Failure prediction for mechanical doors using cheap sound analysis

Master Thesis

Student: Olof Englund, ol7501en-s@student.lu.se
Supervisor at ASSA Abloy: Roger Dreyer
Supervisor at Mathematical Statistics: Maria Sandsten
Examiner: Johan Lindström

Project period: 2021-01-18 - 2021-06-14

June 15, 2021

Abstract

As the field of Internet of Things grows more popular, its potential in fault detection becomes apparent. Much work has been done in the area where the main techniques used are machine learning methods. While powerful, they often require a large amount of computational power. A computationally cheap alternative to analyse recordings of mechanical doors for deterioration is presented and analysed in this report. By utilising repeated measurement of the same door sound, spectral analysis and data reduction, an efficient method to detect long term trends is developed. The method show some potential in detecting deterioration without requiring the same amount of computations as the machine learning techniques. The best precision was achieved using a power spectra estimated with Welch's method. However, the exact precision of the method can not be concluded based on the analysis in this report but needs more testing and collection of more real world data.

Acknowledgements

I wish to say a big thank you to Maria Sandsten, supervisor from Mathematical Statistics at LTH, for helping me steer in a good direction and assisting me in general throughout the project. I also want to thank ASSA Abloy for all support and for allowing me to collect data on their premises. Special thanks to Roger Dreyer and Rebeca Homssi for assisting me as supervisors on ASSA Abloy.

Contents

1 Introduction **1**

1.1 Motive of using sound for this analysis 2

1.2 Previous work 2

1.3 Research questions 2

2 Theory **3**

2.1 Time series and signal sampling 3

2.2 Data distributions 7

2.3 Computational efficiency 8

3 Data collection **10**

3.1 Capture of audio 10

3.2 Recording door sounds 10

3.3 Resulting data sets 13

4 Method **14**

4.1 Changing representations of the input signal 14

4.2 Preprocessing 14

4.3 Proposed detection of deteriorated doors 15

4.4 Testing the performance of the method 19

5 Analysis and results **23**

5.1 Real world data used 23

5.2 Visibility of deterioration in various representations 24

5.3 Notes on automatic detection of deterioration 29

5.4 Real world data *with* audible deterioration (*lab data set*) 30

5.5 Real world data *without* audible deterioration (Live data set) 33

5.6 Simulated data 36

6 Discussion **43**

6.1 Manual detection of deterioration 43

6.2 Automatic detection of deterioration 43

6.3 Difference between real and simulated data 43

6.4 Comparing speed of calculation 44

7 Conclusion **45**

7.1 Future work 45

1 Introduction

During recent years, a spike in usage of IOT-technologies (Internet of Things) can be seen in many industries. Fine monitoring and control of devices is now a requirement to be able to provide the best possible products and service to customers. A large motive to using IOT is the possibility to collect large amounts of data through various sensors. The data can be used to further develop a product or to monitor an installation to ensure adequate performance. The latter is what ASSA Abloy hopes to achieve using sound as a source of information.

A large part of ASSA Abloy's services involves automated doors and other entrance systems. These automated doors need to perform seamlessly where the doors open upon detecting a person fast enough to not hinder the person and silent enough to not cause annoyance. Naturally, such a system involves many moving parts such as engines, rails, belts, cogs etc. Deterioration of one or many of these parts could result in the aforementioned performance requirements not being met, or in the worst case the door fully breaking down. Machine deterioration does not only pose a problem in mechanical door systems. In, for example, compressed air systems, mechanical deterioration leading to air leakage can result in unnecessary high energy costs [1].

To provide the best possible product to the end user, the risk of failure needs to be minimised. The risk can partly be reduced by monitoring the system and taking pre-emptive measures such as changing parts or repairing the door before trouble arise. This is where IOT is a useful tool. If the door can be constantly monitored by sensors, then an imminent failure could be detected and prevented. One measure of the health of the door can be obtained by recording the audio generated when using the door. Often when there is breakage in the moving parts of a mechanic system, sounds or vibration occur which can be recorded with a microphone and analysed [2, 3, 4].

Such analysis has been done successfully numerous of times, often using computationally expensive machine learning techniques [5, 6]. But, since ASSA Abloy has thousands of concurrent door installations and the analysis would ideally be done for each installation, such expensive analysis at the door is not feasible. Additionally, the doors communicate over a metered network which removes the possibility of sending all audio to a centralised cloud where expensive computation could occur. The challenge, therefore, is to efficiently detect recordings which with adequate confidence contain some anomaly.

The mechanical door systems produce sound signals that are non-stationary in nature. These signals will be analysed both in time and in frequency using various methods. The frequency spectrum, spectrogram and reassigned spectrogram will be used for analysis in the frequency domain. Initially, manual analysis will be done to decide if a deterioration sound is present in the door sound but the goal is to perform such analysis automatically using the aforementioned methods.

1.1 Motive of using sound for this analysis

Sound is used on a daily basis by the average person to analyse the surroundings. This is especially true when focus lies on mechanical entities such as cars, elevators, bikes etc. When there is breakage in mechanical systems there is a risk that parts become misaligned which results in friction forces not intended in the initial design of the system which in turn produce sounds. ASSA technicians often use these sounds to perform an initial diagnostics on a door, gaining some insight into the problems before needing to disassemble the door. The method in this report will attempt a similar analysis and therefore be restricted to detecting recordings which contain anomalous sounds detectable by human ear. This allows any findings to be verified by simply listening to the recordings and removes the need to know the actual mechanical wear of the door components. It also focuses the analysis on sounds which can actually cause annoyance to people nearby. Furthermore, the fact that ASSA technicians listen to the door as a part of a diagnosis further motivates this restriction.

1.2 Previous work

Earlier work on failure prediction using time frequency analysis has been done by Feng et. al. where various time frequency approaches to detecting engine failure was done [2]. Another attempt at analysing vibrations in gearboxes using time frequency analysis was done in an article by Chen et. al. where the authors used an iterative approach to increase the resolution in the time frequency distributions [3]. A final method of fault detection for rotating machinery using time frequency analysis is presented by Gan et. al. [4]. A method combining time frequency representations with more expensive deep learning approaches is presented by Zhang et. al. [5]. In photovoltaic systems fault detection based on supervised machine learning through Principal Component Analysis is presented in [7] and auto-encoders are used for sensor-fault detection in [8].

1.3 Research questions

- Can deterioration of a door be *manually* detected using sound analysis tools?
- Can deterioration of a door be *automatically* detected using sound analysis tools?
- Can such an automatic analysis be efficient enough to run on a micro controller?
- Can simulated data be used to test an automatic analysis?

2 Theory

2.1 Time series and signal sampling

A time series is essentially a set of points which are ordered in time. Examples of every day time series are the value of a stock over time, some temperature measured each day or an audio recording which is the type of time series this report will focus on. An audio recording is a series of air pressure measurements done at a fixed interval called the **sample rate** which is usually denoted f_s in academic literature. The most common sample rate for recording audio is 44 100 Hz and dates back to the compact disk (CD) format.

2.1.1 Fourier transform

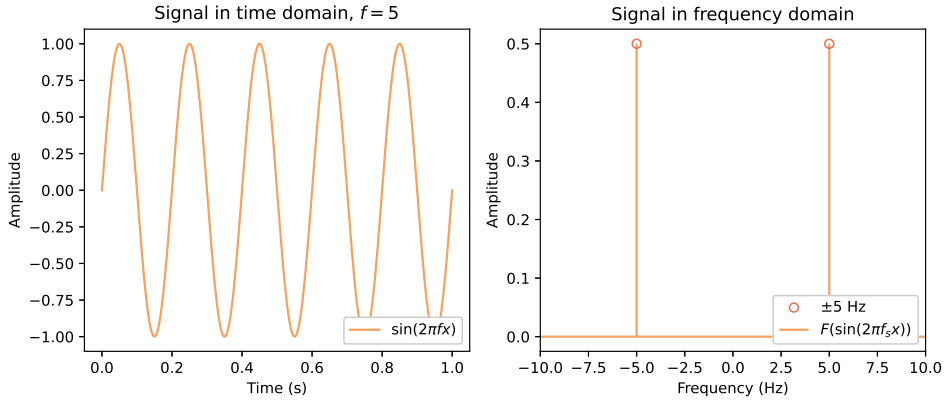

Figure 1: A sine wave with a frequency of 5 Hz and amplitude 1 (left) and the absolute value of its Fourier transform (right).

When audio is captured, it is done so in the time domain and to analyse the frequency contents of the audio one needs to transfer the signal to the frequency domain. This can be done using a Fourier transform and the result is illustrated in figure 1 where a simple sine wave is transformed from the time domain to the frequency domain. The Fourier transform essentially decomposes a function into components and is for an input signal $x(t)$, $-\infty < t < \infty$ formulated mathematically as

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi ft} dt, \ -\infty < f < \infty \tag{1}$$

[9]. By taking the absolute value of the Fourier transform ($|X(f)|$) one gets the magnitude function which is what is plotted in the right graph in figure 1 and $|X(f)|^2$ is called the spectrum. The Fourier transform presented in eq. 1 takes a continuous function $x(t)$ and produces a continuous transform $X(f)$. In any real world example, the incoming signal is not continuous but a set of discrete values as a consequence of the sampling previously described. Therefore, a discrete version of the continuous Fourier transform (DFT) is required and can be

formulated as

$$X_l = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi}{N} ln} \tag{2}$$

where $\mathbf{x_n}$ is a set of N complex numbers [10]. The Fourier transform is essential for the field of spectral analysis and is used in a range of applications - many with a requirement on performance. The (re)introduction of the Fast Fourier transform (FFT) by J.W. Cooley and J.W. Tukey in 1965 allowed for fast computation of the Fourier transform and is considered one of the most important algorithms ever discovered [11]. The discovery of the FFT, however, could have already been done in the early 1800s by Gauss as a means of quickly calculating the movement of planetary bodies [12]. Roughly speaking, the FFT exploits symmetry in the Fourier transform to divide the sum in eq. 2 into equal parts, both containing $M = N/2$ elements. This can be done recursively as long as M is a power of 2. For each division of elements, half the amount of computations are required to find the Fourier transform. As a result of this optimisation, the FFT has a time complexity of $\mathcal{O}(N \log(N))$ while the original DFT has a time complexity of $\mathcal{O}(N^2)$ [13].

The DFT maps N discrete values in the time domain to N discrete values in the frequency domain. To map N discrete values in the time domain to a continuous function in the frequency domain, one can use a discrete time Fourier transform (DTFT) which is found by letting $n \to \infty$ in eq. 2 [14].

From the DTFT, one can calculate what is called a periodogram which is analogous to the spectrum for the continuous Fourier transform. A window function $h(n)$ is used to extract a finite range of points in the time domain. The periodogram is then calculated by squaring the magnitude function calculated from these points together with a scaling factor:

$$P(l) = \frac{1}{N} \left| \sum_{n=0}^{N-1} h(n) x_n e^{-\frac{i2\pi}{N} ln} \right|^2 \tag{3}$$

assuming the window function has a length of N [14]. The periodogram has the same time complexity as the FFT but is done on fewer points since the window length usually is shorter than the total amount of points.

2.1.2 Welch's method

When using real world data, there is usually some amount of noise in the data. This noise can have a large effect on the appearance of the periodogram and result in false peaks or peaks with low resolution. In essence, the periodogram is not well suited for data with high variance. A well known method for reducing variance in data is averaging and it turns out that averaging works well here too. Bartlett's method splits a sequence of N points into K equal parts, windows these parts, calculates the DFT and finally the periodogram for each of these parts. The parts are then averaged to retrieve a single spectrum with a lower variance then the original periodogram in exchange for a lower frequency resolution [15]. This method was

modified by Peter D. Welch to include some overlap D between each part. As a consequence of the shape of window functions, more information is retained in the centre of the window compared to the edges. The overlap reduces the loss of information in the edges and further enhances the resulting spectrum.

The periodogram in eq. 3 can be modified to only include a segment of length L of the total points N. Let this periodogram be called P_m. The segment can overlap with another segment of the same length L with D overlapping points. Create K segments such that $(K-1)*(L-D) = N$, i.e. the segments include all points N. Now calculate the K modified periodogram as

$$P_m(l) = \frac{1}{L}\left|\sum_{n=0}^{L-1} h(n)x_n e^{-\frac{i2\pi}{L}ln}\right|^2 \tag{4}$$

and average them to retrieve the spectral estimate

$$P_{welch}(l) = \frac{1}{K}\sum_{k=1}^{K} P_m^k(l) \tag{5}$$

[16]. Since the periodogram consists of calculating multiple periodograms and averaging them, the time complexity is the same as for the periodogram with a factor K equal to the number of periodograms:

$$\mathcal{O}(P_{welch}) = \mathcal{O}(KL\log(L)) \tag{6}$$

2.1.3 Spectrogram and marginals

The Fourier transform, in its unmodified form, takes exactly one input signal x and generates exactly one frequency spectrum. If one were interested in how the frequency changes over time in an input signal, the Fourier transform in itself would not help. A method to analyse frequency as a function of time is to use a spectrogram which involves dividing the input signal into equal parts and computing the Fourier transform for each of these parts. The parts are then combined to form a function that maps time and frequency to an amplitude. This can be done using the short-time Fourier transform (STFT):

$$X(t,f) = \int_{-\infty}^{\infty} x(\tau)h^*(\tau - t)e^{-i2\pi f\tau}d\tau,\ -\infty < t, f < \infty \tag{7}$$

where h is a window function centred at time t. The window used for the spectrograms in this report is a Hann window. The spectrogram is then found in the same manner as the spectrum in the Fourier transform:

$$S_x(t,f) = |X(t,f)|^2,\ -\infty < t, f < \infty \tag{8}$$

[9]. Picking a proper window size is vital to be able to resolve specific components in the analysed signal. Due to the uncertainty principle, one can not have both perfect time resolution

and frequency resolution. There is therefore always a trade-off between these resolutions. A rule of thumb is, however, to pick a window size similar to the length of the components one is interested in [9].

As with the Fourier transform, the continuous spectrogram has a discrete version more suitable for machine computation. The spectrogram can be discrete in either time, frequency or both time and frequency. The latter, being the only practically useful alternative, is defined as

$$S_x(n,l) = \left| \sum_{n_1=0}^{N-1} x_{n_1} h^*(n_1 - n + M/2) e^{-i2\pi n_1 \frac{l}{L}} \right|^2 \tag{9}$$

where M is the length of the window h. This definition of the discrete spectrogram requires performing an FFT for each value n. This is, however, not required as the calculations can be done for every N_{step} value where N_{step} can be as large as $M/8$ without significant visual change [9]. The resulting time complexity of the spectrogram is $\mathcal{O}(\frac{N}{N_{step}} L \log(L))$.

