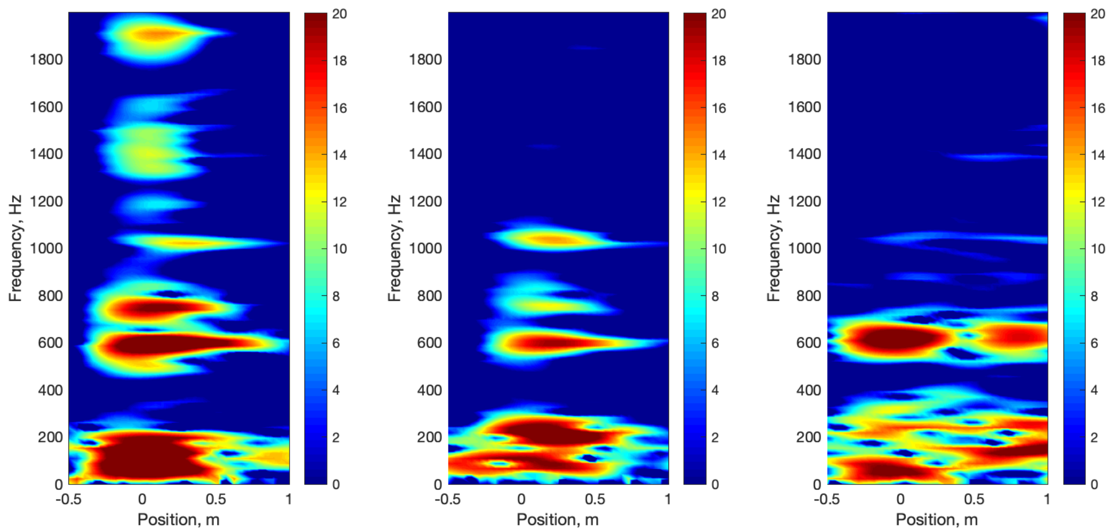


Figure 4.4: Spectrogram image of measured ABA with f_s=22.2 kHz at two different insulated rail joints, Sf6 and Sf15 (left and middle) as well as at a switch Sf2 (right), with a spatial resolution of 5 cm

by their spectrogram images. The squats chosen as representative images are squats Sf8 and Sf12 as Sf8 was geometrically considered a large squat, and Sf12 was geometrically considered to be a small squat. Squat Sf28 is located on the opposite rail of a larger squat (Sf29) and is therefore also of interest. Common for the squats is a broadband excitation at the location of the squat and a variation in damping depending on frequency. The more resonant regions are around 200 Hz, 600 Hz, 1000 Hz and around 1400-1500 Hz. In the case of Sf12, these are also the frequencies that are more heavily excited as the wheel passes the squat.

In the left and middle plots of Figure 4.4, the spectrogram image of the ABA when passing the insulated rail joints Sf6 and Sf15 are shown. The spectrogram images of the insulated joints varies; this is likely due to the condition of the insulated rail joint. Repeated loading at the joint may lead to one or both of the rails dipping, which will lead to acceleration due to the step height and angle of the rail. Sf6 is chosen as a representative insulated rail joint as the effect of the isolation deterioration seems quite clear. However, it is worth noting that not all insulated rail joints have a similar appearance; a milder case is shown in the middle plot in Figure 4.4. From the spectrogram images of the insulated rail joints, it can be seen that the insulated joint has a similar frequency composition compared to squats. This could be related to the geometrical similarity; the dip angle and the step height of the rail joint may be similar to the length and depth of a typical squat. Another hypothesis is that the excitation is due to the resonance of the coupled system between axle-box and rail.

In the case of the switch (right, Fig 4.4), excitation occurs in the low-frequency range between 50 Hz to 400 Hz as well as around 600 Hz. The 600 Hz region has a slower decay. As all three types of surface anomalies result in resonant excitation around

600 Hz, a hypothesis is that 600 Hz resonance is a characteristic of the wheel-rail interface rather than due to geometrical or mechanical characteristics of the different rail surfaces.

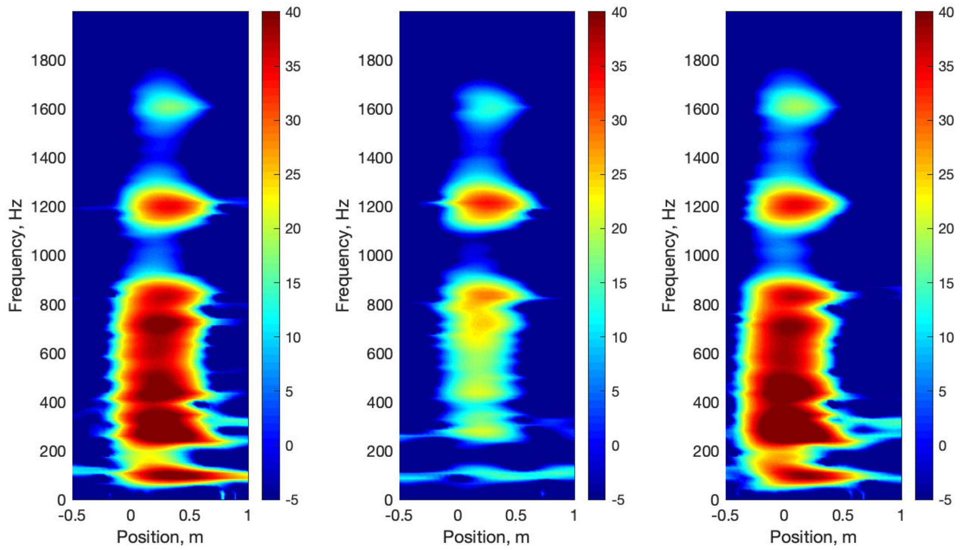


Figure 4.5: Three examples of the short-time Fourier transform on simulated ABA at three squats with f_s=22.2 kHz, with a spatial resolution of 5 cm

In Figure 4.5, three examples of simulated squats are represented by an STFT spectrogram. Between approximately 200 Hz and 1 kHz, the excitation is more broadband compared to the measured examples shown in Figure 4.3. At 1.2 kHz there is a clear peak for all three squats, this could be compared to the peak at 1.4 kHz for the measured squat Sf8 shown to the left in Figure 4.3.

4.1.2 CWT Scalogram Images

The continuous wavelet transform, CWT, was used to create scalograms of the training data in the time-frequency domain. Similar to the STFT, the number of input features increases quadrattically when increasing resolution or range in the time- or scale (frequency) domain. The CWT was performed by using the built-in Matlab function `[cwt f] = wt(filterbank, inputdata)` and the filterbank was determined using the built-in function `fb = cwtfilterbank()` with several input parameters. The wavelet type used was the 'Morse' wavelet, the number of voices per octave was set to 10, and the frequency range was set to between 50 Hz and 2000 Hz. The lower- and upper-frequency limit was chosen to confine to limitations of the model used for creating the simulated acceleration data [21]. The result was a wavelet transform over 54 different scales.

The choice of wavelet type was based on a visual evaluation of three wavelet types included in the function `fb = cwtfilterbank()`. Furthermore, in an earlier study [3], the wavelet type 'Morlet' was used. To broaden the scope of research on the topic, it would be of interest to use a different wavelet type. The three wavelet types included in `fb = cwtfilterbank()` were Morse, Morlet (specified as 'amor') and bump. An example of a scalogram image at a squat produced through wavelet transforms of acceleration data using these three filter banks can be seen in Figure 4.6.

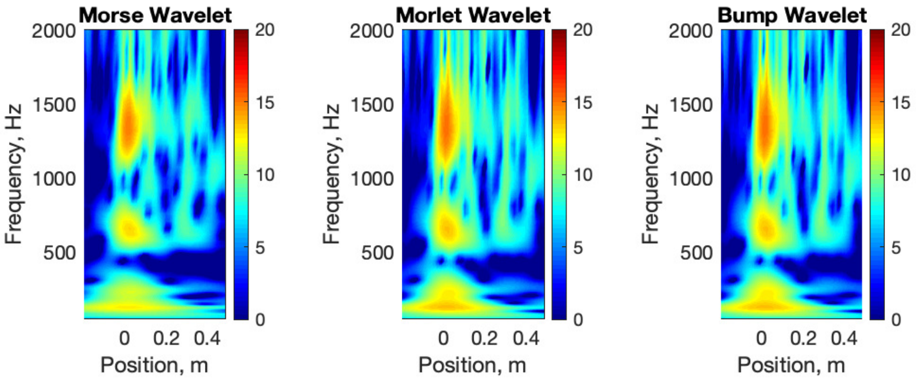


Figure 4.6: Scalogram images of the ABA at squat Sf12 for wheel 3 with the train in forward direction using three different filter banks - Morse, Morlet and bump

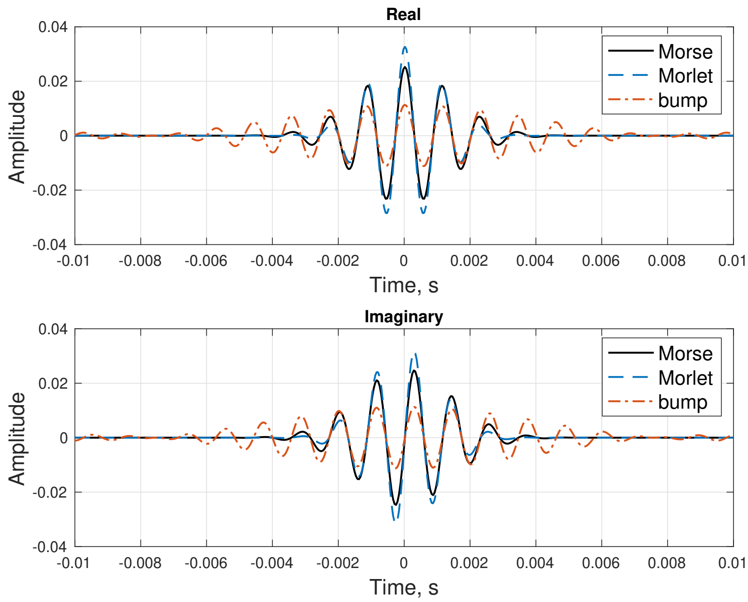


Figure 4.7: Real and Imaginary wavelets of Morse, Morlet and bump wavelet types

The different wavelet families are shown in the time domain in Figure 4.7 for scale 13. Using the given input parameters, scale 13 has a center frequency of 871 Hz. The Morlet wavelet has the highest peak around time zero, with a quick decay. The bump wavelet instead shows a slower decay and a smaller peak at the origin. The Morse wavelet is quite similar compared to the Morlet wavelet, but has a smaller peak at the origin. The width of the decay can be related to the resolution in the frequency domain, as shown in Figure 4.8. The bump wavelet has a faster decay in the frequency domain and is almost discontinuous at the edges. The decay time correlates to the width of the frequency response for all three wavelet types. It is

