Aalto University
School of Engineering

Ludo Meijs

Signal processing techniques related to laboratory measurements of ultrasonic S-wave velocities in rocks

Master's thesis

European Mining Course

Supervisor: prof. Mikael Rinne

Advisor: MSc Risto Kiuru

Otaniemi, Finland

13.9.2019

This page is intentionally left blank

Author Ludo Meijs

Title of thesis Signal processing techniques related to ultrasonic laboratory measurements of S-waves in rocks

Master programme EMC **Code** ENG3077

Thesis supervisor prof Mikael Rinne

Thesis advisor(s) MSc Risto Kiuru

Date 01.09.2019 **Number of pages** 121 **Language** English

Abstract

This thesis aims to evaluate the effect of signal processing techniques related to ultrasonic laboratory measurements of shear waves. Compressional and shear wave velocities play an important role in static elastic rock deformation behaviour estimation. Onsets of compressional and shear wave signal have to be determined in order to calculate the corresponding wave propagation velocity. Onset estimation by automation is especially problematic in shear wave signals due to noise caused by reflections and refractions, which results in inaccurate onset estimations and, therefore, requires manual onset picking which is time-inefficient and, hence, costly.

Akaike Information Criterion (AIC) is the automated picking method applied to the ultrasonic signals in this thesis. By efficiently processing shear wave signals it was tried to optimize the results of the AIC. Ten processing techniques from biomedical engineering, statistical signal processing, audio and speech processing and RADAR applications were thoroughly researched. Their applicability to ultrasonic signals was reasoned based on literature. Six applicable signal processing techniques were eventually applied to 30 synthetic and 30 real ultrasonic signals. The mean and standard deviation of the error related to onset estimation before and after processing was used for evaluation. Visual comparison before and after processing was also executed to evaluate the visual impact of the processing techniques.

Results showed that only a Butterworth high-pass filter visually enhances synthetic and real ultrasonic signals and improves the mean and standard deviation with respect to onset estimation. A Chebyshev high-pass filter also improved onset estimation results, but deteriorated the visual interpretation of the time signals. A simple amplitude filter unexpectedly provided the best results with respect to onset estimation.

It is concluded from this studies that onset estimation by AIC can be improved by application of related signal processing techniques. This could be beneficial in estimation static deformation behaviour. Potential room for improvement is found within parameter optimisation and synthetic signal production.

Keywords Ultrasonic laboratory measurements – Shear waves – Signal processing techniques – Onset estimation - AIC

Preface

This thesis will address the effect of the signal processing methods applied to ultrasonic laboratory signals. It is part of Geophysical and Geochemical Methods for Stope Design project (GAGS). Main goal of the GAGS project is to improve mine planning by novel utilization of geophysical and geochemical methods to characterise rock properties for mine planning and geometallurgical purposes. This thesis should contribute to the doctoral thesis of R. Kiuru who aims to establish a link between in-field measurements, destructive laboratory measurements and non-destructive laboratory measurements of rocks.The thesis was written to fulfill the requirements of a thesis at Aalto University School of Engineering for the Department of Civil Engineering.

The around signal processing was practically unknown to me as a student of mining engineering, which made doing research particularly difficult at the start. Towards the end I could speak with confidence about signal processing which reflects progression. For that I have to thank my advisor MSc R. Kiuru who excellently guided me on a daily basis and never missed any detail. My supervisor prof. M. Rinne who provided structural guidance and made this signal processing thesis fit into the world of mining. And, lastly, my colleagues for on and off the topic discussions and mind-clearing distractions.

I hope you enjoy your reading.

Ludo Meijs

Nijmegen, September 7th, 2019

Table of contents

List of tables

List of figures

List of abbreviations

P-wave	Primary or Compressional wave
S-wave	Secondary or Shear wave
λ	Lamé's first parameter
μ	Lame's second parameter
S1	First polarisation of a S-wave
S2	Second polarisation of a S-wave
AIC	Akaike Information Criterion
f	Frequency
Hz	Herz
ω	radial frequency
rad	radials
s	second
t	time
t_0	time instance zero
a	scaling factor
x(t)	input signal as function of time
y(t)	output signal as function of time
v_s	shear wave velocity
v_p	compressional wave velocity
σ	Poisson's ratio
p.p.	percent point
L	length
F2F	face-to-face
A	amplitude
mV	milli Volt

γ	decay constant in reciprocal time units
mm	millimeter
k	current state (as subscript)
k-1	previous state (as subscript)
x	state variable of a system
$\hat{x}$	best estimate state variable of a system
P	covariance matrix
Var	state variable
Σ	covariance
F	prediction matrix
B	control matrix
μ	control vector
Q	external influences
m	moving average window
k	iteration parameter
a	Chebyshev polynomial
EM	electromagnetic
RADAR	radio detecting and ranging
$H(j\omega)$	frequency response as function of complex frequencies
c	cut-off (as subscript)
p	passband (as subscript)
s	stopband (as subscript)
n	filter order
G	gain
ε	ripple factor
T_n	Chebyshev polynomial of order n

θ	phase of a signal
$\underline{y}$	signal
$\underline{d}$	noise
$\underline{n}$	clean speech
Γ	power spectrum
$\hat{\Gamma}$	estimator of power spectrum
PDS	Power Density Spectrum
w	time window
MSE	Mean Squares Estimate
$h^{\dagger}$	impuls respons
s	input signal
$\hat{s}$	estimator of input signal
stPDS	short term Power Density Spectrum
TDHS	Time Domain Harmonic Scaling
α	compression factor
N_c	crossfade length
µs	microsecond
KZ	Kolmogorov-Zurbenko
KZA	Kolmogorov-Zurbenko Adaptive
St. dev.	Standard deviation
F_{pc}	cut-off frequency of the passband
F_{sc}	cut-off frequency of the stopband

1. Introduction

This section introduces the reader to the relation between mining, stope design and ultrasonic signals. It will then go deeper into rock testing methods, primary and secondary waves and ultrasonic laboratory methods. This will provide a sufficient amount of background information to understand the problem statement. Then the scope of the thesis is discussed where a goal, research questions and hypothesis are defined. The last paragraph is dedicated to general background information on signal processing.

1.1 Mining, stope design and ultrasonic signals

Stopes are underground cavities created to extract ore from an underground ore zone. Purpose of a stope is to extract the optimum amount of valuable material from the subsurface and to remain stable at the same time. Stopes designed too large will collapse, diluting valuable material with waste. Stopes designed too small leave behind valuable material. Stope geometry is dictated by rock mass quality and rock deformation moduli.