The spectrogram produces an representation of how the frequency content changes over time and can, given a specific frequency and time point, return the occurrence of that specific frequency at that specific time point. For some applications where there are changes in many frequencies concurrently at specific time points, an interest might be how the total frequency contents changes with respect to time and not how specific frequencies change. Such a representation can be found by calculating the time marginals by simply integrating the spectrogram over all frequencies:

$$M_x(t) = \int_{-\infty}^{\infty} S_x(t,f) df \tag{10}$$

[9] which for the discrete case involves summing the spectrogram over frequencies:

$$M_x(n) = \sum_{f=-\infty}^{\infty} S_x(n,f). \tag{11}$$

This will result in a measure on how the total frequency content of a signal changes over time and is sensitive to transient changes in all frequencies. The time complexity is similar to the spectrogram but also involves summing all frequencies which results in a time complexity of $\mathcal{O}(\frac{N^2}{N_{step}} L \log(L))$.

2.1.4 Reassigned spectrogram

Due to the uncertainty principle, one can not achieve perfect localisation in either frequency or time. A method to increase the localisation of the spectrogram in either frequency, time or both is the reassigned spectrogram. Given the spectrogram $S_x(t,f)$ from equations 7 and 8 with $\omega = 2\pi f$, the reassigned spectrogram is defined as

$$RS_x(t,\omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S_x(t_1,\omega_1) \delta(t - \hat{t}_x(t_1,\omega_1), \omega - \hat{\omega}_x(t_1,\omega_1)) dt_1 d\omega_1 \tag{12}$$

which results in each value (t, ω) being relocated to corresponding $(\hat{t}_x, \hat{\omega}_x)$. $\delta(t, \omega)$ is the two dimensional Dirac impulse which retrieves the instantaneous time and frequency at a location and is defined as

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,\omega)\delta(t-t_o, \omega-\omega_0)dtd\omega = f(t_0, \omega_0). \tag{13}$$

The reassigned time and frequency components are defined by

$$\hat{t}_x(t,\omega) = t + \Re\left(\frac{X^{th}(t,\omega)}{X^h(t,\omega)}\right),\ \hat{\omega}_x(t,\omega) = \omega + \Im\left(\frac{X^{dh/dt}(t,\omega)}{X^h(t,\omega)}\right) \tag{14}$$

with $X^{th}(t,\omega)$ and $X^{dh/dt}(t,\omega)$ being STFTs (eq. 7) of the signal $x(t)$ using a time multiplied window $t * h(t)$ and the time derivative window $dh(t)/dt$ respectively [9].

2.1.5 Autoregressive models

An autoregressive (AR) model is a model which output depend linearly on some number of previous values. The order p of the model determines how many previous values the output depend on and an AR(p) model is defined as

$$X_t = \alpha + \beta_1 X_{t-1} + \ldots + \beta_p X_{t-p} + \epsilon_t. \tag{15}$$

where α is a constant, $\beta_1, \ldots, \beta_p$ are parameters and ϵ_t white noise. AR models are commonly used to model various time-varying processes but can also be used to generate different types of noise depending on the order of the model. An AR(0) model is a white noise process which spans all frequencies while an AR(2) model can be constructed to output noise with a specific peak frequency. The frequency peak location is determined by the parameters β_1, β_2 and can therefore be moved to produce various types of noise [17].

2.2 Data distributions

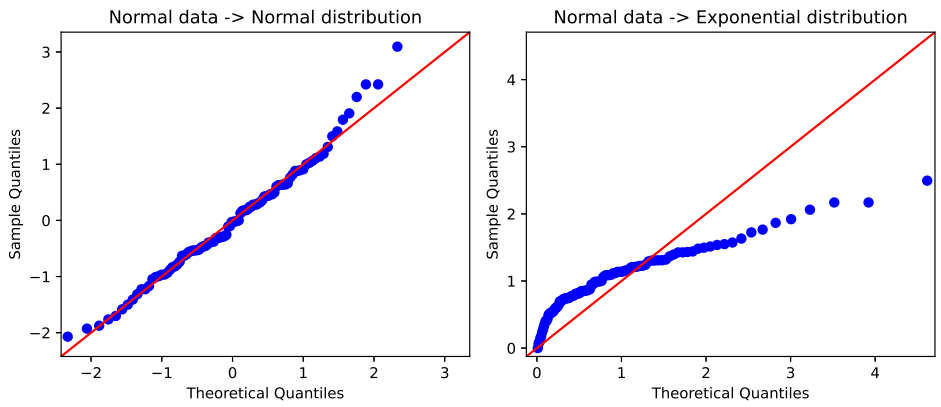

Figure 2: A Q-Q plot of data sampled from a normal distribution against a normal distribution (left) and an exponential distribution (right).

Data retrieved from a process such as door deterioration will conform to some data distribution. Deciding which distribution the data follows is important to be able to make assumptions on the data. It is also important when trying to describe the data with various models and can be used to pick or discard specific models. Finding the distribution of some specific set of data is not always an easy task but likely involves thorough analysis of the data and visualisation of the data with various methods. One such visualisation tool to give an idea on how a set of data follows a specific distribution is the quantile-quantile (Q-Q) plot. A Q-Q plot produces a visualisation of the data with respect to a specific distribution which provide easy to interpret visual feedback on how well the data conforms to the distribution. Producing a Q-Q plot is done by plotting the quantiles of the data against the quantiles of the theoretical distribution in a scatter plot. If the distribution of the data is the same as the theoretical distribution, the points in the scatter plot should form a straight line. This behaviour can be seen in figure 2 where data sampled from a normal distribution is plotted against two different distributions. In the left plot the theoretical distribution is a normal distribution which results in a straight line as expected. In the right plot, however, the theoretical distribution (an exponential distribution) is not the same as the actual one which results in a clearly skewed line.

2.3 Computational efficiency

Since this report assumes calculations are done in an environment where computational power is limited, it is important to have tools to analyse the computational efficiency of a specific set of calculations. A micro controller, for example, does not offer the same computing power as a desktop PC or a dedicated server. An important aspect of analysis on the edge node is therefore the computational efficiency of the implementation. Each step in the analysis process will require some amount of computational effort and care will therefore need to be taken to optimise each step. Generally the largest gain for the least effort can be made by firstly optimising the most computationally heavy parts of the process. However, to be able to compare the computational need of various steps, a measure is needed.

Time complexity is a measurement of the general computational effort needed for a specific algorithm or a function. A so called big O ($\mathcal{O}$) notation is used to compare algorithms and is based on the limit behaviour of the algorithms, i.e. how the algorithm performs when the amount of elements processed grows arbitrarily large. The measure, by definition, also ensures that the worst possible performance of an algorithm is proportional to the $\mathcal{O}$ time complexity [18]. One example is sorting a list which, for some sorting algorithms, has a time complexity of $\mathcal{O}(n^2)$ where n is the amount of elements in the list. The worst case performance of the algorithm is therefore that n^2 comparisons are needed to sort the list but in reality, fewer comparisons are likely required. For example, if the list is already sorted, only n comparisons are needed (to traverse the list and verify that each element is in its correct position).

2.3.1 Vectorisation

It is obvious that the lower the amount of calculations, the faster the algorithm. Reducing the amount of needed calculations should be the first priority when optimising machine instructions but there are other ways to optimise the running time of a specific algorithm or program. One common way to speed up execution of code is to *vectorise* it. Vectorisation essentially means that instead of performing an operation on each element in an array separately, the operation is done on multiple elements simultaneously. These elements, however, need to be adjacent in memory for vectorisation to be possible.

Vectorisation in python can be achieved using the *NumPy* package. NumPy introduces a data structure which is precisely mapped in memory to allow for vectorisation. NumPy also contains a large amount of methods which serve as an interface to C libraries. Since C is a low level language and python is not, these methods execute faster than the same method in python and are essential for achieving adequate efficiency in python [19].

3 Data collection

Two separate data collections were performed during this project, both on the ASSA Abloy premises in Landskrona. The first collection was done in the test lab in a controlled test environment and the resulting data set will be referred to as the **Lab data set**. The other was done in an office space on a live installation and this resulting data set will be referred to as the **Live data set**.

3.1 Capture of audio

All audio captured is done so with a sample rate of 44 100 Hz and a resolution of 18 bits.

3.2 Recording door sounds

Initial real world data was collected by recording door opening and closing cycles with mobile phones of models Samsung Galaxy s9 and Samsung Galaxy s21. The phones were held at distances between a few centimetres to an arms length from the door. Since no stand was used to ensure consistent recording distances, these recordings are inconsistent in volume levels and requires some normalisation before usage.

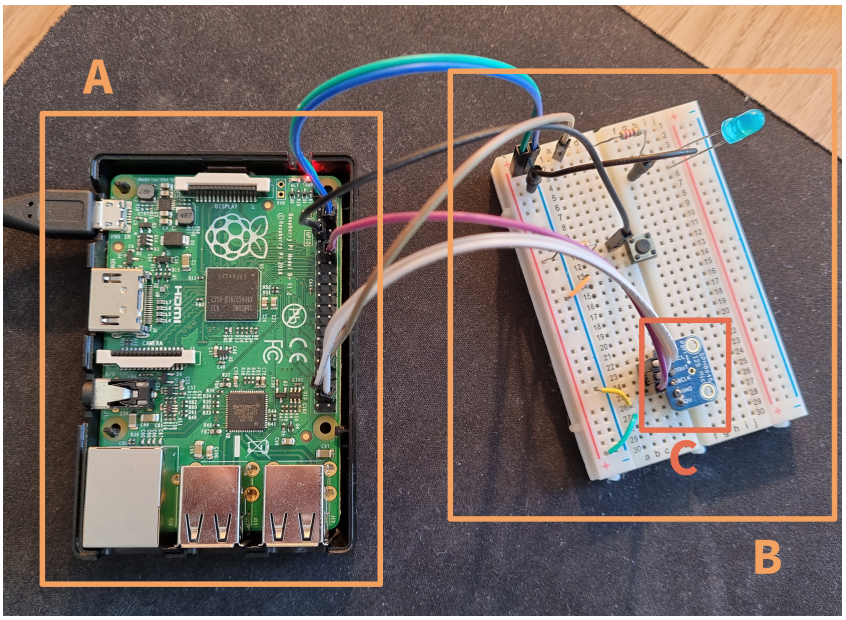

Figure 3: Raspberry PI (A) connected to a breadboard with a recording circuit (B). The blue chip in the lower part of the breadboard (C) is a MEMS-microphone.

Later on, recording was done using a micro-controller with a connected microphone. The hardware set-up is shown in figure 3 and a more detailed description can be found in the next chapter. Initially, start of recording was triggered by a logical signal which also triggered an opening cycle in the door. Data acquired using this method is naturally normalised in both

maximum amplitude and time due to the microphone being at a consistent distance away from the door and the recordings started by a logical signal coupled with the door.

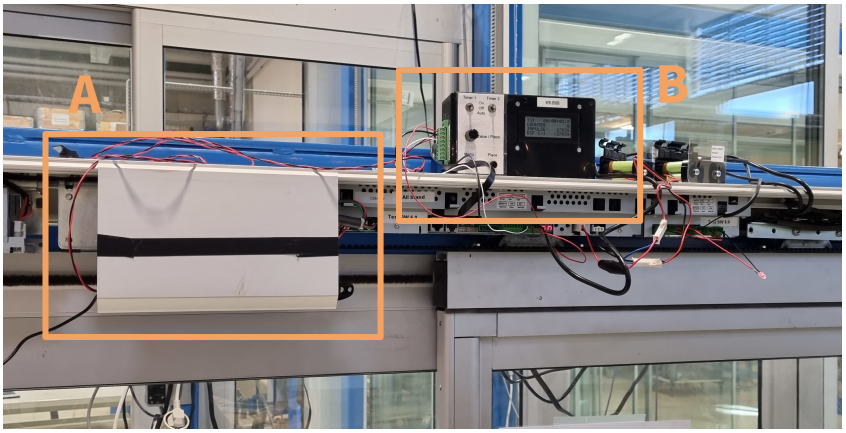

Figure 4: Location of the Raspberry PI (A) together with a pulse signal generator (B) which triggers start of recording and the door opening cycle. The actual slide door can be seen in the lower part of the image and the microphone is pointed towards it. This set-up was used to collect the *lab data set*.

During the recordings in this environment the microphone was placed close to the machinery at the top of the door, pointed towards the door. The approximate vertical distance from the door was 0.1 meters. See figure 4 for an image of the set-up. This logical signal was sent to the door and the MCU in approximately 20s intervals continuously for a couple of days producing several thousands of recordings.

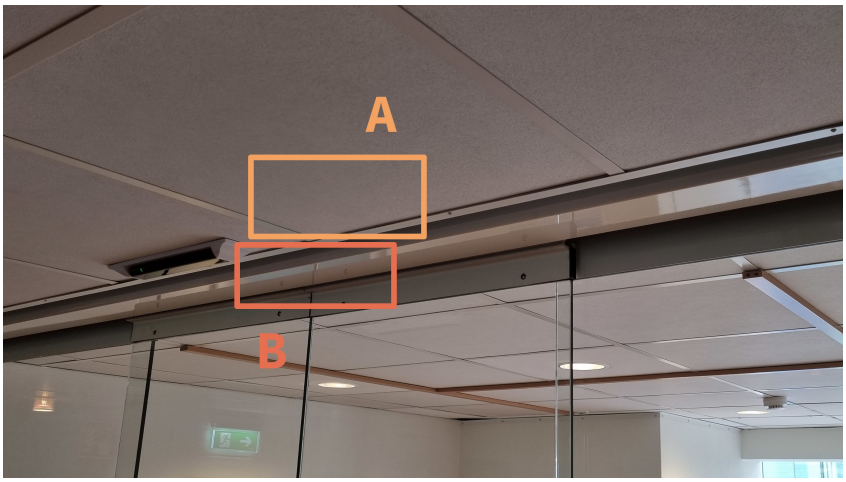

Figure 5: Image of the door used in the *live data set* collection where the Raspberry PI is located on top of the roof tile with microphone pointed towards the centre of the door (A) and the location of the lock sound source (B).

This set-up was only available in a controlled test environment and not at a typical door installation due to requiring an external logical signal for triggering a door cycle. To be able

to gather data in another environment, a simple implementation was developed to trigger on a threshold sound level. Some of the doors have locks which produce a relatively loud and short audio spike when unlocked. The sound stream from the microphone was continuously monitored for such spikes and a recording was started upon detection. Naturally, this set-up is more prone to erroneous recordings due to ambient sounds triggering recordings or slight inconsistencies in lock sound levels leading to missed recordings. An image of this set-up can be seen in figure 5.

Since this collection took place at a door in use in the ASSA Abloy office space (not a test lab), data was only collected when someone passed the door which resulted in much fewer recordings per day compared to the lab environment. Therefore this collection took place during several weeks where data from the micro-controller (MCU) was retrieved multiple times. Retrieving data involved moving the MCU and then placing it back for further collection. Since the MCU was not mounted, this slightly changed the microphone position and orientation between each session which could further increase inconsistencies in the data.