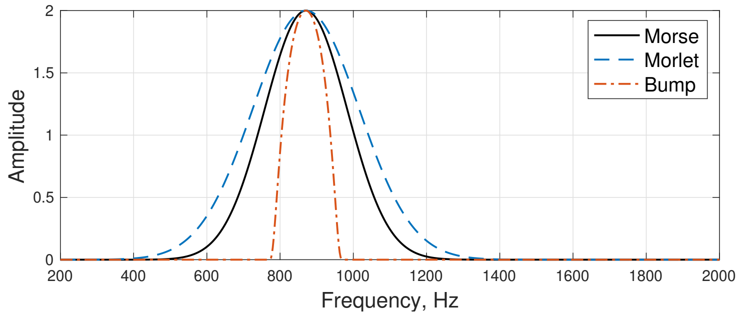


Figure 4.8: Frequency response for filter of wavelet 13 of Morse, Morlet and bump filter banks

worth noting that several parameters can be varied within each wavelet family; the default values were chosen for all three.

Due to the compromise between frequency and time resolution, the Morse wavelet was chosen. Of the three investigated wavelet types, the Morse wavelet had a sharper frequency resolution compared to the Morlet wavelet and a faster decay in the time domain compared to the bump wavelet.

The segmentation of the scalogram images was chosen such that there was a spatial resolution of 1.6 cm. The length of the scalogram in the time domain was 44 samples, leading to the image beginning roughly 0.2 m before a squat and continuing 0.5 m after a squat. The spatial resolution is higher compared to that of the short-time Fourier transform. Instead, the total number of features due to the spatial and frequency resolutions was kept in the same order. The scalogram images, as a matrix of 54 by 44 samples resulted in 2376 features per example compared to 2697 features for the spectrogram.

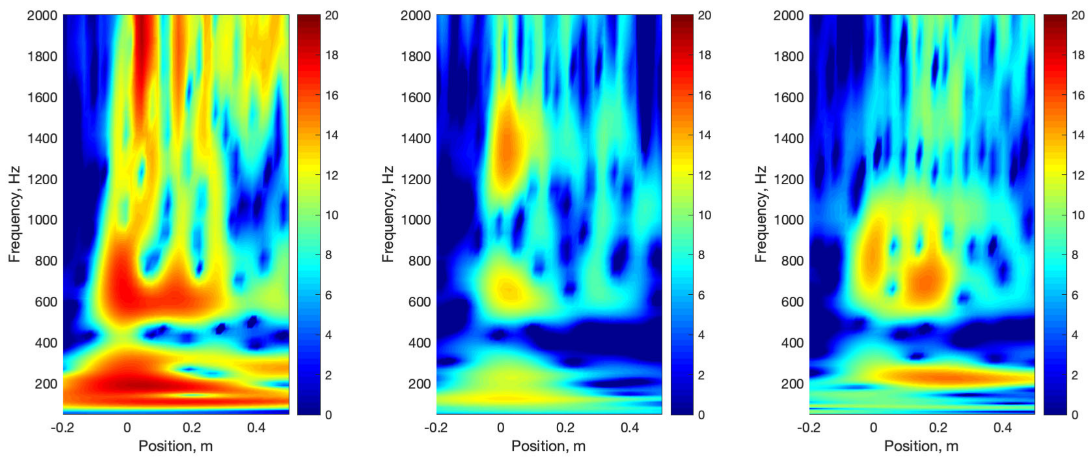


Figure 4.9: Wavelet transform of measured ABA with f_s=22.2 kHz at three different squats. From left, Sf8, Sf12 and Sf28

As can be seen in Figure 4.9, the squat can be identified by a broadband excitation at the location of the squat. The same three squats are used for visual comparison

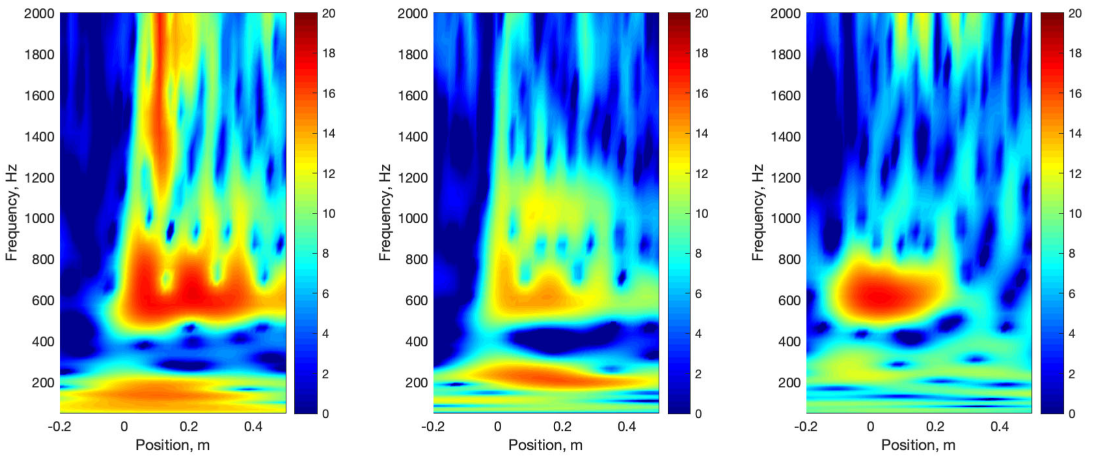


Figure 4.10: Wavelet transform of measured ABA with f_s=22.2 kHz at two different isolated joints, Sf6 and Sf15 (left and middle) as well as at switch Sf2 (right)

as in the case of the short-time Fourier transform. The main visual difference between the STFT and the wavelet scalograms is the apparent frequency modulation above 600 Hz. This modulation is not found using the short-time Fourier transform. Broadband, discrete peaks in amplitude could correspond to bumpiness on the rail resulting in an impact on the ABA.

The wavelet scalogram of two insulated rail joints and a switch is given in Figure 4.10. Visually, the joint is similar to squat Sf8; the difference is a slower decay time. However, this varies between squats and between different joints - a clear conclusion cannot be made from these plots alone. In the right plot of Figure 4.10, it can once again be seen that at switches, the excitement of the wheel/rail interface occurs around 600 Hz. Otherwise, there is little visual similarity between the switches and squats.

In Figure 4.11, three examples of simulated squats are shown by their scalogram representations. The broadband excitation and the slower decay rate for lower frequencies is similar to that of the measured ABA. However, a higher amplitude is found around 800 Hz rather than 600 Hz. The simulated track parameters of the three examples in Figure 4.11 were designed to correspond to the average measured track receptance at the measurement site [1].

Compared to the short-time Fourier transform, the wavelet scalogram images of simulated squats do not show a clear peak at the high frequency, 1.2 kHz. This is likely due to the frequency resolution of the wavelet transform; the resolution in the frequency domain is lower for higher frequencies. On the other hand, the wavelet transform has a higher time resolution, which illustrates the modulation of the high-frequency components in the measured case.

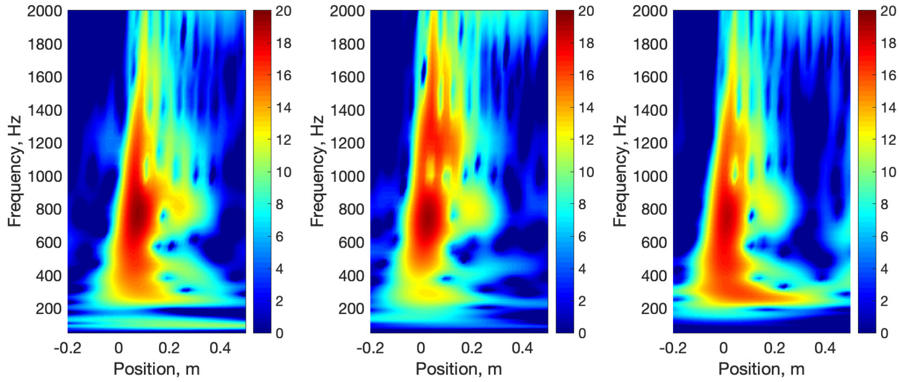


Figure 4.11: Three examples of wavelet transform of the simulated ABA at three squats with f_s= 22.2 kHz, downsampled by a factor 16 in the time domain

4.2 Time Averaged Wavelet Power

In an attempt to minimize the number of features used as input for the machine learning algorithm, the possibility of extracting important features of the wavelet transform was investigated. One method was to average each wavelet in the time domain to achieve a time-average for each of the wavelet scales. The data is not resampled before averaging over time. The number of scales was increased in the function `fb = cwtfilterbank()` to 24 voices per octave, resulting in 128 frequency scales instead of 54. The increase in frequency resolution was justified by the overall decrease in the number of features by a factor 20.