Four main rock deformation moduli are: Shear modulus, bulk modulus, Young's modulus and Poisson's ratio (see Section 1.2). These moduli can be determined by dynamic or static testing. The static approach requires the propagation velocity of primary and secondary waves (see Section 1.3) and the density of the examined rock. Waveforms with a frequency higher than 20 kHz are categorised as ultrasonic waves (The Gale Enceclopedia of Science, 2002). Those are measured in the laboratory on rock samples (see Section 1.3). Via mathematical relationship between rock moduli and the previously mentioned parameters the deformation moduli can be estimated. This thesis will focus on primary and secondary wave propagation of ultrasonic signals as part of geophysical measurements on rocks.

1.2 Rock testing methods

In general, measurements on rocks can be executed in three different ways. The first type of methods are in-situ measurements. Those measurements are performed in a borehole and they measure the integrated rock mass. The integrated rock mass is different from the intact rock mass, because it includes fractures that are present in the field and affect the measurements (Kahraman, 2002; Schoenberg & Sayers, 1995). The second type of methods are destructive laboratory methods. They measure rock mechanical properties and are used for dynamic deformation moduli estimation (Sabbagh, et al., 2002). Dynamic refers to the transient nature of loading (Kaiwen & Wei, 2015). They are labour-intensive, slow, and, hence, expensive. The type of methods are is non-destructive laboratory methods. They measure geophysical properties which can be used for determination of static deformation moduli (Naser, 2004). They are repeatable, relatively less labour-intensive and more rapid than destructive methods. Consequently, they are cheaper. Static and dynamic deformation moduli describe the same deformational behaviour, but tend to give different results (Sabbagh, et al., 2002).

1.3 Primary and secondary waves

Acoustic wave velocities are an important tool for estimating elastic rock properties. There is a direct relationship between P- and S-wave velocities and the elastic moduli that express the deformability of a rock. The main difference between a P- and S-wave is supported by Fig. 1.

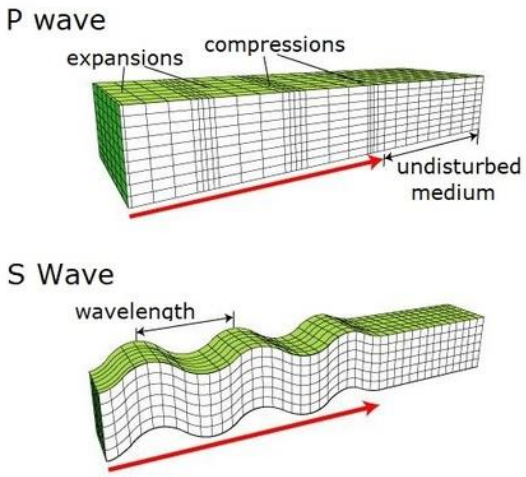


Fig. 1: P-wave (top) and S-wave (bottom). Red arrow indicates direction of propagation.

A compressional wave has an expansion phase and a compression phase. The particle motion and wave propagation direction are in the same direction. A shear wave is a wave due to shear movement of a particle. Shear waves can only exist in solids, due to the bond between particles in the lattice. Molecules in a fluid follow a random motion and do not have a lattice structure and therefore a particle cannot exhibit shear stress on an adjacent particle in the lattice. The movement of a particle in a shear wave is perpendicular to the wave propagation direction. This is typical for a shear wave. Compression, extension and shearing are all the basic deformation possibilities that rock has and therefore a combination of both waves gives an indication of the rock elastic properties. For every modulus there is a separate formula. The focus of this thesis, however, is not to determine rock elastic properties, but to optimize arrival time estimation of a shear wave through a rock sample (See Section 1.4). A mathematical notation and description of the elastic moduli is therefore not included.

1.4 Ultrasonic laboratory measurements

P-and S-wave velocities are important parameters for static moduli estimation assuming a medium to be isotropic, homogeneous and linear. P-and S-wave velocity allow for calculation of Lamé's first and second parameter λ and μ. Using Newton's second law of motion and Hooke's Law the Young's Modulus and Poisson's ratio can be calculated, which are two of of the four main elastic deformation moduli. A comprehensive mathematical derivation is provided by (Yilmaz, 2001; Langenberg, et al., 2012). This thesis will focus on ultrasonic laboratory measurements of primary and secondary ultrasonic waves in rocks.

A cylindrical shaped rock sample is placed between two transducers. A transducer converts energy from one form to another (Agarwal & Lang, 2005). In perspective of ultrasonic measurements a transducer converts an electrical pulse to a mechanical wave and vice versa. The transmitting transducer can send a P-wave or a S-wave wave. Particle motion of a S-wave is perpendicular to the propagation direction. (Poisson, 1831). Two polarizations of S-waves can be transmitted and recorded. Due to anisotropy velocities of different polarizations can differ, see Fig. 2. Even though isotropy is assumed in order to calculate the static elastic moduli in reality a rock is anisotropic and heterogeneous due to surface boundaries, foliation, density variations, cracks, pore filling etc. The two polarisations transmitted in laboratory measurements are named S1 and S2 which is contracted from S-wave one and S-wave two. S1 is always perpendicular to S2.

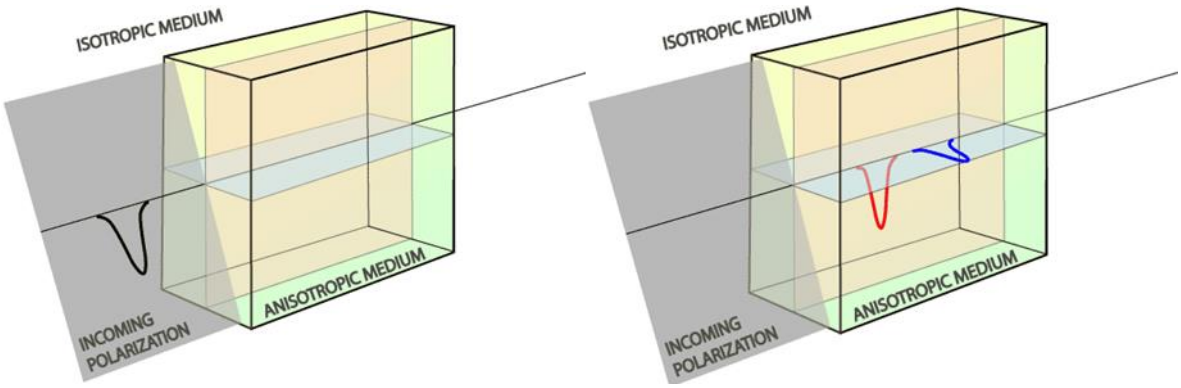


Fig. 2: Incoming S-wave (left) and two polarisations of the same S-wave in an anisotropic medium (right). (Garnero, 2010)

A wave is transmitted by the piezo crystal located in the transducer. It propagates through multiple interfaces e.g. piezo crystals-transducer interface, transducer-couplant interface, couplant-rock sample interface, and interbedding. See Fig. 3.