3.2.1 Hardware components and protocols

A **Micro-controller (MCU)** is essentially a small computer, often with a fraction of the computing power of a normal desktop computer. Due to a small size and low cost, MCUs are used in a range of industries for various tasks ranging from control of machinery to gathering of data. The MCU used in this project is a Raspberry Pi B+ and is visible in figure 3.

To communicate between MCUs and other chips, a common interface to use is **General-purpose input/output (GPIO)**. The Raspberry Pi has 48 GPIO pins which are located on the right end of the chip in figure 3.

Several of these GPIO pins are occupied by cables connected to a **breadboard** (*B* in figure 3) which is a component widely used in development environments to test various hardware circuits.

Connected to this breadboard is a **Micro-electro-mechanical systems (MEMS) microphone** which is a hardware component for recording audio. MEMS are essentially a group of components having sizes in the micrometer range. Being small in size results in MEMS components common usage in mobile phones. MEMS microphones is also easily manufactured and provide an adequate recording quality [20].

Audio was recorded as per the **I2s** standard which is a protocol for serial communication between audio devices in integrated circuits. The microphone used in this project had a precision of 18 bits [21].

3.3 Resulting data sets

The *lab data set* consists of almost 12 000 recorded door cycles over 3 days. Due to the door being situated in a test environment together with several other doors, the amount of external noise in this data set is high. There are sounds from other doors opening and closing but also voices and other external sounds.

The *live data set* contains much fewer recordings since recordings are only captured when someone passes the door. The data collection ran for multiple weeks and resulted in around 2400 recordings. The amount of external noise in this environment is low compared to the test lab. Examples of common external sounds are voices or footsteps.

To be able to verify the results found in the analysis, recordings late and early in the data collection was manually compared by listening to the recordings. Some care is also taken to pick recordings with minimal external noise. The recordings picked for comparison will also be used in part of the analysis and be referred to as the **early recording** and **late recording**.

In the *lab data set* an audible difference is present between the early and late recording as a rattling sound in the first half of the recordings had appeared. In the *live data set*, no audible difference is present.

4 Method

This chapter mainly describes the proposed method for detecting deterioration of door but also how collected real world data and simulated data was used to test the method.

4.1 Changing representations of the input signal

The proposed method will not only be used on the time domain signal but also on the frequency contents of the signal and the time marginals of the signal spectrogram. The method will only be described for the time domain but is applied in the exact same way in all representations.

4.2 Preprocessing

4.2.1 Normalisation

Prior to using the data for analysis, some of the data required normalisation in sound amplitude and time offset. Each recording had amplitudes scaled to the interval $[-1, 1]$ by dividing each recording by its maximum amplitude. The recordings were also shifted in time to ensure that door sequences started at the same point in time. This was done either manually or by using common patterns in the recording such as amplitude peaks present in all recordings.

4.2.2 Extraction of deterioration sounds

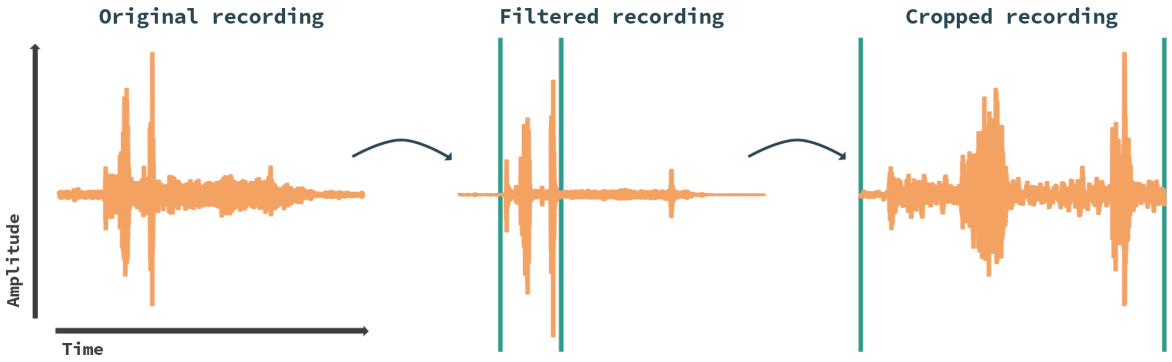

Figure 6: Recording of a door opening sequence containing a deterioration (squeaking) sound (left). A high pass filter is applied and the location of the deterioration sound decided (middle). Lastly, the recording is cropped to only include the deterioration sound (right).

To simulate deterioration in the door, recordings from already deteriorated doors was used. The deterioration sounds were located and extracted by cropping and filtering the original recording as is shown in figure 6.

4.3 Proposed detection of deteriorated doors

Firstly, the data will be manually analysed using five different representations. The representations are *energy* - the energy of the time signal, *Welch* - the frequency spectrum estimated with Welch method from eq. 5 using a window size of 20 ms, *spectrogram* - the discrete spectrogram defined in eq. 9 with a window size of 10 ms, *marginals* - the time marginals calculated from a spectrogram with a window size of 20 ms as per eq. 11 and the *reassigned spectrogram* - the reassigned spectrogram from eq. 12 using a window size of approximately 30 ms. After manually analysing the data with these methods, the *energy*, *Welch* and *marginals* will be used for automatic analysis. Since the automatic analysis is independent on representation, only the method applied to the time domain will be described in the upcoming chapters.

4.3.1 Grouping and aggregation of recordings

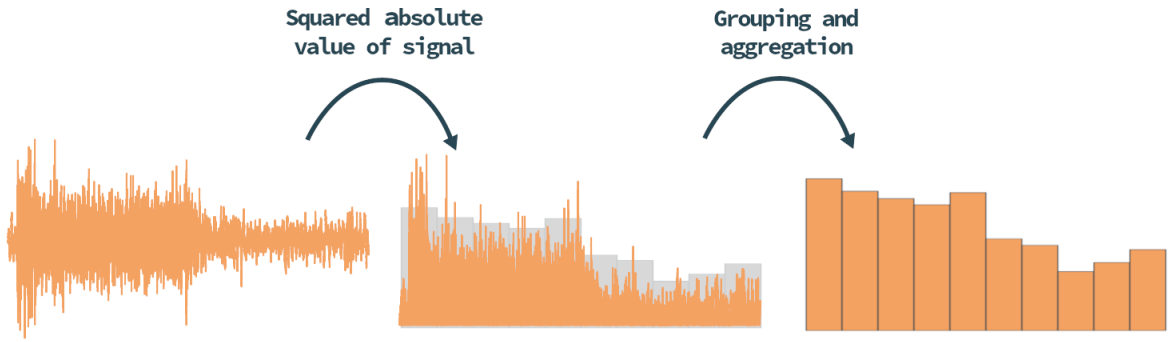

Figure 7: Illustration of the grouping and aggregation of a recording in time. The left graph visualises the time audio signal which a door produces during its opening sequence. The middle graph displays the energy of this signal which is split into ten parts where the mean energy of each part is calculated to produce the histogram in the right graph. The middle figure is overlayed with the right figure to illustrate how the recording is split and aggregated.

The basis of all proposed methods is the data reduction achieved by grouping and aggregating data points into fewer data points. Figure 7 illustrates the process on a typical recording of a door opening sequence. The data points are split into n equal parts and some aggregation is done on each part. Some examples of aggregation are taking the mean, median, maximum or min of the data points in each part. Typically, n is much smaller than the actual amount of data points which results in a relevant reduction in the amount of data points. This data reduction results in fewer required calculations and increases the efficiency of the method. Splitting the data into parts will also capture localised differences in the data while reducing noise as can be seen in figure 7 where the rough differences in amplitude in the original signal still is visible in the histogram.

4.3.2 Comparing processed recordings

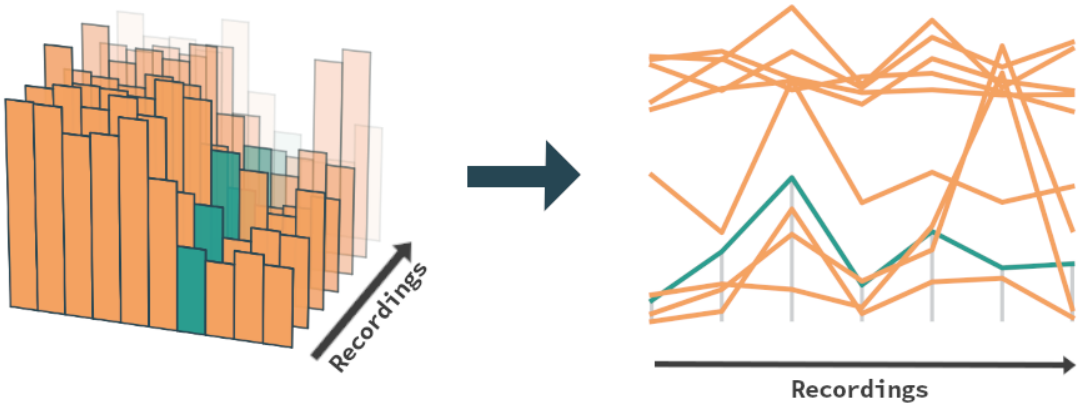

Figure 8: Representation of how aggregated groups in individual recordings are compared. Each group consists of multiple bars in the left plot and one corresponding line in the right plot. One group is highlighted in green in both plots to more clearly visualise the connection between the representations.

After grouping and aggregating, each recording is represented only by its n groups. The rightmost plot in figure 7 shows the final representation of one recording. Each group has a value which, given consistent grouping and aggregation, is comparable to the same group in another recording. By grouping and aggregating a series of recordings, one can construct a series of values for each group. Figure 8 illustrates how multiple grouped and aggregated recordings can be used to construct multiple time series. These time series now constitute a measure on how various parts of the original data changes over multiple recordings. Using the grouping and aggregation visualised in figure 7, if there is a change in amplitude between recordings, this change will be visible in the time series.

4.3.3 Detecting changes between recordings

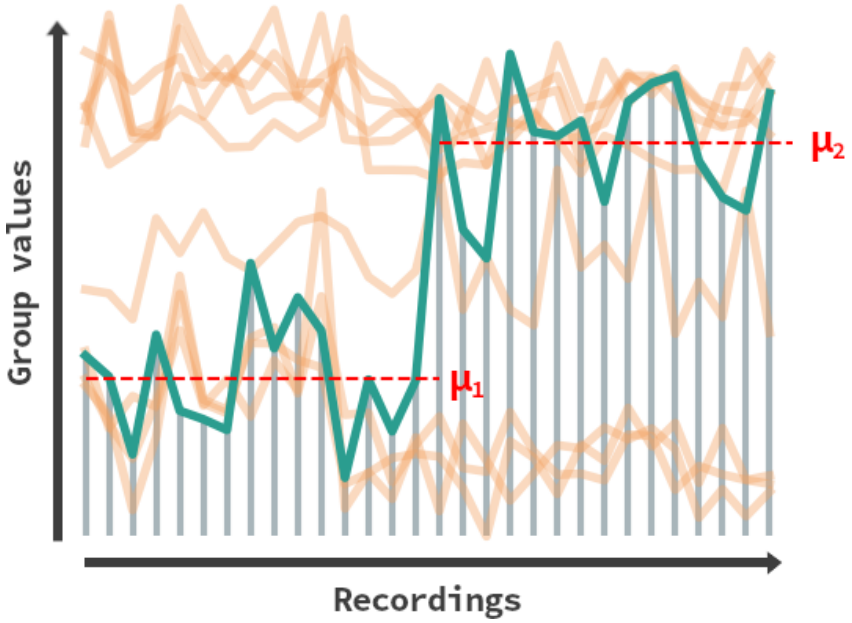

Figure 9: Several time series of grouped recordings where one specific group is highlighted in green. This group initially has a mean μ_1 over some recordings but after a while the mean for this specific group is shifted upwards to a new mean of μ_2. One can also see smaller, seemingly random, variation for both the highlighted group and the yellow groups which does not indicate a shift in the mean. This is not real data but an exaggerated example to clearly show a shift in the mean.

The time series in figure 8 contain some natural variation due to external sounds such as voices or other ambient sounds. The mean of each group, however, is assumed to be constant over many recordings when the door functions normally without anomalous sounds. If the door starts to deteriorate and anomalous sounds starts to occur in each recording, the mean should be shifted as a result. Essentially, each group should follow some distribution with mean μ_1 and variance σ^2 during normal operation. If the door starts to deteriorate, the mean of any number of groups should start shifting to some new mean μ_2. An example of this can be seen in figure 9 where one group has its mean shifted upwards. Consistent changes in group mean such as in the figure should be detected while ambient variation (which is also visible in the figure) should be ignored. Deciding when the time series reach a new mean μ_2 is arbitrary and therefore the value x of each point will be used as a more specific measure of the deviation from the original mean.

A baseline model of each group can be created by calculating the mean and variance of each group over a number of recordings. A measure of how much a new point x deviates from the baseline model mean μ can then be determined using Z-scores:

$$z = \frac{x - \mu}{\sigma} \tag{16}$$

where σ is the standard deviation of the baseline model. If z is close to 0 the new point likely belongs to the baseline model while any deviation from 0 indicates a change in the sound of the door. For $|z| > k$ the point deviates k standard deviations σ from the original mean. A large z can arise from a temporary external sound and does not necessarily indicate a deteriorated door. The challenge is to detect a consistent change in z over many recordings which minimises the possibility of an external sound causing the increase.