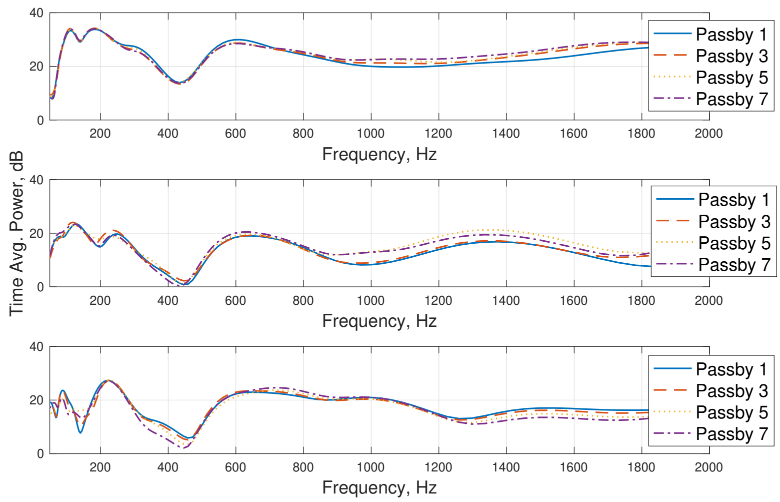


Figure 4.12: Time averaged wavelet power of measured ABA at three different squats. From top Sf8, Sf12 and Sf28

When extracting simulated data, the time-averaged power was found for every 16$^{\text{th}}$ sample to maintain the same distance between each training example as the second method of wavelet scalograms of 1.6 cm.

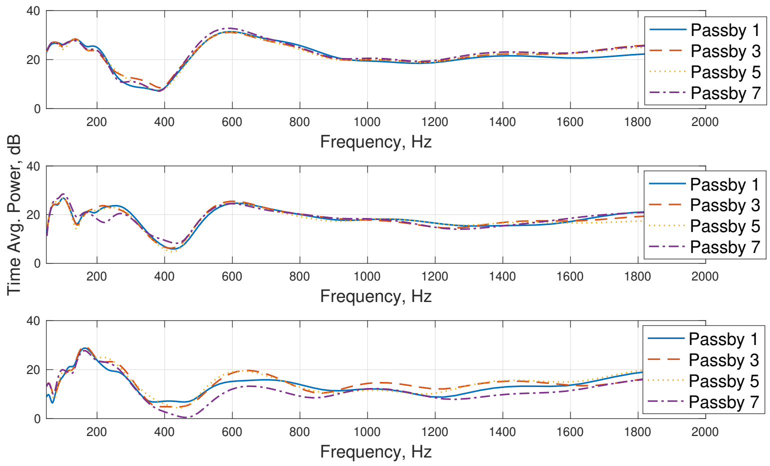


Figure 4.13: Time averaged wavelet power of measured ABA at two different isolated joints, Sf6 and Sf15 (top and middle) as well as at switch Sf2 (bottom)

By averaging over time, the importance of the two peaks of the squats at 600 Hz and around 200 Hz, as shown in Figure 4.12 became apparent. The high level at higher frequencies is also clear. In Figure 4.13, both peaks are clear for the insulated rail joints as well. For the switch given in the example, the 600 Hz peak is not as clear, but overall the time-averaged wavelet power is still similar.

4.3 Scale Averaged Wavelet Power

From Figure 4.12, it can be seen that squats generally show a peak in acceleration in the frequency regions around 200 Hz and 600 Hz. However, the low-frequency component seems to be common even for healthy rail, perhaps that it is an easily excited eigenfrequency of the wheel-rail system. As such, the time variation of the acceleration at the scales between 500 Hz and 707 Hz is investigated through scale average wavelet power.

In Figure 4.14, it can be seen that most of the energy at these scales is centered around the first 0.2 meters after passing a squat. However, the increase in energy is seen just before the wheel passes the squat. This is a general observation for the squats. However, the rate of decay after passing the squat varies for some of the squats. In identifying squats through scale averaged wavelet power, how this rate of decay affects the machine learning algorithm will likely be important to the overall success of the algorithm.

Furthermore, correctly identifying the start of the squat for the training data is imperative for success in the scale average wavelet power. As can be seen for squat Sf8 in Figure 4.14, the shape of the curve is very similar for all passbys. However, the position at which the power starts to increase is shifted up to 0.1 m between different measurements. This improper alignment increases the variation between the examples. However, if only examples of perfectly aligned squats are given to the

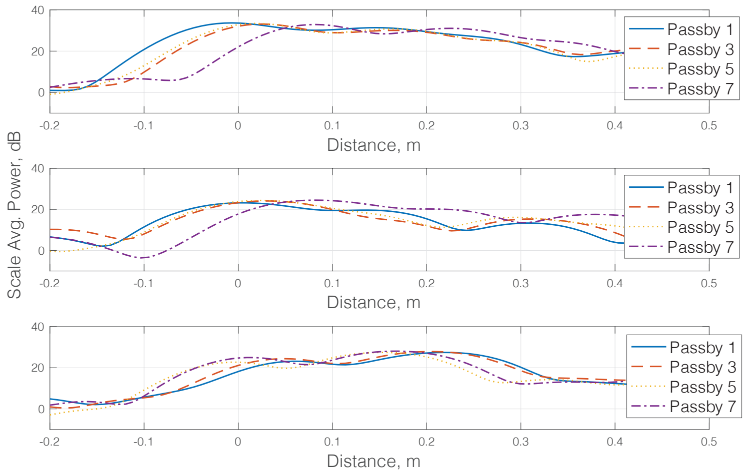


Figure 4.14: Scale averaged wavelet power of the scales centered between 500 Hz and 707 Hz, at three different squats. From top Sf8, Sf12 and Sf28

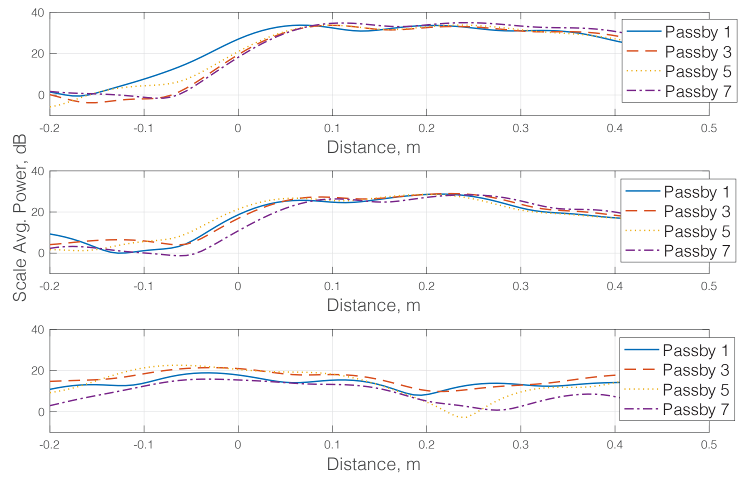
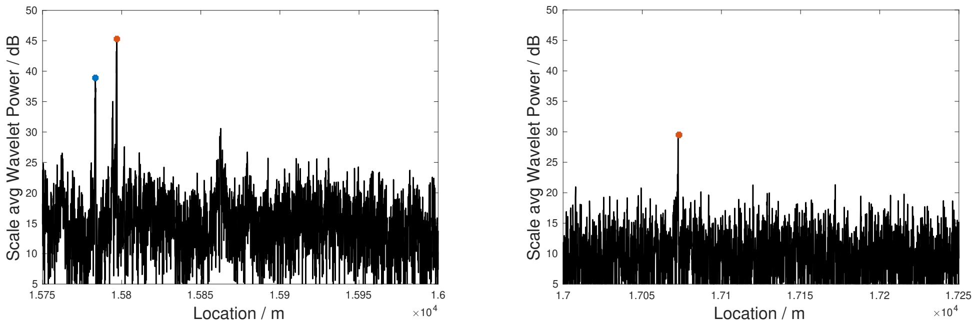


Figure 4.15: Scale averaged wavelet power of the scales centered between 500 Hz and 707 Hz, at two insulated rail joints Sf6 (top) and Sf15 (middle) as well as at a switch Sf2 (bottom)

training data, the implementation of the measurement system would require a very high time resolution between each SAWP.

It can be seen in both Figures 4.14 and 4.15 that the acceleration level is consistently high at least the first half meter after passing each surface anomaly. In the case of Sf12 (middle of Fig 4.14) and a switch (bottom of Fig 4.15), the level varies quite significantly between different passbys.



(a) test sequence 1 **(b)** test sequence 2

Figure 4.16: Scale average wavelet power in dB for wavelet scales with center frequencies 500 Hz through 707 Hz for the two test sequences. The locations marked with a red star are squats and the location marked with a blue star is an insulated rail joint.

By observing the scale averaged wavelet power over longer periods, the changes in level can more clearly be observed. This was done for test sequences 1 and 2 (Figure 4.16). However, this method was not investigated further as this type of data was not of use for training the logistic regression or neural network algorithms.

4.4 Conclusions for Optimization

Observing the acceleration data at a squat, it is found that there is some damping which results in the acceleration decaying oscillation for a few meters. As such, the data is cut so that the localization of the squat is the sample after one-quarter of the segment; a larger portion of the data consists of acceleration data after the location of the squat.

Although the time-average power shows clear peaks for squat measurements around 100-250 Hz as well as around 600 Hz, this is also true for insulated joints and switches. As such, it cannot be expected that the method should increase successful discrimination between the two.

The success of the scale average power depends on the patterns in the time domain. As was found by the previous study on the same data [1], the simulated data was generally more highly damped than the measured data. As such, it can be expected that the scale average power may fail in finding commonality between measured and simulated squats, leading to difficulty in training the algorithm. Furthermore, from the figures in section 4.3, it is difficult to visually distinguish a squat from different types of track data.

The long-time observation of the scale average wavelet power shown in Figure 4.16 shows promise in identifying squats from their wavelet power between 500 Hz and 707 Hz. Although this method minimizes the complexity of the detection method, Figure 4.14 shows that both squats Sf12 and Sf28 have moderately low levels compared to the rough track of test sequence 1.