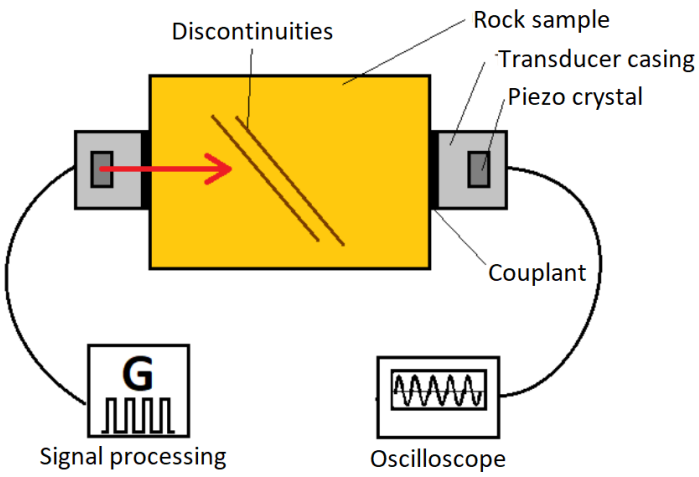


Fig. 3. Schematic drawing of a laboratory setup for ultrasonic testing of a rock sample. Wave propagation highlighted in red. (Idea from: (Alejano, et al., 2018))

The time it takes for a wave to travel through a medium is called the travel time. Dividing the sample length by the travel time gives the propagation velocity and that is the parameter required for static elastic moduli calculations. This calculation is easy to execute and quick.

1.5 Problem statement

The problem with body wave velocity estiation lays within the physical aspect of wave propagation through boundaries. As a wave interacts with an interface it is converted into different waveforms. For example, a S-wave encountering the discontinuities converts to a P-and S-wave refraction and a S-wave reflection. P-waves always travel faster than S-waves and therefore the P-wave onset is easily determined, because it is the first major amplitude displacement measured.

Because shear waves are slower than P-waves, P-waves get ahead of the initial S-wave and refract into shear wave polarisations again. Those are of much smaller amplitude than the initial S-wave, but arrive before the initial S-wave at the receiver. This causes the initial S-wave to be buried in noise due to refractions and reflections. An example of a S-wave is shown in Fig. 5.

Akaike Information Criterion (AIC) picking method (Sedlak, et al., 2008; St-Onge, 2011) can be applied to find the onset in ultrasonic P-waves and S-waves (See 2.2.2 for more information about the AIC). The problem is that the AIC often picks incorrect shear wave onsets, because they are buried in noise, see Fig. 6. The challenge is to reduce the noise and improve the success rate of the AIC. Unlike the majority of signal processing application where the noise source is often known and, hence, noise can be characterised ultrasonic acoustic waves in rocks tend to be altered strongly making it difficult to distinguish the original signal from noise.

This AIC method has given fair approximations of S-wave arrival time. Nevertheless, errors made on small scale laboratory measurement significantly influence the outcome. It is expected that noise reduction in shear wave signals will improve the precision and accuracy of the AIC picking method. Every field dealing with signals, regardless of the nature of the

signals, encounters noise related problems. There are many signal processing techniques to reduce noise, but the question is which processing techniques could improve the quality of ultrasonic laboratory signals.

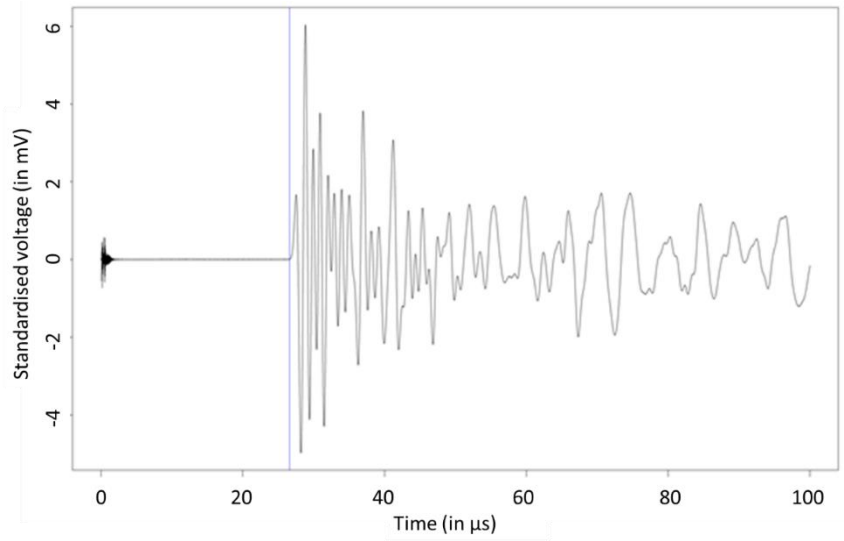


Fig. 4: P-wave signal on 42 mm diameter rock sample. P-wave onset highlighted by the blue line.

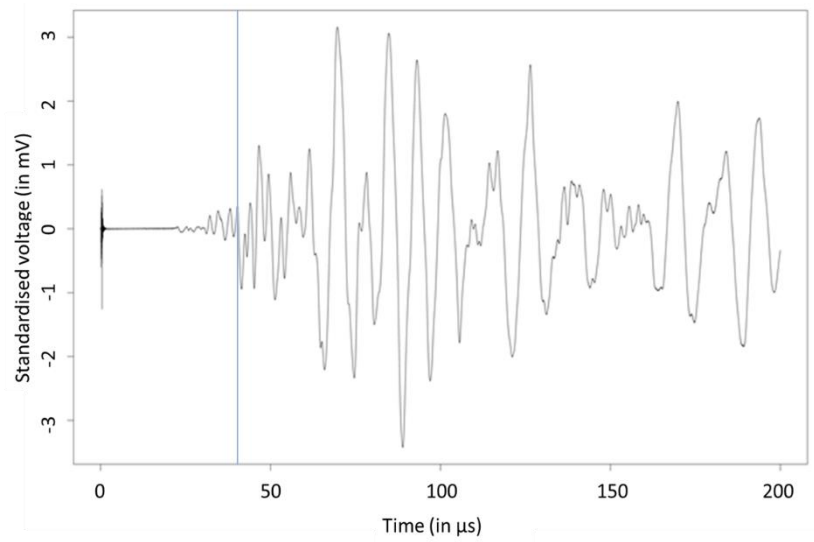


Fig. 5: S-wave signal on 30 mm diameter rock sample. S-wave onset highlighted by the blue line.