4.3.4 Detecting a consistent change while eliminating temporary variance

To be able to detect a consistent change from the mean, some cumulative measure has to be used. Such a measure can be constructed in a number of ways, but in this report a method called *cumulative sum control chart* (CUSUM) will be used. A cumulative measure z_c^i is initially set to $z_c^0 = 0$ and updated for each recording x_i accordingly:

$$z_c^{i+1} = \max(0, z_c^i + \frac{x_i - \mu}{\sigma} - \alpha) \tag{17}$$

where α is some user defined constant which regulates how sensitive the algorithm is [22]. If x_i is close to the mean μ, the middle term is close to 0 which transforms the expression into

$$z_c^{i+1} = \max(0, z_c^i - \alpha) \tag{18}$$

where z_c^i will decrease to 0. If, however, x_i is significantly greater than μ, the middle term will be positive and contribute to z_c^{i+1}. Assume the middle term has the positive value β, then the expression can be written as

$$z_c^{i+1} = \max(0, z_c^i + \beta - \alpha) \tag{19}$$

and whether z_c^i increases or decreases is determined by the following inequalities

$$\begin{cases} z_c^{i+1} - z_c^i > 0 \iff \beta > \alpha \\ z_c^{i+1} - z_c^i \leq 0 \iff \beta \leq \alpha. \end{cases} \tag{20}$$

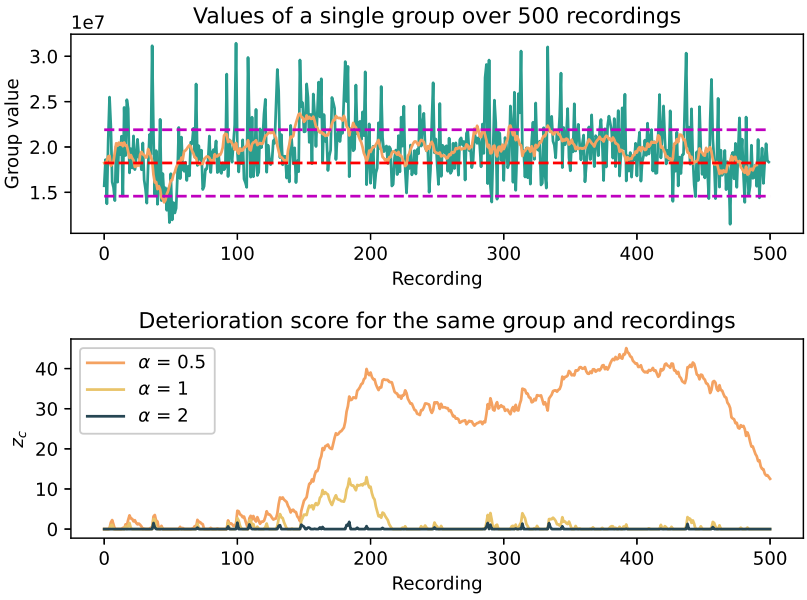

Figure 10: Visualisation of how z_c develops over 500 recordings for one group. The top plot shows the group values of one specific group together with a mean μ (red dashed line) and one standard deviation σ (magenta dashed line) calculated from the first 10 recordings. A 10 point moving average is also displayed in yellow. The bottom plot shows how z_c changes over the recordings with varying α.

Essentially, if the value x_i starts to deviate consistently from the mean μ, then z_c^i will start to increase, signifying a consistent shift from the original mean. By picking a large α, the implementation allows a large deviation from the original mean before z_c^i starts to increase which could be useful in areas where large amounts of noise is to be expected. Picking a small α, however, will result in the implementation being highly sensitive to changes in input levels.

Deciding whether a specific door is in a deteriorated state based on z_c^i can be done in a number of ways. A simple threshold could be used to determine deterioration once z_c^i grows larger than this threshold. A bit more advanced approach where one requires z_c^i to be larger than a threshold over several recordings could also be used. Implementation of a decision system is unfortunately out of scope for this report and will not be studied further.

4.4 Testing the performance of the method

The proposed procedure was tested on both simulated data and on real world data collected from two different slide doors.

4.4.1 Using real data as input

Firstly, recordings captured early and late in the data sets were manually compared as described in chapter 3.3. After that, manual spectral analysis was done in the representations

presented in the beginning of chapter 4.3 to determine the visibility of deterioration in each of the representations. Lastly, in the automatic analysis, the previous manual analysis was used to determine whether the automatic analysis was successful by comparing the results.

4.4.2 Simulation of door deterioration to generate input data

Since deterioration of a door occurs slowly and with small incremental tear, collecting real world data where the deterioration is visible is cumbersome. Therefore, to make the assessment of the method more robust, a small amount of data was used together with some assumptions on the deterioration process to simulate multiple real world processes.

A few recordings of the same door opening cycle was used as a baseline for the simulation. A "ground truth" recording was chosen as a foundation to generate simulated samples. This recording was manually analysed to ensure no anomalous sounds were present. Using this recording, various types of disturbances could then be added. These disturbances can be split into true and false disturbances were true disturbances are sounds arising from deterioration in the door and false disturbances are sounds from the environment.

True disturbance was simulated by incrementally adding an actual deterioration sound extracted from another door. A deterioration factor was used to decide the amplitude gain of the added deterioration and was typically increased incrementally up to a certain level over a simulation to mimic a real deterioration process. The deterioration sound level was capped to reach a maximum value of 20 % of the largest amplitude in the base recording. This level resulted in a faint, but audible, sound of the deterioration when added to the recording. The deterioration sound used is a squeaking sound which is visualised in the rightmost plot in figure 6. Since the deterioration sound is much shorter than the actual door recordings, the location of the sound can be varied and the expected disturbed groups decided. This is done by simply placing a high amplitude disturbance at a specific location and performing the grouping and aggregation. The differing groups are determined to be affected by the deterioration sound. This is done to decide if the deterioration score actually reflects a disturbance (true positive) or if the score is high as a result of added artificial noise (false positive). The three groups with largest response to the raw true disturbance in each representation was used. If the method decided deterioration occurred in any of those groups it is deemed successful in identifying the deterioration. Since no specific score breakpoint is defined, a non-zero score during the last 20 % recordings is deemed a detection of deterioration. This measure ensures that the z-score is larger than α in eq. 17 over many iterations.

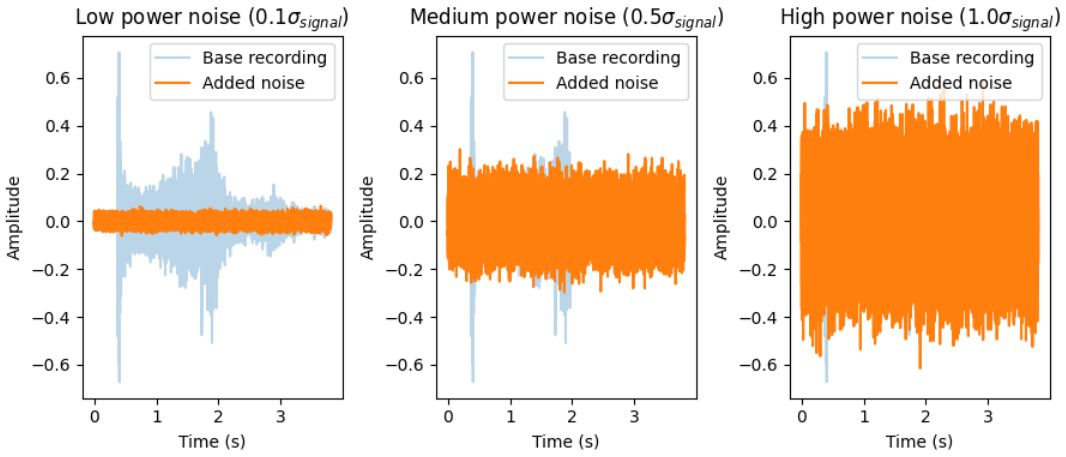

Figure 11: Disturbance in the form of AR(2) noise added to a recording in various amounts. The percentage refers to the maximum amplitude of the noise in the range $[0, 1]$.

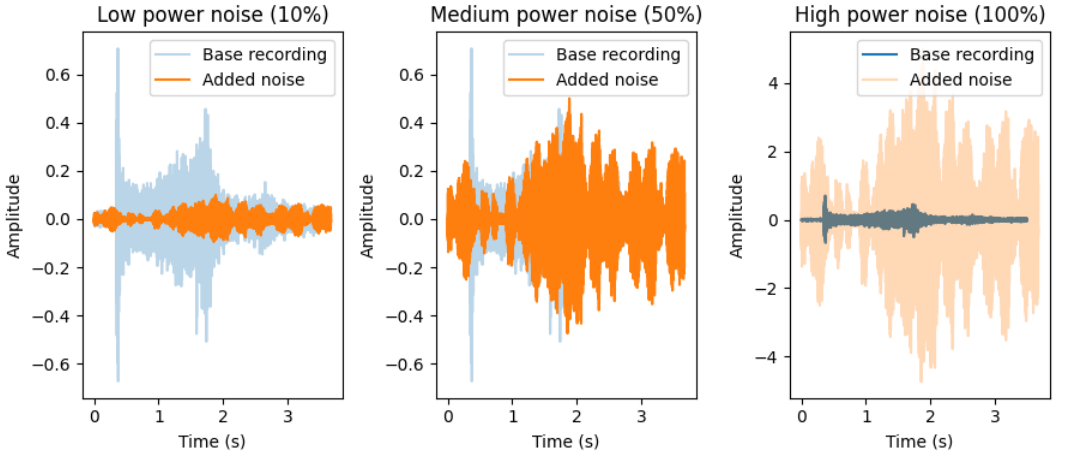

Figure 12: Disturbance in the form of a child playing added to a recording in various amounts. The percentage refers to the maximum amplitude of the noise in the range $[0, 1]$. In the rightmost plot, the disturbance sound is made transparent to reveal the recording.

Various false disturbances was used during the simulation to assess the robustness of the proposed algorithm. AR(2) noise with a peak frequency of 350 Hz was added together with various recordings from the *UrbanSound8K* data set of noises often found in urban environments such as cars, car horns, voices, etc. [23]. Three levels of AR(2) and Urban noise power was tested to asses how loud the noise can be before the method fails. The three levels tested for the AR(2) noise were 10%, 50% and 100% of the standard deviation σ of the base recordings and is visualised in figure 11. The three levels tested for the urban noise were 10%, 50% and 100% of the maximum possible noise amplitude in the range [0,1] i.e. 20% will result in a maximum amplitude of 0.2. One such recording was added for each simulated door recording and an example of urban noise levels can be seen in figure 12.

Since the performance could depend on the location of the added true disturbance, 61 trials with varying disturbance location were run and the results summarised.

4.4.3 Measuring method success or failure

Measuring whether the method did not detect deterioration is done based on the proposed scoring system. Since a score that deviates from zero during many iterations indicates some permanent change in the sound of door, the minimum value of the score during the last recordings will be used as a measure. Let z_d^{min} be the minimum score for the last 20% of recordings for a specific group. If if z_d^{min} is different from zero then the method has detected a deterioration and the deteriorated group is decided as the group with highest z_d^{min}. If z_d^{min} is zero then no deterioration is present in the specific group and if z_d^{min} is zero for all groups the door is deemed healthy. As a measure of the amount of deterioration for a specific group, the mean score is used for the last 20% recordings.

5 Analysis and results

In the automatic analysis using the method described, 10 groups were used for each representation. The aggregation method used is the mean in each group.

5.1 Real world data used

The two real world data sets collected were used to test the method. One recording from early in the data collection was picked and compared to one recorded in the end of the data collection by manually listening to the recordings. The recordings picked for comparison will also be used in part of the analysis and be referred to as the **early recording** and **late recording**.

In the *lab data set* an audible difference was present between the early and late recording as a rattling sound in the first half of the recordings had appeared. In the *live data set*, no audible difference is present.

5.1.1 Distribution of the real world data

The distribution of the individual groups in the real world data will be analysed by partly using a Q-Q plot where the data in each group is plotted against a lognormal distribution but also by plotting a histogram with the data. Before producing the plots, an attempt to remove outliers was made by removing the 2% largest and 2% smallest values in each group. The *energy* representation is used and the group that fits the best and the worst to the distributions are visualised for both data sets.

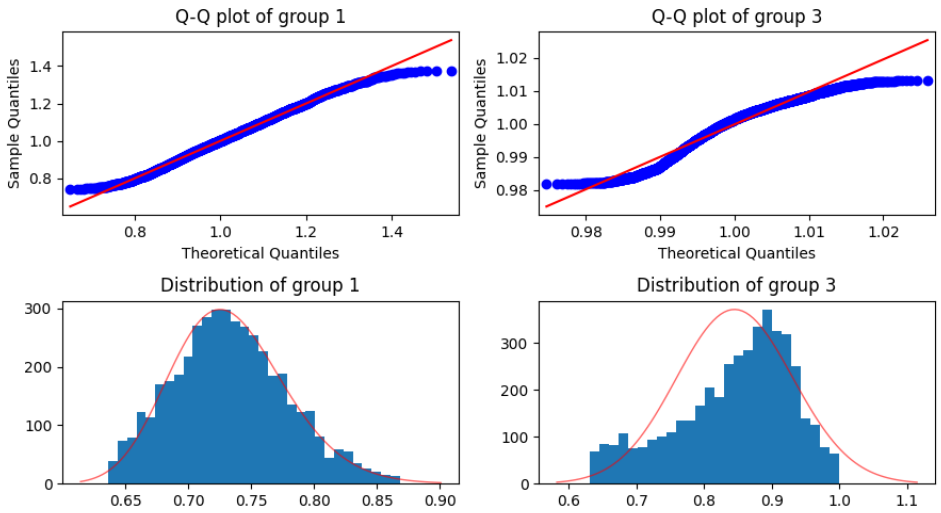

Figure 13: Q-Q plots and distribution plots together with a fitted distribution for the *energy* representation using data from the *lab data set*. The left column contains an example of a group which fit the distribution well and the right column contains an example of one that fits less well.

Figure 13 visualises the fit of two different groups to the lognormal distribution. The groups are based on data from the *lab data set*. Group 1 seems to fit well to the distribution while group 3 seems to fit poorly. This might indicate that group 3 should belong to some other distribution.

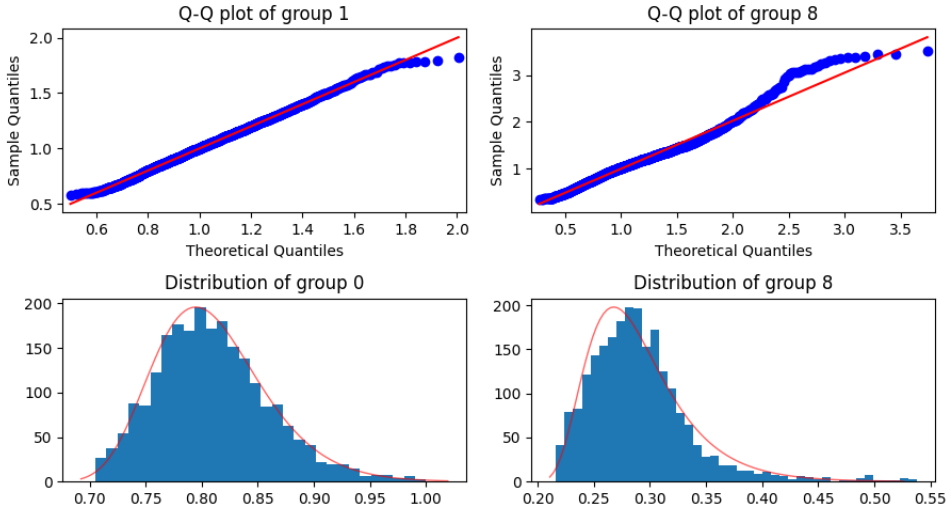

Figure 14: Q-Q plots and distribution plots together with a fitted distribution for the *energy* representation using data from the *live data set*. The left column contains an example of a group which fit the distribution well and the right column contains an example of one that fits less well.

Figure 14 visualises how two groups fit to the lognormal distribution. The groups are based on data from the *live data set*. Group 0 fits well with the distribution while group 8 fits less well. The worst fitting group for this data set, however, has a much better fit than the worst group from the *lab data set* in the right column in figure 13. The poor fit for group 3 in the *lab data set* likely appears as a consequence of deterioration which is not present in the *live data set*.