Seemingly there are resonances of the track or the wheel or the interaction of the two at 600 Hz as a peak is clear in almost all cases of increased force between the rail and track. The theoretical frequency response of the wheel or track does not signify any clear correlation to the measured acceleration. Furthermore, the system is coupled, so investigating only the track or only the wheel is a simplification. No clear inferences regarding the resonances of the system can be made.

To conclude, although the squats can visually be recognized both from spectrogram and scalogram images, there is no clear advantage of either method found. Furthermore, neither averaging method seems to more clearly distinguish squats. Finally, it can be concluded that distinguishing deteriorating insulated rail joints from squats will be cumbersome with any of the chosen preprocessing methods.

5
Optimization of Machine Learning Algorithm

Designing a machine learning algorithm to perform well on a specific set of data requires tuning a large number of parameters. Some of these are inherent to the choice of the algorithm, such as regularization for logistic regression and neural networks, or the hidden layer size for the neural network. Furthermore, choosing how to label data, which data to include, and the number of features in the data set relates to the input data itself. In optimizing the machine learning algorithm, the different parameters were adjusted and evaluated iteratively. The different possible variations of the algorithm explored are given in Table 5.1.

Focus on this thesis was to assess the results of different preprocessing methods. A few attempts were made using the spectrogram data with the location of the squat shifted, as explained in section 4.1.1, to determine if shifting the origin would significantly affect the results. Furthermore, by shifting the origin of the STFT, the results could more rigorously be compared to the results of the other preprocessing methods used in the thesis.

Table 5.1: Possible variations of machine learning algorithm

Variable	No. of Variations
Classification	4
Algorithm	2
Preprocessing Method	4
Exclusion of data	n/a
Regularization	n/a
Hidden layer size	n/a
Ratio healthy/squat rail	n/a
Ratio training/testing data	n/a
Number of Iterations	n/a

Although it is possible to evaluate the different possible variations systematically, a more organic approach was taken. The knowledge gained from applying a certain variation was applied to later optimization attempts. To illustrate, classifying according to tests 2, 3, and 4 according to Table 3.1 did not lead to better identification. As such, assessing the effect of classification had a lower priority in later optimization methods.

After some iterations, it was found that correctly distinguishing between insulated rail joints and squats was cumbersome. As knowledge of the location of both insulated rail joints and railway switches is known, data at these locations can easily be excluded from monitoring. Later attempts at optimizing the machine learning algorithm excluded joints and switches from the training and testing data. In these cases, only classification tests 1 and 4 according to Table 3.1 were relevant.

The relationship between the number of examples of healthy rail and squats was relevant to the success of the algorithm. Including a large percentage of healthy rail led to high training and test accuracy, but in many cases led to lower success in identifying squats using the leave-one-out validation. Furthermore, decreasing the number of healthy rail examples also generally increased the rate of false alarms for the test sequences.

After varying the architecture of the two machine learning algorithms, a final design was chosen. The final setup of the two algorithms is shown in Table 5.2. For consistency, the different data and classification systems were trained using this final design of the two algorithms. Unless otherwise stated, the variables in Table 5.2 were used. The focus of the results is therefore on the preprocessing methods and classification method rather than the architecture of the two machine learning algorithms.

The total number of examples used varied between different input data, due to the amount of simulated data used for each preprocessing method as well as the ratio between examples of healthy rail and squats. The number of training examples was between approximately 5000 and 20000 examples. The amount of measured data was kept consistent at 400 examples. This method is not recommended, as this was an un-monitored variable between the different preprocessing methods. However, the ratio between healthy rail and squats seemed more important for succesful algorithm training than the total number of examples, which is shown in the section.

Table 5.2: Design of variable parameters of the machine learning algorithms. The hidden layer size is only applicable to the neural network algorithm

Variable	Setup
Number of Iterations	200
Regularization, λ	0.5
Hidden layer size	20
Ratio training/testing data	90/10

5.1 Spectrogram as Input Data

The neural network algorithm was trained with the spectrogram data using the variable parameters given in Table 5.2. Furthermore, the percentage of the data which were examples of squats was approximately 4.1%. The algorithm showed moderate success in identifying the squats, yet most examples of insulated rail joints were falsely identified as squats. The results of the squat identification is given in Table 5.3. The trained algorithm was tested on the two test sequences. For test sequence one, the squat Sf20 was correctly identified along with 2 possible false alarms. Of the false alarms, one was the identified joint Sf21. For test sequence two, no false alarms were found. However, the squat Sf13 was not identified for this passby, which is reasonable as only 3 of the 7 examples of squat Sf13 were correctly identified.

Table 5.3: Results of neural network algorithm on squat data using spectrograms as input for test one according to variable parameters given in Table 5.2

	Sf8	Sf10	Sf12	Sf13	Sf20	Sf28	Sf29
Successful Identification	4	5	4	3	8	8	14
Total Examples	4	8	8	7	8	8	16

In the case of logistic regression, similar success was found for identifying squats. However, the number of false alarms increased significantly. The number of false alarms for test sequence one was up to 44, and the number of possible false alarms for test sequence two was 8. However, both of the squats were identified. The result is somewhat surprising, since the previous paper by A. Pieringer et. al [1] used the logistic regression with the descent algorithm `fminunc` as well and had fewer false alarms.

5.2 Scalogram as Input Data

5.2.1 Classification of Squats Only

Before deciding on the final machine learning architecture in Table 5.2, a first iteration of the test used classification according to test 1 in Table 3.1. The training data and testing data consisted of 70% and 30% of the total data respectively. The chosen variable parameters are given in Table 5.4.

Table 5.4: Variable parameters of first test of scalogram

Variable	Setup
Classification	1
Algorithm	Neural Network
Number of Iterations	400
Regularization	5
Hidden layer size	30
Percentage examples of squats	1.8%
Exclusion of data	none

The first test resulted in the perfect fitting of the training data (100%) and the correct classification of the testing data between 93-97% depending on which data were excluded. The correct identification of squats and insulated rail joints is given in Tables 5.5 and 5.6 respectively. For the test sequences, there were many false alarms.

Table 5.5: Results of neural network algorithm on squat data using scalograms as input for test one according to variable parameters given in Table 5.4

	Sf8	Sf10	Sf12	Sf13	Sf20	Sf28	Sf29
Successful Identification	4	7	8	7	8	3	11
Total Examples	4	8	8	7	8	8	16

Table 5.6: Results of machine learning algorithm on insulated rail joints data using scalograms as input for test one according to variable parameters given in Table 5.4

	Sf6	Sf15	Sf16	Sf18	Sf21
Successful Identification	2	7	4	6	8
Total Examples	4	8	8	8	8

A second test excluded examples of insulated rail joints and switches. The design of the variable parameters was once again set according to Table 5.2. The identification of squats for the second test are given in Table 5.7. Aside from squat Sf10, which was correctly identified for 75% of the examples, the algorithm correctly identified all examples of measured squats. However, the algorithm resulted in many false alarms when tested on the two test sequences.

Table 5.7: Results of neural network algorithm on squat data using scalograms as input with variable parameters according to Table 5.2 with examples of insulated rail joints and switches excluded

	Sf8	Sf10	Sf12	Sf13	Sf20	Sf28	Sf29
Successful Identification	4	6	8	7	8	8	16
Total Examples	4	8	8	7	8	8	16

The number of false alarms for the two test sequences was much higher for the scalograms compared to the short-time Fourier transform as well as compared to the results from the previous study [1]. The false alarms which occurred on the track where the ABA was less than 40 m/s^2 could be removed. This resulted in fewer false alarms on test sequence 2 but a large number of false alarms on the rougher track of test sequence 1 remained.

An idea to improve the algorithm further was to add the maximum acceleration observed during each segment as a feature in the input data. The results of the third attempt on the scalogram data are given in Table 5.8. Aside from squat Sf28, all examples of measured squats were correctly identified. For squat Sf28, half of the examples were correctly identified.

The algorithm was tested on the two test sequences. All data which was classified as a squat but had acceleration values below 40 m/s^2 were reclassified as a healthy

Table 5.8: Results of neural network algorithm on squat data using scalograms as input for test three with variable parameters according to Table 5.2 excluding images of insulated rail joints and switches and adding maximum acceleration as a feature

	Sf8	Sf10	Sf12	Sf13	Sf20	Sf28	Sf29
Successful Identification	4	8	8	7	8	4	16
Total Examples	4	8	8	7	8	8	16

rail. Running the algorithm on the first test sequence identified the squat Sf20 in the data. Furthermore, the joint Sf21 was identified as a squat and three other possible false alarms were identified. For the second test sequences, the squat Sf13 was correctly identified and there were no false alarms.

5.2.2 Other Classification Methods

The three later classification methods according to Table 3.1 were investigated for their success in training the neural network algorithm. For classification methods 2 and 3, the insulated rail joints and switches were included in the data. For classification method four, the joint and switch data were removed.

Classification method 2 resulted in the correct identification of insulated rail joint Sf21. Both squats were identified in the test sequence. However, test sequence 1 found 16 possible false alarms, and test sequence 2 found 5 possible false alarms. Classification method 3 led to the highest success of correctly classifying insulated joints, although neither method clearly distinguished squats from joints. Both classification methods 2 and 3 resulted in many false alarms; the false alarms in test sequence 2 were minimized by setting a minimum acceleration of 40 m/s^2. However, the false alarms for test sequence 1 remained.

Classification method 4, which excluded joints and switches, led to overall good identification of the squats. Only one example of squat Sf10 and two examples of squat Sf12 were falsely identified. Noticeably, the success rate of the testing data at 77%-82% was much lower than for the training data at 88%-92%. It can be inferred that the algorithm is poor at generalizing new data.

5.2.3 Logistic Regression Method with Scalogram as Input

The logistic regression algorithm was used to test the success of the scalograms as input data. The focus was set on classification method 1, i.e. classifying squats only. The algorithm performed better by excluding joints and switches from the data. By setting the requirement that only acceleration levels above 40 m/s^2 could be classified as squats led to zero false alarms for test sequence 2, but many seemingly random hits for test sequence 1. The results of identifying the squats are given in Table 5.9.