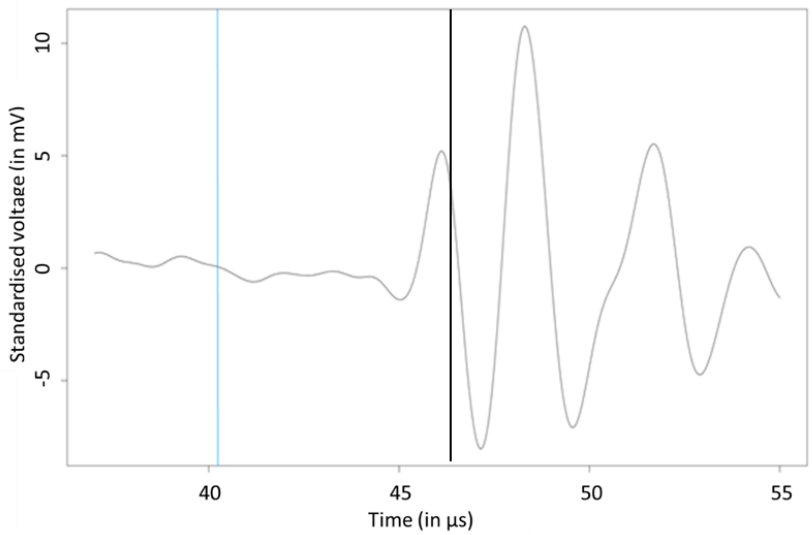


Fig. 6: S-wave signal on 42 mm diameter rock sample. Wrong estimation by AIC highlighted by the blue line. Correct manual estimation highlighted by the black line.

1.6 Scope of the thesis

1.6.1 Research goal

The goal is to increase the precision and accuracy of the AIC picking method regarding S-waves by improving the quality of S-wave signals through noise reduction. Two main purposes of signal processing are signal modelling to reduce data size and noise reduction to improve signal quality. This thesis will focus on the latter. The potential benefit of processing ultrasonic data is to make it more suitable for automated onset picking by the AIC.

1.6.2 Research questions

The research questions that guide this thesis are defined at listed numbers 1 and 2. The first research question aims to make a selection of signal processing methods that can be applied to ultrasonic laboratory signal of S-waves. The outcome of research question number one is used for evaluation which is stated by research question number two. They are listed below. See Section 2.1 for a comprehensive description of the approach to find answers to these questions.

1. *Which signal processing methods can be applied to ultrasonic laboratory signals?*
2. *How do these methods influence the ultrasonic shear wave onset estimations?*

1.6.3 Hypothesis

Many applications are expected to deal with signals that are dissimilar to ultrasonic laboratory signals or that their processing methods are too complex for implementation. For example, in sonar and radar applications signals are received in sensor arrays and moving

targets are often the objective to trace. This makes their practical implementation probably too complex and their relation to the problem in many cases inadequate, since tracing moving objects is not the objective of ultrasonic laboratory measurements. The fields that are expected to have potential are biomedical signal processing (Tranquillo, 2014), audio processing (Zölzer, 1998), and speech processing (Deller, et al., 2000), because they deal with two-dimensional time signals. Frequency filters are expected to provide good results, because they can filter out the frequencies of unwanted noise and they allow for automation. Furthermore, statistical signal processing techniques are expected to improve the quality of the ultrasonic signals. Because they are based on statistical parameters the origin of a signal should not matter (Gray & Davisson, 2004; Hayes, 1996).

1.7 Background information on signal processing

1.7.1 Signals

Sound waves, electrical pulses, electromagnetic waves, measurements of populations and images are examples of signals. Although they do not share the same nature, they are common in the sense that they are the input, output or internal functions that a system processes or produces (Jackson, 1991). However, a signal has an intuitive meaning which might be better understandable than the exact definition.

Signals change in nature every infinitely small time steps, they are time-continuous. To measure a signal its value at certain time instances is extracted. The signal is now a vector of values along a time scale, it has become discrete (Zölzer, 1998). When signal processing is applied to a signal it is usually a discrete signal, because it concerns the processing of an extraction of a continuous signal. Moreover, the signal of interests has to be stored and storing on a computer or electrical device can only happen in data points, which have a discrete basis.

Then there are periodic and aperiodic signals. Periodic signals tend to repeat themselves after a certain time, the cycle time, which is often denoted by T_0 and expressed in seconds (Jackson, 1991). The number of cycles a signal goes through per second is called the frequency, which is denoted by f with unit Hz or denoted by ω with unit rad/s. The ultrasonic shear wave signal used in laboratory measurements performed for this study has a frequency of 1 MHz. Aperiodic signals do not repeat themselves and are mostly found as noise. The most common type of noise is white noise.

1.7.2 Systems

Due to the wide application of signal processing there are many shapes and forms of systems (Jackson, 1991). Though, they have analogous properties that can be used to categorise them. The main properties are causal vs non-causal, linear vs non-linear, and time-invariant vs time-variant, which are explained comprehensively in the next few paragraphs. There are more properties used to describe a system being memory, stability or invertibility which are not explained into detail (Jackson, 1991).

A system is called causal when its output only depends on past and/or present inputs and does not anticipate future inputs. Most natural systems are causal, because they have the

inability to anticipate the future. A signal is not always a function of time, but can also be a function of distance, for example (Jackson, 1991).

A system is categorised as linear when it meets two principles: Additivity and Scaling. When an input signal $x_1(t)$ produces an output signal $y_1(t)$ and a $x_2(t)$ produces $y_2(t)$, then the input signal $x_1(t)+x_2(t)$ results in output signal $y_1(t)+y_2(t)$, *that is called additivity*. Scaling indicates that when a certain signal is scaled with a scaling factor *a* the output of the system should be equal to the original outcome multiplied by the same scaling factor *a*. In proper notation it would be described as $ax(t)=ay(t)$ for any $x(t)$ or *a* (Jackson, 1991).

Time invariance means that the output of a system is independent on time. A system that is time-invariant always produces the same output for the same input regardless of the time of input. If an input signal $x(t)$ produces output signal $y(t)$ then the time shifted version $x(t-t_0)$ produces output signal $y(t-t_0)$ for any value of t_0 (Jackson, 1991).