5.2 Visibility of deterioration in various representations

The *early* and *late* recordings picked earlier will be used to analyse the visibility of deterioration in the various representations.

5.2.1 Energy representation

To more clearly see differences in the time domain, the energy of the recordings will be calculated and displayed.

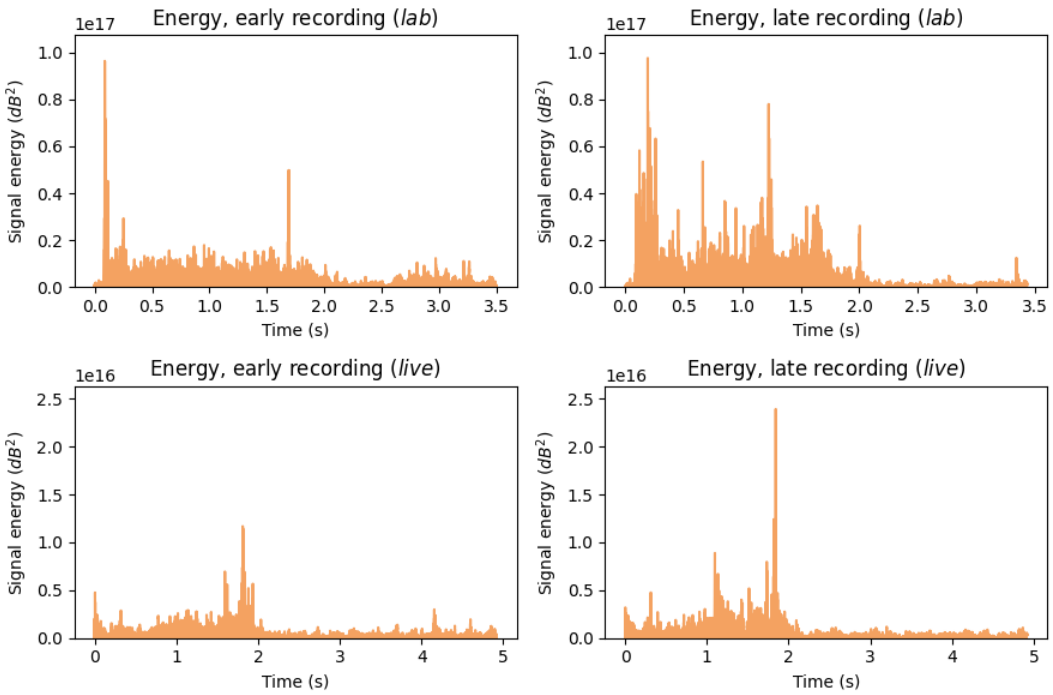

Figure 15: The *energy* representation of the two picked recordings from each real world data set. The top row contain recordings from the *lab data set* and the bottom row from the *live data set*. The y-axis is equal for recordings from the same data set.

Figure 15 shows the *energy* representation of the early and late recordings from each data set. For the *lab data set*, there is a clear increase in energy in the late recording when compared to the early recording. This increase corresponds with an audible rattling in the later recording not present in the early recording and should therefore be regarded as a result of the deterioration sound. The late recording in the *live data set* also has a slight increase in energy which could indicate a shift in the door sound. The difference is, however, too small to regard it as evidence for deterioration.

5.2.2 Welch representation

To analyse how the frequency contents of the door sound develop over time, a frequency spectrum estimated using the Welch method in eq. 5 is analysed. If audible deterioration occurs between the early and late recordings it should also be visible in this representation.

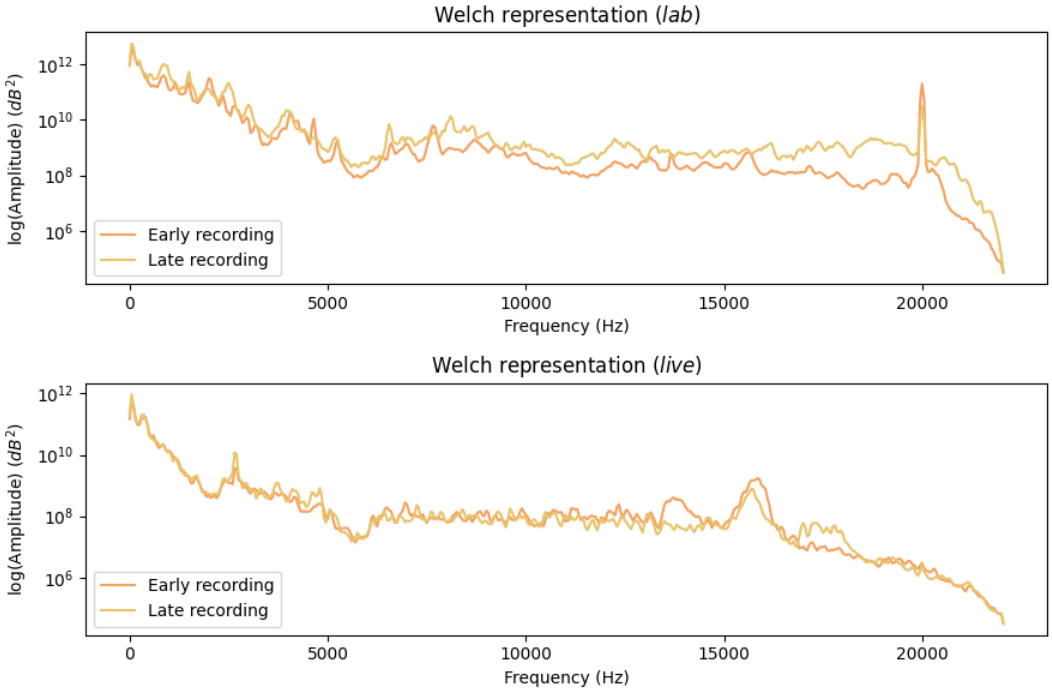

Figure 16: The *Welch* representation of the two picked recordings from both data sets. The top plot contains the recordings from the *lab data set* while the bottom plot contains the recordings from the *live data set*. The y-axis is in logarithmic scale.

Figure 16 shows the *Welch* representation of the late and early recordings from each data set. The two spectra differ more for the *lab data set* than for the *live data set*. For the *lab data set* (upper plot) there seems to be overall more frequency contents in the late recording compared to the early. This is especially visible for the higher frequencies. For the *live data set* (lower plot) the only clearly visible differences are the peaks in the higher frequencies where the late recording has a new peak just below 15 000 Hz but also no peak around 17 500 Hz which the early recording has.

It is hard to draw any conclusions regarding deterioration from these plots but it is clear the overall frequency contents have increased in the *lab data set*, indicating more sounds present in the recording. The differences between the recordings could also be under-exaggerated due to the logarithmic axis. Using a linear axis would require some high pass filtering due to the amount of disturbance in the low frequencies which is why a logarithmic axis is chosen.

5.2.3 Spectrogram representation

Since the deterioration sounds can consist of short transient sounds which span many frequencies it can be interesting to create a spectrogram of the recording as per eq. 9. The spectrogram should visualise these short sounds clearly as vertical lines if they span a large range of frequencies.

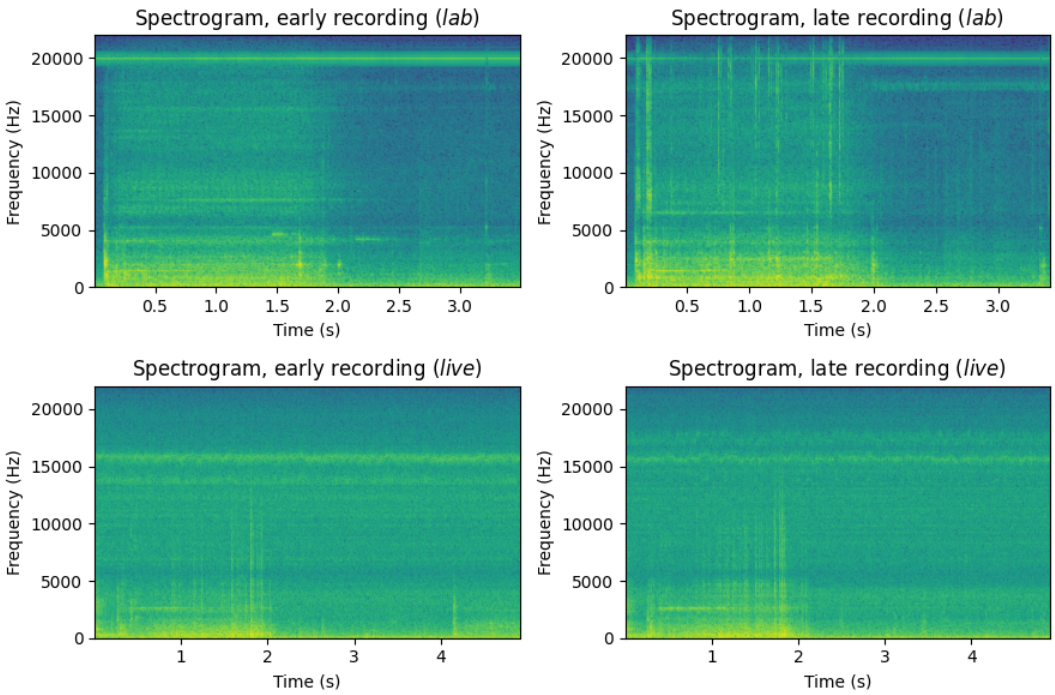

Figure 17: The *spectrogram* representation of the two picked recordings from the data sets using a window size of 10 ms. The top row contains recordings from the *lab data set* and the lower row from the *live data set*. A lighter colour represent a larger amplitude for the specific time and frequency. The colour ranges are equal for recordings from the same data sets.

Figure 17 shows the *spectrogram* representation of two recordings from each data set. The spectrograms for the two data sets both contain vertical lines which indicate short sounds spanning all frequencies. There is, however, a visible difference between the spectrograms for the early and late recordings for the *lab data set* which is not visible when comparing the recordings in the *live data set*. The difference between the recordings in the *lab data set* consist of multiple clearly visible vertical lines in the first half of the recording. These lines corresponds well with the audible rattling in the recording and has therefore likely appeared as a consequence of the deterioration.

5.2.4 Marginal representation

The vertical lines in the spectrogram are highly localised in time and spread out in frequencies. It could therefore be interesting to sum the spectrogram over frequencies which will produce peaks where the vertical lines are. This is done by calculating the time marginals as defined in eq. 11.

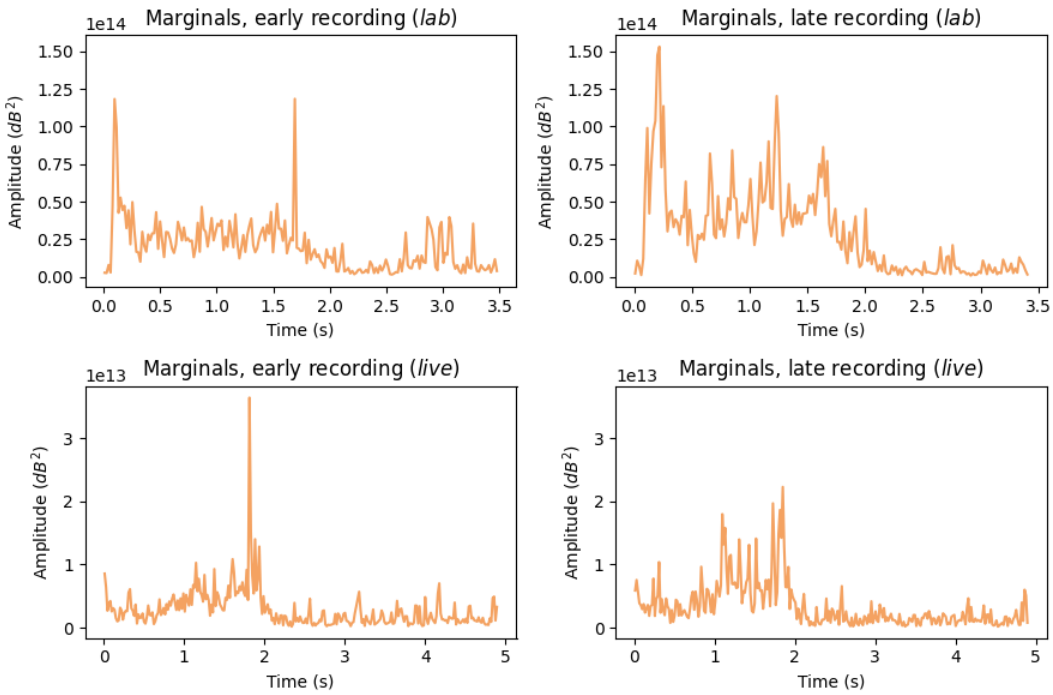

Figure 18: The *marginals* representation of the two picked recordings from the data sets using a window size of 20 ms. The top row contains recordings from the *lab data set* and the lower row from the *live data set*. The y-axis is equal for recordings from the same data set.

Figure 18 shows the *marginals* representation of two recordings from each data set. For the *lab data set* the marginals contain significant differences in the same time region as in figure 17 where the vertical lines in the spectrogram are summed to peaks in the time marginals. Using the same reasoning as for the *spectrogram* representation, these peaks indicate some new sound in the recording. As for the *live data set* the marginals differ more clearly than the *spectrogram* representation which might indicate some change in the door sound. It is possible that a slight rattling sound similar to the one in the *lab data set* has appeared but is too faint to detect by listening to the recording. Unfortunately, this is hard to determine since one would need more data from the door which is not available at the time of writing. This should, however, definitely be further investigated since it shows potential in detecting deterioration before it is audible.

5.2.5 Reassigned spectrogram representation

Due to the uncertainty principle, highly time localised sounds can be smeared over multiple time points in order to achieve adequate frequency resolution. By time reassigning the spectrogram of the recordings using eq. 12 one might achieve even more localised vertical lines where deterioration sounds are present.