By adding the maximum acceleration found in each segment as a feature, the false alarms in test sequence 1 decreased significantly to four possible false alarms, of

Table 5.9: Results of logistic regression algorithm on squat data using scalograms as input for test one with variable parameters according to Table 5.2

	Sf8	Sf10	Sf12	Sf13	Sf20	Sf28	Sf29
Successful Identification	4	3	8	7	8	8	16
Total Examples	4	8	8	7	8	8	16

which one was an insulated rail joint. The false alarms were all within 7 meters from the insulated rail joint or the squat, except for one false alarm which was about 40 meters from the squat. There were no false alarms for test sequence 2. The results of the detection of measured squats are given in Table 5.10.

Table 5.10: Results of logistic regression algorithm on squat data using scalograms as input for test one with variable parameters according to Table 5.2 with maximum acceleration added as a feature of the input data

	Sf8	Sf10	Sf12	Sf13	Sf20	Sf28	Sf29
Successful Identification	4	7	8	4	8	4	15
Total Examples	4	8	8	7	8	8	16

Many of the methods found false alarms for the first test sequence. As the segment covered had high acceleration levels, these false alarms were not eliminated by setting a minimum acceleration requirement. A test was made by using the first 3000 test examples from test sequence 1 as training data of healthy rail. Although there was no guarantee that all these examples were of healthy rail, none of the identified squats, joints, or switches were included. The result of including these test examples significantly minimized the number of false alarms on the remainder of test sequence 1 while maintaining an equally high success rate of identifying measured squats. The original data contained track with varying roughness; however, this test indicates that more examples of rough track may improve training of the algorithm.

5.2.4 Discussion and Conclusions Regarding Scalogram-based Algorithm

Both the neural network and logistic regression algorithms were successful after adjusting the design of the algorithm. Furthermore, the algorithms performed better by adding a feature of the maximum observed acceleration to the input data as the number of false alarms decreased. The two squats which were not identified were generally either Sf10 or Sf28 which both were considered as small squats by visual investigation of the indentations.

Although the input data included a healthy track of varying roughness, the algorithm found fewer false alarms when adding some of the rough track from test sequence 1 to the training data. This could indicate that there is still room for improvement by adding more examples of different track parameters to the training data.

From the perspective of machine learning efficiency, an issue with the scalogram as input data is the number of features. The scalogram is two-dimensional, which leads to a squared increase in the number of features when increasing range or resolution in either the time or frequency domain. As there are not a lot of examples of squats or other track anomalies, a few errors may occur. If almost all data is examples of healthy rail, the algorithm may skew toward the healthy rail, by learning that the cost generally decreases by assuming all examples are of healthy rail. Another issue, if one is careful to not overload the training set with examples of healthy rail, is that there is too little data to properly fit.

The complexity of the algorithm compared to an unbalanced number of examples of each class may lead to skewing. Although the difference between the training data and testing data success was only a few percentage points at most, the percentage of the data which consisted of squat data was quite low, such that statistically, a success lower than 98.2% may consist of the algorithm constantly predicting healthy rail. The combination of validating measured squats to determine success in finding squats, as well as the two test sequences, which should not have many false alarms, was a sufficient method for assessing the skewing.

5.3 Scale Average Wavelet Power as Input Data

The scale average wavelet power as input showed little success for identifying squats and found many false alarms when using the test sequence. The results of using SAWP as input data with the neural network algorithm are given in Table 5.11

Table 5.11: Results of neural network algorithm on squat data using scale average wavelet power as input and classification test 1 according to Table 3.1

	Sf8	Sf10	Sf12	Sf13	Sf20	Sf28	Sf29
Successful Identification	3	4	7	6	0	2	0
Total Examples	4	8	8	7	8	8	16

By using the logistic regression algorithm, compared to the results in Table 5.11, the algorithm identified one more case of squat Sf10 and two more cases each of squats Sf20 and Sf28 and finally identified four more cases of squat Sf29. The same success was found by adding the maximum acceleration data as a feature. However, the number of false alarms remained high for all three attempts.

5.4 Time Average Wavelet Power as Input Data

The time-averaged wavelet power showed varying success in identifying squats, depending on the different variable parameters. The most common error was the inability to identify squats Sf10 and Sf28, which are two of the smaller squats. This error occurred consistently for the different methods using the neural network algorithm. The results of using time-averaged wavelet power on the logistic regression algorithm with parameters according to Table 5.2 is presented in Table 5.12.

Table 5.12: Results of the logistic regression algorithm on squat data using time-average wavelet power as input for classification test 1 according to Table 3.1

	Sf8	Sf10	Sf12	Sf13	Sf20	Sf28	Sf29
Successful Identification	3	0	2	2	8	2	16
Total Examples	4	8	8	7	8	8	16

Using time-averaged wavelet power to detect squats resulted in the trained algorithm finding many false alarms for the test sequence. The descent algorithm was monitored during training the different training examples. From the monitoring, it was found that the descent was either very slow or oscillating around a consistent level. The failure to descent indicates either that the method was finding a local minimum or that the algorithm was failing to find a pattern in the data. As the same effect was seen for many tests, the latter of the two seems more likely.

5.5 Comparison of Methods

5.5.1 Setup of the Machine Learning Algorithm

As can be seen in Table 5.1, the possible variations in the design of a machine learning algorithm are infinite. As was presented by Andrew Ng on his online machine learning course [19], there are some simple diagnostic tests to evaluate the success of the machine learning algorithm.

The number of features for all different preprocessing methods is high, the expected issue was that the algorithm would overfit to the data. This error is plausible if the success of the training data is much higher compared to the success of the test data. Some methods of mitigating the high variance are finding more training examples, lowering the number of features, increasing the regularization parameter λ, and stopping the training early. The first two of these methods are inherent to the input data itself, meaning that the possible methods of lowering the variance without altering the data were altering the regularization and lowering the number of iterations.

The regularization parameter chosen was 0.5. With a higher regularization, the variance was minimized for some of the different preprocessing methods, whereas for other input data the descent method did not perform as well. With a lower regularization parameter, it was found that the variance increased for most preprocessing methods. Overfitting due to training the algorithm for too many iterations was noticeable only for some of the preprocessing methods. However, it was found that after 200 iterations the successful identification of testing data and validation data did not increase noticeably for any of the preprocessing methods. As such, the number of iterations chosen was 200.

The size of the hidden layer was chosen for the neural network. The size was kept between the number of classifications and the size of the input layer. By increasing the hidden layer size, the flexibility of the algorithm increased, which may lead to better results. However, increasing the hidden layer size leads to longer computation time and may also lead to overfitting of the data. The choice of the hidden layer size was a compromise between these factors.

5.5.2 Classification of the Data

The different methods of classifying the data according to Table 3.1 were designed to test methods of distinguishing squats from insulated rail joints. However, distinguishing between the two was difficult. The insulated rail joints excite the axle boxes with very similar frequency and time patterns compared to squats. This could be related to the geometries of the squats and insulated rail joints. According to Thompson [5], the step height and dip angle of the insulated rail joint are defining factors of the generated wheel-rail noise. Another hypothesis is that the resonances of the wheel-rail system which effects the ABA are very strong. This hypothesis is supported by the ABA of the first test sequence, shown in Figure 3.3. Compared to the smoother second test sequence, the scalogram image of test sequence 1 shows excitation in the same frequency ranges as the squats and insulated rail joints.

Attempting to train the algorithm to differentiate between two very similar signals may be counterproductive; after the difficulty in differentiating squats from insulated rail joints was established, the insulated rail joints were removed from the data. The examples of switches were removed as well, as the switches can easily be removed from the data when implementing the algorithm in real-time.

5.5.3 Preprocessing Method

Both the time-averaged and scale-averaged wavelet power performed poorly for the measured squats and the test sequences. The results varied heavily depending on the setup of the machine learning algorithm. The conclusion can be made that the ABA of a squat is most properly identified by a simultaneous representation in the time and frequency domain.

Although the scale-averaged wavelet power performed poorly with the two machine learning algorithms, the Figure 4.16 highlights the possibility of using scale-averaged wavelet power for another method of automatic detection. This method was used in the paper by M. Molodova et. al [13]. This method was not investigated further, but due to the low complexity this method may be advantageous to investigate further.

The scalogram based algorithms did not perform better than the short-time Fourier transforms from the previous study by Pieringer et. al [1]. However, the algorithm performed much better than the STFT-based algorithms of this paper. Furthermore, the number of false alarms for the test sequences decreased.

5.6 Final Design of Machine Learning Algorithm

Success in identifying squats along with low levels of false alarms were determining in the final design of the machine learning algorithm. A lower complexity saves computation time and has less risk of overfitting. However, the highest overall success was found for a neural network algorithm.

The algorithm chosen for the final design was the neural network algorithm with parameters according to Table 5.2. The input data chosen were the scalogram levels along with the maximum acceleration found in the signal segment as an added feature. For each squat, the successful identification is therefore found in Table 5.8. For the medium to large squats Sf8, Sf13, Sf20, and Sf29, 100% successful identification was found. For the small squats, Sf10, Sf12, and Sf28, 87% successful identification were found due to the four misses of squat Sf28.

5.7 Discussion

The size of the squats was directly related to the success of identifying the squat. The only method which was not able to identify the large squats Sf8 and Sf20 was the scale-averaged wavelet power. However, it was expected that this data would perform poorly compared to the other input data due to the large variation in input data. This was established in section 4.3.

At the occasion of either surface irregularity, a broadband excitation occurs. However, some mid-frequency ranges are excited for an extended period, namely frequencies around 200 Hz and 600 Hz. Furthermore the ABA at 1.4 kHz has high levels for both squats and some of the examples of insulated rail joints. These could either be seen as related to the eigenfrequencies of the wheelset-track system, or related to the size of the irregularity combined with the speed of the train. The first assumption is highlighted by the similarity between rough track shown in Figure 3.3(a) and for squats. If the frequency response at a defect is mainly determined by the wheel-track mobility, differentiating between different surface defects would be difficult, even with more measurement data. The frequency response at switches indicates that the geometry of the surface irregularity is also relevant. A combination of both phenomenon is probable.