Systems can also have a memory, be stable or instable, and be invertible or noninvertible (Jackson, 1991), which are also meaningful properties, but not applicable in the context of this thesis. This thesis deals with ultrasonic laboratory measurements that are performed on rock samples. The rock functions as the system which has straightforward properties. It is assumed to be causal, because there are no measurement recorded before an input is generated. It is also assumed to be linear and time invariant, because the input does not depend on the time instance of the input and the output is proportional to the input.

2. Research approach and methodology

This section addresses all the methods used in this thesis that can be linked to results generation and evaluation. Signal processing methods are not discussed here. Those will be discussed in Section 3. The first subsection of this section will cover a detailed plan of the research approach. Data generation and evaluation methods named in the research plan are discussed in the second and third subsection. How the synthetic data is generated is comprehensively explained in the fourth subsection.

2.1 Research approach

First a literature study was conducted in which familiarisation with signal processing was fundamental. Books and internet were mainly used in this phase of the study. Elementary signal processing books (Gray & Davisson, 2004; Hayes, 1996; Jackson, 1991; Tranquillo, 2014) often introduce the reader to signal processing by giving examples of applications of the study. These examples gave a good insight into the most common fields of signal processing. Those fields were explored more thoroughly to look for noise reduction methods used there. The relation to ultrasonic signals on laboratory scale was assessed and applicable methods were considered in the practical implementation.

The practical part started with implementing the applicable methods. If the methods would take more than one week to program in R, an open-source statistical and graphical software environment, they were disregarded. This estimation was made consciously, because writing a function in R could be too time-consuming, counteracting the goal of this thesis. Methods that could be programmed in roughly one week or that were already programmed by others were prioritized over computationally time-consuming methods. The methods that required no excessive education in programming or that were already provided in R were used for evaluation on synthetic data set. This data set was build using characteristics of the real ultrasonic signals, such as noise components of different frequencies and onsets and an approximation of the clean signal. After processing the synthetic signals with the signal processing techniques the performance of the methods was evaluated using the AIC for S-wave onset picking. The AIC would show the S-wave onset after processing of the signal. The main goal during this phase of the practical part was to see how the signal processing methods influence the signal. The methods that positively influenced the signal were then applied to real ultrasonic signals. This data was measured on rock samples from Olkiluoto veined gneiss rock mass. The AIC picking method was applied to the processed and unprocessed real signals. The performance of the filters was evaluated by comparing the AIC onset after processing to the AIC onset before processing. The exact onset in the real data is unknown, therefore, the onset was once picked manually by three individuals and once approximated by the theoretical relationship between P- and S-wave velocity (see Section 2.2). The mean and standard deviation of the relative difference between the manual or theoretical onset and the onset picked by the AIC picking method was used for evaluation. A ranking of the best to worst performing filters was generated. The research approach is illustrated by a flow chart in Fig. 7. The next subsection will go into more detail about the theoretical relationship between P-and S-wave velocities, the AIC and the chosen methods for result evaluation.

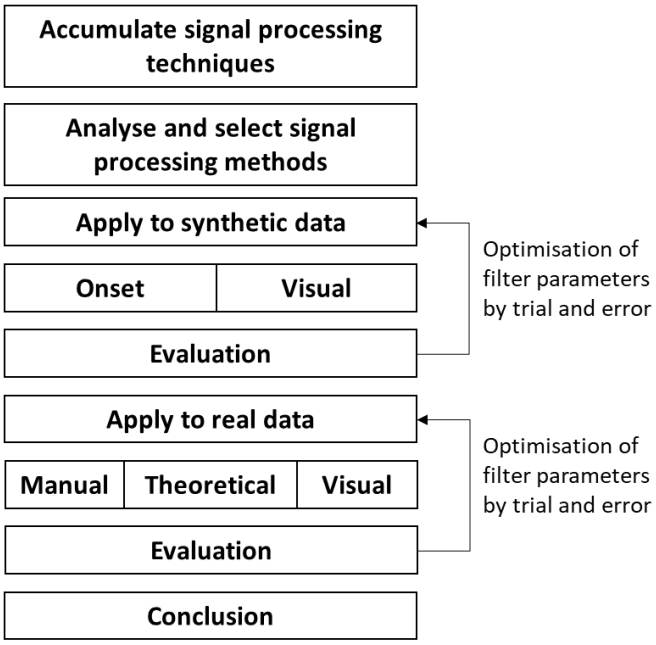


Fig. 7: Research approach flow chart

2.2 Methods for P-and S-wave velocity estimation

2.2.1 Theoretical relationship

A simple way to estimate S-wave velocity is via the theoretical relationship between P-and S-waves.

$$V_s = V_p * \frac{1}{\sqrt{3}} \tag{1}$$

where V_s is the S-wave velocity and V_p is P-wave velocity.

The theoretical relationship is derived by (Yilmaz, 2001) and (Langenberg, et al., 2012). They state that

$$\frac{V_s}{V_p} = \sqrt{(1-2\sigma)/(2(1-\sigma))} \tag{2}$$

where σ is Poisson's ratio.

Typical Poisson's ratio for rocks are found between 1/6 and 1/3 for medium range rocks (Gercek, 2007). Plugging in these values in (2) gives a *Vs/Vp* ratio of between 0.62 and 0.50. In general this ratio is accepted as (1).

The ultrasonic tests that are performed on the rock specimen used for this thesis have a V_s/V_p ratio of 0.59 (Alejano, et al., 2018). The theoretical relationship is stable for a first indication of shear wave velocity based upon the easily measurable pressure wave velocity.

2.2.2 Akaike Information Criterion

This AIC, developed by the Japanese Hirotugu Akaike in the early 1970's (Akaike, 2nd International Symposium on Information Theory), treats the signal in the time domain and uses a mathematical approach based on the variance (Cavanaugh & Neath, 2019). A pre-existing code was used for the practical application of AIC created by R. Kiuru in 2017, which is appended in Appendix 1 (1/31).

In practice the AIC aims to select a point in the time domain that coincides with a rapid change in tangent and a strong amplitude displacement. Visually checking the chosen S-wave onsets by the AIC could lead to different interpretations. The first case scenario is where the AIC picks the onset at a arrival time in a signal that would be manually fairly easy to detect, see Fig. 8. The black line indicates the clear S-wave onset and the blue line indicates the onset picked by the AIC.

See Fig. 9 for a debatable arrival time. There are two sudden declines that could be related to the S-wave onset, those are indicated with black lines. The AIC pick is indicated with a blue line. In this case the AIC has picked the wrong time instance, but between the manual picks is also uncertainty.