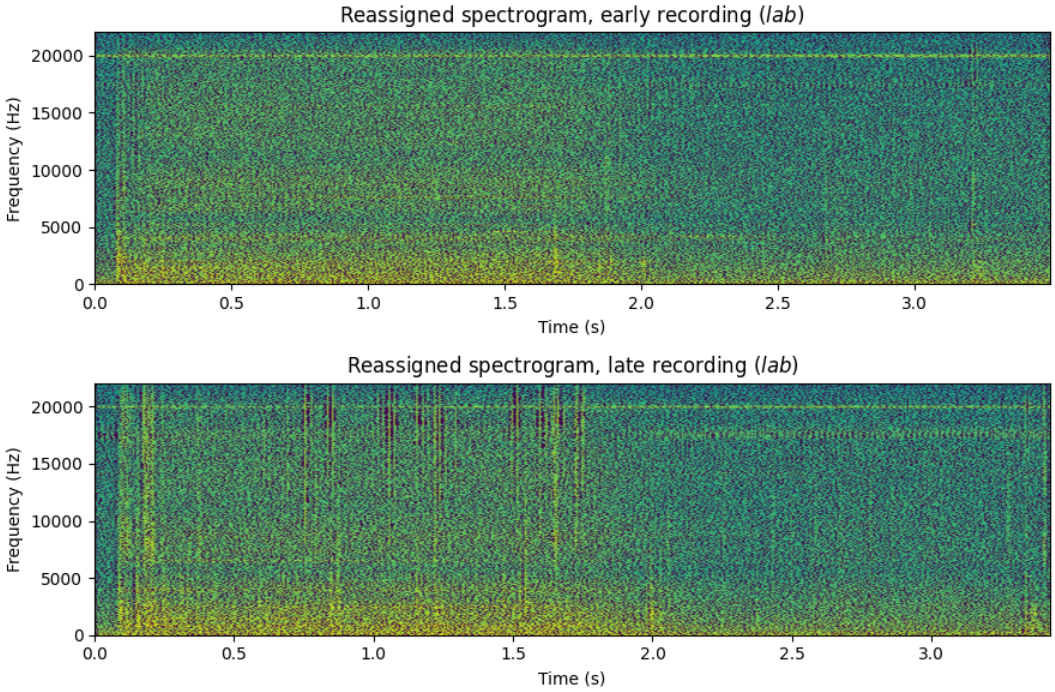

Figure 19: The *reassigned spectrogram* representation of the two picked recordings from the *Lab dataset* using a window size of approximately 30 ms. A lighter colour represent a larger amplitude for the specific time and frequency.

Figure 19 shows the *reassigned spectrogram* representation of two recordings from the *lab data set*. The reassigned spectrogram is noisy but the frequency lines are sharper and more localised compared to the spectrogram in figure 17. The reassigned spectrogram for the late recording contain highly visible vertical lines which are not apparent in the reassigned spectrogram for the early recording. The difference in the reassigned spectrograms is even more apparent than in the normal spectrograms, further indicating some new sound in the late recording.

No reassigned spectrograms is displayed for the *live data set* since they are practically identical for the *early* and *late* recordings.

5.3 Notes on automatic detection of deterioration

Automatic detection of deterioration was done using the *energy*, *Welch* and *marginal* representations. To allow for fair comparison between the performance between the representations, α was set to 1 and the highest z_d^{min} was used as a comparison measure. The same amount of training points (200) was also used in each case to find σ and μ for the base model in eq. 17. Only moving averages of group values will be visualised for each representation and specific scores of interest highlighted later. Since one data set contains an audible difference while the other does not, a performance measure differs between these data sets. Where an audible difference is present (*lab data set*), a high deterioration score indicates good performance while

an overall low deterioration score is preferable when no deterioration can be heard (*live data set*).

5.4 Real world data *with* audible deterioration (*lab data set*)

5.4.1 Grouping the *energy* representation

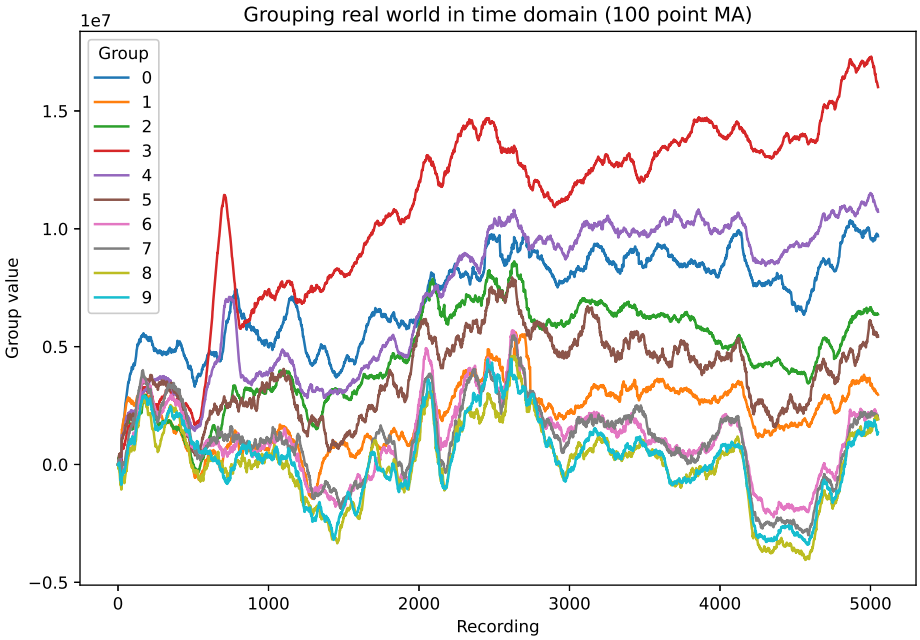

Figure 20: A 100 point moving average of group values over some 5000 recordings from the lab data set. The grouping is done on the *energy* representation with mean as aggregation. Each time series has its values adjusted so that the first value is zero to see difference between groups more easily.

Figure 20 shows the moving averages of groups based on the *energy* representation. The recordings come from the lab data set but every other recording was removed due to limitations in computing memory. The moving averages are calculated with a window of 100 points and each time series is adjusted to start at zero to increase visibility of differences.

Some of the time series revolve around zero over all recordings (groups 6-9) while the other groups see a varying increase in group value with groups 0, 3 and 4 having the largest increases. The fact that the moving averages for some of the groups increase during the process strongly indicates an increase in the sound amplitude of the door. External noise should not cause a steady increase such as this since the amount of noise should increase and decrease throughout the process.

5.4.2 Grouping the *Welch* representation

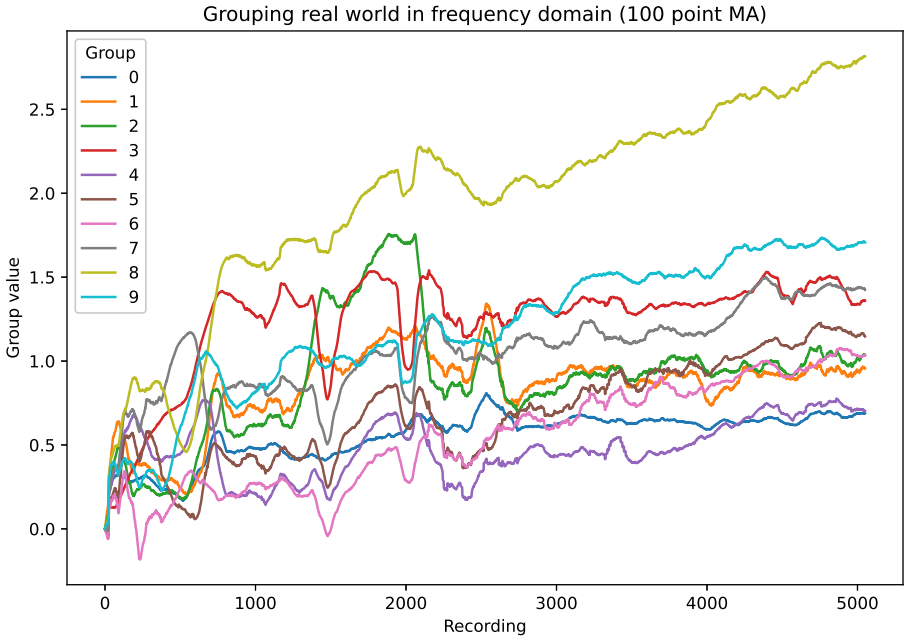

Figure 21: A 100 point moving average of group values over some 5000 recordings. The grouping is done on the *Welch* representation with mean as aggregation. Each time series has its values adjusted so that the first value is zero to see difference between groups more easily.

Figure 21 is processed an visualised similar to figure 20 but with the logarithm of the *Welch* representation. All groups seem to deviate from zero in the end of the process but group 8 has the largest deviation. Group 0 deviates the least but still has a clearly visible deviation. This increase in moving average for all groups reflects the phenomena seen in the spectrogram in figure 17 where the vertical lines present in the late recording span all frequencies and should therefore contribute to an increase in all moving averages. Since group 8 includes the later part of the spectrum in figure 17 it is also expected to have the largest increase.

5.4.3 Grouping the *marginal* representation

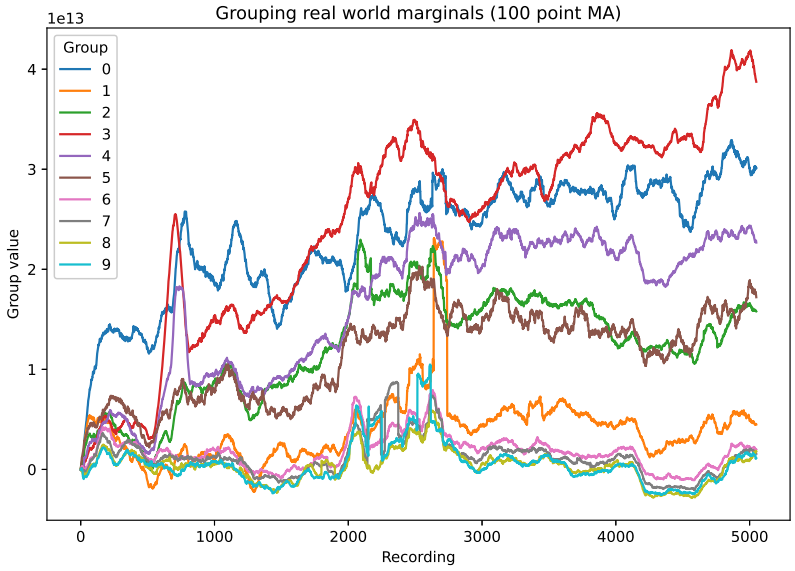

Figure 22: A 100 point moving average of group values over some 5000 recordings. The grouping is done on the *marginal* representation with mean as aggregation. Each time series has its values adjusted so that the first value is zero to see difference between groups more easily.

Figure 22 is processed an visualised similar to figures 20 and 21 but with the *marginal* representation. The moving averages are very similar to the ones for the *energy* representation in figure 20 which is expected since there is some localisation in time contrary to the *Welch* representation where only the frequency contents are analysed. The groups which remain close to zero seem to vary less for the *marginal* representation when compared to the *energy* representation which could indicate that the *marginal* representation is better at suppressing external noise.

5.4.4 Summary

Table 1: Summary of the performance on the *lab data set* where the group with the highest z_d^{min} is presented together with the mean score for the group and the total mean deterioration score of all groups. $\alpha = 1$ in eq. 17 has been used for each domain.

Representation	**Group with highest z_d^{min}**	**z_d^{min} for this group**	**Total mean score**
Energy	3	10 311	976
Welch	8	32 737	3 753
Marginal	3	17 432	2 092

Table 1 summarise the score results from the *lab data set* where an audible difference between early and late recordings exist. The largest z_d^{min} and mean total score is both produced in the *Welch* representation with the *energy* and *marginal* representations having significantly lower scores.

5.5 Real world data *without* audible deterioration (Live data set)

5.5.1 Grouping the *energy* representation

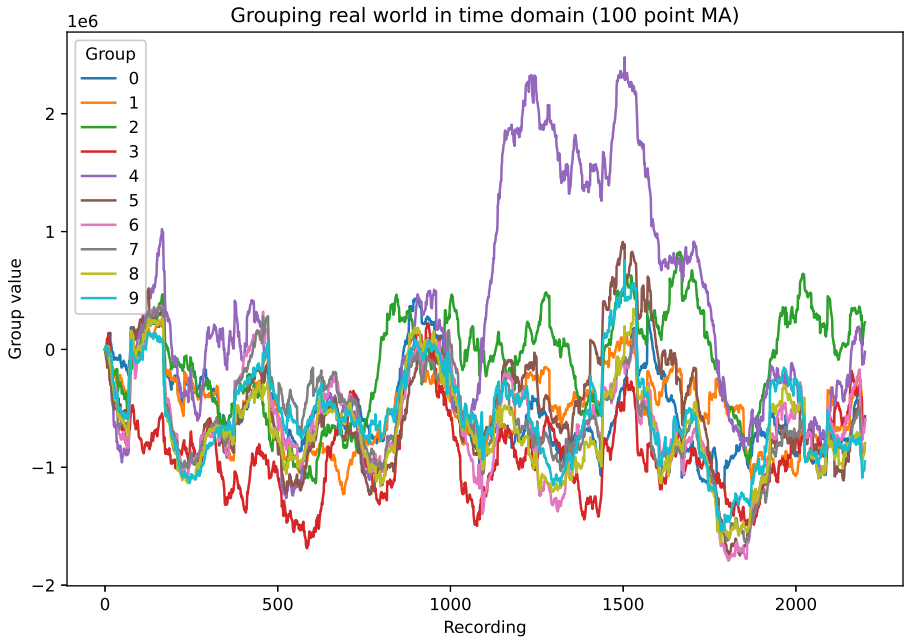

Figure 23: A 100 point moving average of group values over some 2500 recordings from the live data set. The grouping is done on the *energy* representation with mean as aggregation. Each time series has its values adjusted so that the first value is zero to see difference between groups more easily.

Figure 23 shows the moving averages of groups based on the *energy* representation from recordings in the *live data set*. The moving averages are calculated with a window of 100 points and each time series is adjusted to start at zero to increase visibility of differences. All of the time series hover around zero with group 4 deviating temporarily in the later recordings. This increase in moving average is likely not connected to the door since it is only temporary. It could occur due to external noise or as a result of relocation of the micro controller and microphone. External noise would, however, likely affect more than one group since it should happen at random times during the recording. It could also appear as a result of some door mechanism which produces a sound at a similar time point each recording but since the group moving average decreases again after a few hundred recordings it should not represent some permanent

deterioration.