Many of the attempts resulted in many false alarms for test sequence one. Although the final design does not have this issue, it highlights the importance of including more rough track data and variations in track parameters in the training data. Furthermore, using two test sequences with significant difference in roughness was a suitable method for troubleshooting.

For the final design, all squats were identified for at least once. In general, this is a satisfying result, as the algorithm can identify even the smaller squats. Compared to previous research, the main benefit of the optimization is a lowered number of false alarms, making the system more practical for implementation.

5.7.1 Future Research

The study focused on developing and optimizing an algorithm for detecting squats measured by a measurement train on the German railway. As a single measurement car is used, there is the advantage that the coupling between the surface defect and the axle-box acceleration is consistent for each measurement compared to if many different cars are used. However, the advantage of being able to use many different wagons is that larger portions of the track may be measured and monitored with this method. Future research could include either training the algorithm to generalize regardless of wheel and car type or by processing the acceleration data to account for the different couplings.

Another simplification of the measurement data is the use of data only within a short range of velocities. The benefit of this is that the forces and frequencies of passing each squat are kept more consistent; however, it requires that the passing measurement car keeps this velocity. As could be seen in Figure 4.2, by increasing the velocity to 100 km/h (27.8 m/s) some peaks in the frequency domain became more pronounced and some of the peaks shifted upward in frequency. A method of handling this could be investigating how to account for the velocity in the frequency domain. However, many of the resonances seemed to be independent of the vehicle velocity.

There are a number of different machine learning algorithms available to use, among which the logistic regression algorithm and the single hidden layer neural network was used for this thesis. An area of future research is testing other algorithms. One algorithm, which is useful for image identification, is the deep convolutional neural network. As the feature identification presented in Chapter 4 visually assessed the data, image recognition algorithms may more closely relate to how squats are identified manually. An example of such research is the paper presented in section 2.4.2. Spectrogram and scalogram images could be used as input, rather than ultrasound as suggested in the paper, to detect the small squats earlier.

6

Conclusion

The final implementation of a neural network machine learning algorithm to identify squats from ABA was successful. The algorithm correctly identified all medium and large squats, as well as 87% of the examples of small measured squats. Furthermore, the number of possible false alarms for the test sequence was lowered compared to previous attempts, to a total of four possible false alarms. The algorithm was not able to differentiate between squats and insulated rail joints.

A lot of focus on optimizing the machine learning algorithm was set on differentiating between squats and insulated rail joints. However, there was little measurement data and no simulated data of the insulated rail joints. As such, little success was found in this area of optimization. In order to differentiate between the two, more data would be needed. Furthermore, the acceleration data from a squat and from an insulated rail joint had many similarities.

Oversaturating the training set with examples of healthy rail led to more false classification. However, if not enough examples of healthy rail were given, the two test sequences found a higher number of possible false alarms. For the final implementation, approximately 1.8% of the data were examples of measured or simulated squats.

For further research, different machine learning algorithms could be used. The squats were manually assessed through images of the spectrograms and scalograms as well as through the plots of the time- and scale-averaged wavelet power. As such, an algorithm that finds patterns in images rather than treating every sample as a feature may find higher success.

Finally, in order to implement the algorithm, the algorithm should be trained on many different types of track. Some variations would be in the railpad stiffness, the spacing of the sleepers, and to train the algorithm on ballastless tracks. To conclude, although the optimization attempt did not increase the successful identification of small squats, the success of using a different preprocessing method on the ABA data substantiates the hypothesis that the axle box acceleration data is a promising method to detecting small squats.

Bibliography

[1] A. Pieringer, M. Stangl, J. Rothhämel, and T. Tielkes, "Acoustic monitoring of rail faults in the German railway network," in *IWRN13*, Munich, Gothenburg, 2019, p. 8.

[2] Z. Li, "Squats on Railway Rails," in *Wheel-Rail Interface Handbook*, 2009.

[3] M. Molodova, Z. Li, A. Nunez, and R. Dollevoet, "Monitoring the railway infrastructure: Detection of surface defects using wavelets," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2013.

[4] A. Jamshidi, S. Faghih-Roohi, S. Hajizadeh, A. Núñez, R. Babuska, R. Dollevoet, Z. Li, and B. De Schutter, "A Big Data Analysis Approach for Rail Failure Risk Assessment," *Risk Analysis*, 2017.

[5] D. Thompson, *Railway Noise and Vibration.* Elsevier Science, 2009.

[6] "Rails Train Railway - Free photo on Pixabay," 2018. [Online]. Available: https://pixabay.com/photos/rails-train-railway-gleise-travel-3624853/

[7] R. Lewis and U. Olofsson, *Wheel-Rail Interface Handbook.* Woodhead Publishing Limited, 2009.

[8] K. H. Oostermeijer, "Review on short pitch rail corrugation studies," *Wear*, 2008.

[9] E. Hansen, *Fourier Transforms : Principles and Applications*, 1st ed. New York: John Wiley & Sons, Incorporated, 2014. [Online]. Available: https://ebookcentral.proquest.com/lib/chalmers/detail.action?docID=1810513

[10] L. Debnath and F. A. Shah, *Wavelet Transforms and Their Applications*, 2nd ed. Boston: Birkhäuser, 2015.

[11] M. G. Christensen, *Introduction to Audio Processing*, 2019.

[12] J. Bergh, F. Ekstedt, and M. Lindberg, *Wavelets.* Lund: Studentlitteratur, 1999.

[13] M. Molodova, "Detection of early squats by axle box acceleration," pp. 1–138, 2013. [Online]. Available: http://repository.tudelft.nl/islandora/object/uuid%3Aee403235-2ddf-4ce3-9f6d-e0775448f428

[14] D. L. Hartmann, “Section 9: Wavelets,” St Louis, 2016. [Online]. Available: https://atmos.washington.edu/~dennis/552_Notes_9.pdf

[15] M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi, “Wavelet Toolbox ™ User ’ s Guide,” *The MathWorks Inc.*, 2020. [Online]. Available: https://se.mathworks.com/help/pdf_doc/wavelet/wavelet_ug.pdf

[16] C. Torrence and G. P. Compo, “A Practical Guide to Wavelet Analysis,” *Bulletin of the American Meteorological Society*, 1998.

[17] Z. Su and L. Ye, *Identification of Damage Using Lamb waves.* Springer, London, 2009.

[18] G. Rebala, A. Ravi, and S. Churiwala, *An Introduction to Machine Learning*, 1st ed. Springer International publishing, 2019.

[19] A. Ng, “Machine Learning. Online course at Coursera Platform, Stanford University,” 2013. [Online]. Available: https://www.coursera.org/learn/machine-learning

[20] B. Asmussen, H. Onnich, R. Strube, L. M. Greven, S. Schröder, K. Jäger, and K. G. Degen, “Status and perspectives of the "Specially Monitored Track",” *Journal of Sound and Vibration*, vol. 293, pp. 1070–1077, 2006.

[21] A. Pieringer, “Time-domain modelling of high-frequency wheel/rail interaction,” Ph.D. dissertation, Chalmers University of Technology, 2011.

[22] A. Ng, “Machine Learning | Coursera,” 2013.

A

Preprocessing of Measured Acceleration Data

CHOOSE MEASUREMENT

```
% Choose squat or joint insulation
% Squats: Sf [8 10 12 13 20 27 28 29] -- exclude 27
% Isos: Sf [6 15 16 18 21]
% Anys: [1 2 3 4 5 6 7 8 9 10 11 12]
% Switch: Sf [1 2]
sf = 8;
% set 'squat', 'iso, 'any', or 'switch'
type = 'squat';

% Define direction
dir = 'FR'; % FR - forward, GFR - backward
% Define wheel
wh = 2; % 1:3L, 2:4L, 3:3R, 4:4R
```

LOAD MEASUREMENTS

```
% localize data
path_mea = 'MeasuredData';

if strcmp(dir,'FR')
    pens=[1,3,5,7];
elseif strcmp(dir,'GFR')
    pens=[2,4,6,8];
end

order_r = [28 20 13 8];
order_riso = [0 21 18 16 15 6];
order_l = [29 20 12 10];

pos_any =  [15790 17084 17020 16430 16700 15440 15300 ...
    15650 16000 17400 15605 15900]; % belong to all wh
```

```
pos_switch = [16129 16053]; % belong to wh 3 & 4

% pre-allocate
X=zeros(4,29*93);
xps=zeros(4,29);
fps=zeros(4,93);
vps=zeros(4,29);

for pp=1:4
    pendel=pens(pp); %pendel
    load([path_mea,'/vorbeifahrt',...
        num2str(pendel)],'acc','x','tt','v3','acc_label')

    if strcmp(type,'squat') || strcmp(type,'iso')
        if wh == 2
            load([path_mea,'/vorbeifahrt',...
                num2str(pendel)],'pos_squ_r','pos_iso_r')

            if strcmp(type,'iso')
                pos_sf = pos_iso_r(order_riso == sf);
            else
                pos_sf = pos_squ_r(order_r == sf);
            end

        else
            load([path_mea,'/vorbeifahrt',...
                num2str(pendel)],'pos_squ_l')

            pos_sf = pos_squ_l(order_l == sf);
        end
    elseif strcmp(type,'any')
        pos_sf = pos_any(sf);

    elseif strcmp(type,'switch')
        pos_sf = pos_switch(sf);
    end

    % manual locations of "any" and "switch"

    % Labels
    % tt - time data
    %%%%dt = tt(2) - tt(1);
    % x - spatial data
    % v - velocity data
    % acc - acceleration data
    % acc_label - label of acceleration data:
```