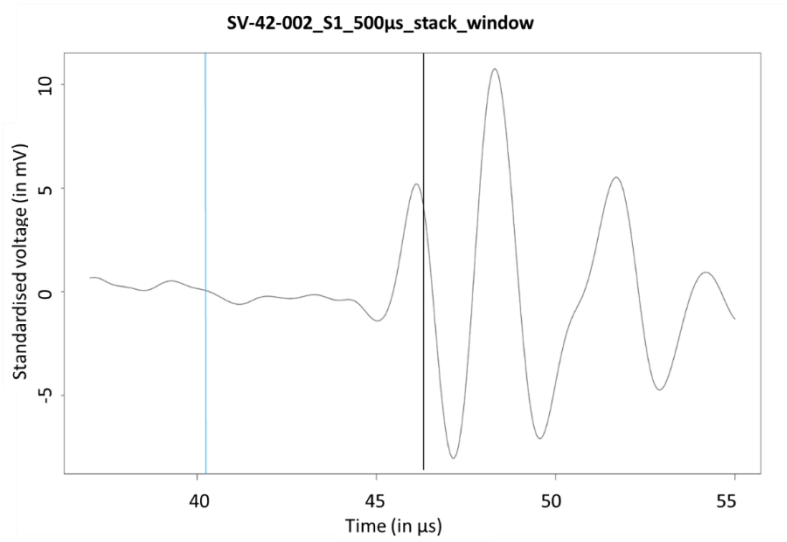


Fig. 8: S-wave signal on 42 mm diameter rock sample. Wrong onset estimation by AIC is highlighted by the blue line. Correct manual onset estimation is highlighted by the black line.

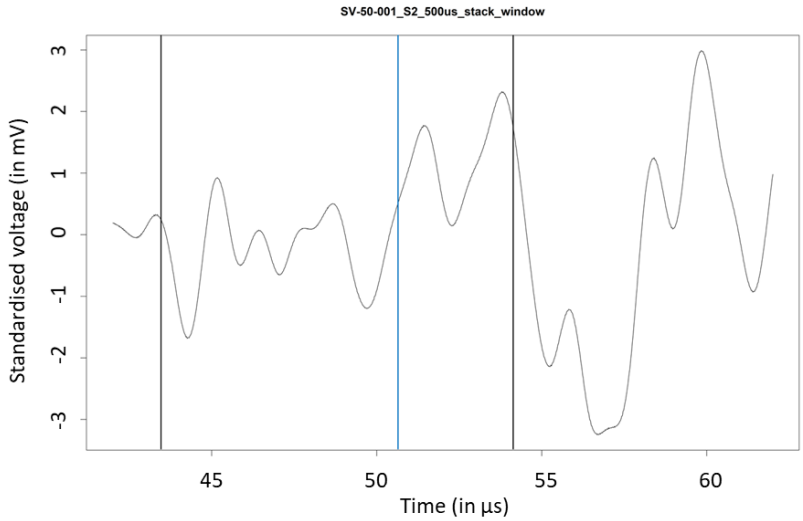


Fig. 9: S-wave signal on 50 mm diameter rock sample. Incorrect onset estimation by AIC highlighted by the blue line. Two debatable manual onset estimations highlighted by the black lines.

2.3 Methods for evaluation of results

2.3.1 KS-test

The Kolmogorov-Smirnov test (KS-test) evaluates the chance that two signals are from the same population. Three signals are distinguished between in this matter. The synthetic signal (damped sinusoid + noise components), clean signal (damped sinusoid sinusoid only), and filtered signal (signal after processing). The ultimate goal of signal processing in this thesis is to obtain a signal after processing that exactly matches the clean signal. If this is the case, then the filter can be referred to as ideal. Clearly, ideal cases do rarely exist in science and therefore the filtered signal, clean signal, and synthetic signal are compared using the KS-test. The output of the KS-test with inputs synthetic signal and clean signal is the base case. After processing the filtered signal is compared to the clean signal and this result the output of the KS-test should indicate a stronger relationship between those two signals than between the synthetic and clean signal. If that is the case, the filter does improve the quality of the signal in terms of similar population probability. If the KS-test output is lower than before the filter makes the signal worse.

The outcomes of the KS-test showed remarkable results. All p-values where in the range 0.0005-0.0001, which indicates that the chance that two samples are not from the same population is very small. Due to the strong similarity in signals, oscillations around zero mean with multiple sinusoidal components, the KS-test could point out a severe difference between ‘clean vs synthetic’ and ‘clean vs processed’. The measured differences are not significant enough to draw a conclusion from. For that reason the KS-test results are not taken into account in the filter evaluation. The code for input production for the KS-test in attached in Appendix 2 (3/31). Appendix 3 (4/31) contains R code for KS-testing.

Other signal comparison tests, such as the t-test, Wilcox-test and Cucconi-test have been considered for application to the ultrasonic waves, but those tests do not measure signal characteristics that are valuable for this research purpose.

2.3.2 Onset estimation

The key feature of the this research is onset estimation. The goal is to improve the precision and accuracy of the onset picked by the AIC. It is easy to evaluate the improvement of the onset on the synthetic signals, because the true onset is known. The mean and standard deviation of the onset of thirty synthetic signals is derived for every filter type. Those values are compared to the mean and standard deviation of the picks before processing. The filters are than ranked from best to worst based on the mean relative difference. When two mean differences are within 1 p.p. of one another the filter having a smaller standard deviation shall be ranked higher.

The real data is evaluated according to the same principle, the only difference being an unknown true onset which needs to be approximated by estimation. This approximation is carried out in two ways. The first approximation is based on the theoretical relationship between V_p and V_s. (See Section 2.2.1.) The subset of the real data used for evaluation contains 15 samples. An accurate P-wave onset is available for every sample, which allows for S-wave onset estimation based on the following equations:

$$V_p = \frac{t_p - F2F_p}{L_{sample}} \tag{3}$$

Where V_p is the P-wave velocity, t_p is the P-wave onset, $F2F_p$ is the face-to-face time of the P-wave component, and L_{sample} is the sample length.

The P-wave velocity is then used for theoretical onset estimation by factor $1/\sqrt{3}$ using the next equation:

$$t_{s1,s2} = \frac{L_{sample}}{(V_p/\sqrt{3})} + F2F_{s1,s2} \tag{4}$$

Where $t_{s1,s2}$ is the theoretical onset of S1 or S2, and $F2F_{s1,s2}$ is the face-to-face time of either the S1 or S2 component.

The theoretically obtained onsets are compared to the AIC onsets before and after processing. Their mean and standard deviations are compared.