5.5.2 Grouping the *Welch* representation

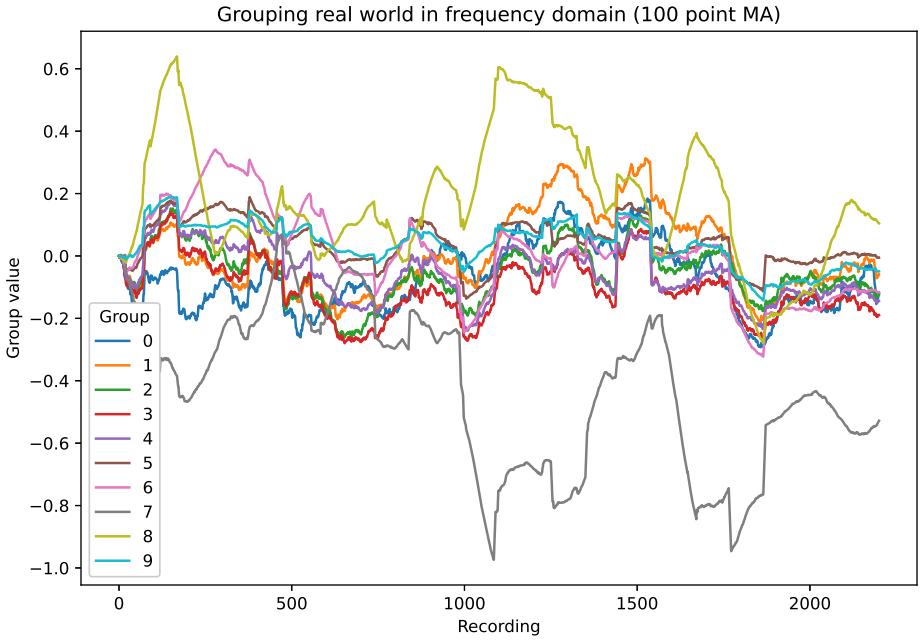

Figure 24: A 100 point moving average of group values over some 2500 recordings. The grouping is done on the *Welch* representation with mean as aggregation. Each time series has its values adjusted so that the first value is zero to see difference between groups more easily.

Figure 24 is processed and visualised similar to figure 23 but with the logarithm of the *Welch* representation. No group seems to deviate from zero significantly apart from group 7 which seems to deviate towards lower group values. Group 8 also seems to occasionally deviate temporarily from zero. Variation for both these groups is only temporary and very sporadic. If the variation was due to deterioration one would expect the variation to occur more slowly and then remain above some new level permanently. As for the *energy* representation, the variation seen in these groups is likely not due to door deterioration. Since the decrease for group 7 occurs throughout all remaining recordings one can not know whether the moving average returned to zero after a while. The variation seems to start after approximately the same number of recordings as for the *energy* representation in figure 23 which could indicate a similar source.

5.5.3 Grouping the *marginal* representation

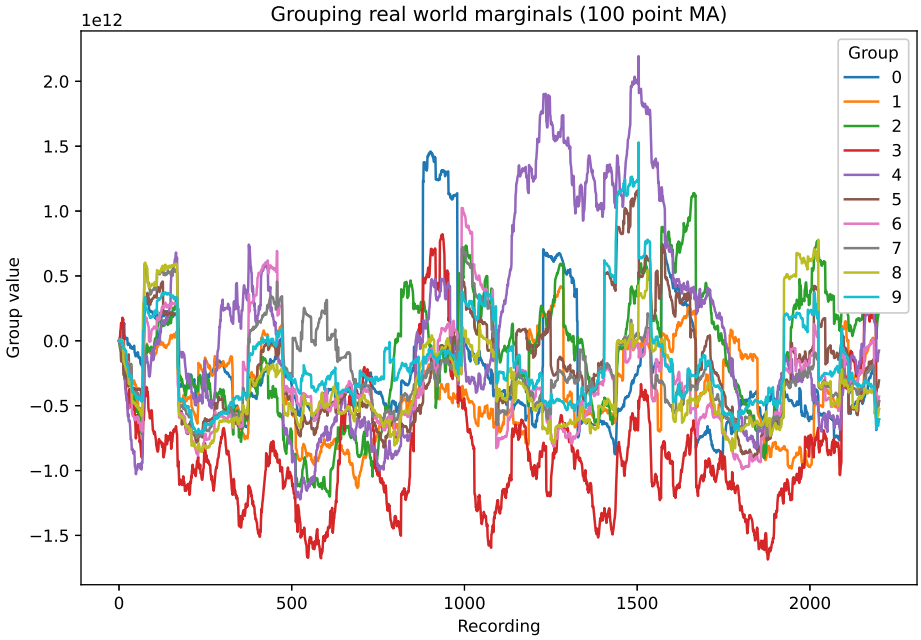

Figure 25: A 100 point moving average of group values over some 2500 recordings. The grouping is done on the *marginal* representation with mean as aggregation. Each time series has its values adjusted so that the first value is zero to see difference between groups more easily.

Figure 25 is processed an visualised similar to figures 23 and 24 but with the *marginal* representation. The group values for this representation behave similar to the group values for the *energy* representation where most groups have values close to zero throughout the process and group 4 deviates temporarily. For the *energy* representation in figure 23, the groups seem to vary together where any increase or decrease sometimes affects all groups. This is not equally apparent for the *marginal* representation where variation in different groups seems more independent. As for the other representations, the moving averages for the *marginal* representation show no sign of permanent deterioration.

5.5.4 Summary

Table 2: Summary of the performance on the *Live data set* where the group with the highest z_d^{min} is presented together with the mean score for the group and the total mean deterioration score of all groups. $\alpha = 1$ in eq. 17 has been used for each domain.

Representation	**Group with highest z_d^{min}**	**z_d^{min} for this group**	**Total mean score**
Energy	-	0.0	0.2
Welch	-	0.0	0.1
Marginal	-	0.0	0.4

Table 2 present the group with the largest mean deterioration score together with the total mean deterioration score for each representation. For all representations, z_d^{min} is zero which means no deterioration is detected in any group. This conforms well with the appearance of the moving averages for this data set where no group deviates from zero significantly.

5.5.5 Comparing performance on the data sets

The *lab data set* was determined to contain a deterioration process which is what the proposed method suggested in all representations as per table 1. The method can therefore be regarded as successful in detecting the change in door sound during the period of collecting the data set.

The *live data set* had no audible deterioration which is also what the method concluded as per table 2. The method can therefore ignore the random variation in the data set and does not falsely regard the door as deteriorated.

5.6 Simulated data

Due to the large amount of simulations run (3 AR noise levels and 3 urban noise levels which results in 9 combinations), the simulation results will only consist of tabulated deterioration score information and only examples of moving averages of the group values. As described in the method formulation, the expected disturbed groups are found and used to decide if the method was successful in detecting the deterioration or if artificial noise resulted in a high score. Since the deterioration is shifted in time, the frequency contents do not change. This results in the same groups deteriorating each run for the *Welch* representation. These groups are 4, 1 and 5 in order of decreasing deterioration. All scores calculated use an $\alpha = 1$ for comparability.

5.6.1 Grouping the *energy* representation

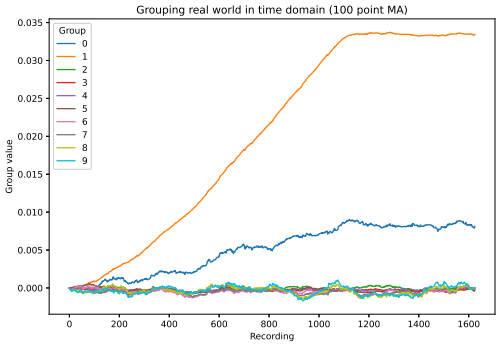

Figure 26: Moving average of an *energy* representation with the lowest amount of urban and AR noise (10 %).

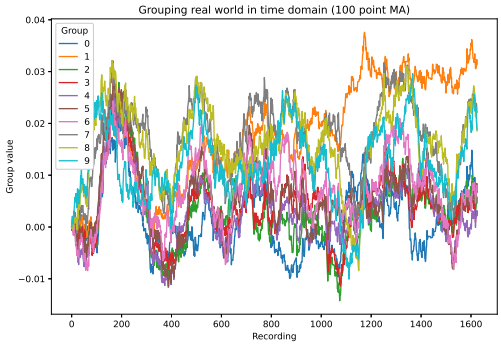

Figure 27: Moving average of an *energy* representation simulation with the highest amount of urban and AR noise (100 %).

Table 3: Success rate of the *energy* representation in detecting deterioration for various noise levels.

Urban \ **AR**	**10%**	**50%**	**100%**
10%	1.00	1.00	1.00
50%	0.41	0.49	0.57
100%	0.00	0.00	0.00

Table 4: z_d^{min} on simulated data with various noise levels using the *energy* representation.

Urban \ **AR**	**10%**	**50%**	**100%**
10%	10696.37	11401.94	13328.87
50%	43.96	48.00	59.06
100%	0.00	0.00	0.00

Figures 26 and 27 show how a simulation with minimal noise differ from a simulation with maximal noise. In the minimal example one can see how the added true deterioration is clearly visible as it shifts two group values upwards without much variation. In the high noise example the true deterioration is barely visible due to the amount of variation from the added noise.

Only a slight increase above the rest of the groups can be seen for group 1 in the end of the simulation. In comparison to the real world moving averages, the low noise example seems to have too little variation due to external noise while the high noise example have perhaps too extreme amounts of noise.

Table 3 displays the performance of the *energy* representation on simulated data where deterioration is present. The representation seems to suffer greatly from the urban noise while the AR noise seems to not affect the performance. The AR noise seems to even improve performance in the medium urban noise level. This is likely due to the increase in training standard deviation σ in eq. 17 which in turn suppress some variation in the data. Consequently, the more consistent change in group values is therefore captured while random noise ignored, leading to a higher success rate. The scores in table 4 also increase for higher levels of AR noise which is not consistent with the previous argument of higher σ. A higher σ should lower the score in eq. 17 but an explanation could be that the added AR noise increases the energy of the recording which then increases x in the equation, leading to a larger score. The fact that AR noise increases performance could be an indication that a larger α is needed to reduce the sensitivity of the method.

Overall, the *energy* representation seems to perform well only in low urban noise levels and is unaffected by AR noise.

5.6.2 Grouping the *Welch* representation

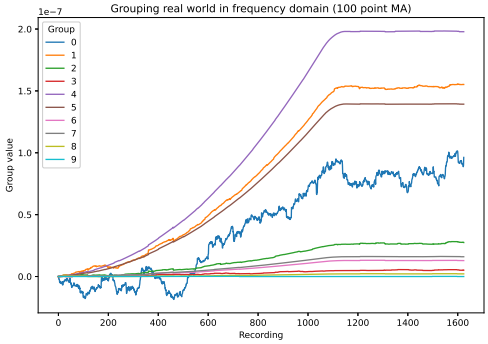

Figure 28: Moving average of a *Welch* representation simulation with the lowest amount of urban and AR noise (10 %).

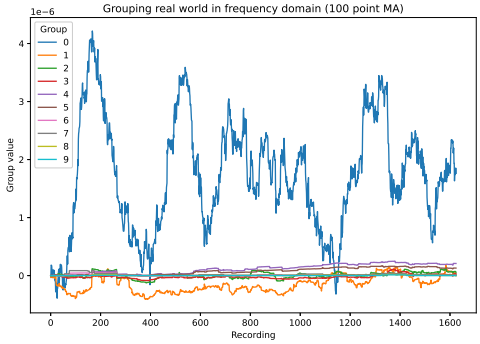

Figure 29: Moving average of a *Welch* representation simulation with the highest amount of urban and AR noise (100 %).

Table 5: Success rate of the *Welch* representation in detecting deterioration for various noise levels.

Urban \ **AR**	**10%**	**50%**	**100%**
10%	0.95	1.00	1.00
50%	0.89	0.95	0.92
100%	0.34	0.43	0.52

Table 6: z_d^{min} on simulated data with various noise levels using the *Welch* representation.

Urban \ **AR**	**10%**	**50%**	**100%**
10%	63619.79	58909.81	48535.69
50%	3456.55	3448.56	3373.54
100%	363.60	308.48	467.92

Figures 28 and 29 show the moving averages of a low noise and high noise example in the *Welch* representation. Similar to the *energy* representation, in the low noise example the deteriorating groups are clearly visible. In the high noise example, group 0 seems to contain extreme amounts of variation in comparison to the other groups. It is likely that the main frequency contents of the added noise is in the low frequency range which is reasonable since part of the noise comes from machinery or vehicles.

Table 5 displays the performance of the *Welch* representation on simulated data where deterioration is present. The representation performs well for both low and medium urban noise and similarly to the *energy* representation, the performance increases with more AR noise. The explanation is likely similar to the *energy* representation where the increase in σ results in less proneness to noise. The scores in table 6 decrease with more AR noise which is what one would expect as per the discussion in the previous chapter (5.6.1). Overall, the *Welch* representation seems to be able to identify a deteriorating door with high accuracy in the low and medium urban noise levels and with lower accuracy for the high urban noise level.

One might expect group 0 in figure 29 to be determined as the most deteriorating group by the method. This is, however, often not the case. Since group 0 is not a part of the true deteriorating groups as described in chapter 5.6, determining this group as the deteriorating group will result in a false positive. The success rate for the high noise scenario in table 5 is

fairly high, indicating that group 0 is not determined deteriorated most of the time. It seems the variance in the group is taken into account which would reduce the score sensitivity for the group.

5.6.3 Grouping the *marginal* representation

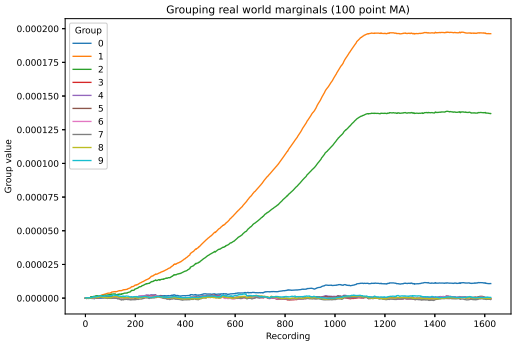

Figure 30: Moving average of a *marginal* representation simulation with the lowest amount of urban and AR noise (10 %).

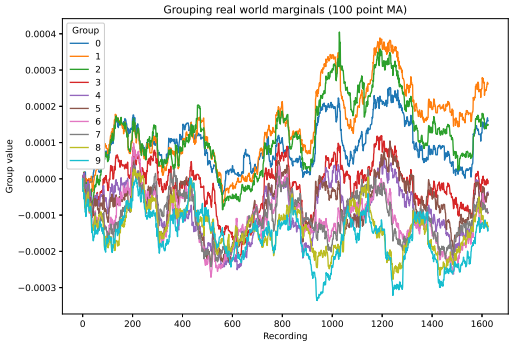

Figure 31: Moving average of a *marginal* representation simulation with the highest amount of urban and AR noise (100 %).

Table 7: Success rate of the *marginals* representation in detecting deterioration for various noise levels.

Urban \ AR	10%	50%	100%
10%	1.00	1.00	1.00
50%	0.97	0.89	0.93
100%	0.00	0.00	0.02

Table 8: z_d^{min} on simulated data with various noise levels using the *marginal* representation.

Urban \ AR	10%	50%	100%
10%	25226.08	24660.67	24655.31
50%	234.76	267.42	243.02
100%	0.00	0.00	0.13

As for the simulations in the *energy* and *Welch* representations, figures 30 and 31 contain amounts of noise which differs greatly. The low noise example seems even less affected by the noise compared to the other representations which could be a result of added deterioration sound spanning many frequencies resulting in larger marginals. In the high noise example the same groups can be seen growing large in the end of the simulation. They, however, only barely stand out from the external noise.