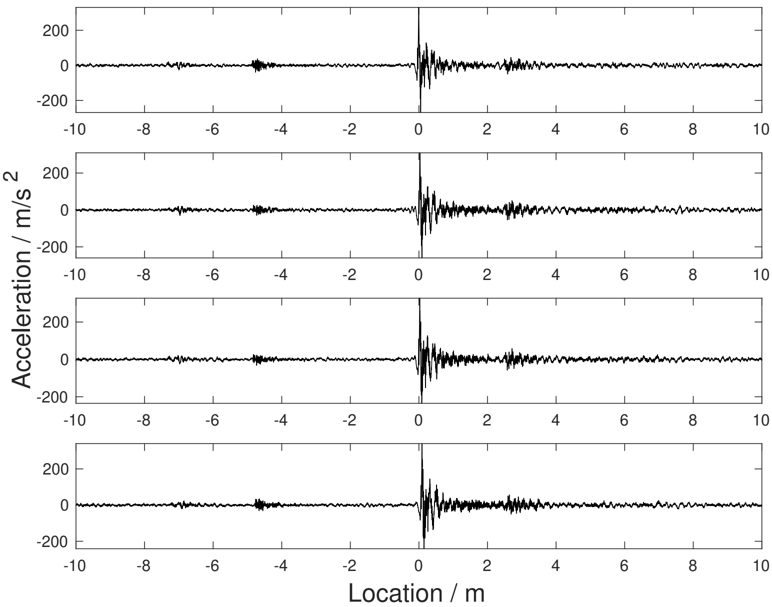
```
% index 1:3L, 2:4L, 3:3R, 4:4R


% Manually adjust squat position
% Currently ideal for Sf8
if strcmp(type,'squat')
    if sf == 8
        if strcmp(dir,'GFR') && wh == 2
            pos_sf = pos_sf + 2.3 - 0.70;
        end
        if strcmp(dir,'FR') && wh == 2
            pos_sf = pos_sf + 0.31;
        end
    elseif sf == 10
        if strcmp(dir,'FR') && wh==3
            pos_sf=pos_sf+0.1-2.35;
        end
        if strcmp(dir,'FR') && wh==4
            pos_sf=pos_sf+0.15;
        end
    elseif sf == 12
        if strcmp(dir,'FR') && wh==3
            pos_sf=pos_sf-3.065;
        end
        if strcmp(dir,'FR') && wh==4
            pos_sf=pos_sf-0.57;
        end
    elseif sf == 13
        if strcmp(dir,'FR') && wh==2
            pos_sf = pos_sf-1.34;
        end
        if strcmp(dir,'GFR') && wh==2
            pos_sf=pos_sf - .55;
        end
    elseif sf == 20
        if strcmp(dir,'GFR') && wh == 2
            %                 pos_sf = pos_sf ;%- 2.5;
        end
    elseif sf == 28
        if strcmp(dir,'FR') && wh == 2
            pos_sf = pos_sf - 3.15;
        end
        if strcmp(dir,'GFR') && wh == 2
            pos_sf = pos_sf - 3.55;
        end
    elseif sf == 29
```

```
            if strcmp(dir,'FR') && wh==4
                pos_sf=pos_sf-3.21;
            end
            if strcmp(dir,'FR') && wh==3
                pos_sf=pos_sf-5.71;
            end
            if strcmp(dir,'GFR') && wh==4
                pos_sf=pos_sf-3.65;
            end
            if strcmp(dir,'GFR') && wh==3
                pos_sf=pos_sf-6.1;
            end
        end
    elseif strcmp(type,'iso')
        if sf == 6
            if strcmp(dir,'FR') && wh==2
                pos_sf=pos_sf+.58;
            end
        elseif sf == 15
            if strcmp(dir,'GFR')
                pos_sf=pos_sf-0.53;

            end
            if strcmp(dir,'FR') && wh==2
                pos_sf=pos_sf-0.65;
            end
        elseif sf == 16
            if strcmp(dir,'GFR')
                pos_sf=pos_sf-0.15;
                if pp == 4
                    pos_sf = pos_sf+2.7;
                end
            end
            if strcmp(dir,'FR') && wh==2
                pos_sf=pos_sf-.17;
            end
        elseif sf == 18
            pos_sf = pos_sf + 15.7;
            if strcmp(dir,'GFR')
                pos_sf = pos_sf - 0.15;
                if pp == 4
                    pos_sf = pos_sf - 1.5;
                end
            end
        elseif sf == 21
            if strcmp(dir,'GFR')&& wh==2
```

```
            pos_sf=pos_sf - 0.55;
            if pp == 4
                pos_sf = pos_sf + 15.55;
            end
        end
        if strcmp(dir,'FR') && wh==2
            pos_sf=pos_sf - 0.3;
        end
    end
elseif strcmp(type,'switch')
    if sf == 1
        if strcmp(dir,'GFR') && wh==4
            pos_sf=pos_sf+2.9;
        end
        if strcmp(dir,'FR') && wh==4
            pos_sf=pos_sf+2.825;
        end
        if strcmp(dir,'FR') && wh==3
            pos_sf=pos_sf+0.3;
        end
        if strcmp(dir,'GFR') && wh==3
            pos_sf=pos_sf+0.4;
        end
    elseif sf == 2
        if strcmp(dir,'GFR') && wh==4
            pos_sf=pos_sf+1.17;
        end
        if strcmp(dir,'FR') && wh==4
            pos_sf=pos_sf+2.4;
        end
        if strcmp(dir,'FR') && wh==3
            pos_sf=pos_sf-0.3;
        end
    end
end


% Place origin at centre of squat
if strcmp(dir,'FR')
    x = -(x - pos_sf);
elseif strcmp(dir,'GFR')
    x = x - pos_sf;
end

% ideal conditions of vel, dx, dt for interpolation
dx = 1e-3;
```

```
vel = 80 / 3.6;
dt = dx / vel;

% interpolate measured aba to resolution of simulated
FF = griddedInterpolant(tt, acc(:,wh));
t2 = tt(1):dt:tt(end);
acc2 = FF(t2);
FF = griddedInterpolant(tt, x);
x2 = FF(t2);
FF2 = griddedInterpolant(tt, v3);
v2 = FF2(t2);

%find index for position of rail fault
[~,ipos] = min(abs(x2));

% Plot Time Data
AccFig = figure(1);
subplot(4,1,pp);
plot(x2,acc2,'k');
xlim([0-10,0+10])
% labels set outside of loop

%%%%%%%%%%%%%% SPECTROGRAM %%%%%%%%%%%%%%
Nw=1024;
%calculate spectrogram
ind1 = ipos - 10000; % 10 meter before squat
ind2 = ipos + 10000; % 10 meters after squat
inds = ind1:50:ind2; % resolution 0.5 mm
Nsp = length(inds); % Number of points in spatial resolution
spectro = zeros(Nsp, Nw/2-1); % pre-allocate

for i=1:Nsp
    Mmid=inds(i);
    [freq_aba,AG] = do_spectra_av(acc2(:,Mmid-Nw/2-1:Mmid+Nw/2), ...
        dt,1,Nw,Nw);
    spectro(i,:)=AG.';
end

% Resize around x = 0 for ML-alg
xp = x2(inds);
vp = v2(inds);
i = 200;
int1 = i-7:i+21;
Xex = 10 * log10(abs(spectro(int1,1:93)));
xp2 = xp(int1);
vp2 = vp(int1);
```

```
    xps(pp,:) = xp2;
    vps(pp,:) = vp2;
    fps(pp,:) = freq_aba(1:93);

    % Unwrap Data for export for ML-alg
    X(pp,:)=Xex(:)';

end

han = axes(AccFig,'visible','off');
han.YLabel.Visible = 'on';
han.XLabel.Visible = 'on';
ylbl = ylabel(han,'Acceleration / m/s^2','FontSize',16);
xlbl = xlabel(han,'Location / m','FontSize',16);
ylbl.Position(1) = ylbl.Position(1) - 0.01;
```

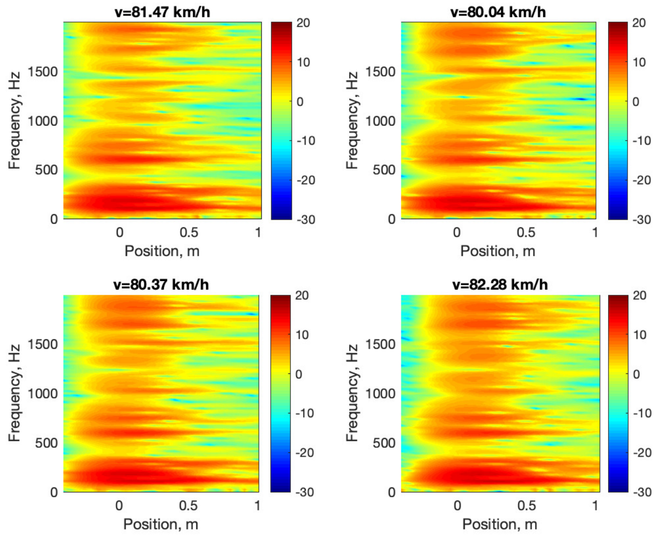


Plot

```
figure(4)
for i=1:4
    Xex=reshape(X(i,:),29,93);
    subplot(2,2,i)
    pcolor(xps(i,:),fps(i,:),Xex'/2)
    shading interp
    colormap jet
    colorbar;
    set(gca,'CLim',[-30 20]);
    xlabel('Position, m')
    ylabel('Frequency, Hz')
    title(['v=',num2str(round(mean(vps(i,:))*3.6,2)),' km/h'])
```

```
end


vok=(abs(vel-mean(vps,2))/vel)<0.05;
%make sure the velocity is within 5% from 80 km/h
X=X(vok,:);
disp([num2str(sum(vok)),' v''s ok'])

save(['ProcessedData/stft_',type,num2str(sf),dir,'_wh',...
num2str(wh),'.mat'],'X','xps','fps','vps','vok')
```

```
4 v's ok
```



B

Logistic Regression Algorithm

This code is based on the example code given for the assignments in the online course "Machine Learning" available on Coursera [19].