The second method for estimating the ‘true’ onset is manual picking. For this case the S1 and S2 onsets of 15 samples were picked by three individuals of whom two experts and one beginner. These individuals have chosen their onsets independently of the picks of the others. All picks are taken into account with equal weights and an average is calculated. The average onset of every sample is used to calculate the relative difference to the onset picked by the AIC before and after processing. In Appendix 4 (5/ a code for onset accumulation is appended.

2.3.3 Visual comparison

The synthetic and real signals are classified based on a visual basis. This classification reflects the manual picking ease of the onset in the time signal. Three classes are defined, class one to three. A signal is assigned to class one if the onset is clearly visible and or multiple points within a 1 % relative distance from each other are considered as the onset. Class two comprises all signals where multiple onsets of more than 1 % relative time difference from each other are considered and expertise is needed to select the correct onset. Finally, class three comprises all signals where manual picking of the correct onset is impossible with certainty. Fig. 10Fig. 12 show one example signal per class. The signals are zoomed in on a time window of interest.

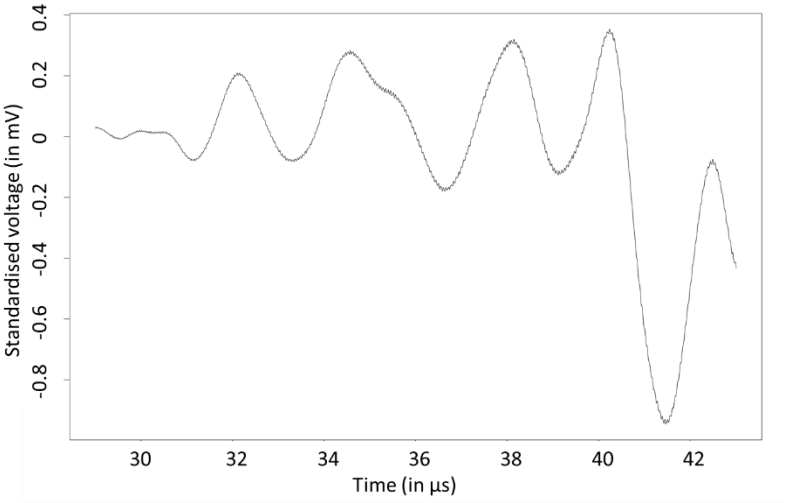


Fig. 10: Example of a signal assigned to class one

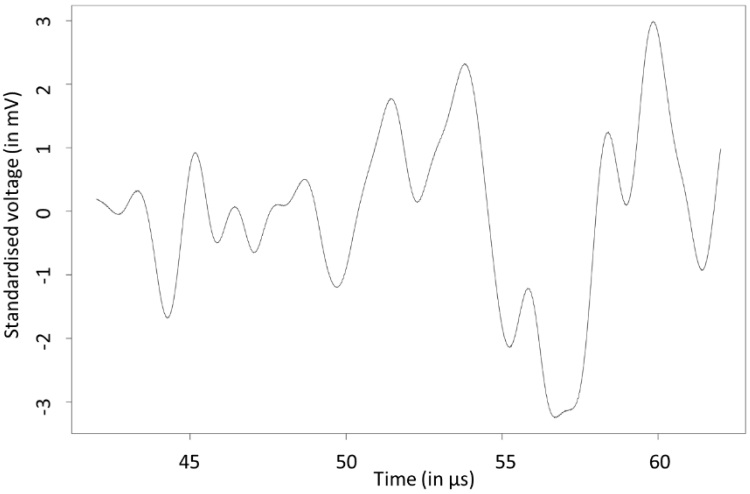


Fig. 11: Example of a signal assigned to class two

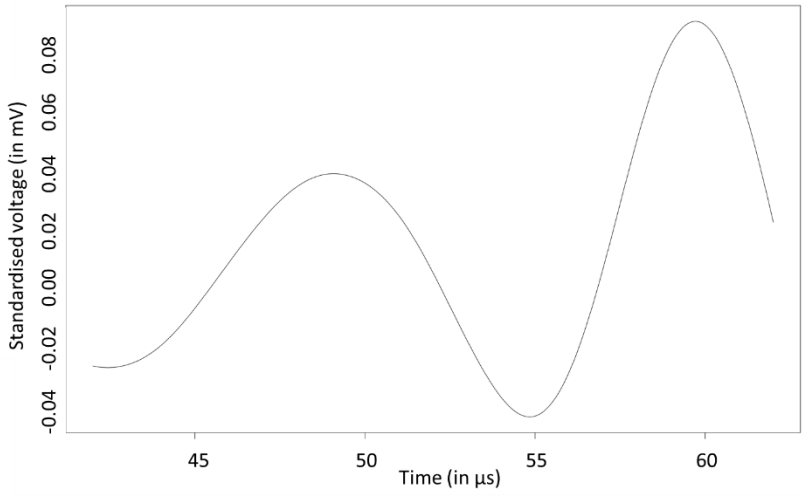


Fig. 12: Example of a signal assigned to class three

2.4 Synthetic signal production

An ultrasonic signal has two main components: a clean signal and noise components. The clean signal is the signal that is sent in the first place. This is the signal that one looks for when searching for the shear wave onset. Due to e.g. reflection and refraction noise is induced. In real signals there is a broad bandwidth of noise present. In this thesis 7-8 noise components are added to the clean signal for synthetic signal production. In Section 2.4.1 the methodology for creating a clean signal is explained. In Section 2.4.2 the procedure for noise selection and addition is discussed. The code written for synthetic signal production is appended in Appendix 5 (8/31).

2.4.1 Clean signal

Every signal has a frequency spectrum. The highest peak in that spectrum is referred to as the dominant frequency of that signal. This frequency is accepted as the frequency of clean signal. Therefore, it is also used as the base for the clean signal in synthetic signal production. In order to find the dominant frequency for the synthetic signal the face-to-face (F2F) signal is used. This signal is shown in Fig. 13. It is obtained by ultrasonic wave propagation through the transducers only. This is the cleanest form of the shear wave that can be obtained.

The first large amplitude displacement which indicates the arrival of the target shear wave is analysed in the frequency domain to obtain a better understanding of the dominant frequency in the F2F signal. This is displayed in Fig. 14.