Table 7 shows the performance of the *marginal* representation on simulated data where deterioration is present. The representation performs well for low and medium urban noise and seems unaffected by the AR noise which neither increases nor decreases performance by any significant amount. The *marginal* representation seems to even perform slightly better than the *Welch* representation on the medium urban noise environment. Table 8 shows the corresponding scores where a similar conclusion can be drawn.

5.6.4 Summary of performance on simulated data

Overall, the *Welch* representation seems to perform the best among the three with the *marginal* representation performing slightly worse and the *energy* performing well below the other two. The only representation with an ability to detect deterioration in the high urban noise level is the *Welch* representation where, in the best scenario, half of the runs are correctly labelled.

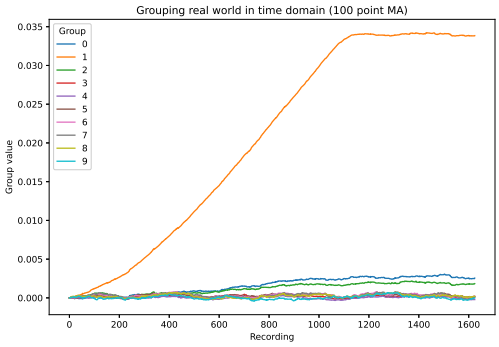

Figure 32: Moving average of an *energy* representation simulation with the lowest amount of urban noise (10 %) and highest amount of AR noise (100 %).

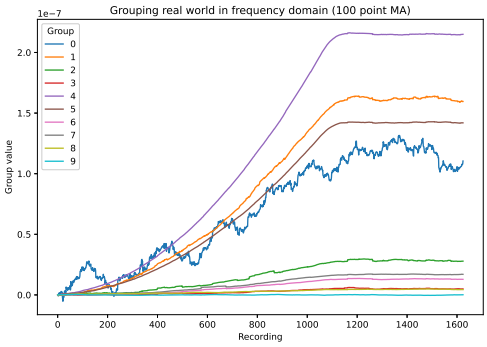

Figure 33: Moving average of a *Welch* representation simulation with the lowest amount of urban noise (10 %) and highest amount of AR noise (100 %).

Using simulated data allows for customisation of the noise levels and more control over the deterioration process. The representations have varying performance with the *Welch* representation having the best performance. It seems the frequency spectrum has some resilience to both the AR and urban noise.

Figures 32 and 33 visualise a simulation moving average with low urban noise and high AR noise for the *energy* and *Welch* representations. For the *energy* representation the added noise

is barely visible at all while slight variation in group 0 can be seen in the *Welch* representation. The AR noise has a peak in $f = 350Hz$ which concentrates the noise to the first group in the *Welch* representation which is why the noise is visible in that representation. But overall, the AR noise seems to have little to no effect on the simulations even though figure 11 gives the impression of high noise levels. For the *energy* representation the noise is likely suppressed due to the averaging in the group. Since the noise is added uniformly to the whole recording, it simply averages to approximately zero in each group. For the *Welch* representation the averaging described in eq. 5 can be the reason. The method is designed to try to reduce variance by smoothing and seems to work as intended. Evidently, the representations are not particularly susceptible to uniform noise. This is also reflected in the performance tables where the added AR noise seems to have no impact on performance.

6 Discussion

6.1 Manual detection of deterioration

For multiple representations the expected deterioration in the *lab data set* is visible. Since the change in door sound is actually audible, one can verify that the new patterns in the various representations actually correspond with deterioration. For the *live data set* it is harder to give a verdict. The writer concluded, by listening to the recordings, that no deterioration had occurred and in most representations only faint signatures of change are visible. Since there is no clear definition of how audible a deterioration needs to be it boils down to a subjective decision by the writer. This strongly increases the overall proneness to error in the manual analysis since a reader of this report might suggest that deterioration can be seen in the representations. This problem could and should be fixed by creating a better definition of what counts as a deterioration and where the line between normal and faulty operation should be drawn.

6.2 Automatic detection of deterioration

The automatic detection of deterioration is successful in low to medium levels and fails for higher noise levels. A number of parameters which could improve performance are not tested in this report. For example, a larger amount of groups could likely increase the performance since the deterioration sounds seem highly localised. This would, however, result in more calculations and is therefore a trade-off. Different choices of aggregation can also be tested together with other choices of α in eq. 17.

When comparing the deteriorated groups in table 1 with the distribution plots in figure 13 one can see that the group that fit the worst to the distribution (group 3) is also the most deteriorating group according to the analysis. One could therefore perhaps exploit this to further improve the method. Overall, the data is determined to be lognormally distributed but the scoring system presented does not take this into account. Adjusting the score based on this should therefore also be examined.

6.3 Difference between real and simulated data

The environment in which the method is supposed to operate is highly unpredictable. A single door in an office environment can be subject to a range of different sounds such as voices, machines, alarms or even other doors nearby. Including doors at other locations such as train stations or shopping malls further increases the variety in the ambient sounds experienced. Such an environment is hard to simulate since one would need a large amount of varying variables together with countless amounts of simulation runs to cover a broad amount of cases. It is therefore likely that the simulated data is not representative of how the average real world deterioration process behaves. The simulated data created in this report only includes one kind of deterioration sound (squeaking) and therefore does not cover a broad range of possible

deterioration sounds. Instead, focus has been put on simulating a surrounding environment with varying amounts of noise. Both entirely artificial signals in forms of AR processes was added but also actual recordings of urban noise with varying strengths which might provide a more realistic ambient environment.

When comparing moving averages from simulated data to the real world results there are some differences. Mainly, the real world data seems to contain slower variation where the moving average changes more slowly, indicating some correlation between recordings not captured in the simulated data. This could, for example, be the fact that recordings during a specific time period of the day (or day of the week and so on) contain more of a specific type of noise. Temporary deviations such as the one seen for group 4 in figure 23 is not produced by the simulation either. Adequate testing of how those types of variation affect the method can therefore not be tested with the simulations performed in this report. A relevant question to ask is also whether this variation is due to an actual external noise source or due to repositioning of the microphone. If the latter is the case then simulations should not include such variation unless repositioning of the microphone is something which needs to be accounted for when performing the analysis.

The quality of the *live data set* should therefore be questioned since the recording is triggered by a lock sound and the position and orientation of the microphone changed during the data collection. This data set could therefore contain invalid recordings which are triggered by ambient sounds, or missing recordings when the triggering failed. The slight orientation changes might also impact the recordings which could be why some groups for this data set deviate sporadically from zero.

6.4 Comparing speed of calculation

The quickest representation to calculate is the *energy* representation since it only requires taking the absolute value and element-wise squaring. It, however, performs the worst out of the three representations tested. The *marginal* representation is most costly to calculate but provides a better performance than the *energy* representation while being outclassed by the *Welch* representation. The *Welch* representation is also cheaper to compute than the *marginal* representation which perhaps constitute a good balance between performance and computational cost. Since no practical measurement of the computational needs was done, one can not conclude if the method can be run on a micro controller. This is something that should be done to assess the practical viability of the method.

7 Conclusion

The data representations used in this report can all capture signs of deterioration when it is audible. Since no exact definition of deterioration is stated it is hard to deduce whether non-audible deterioration could be detected using any of the representations. Automatic detection of deterioration was attempted using three representations and seems possible in each representations even if the *Welch* representation seems to perform the best. The efficiency of the method is theoretically fast but no effort has been put into practically timing the implementation. It is therefore difficult to decide if the analysis can be run on a micro controller without further investigation. The simulated data produced is useful when doing an initial benchmark of the method. The amount of possible variation in the real world data is, however, hard to produce in a simulation which is why more effort should be put into collecting real world data for further testing.

7.1 Future work

This being a report with high emphasis on developing a method it leaves several question marks which should be further researched. Most importantly, more testing needs to be done to assess the potential in the method. More data needs to be collected to reduce the need for reliance on mostly simulations as a measure of the method performance. The data collected also needs to be diverse and include multiple deteriorating and non-deteriorating doors to cover a larger range of possibilities. More advanced simulation methods such as Generative Adversarial Networks (GANs) could also be tested to see if more realistic data can be generated.

No effort has been put into practically timing the execution of the algorithms proposed in this report which is highly relevant. Only the theoretical time complexity has been presented. The time complexity should give an idea on the execution time but will not reflect the effective execution time since mechanisms such as vectorisation and other factors are not put into account. If the execution time is measured then future optimisation of the code is also easier. Also related to performance is down-sampling of the recording which likely could be done to further increase the speed of the method.

Only one scoring systems is examined in this report. There are, however, likely better alternative scoring systems which provide a better feedback on the amount of deterioration. A score based on a combination of the representations could also be constructed and tested. The report does not provide a clear way of interpreting the scores either but only states that a high score should indicate deterioration. Taking a decision based on the score should therefore also be examined.

As a way of reducing the variance in the data and avoiding erroneous results, an attempt to detect noisy recordings and ignore them could also be tested. This might allow a more sensitive scoring system since one could rely more on the recordings representing the state of the door and not external noise.

References

[1] R. Dindorf. "Estimating Potential Energy Savings in Compressed Air Systems". In: *Procedia Engineering* 39 (2012). XIIIth International Scientific and Engineering Conference Hermetic Sealing, Vibration Reliability and Ecological Safety of Pump and Compressor Machinery - HERVICON-2011, pp. 204–211. ISSN: 1877-7058. DOI: https://doi.org/10.1016/j.proeng.2012.07.026. URL: https://www.sciencedirect.com/science/article/pii/S1877705812024125.

[2] Z. Feng, M. Liang, and F. Chu. "Recent advances in time–frequency analysis methods for machinery fault diagnosis: A review with application examples". In: *Mechanical Systems and Signal Processing* 38.1 (2013). Condition monitoring of machines in non-stationary operations., pp. 165–205. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2013.01.017. URL: http://www.sciencedirect.com/science/article/pii/S088832701300071X.

[3] X. Chen and Z. Feng. "Iterative generalized time–frequency reassignment for planetary gearbox fault diagnosis under nonstationary conditions". In: *Mechanical Systems and Signal Processing* 80 (2016), pp. 429–444. ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2016.04.023. URL: http://www.sciencedirect.com/science/article/pii/S0888327016300620.

[4] M. Gan, C. Wang, and C. Zhu. "Fault feature enhancement for rotating machinery based on quality factor analysis and manifold learning". In: *Journal of Intelligent Manufacturing* 29.2 (July 2015), pp. 463–480. DOI: 10.1007/s10845-015-1125-6. URL: https://doi.org/10.1007/s10845-015-1125-6.

[5] Y. Zhang et al. "An enhanced convolutional neural network for bearing fault diagnosis based on time–frequency image." In: *Measurement* 157 (2020). ISSN: 0263-2241. URL: http://ludwig.lub.lu.se/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edselp&AN=S0263224120302050&site=eds-live&scope=site.

[6] C. Vununu et al. "Sound Based Machine Fault Diagnosis System Using Pattern Recognition Techniques". In: *Journal of Korea Multimedia Society* 20 (Feb. 2017), pp. 134–143. DOI: 10.9717/kmms.2017.20.2.134.

[7] M. Hajji et al. "Multivariate feature extraction based supervised machine learning for fault detection and diagnosis in photovoltaic systems." In: *European Journal of Control* 59 (2021), pp. 313–321. ISSN: 09473580. URL: http://ludwig.lub.lu.se/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edo&AN=150184249&site=eds-live&scope=site.

[8] S. U. Jan, Y. D. Lee, and I. S. Koo. "A distributed sensor-fault detection and diagnosis framework using machine learning." In: *Information Sciences* 547 (2021), pp. 777–796. ISSN: 0020-0255. URL: http://ludwig.lub.lu.se/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edselp&AN=S0020025520308422&site=eds-live&scope=site.

[9] M. Sandsten. "Time-frequency analysis of time-varying signals and non-stationary processes". In: *Lund University* (2016).

[10] E. W. Weisstein. *Discrete Fourier Transform*. MathWorld–A Wolfram Web Resource. May 2021. URL: https://mathworld.wolfram.com/DiscreteFourierTransform.html (visited on 05/12/2021).

[11] J. Dongarra and F. Sullivan. "Guest Editors Introduction to the top 10 algorithms". In: *Computing in Science Engineering* 2.1 (2000), pp. 22–23. DOI: 10.1109/MCISE.2000.814652.

[12] M. Heideman, D. Johnson, and C. Burrus. "Gauss and the history of the fast Fourier transform". In: *IEEE ASSP Magazine* 1.4 (1984), pp. 14–21. DOI: 10.1109/MASSP.1984.1162257.

[13] J. W. Cooley and J. W. Tukey. "An Algorithm for the Machine Calculation of Complex Fourier Series". In: *Mathematics of Computation* 19.90 (1965), pp. 297–301. ISSN: 00255718, 10886842. URL: http://www.jstor.org/stable/2003354.

[14] J. O. Smith. *Mathematics of the Discrete Fourier Transform (DFT)*. 2007. URL: http://ccrma.stanford.edu/~jos/mdft/ (visited on 05/12/2021).

[15] M. Bartlett. "Smoothing Periodograms from Time-Series with Continuous Spectra". In: *Nature* 161 (1948), pp. 686–687.

[16] P. Welch. "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms". In: *IEEE Transactions on Audio and Electroacoustics* 15.2 (1967), pp. 70–73. DOI: 10.1109/TAU.1967.1161901.

[17] G. Lindgren, H. Rootze'n, and M. Sandsten. *Stationary stochastic processes for scientists and engineers*. CRC Press, 2013. ISBN: 978-1-4665-8618-5.

[18] J. Kleinberg and E. Tardos. *Algorithm Design*. Pearson Education, 2022. ISBN: 9780132131087. URL: https://books.google.se/books?id=GORecgAACAAJ.

[19] S. van der Walt, S. C. Colbert, and G. Varoquaux. "The NumPy Array: A Structure for Efficient Numerical Computation". In: *Computing in Science & Engineering* 13.2 (Mar. 2011), pp. 22–30. ISSN: 1521-9615. DOI: 10.1109/mcse.2011.37. URL: http://dx.doi.org/10.1109/MCSE.2011.37.

[20] Z. Wang et al. "The era of silicon MEMS microphone and look beyond". In: June 2015, pp. 375–378. DOI: 10.1109/TRANSDUCERS.2015.7180939.

[21] *I2S Output Digital Microphone*. SPH0645LM4H-B. Rev. B. knowles. 2015. URL: https://cdn-shop.adafruit.com/product-files/3421/i2S+Datasheet.PDF.

[22] E. S. Page. "Continuous Inspection Schemes". In: *Biometrika* 41.1/2 (1954), pp. 100–115. ISSN: 00063444. URL: http://www.jstor.org/stable/2333009.

[23] J. Salamon, C. Jacoby, and J. P. Bello. "A Dataset and Taxonomy for Urban Sound Research, 22nd ACM International Conference on Multimedia". In: (Orlando USA). Nov. 2014.