Part 1: Loading and Visualizing Data

```
% load training data
load ../ProcessedData/Prep_tap300_comb_order.mat
X = Xin'; % 10 * log10(abs(Xin) + eps);
y = y';
```

Leave One Out and Shuffle Data

```
% loop for the Sfs to use
% 0 -> test run
% joints: [6 15 16 18 21]
% squats: [8 10 12 13 20 28 29]
for num_sf = [0 8 10 12 13 20 28 29]

    % set shuffle
    shuffle = 1;

    % set case
    % 1) y = 1 if squat, y = 0 else
    % 2) y = 1 if squat, y = 2 if iso, y = 0 else
    % 3) y = 1 if squat, y = 2 if slight, y = 3 if iso, y = 0 else
    % 4) y = 1 if squat, y = 2 if slight, y = 0 else
    classification = 1;
    % num_labels by case 1)1, 2)2, 3)3, 4)2
    num_labels = 1;

    [X_train, X_test, X_val, y_train, y_test, y_val] =  ...
        leaveOneOut(X, y, num_sf, shuffle, classification, 1);
```

Part 2: Regularized Logistic Regression

```
    % Initialize fitting parameters
```

```
 initial_theta = zeros(size(X, 1), 1);

% Set regularization parameter lambda
lambda = 0.5;

% Compute and display initial cost and gradient for
% regularized logistic regression
[~, grad_h] = lrCostFunction(initial_theta, X_train, y_train, lambda);
```

Part 3: One-vs-All Training

```
% Set Options
options = optimset('GradObj', 'on', 'MaxIter', 800,'Display','iter');
% Optimize

[all_theta, J, exit_flag] = ...
    fminunc (@(t)(lrCostFunction(t, X_train, y_train, lambda)), ...
    initial_theta, options);
```

Part 4: predict Train

```
[pred_train,~] = predict(all_theta, X_train);
train_val = double(pred_train == y_train) * 100;

fprintf('\nTraining Set Accuracy: %f\n', ...
    mean(double(pred_train == y_train)) * 100);
```

Part 5: predict Test

```
[pred_test,~] = predict(all_theta, X_test);
test_val = double(pred_test == y_test) * 100;

fprintf('\nTesting Set Accuracy: %f\n', ...
    mean(double(pred_test == y_test)) * 100);

info.lambda = lambda;
info.iters = options.MaxIter;
info.class = classification;
info.extranotes = "90% training data, excluding isos and switches";
```

Part 6: Validation

```
if num_sf ~= 0
[pred_val,sig_val] = predict(all_theta, X_val);
val_val = double(pred_val == y_val) * 100;

fprintf('\nValidation Set Accuracy: %f\n', ...
```

```
        mean(double(pred_val == y_val)) * 100);

    save(['Results/logreg/tap/tap_vals_wosf',num2str(num_sf),'.mat'],...
        'train_val','test_val','val_val','info','sig_val')
    end

    % run test sequences
    if num_sf == 0
        Xtemp = X;
        load ../ProcessedData/tap300_test_seq1.mat
        [pred_testseq1,~] = predict(all_theta, X);
        % if excluding acc < 40
%       pred_testseq1(X(:,end) < 40) = 0;

        check_test_seq1 = find(pred_testseq1 == 1);
        disp([num2str(length(check_test_seq1)),' squats detected at'])
        xs(check_test_seq1)
        xs1 = xs;

        load ../ProcessedData/tap300_test_seq2.mat
        [pred_testseq2,~] = predict(all_theta, X);
        % if excluding acc < 40
%       pred_testseq2(X(:,end) < 40) = 0;

        check_test_seq2 = find(pred_testseq2 == 1);
        disp([num2str(length(check_test_seq2)),' squats detected at'])
        xs(check_test_seq2)
        xs2 = xs;

        save('Results/logreg/tap/tap_vals_all.mat',...
            'train_val','test_val','info',...
            'pred_testseq1','pred_testseq2','xs1','xs2',...
            'check_test_seq1','check_test_seq2')

         X = Xtemp;
    end

end
```

C

Neural Network Algorithm

This code is based on the example code given for the assignments in the online course "Machine Learning" available on Coursera [22].

Part 1: Loading and Visualizing Data

```
% load training data
load ../ProcessedData/Prep_tap_acc_comb_order.mat
% X = 10 * log10(abs(Xin) + eps); % Xin'; %
% X = Xin(:,1:2376)';
% acc = Xin(:,2377);
X = Xin(:,1:end-1)';
y = y';
```

Leave One Out and Shuffle Data

```
% set which to leave out
% for inds = 1:12
%     nums = [8 10 12 13 20 (27) 28 29 6 15 16 18 21];
%     num_sf = nums(inds);
% end
for num_sf = [8 10 12 13 20 28 29] %.  6 15 16 18 21]

%     num_sf = 8;
    % set shuffle
    shuffle = 1;

    % set case
    % 1) y = 1 if squat, y = 0 else
    % 2) y = 1 if squat, y = 2 if iso, y = 0 else
    % 3) y = 1 if squat, y = 2 if slight, y = 3 if iso, y = 0 else
    % 4) y = 1 if squat, y = 2 if slight, y = 0 else
    classification = 1;
    % num_labels by case 1)2, 2)3, 3)4, 4)3
    num_labels = 2;
```

```
[X_train, X_test, X_val, y_train, y_test, y_val] =  ...
    leaveOneOut(X, y, num_sf, shuffle, classification, 0);
```

Set up Parameters

```
input_layer_size = size(X_train,2);
hidden_layer_size = 60;
```

Part 2: Loading Parameters

```
% set so Theta values are any random number between -0.5 and +0.5
init_epsilon = 0.5;

Theta1 = rand(hidden_layer_size,input_layer_size + 1) * 2 * init_epsilon ...
    - init_epsilon;
Theta2 = rand(num_labels,hidden_layer_size + 1) * 2 * init_epsilon ...
    - init_epsilon;

nn_params = [Theta1(:) ; Theta2(:)];
```

Part 3: Compute Cost (Feedforward)

```
lambda = 0.1;

J = nnCostFunction(nn_params, input_layer_size, hidden_layer_size, ...
    num_labels, X_train, y_train, lambda);
```

Part 6: Training NN

```
% size of the training set
options = optimset('MaxIter', 200);

% Create "short hand" for the cost function to be minimized
costFunction = @(p) nnCostFunction(p, ...
    input_layer_size, ...
    hidden_layer_size, ...
    num_labels, X_train, y_train, lambda);


initial_Theta1 = randInitializeWeights(input_layer_size, hidden_layer_size);
initial_Theta2 = randInitializeWeights(hidden_layer_size, num_labels);

% Unroll parameters
initial_nn_params = [initial_Theta1(:) ; initial_Theta2(:)];


% Now, costFunction is a function that takes in only one argument (the
```

```
% neural network parameters)
[nn_params, cost] = fmincg(costFunction, initial_nn_params, options);

% Obtain Theta1 and Theta2 back from nn_params
Theta1 = reshape(nn_params(1:hidden_layer_size * (input_layer_size + 1)), ...
    hidden_layer_size, (input_layer_size + 1));

Theta2 = reshape(nn_params((1 + (hidden_layer_size * (input_layer_size + 1))):e
    num_labels, (hidden_layer_size + 1));
```

Part 7: Implement predictNN

```
After training the neural network, we would like to use it to predictNN
the labels. You will now implement the "predictNN" function to use the
neural network to predictNN the labels of the training set. This lets
you compute the training set accuracy.
```

```
[~, pred_train] = predictNN(Theta1, Theta2, X_train);
train_val = mean(double(pred_train == y_train)) * 100;

fprintf('\nTraining Set Accuracy: %f\n', mean(double(pred_train == y_train)) *

% SET SAVE NAME


Xtemp = X;
```

Part 8: Testing

```
[~, pred_test] = predictNN(Theta1, Theta2, X_test);
test_val = mean(double(pred_test == y_test)) * 100;

fprintf('\nTesting Set Accuracy: %f\n', mean(double(pred_test == y_test)) * 100
```

Part 9: Validation

```
% Information & save name
info.hiddenlayer = hidden_layer_size;
info.lambda = lambda;
info.iters = options.MaxIter;
info.class = classification;
info.extranotes = "80% training data, w isos and switches";

if num_sf ~= 0

    save(['Results/tap/tap_thetas_wosf',num2str(num_sf),'.mat'],...
    'Theta1','Theta2','cost','X')
```

```
        [h2_val, pred_val] = predictNN(Theta1, Theta2, X_val);
        val_val = double(pred_val == y_val) * 100;

        fprintf('\nValidation Set Accuracy: %f\n', mean(double(pred_val == y_val))


        save(['Results/tap/tap_vals_wosf',num2str(num_sf),'.mat'],...
            'train_val','test_val','val_val','info','h2_val')
    end
```

Part 10: Test Sequences

```
    if num_sf == 0
        load ../ProcessedData/tap_acc_test_seq1.mat
        [~,pred_testseq1] = predictNN(Theta1, Theta2, X(:,1:end-1));
         pred_testseq1(X(:,end) < 40) = 1;
        check_test_seq1 = find(pred_testseq1 == 2);
        % automatically set all values where acc < 40 to not squats
        % remove/rewrite if the acc data is not one of the features


        disp([num2str(length(check_test_seq1)),' squats detected at'])
        xs(check_test_seq1)

        load ../ProcessedData/tap_acc_test_seq2.mat
        [~,pred_testseq2] = predictNN(Theta1, Theta2, X(:,1:end-1));
         pred_testseq2(X(:,end) < 40) = 1;
        % automatically set all values where acc < 40 to not squats
        check_test_seq2 = find(pred_testseq2 == 2);


        disp([num2str(length(check_test_seq2)),' squats detected at'])
        xs(check_test_seq2)

        save(['Results/tap/tap_acc_vals_all.mat'],...
            'train_val','test_val','info','check_test_seq1','check_test_seq2')

        X = Xtemp;
        save Results/tap/tap_thetas_all Theta1 Theta2 cost X
    end

end
```