The dominant frequency is known to be approximately 1 MHz. Fig. 13 shows that the measured frequency is around 875 kHz. The measured frequency is strongly dependent on the chosen window. It is not of great importance for synthetic clean signal deviate from the known and measured frequency. As it can be seen in Fig. 13 the first dominant shear wave arrival is of high amplitude. The strength of the signal decays over time and therefore a damping factor is added to the synthetic signal to account for this phenomena. The time domain signal is cut off until the first shear wave arrival. From there a damped sine curve is manually fitted to the rest of the signal. It is not necessary to have a minimum least square

solution for the fitting process, because the goal of the synthetic signal is not to make an exact replica of the real signal, but rather a similar signal with known parameters.

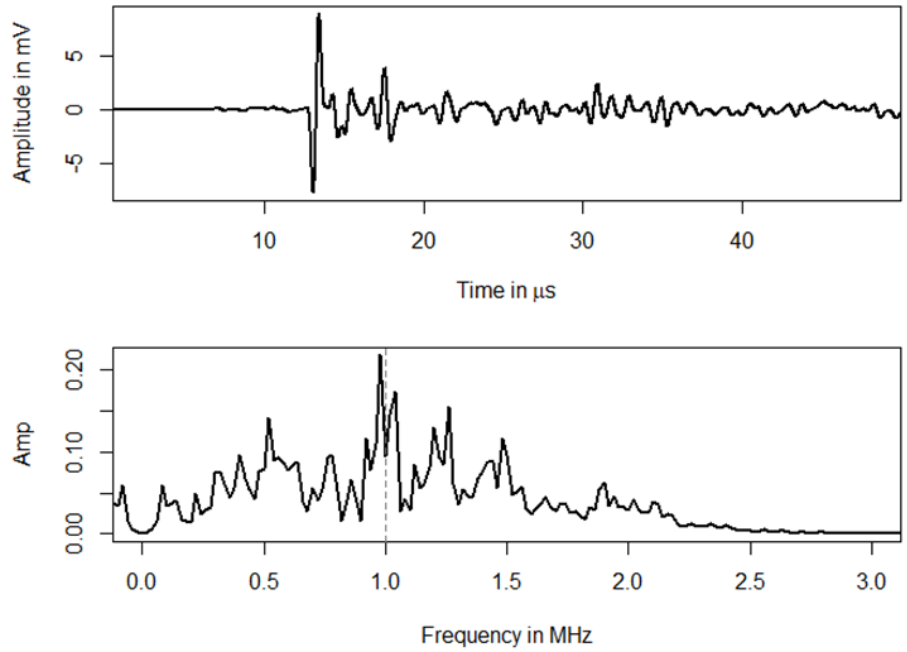


Fig. 13: Time (top) and frequency (bottom) plot of F2F signal of S1. Dashed line indicates 1 MHz.

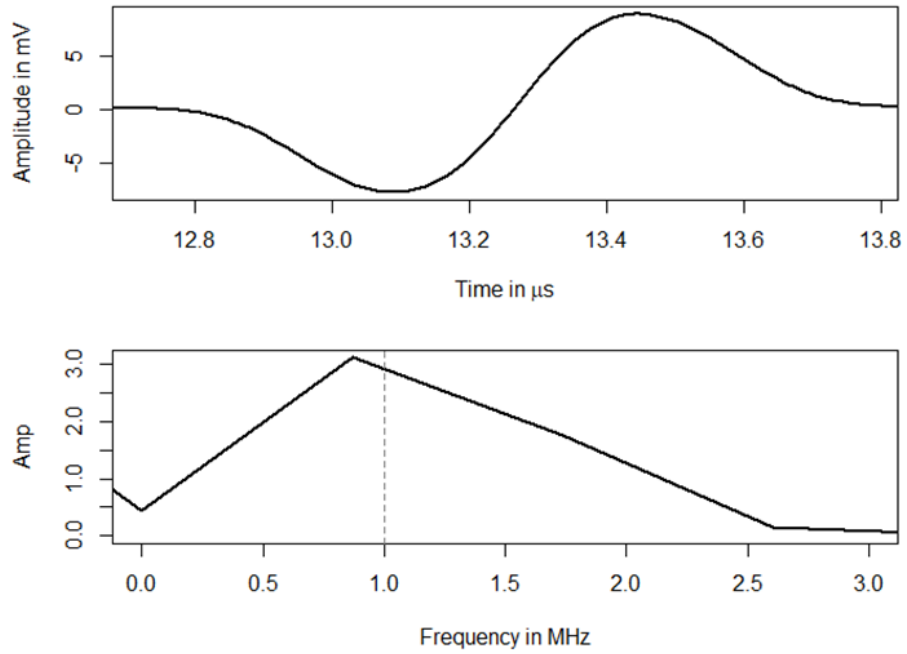


Fig. 14: Dominant peak in time (top) and frequency (bottom) domain of F2F signal S1. Dashed line indicates 1 MHz.

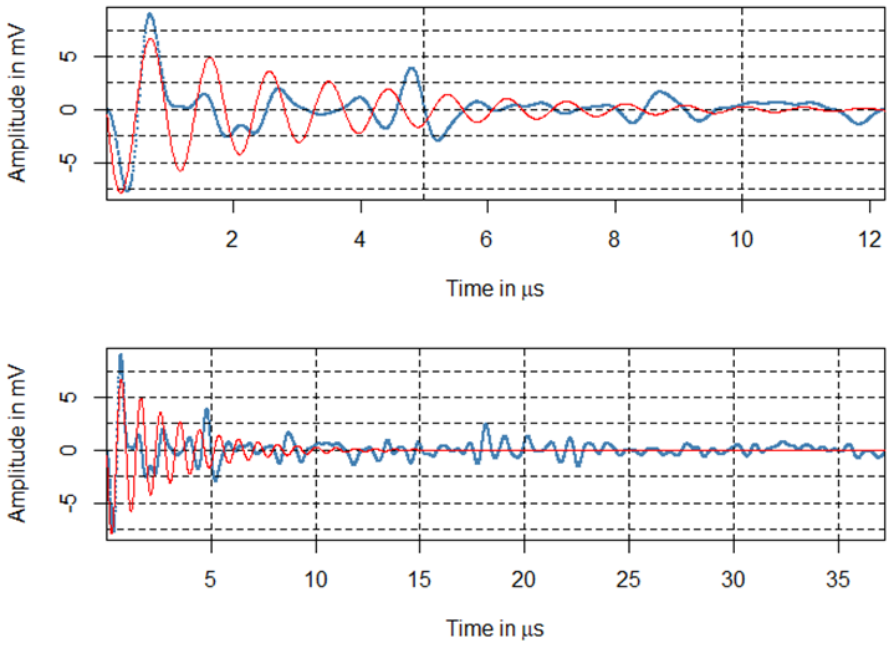


Fig. 15: Damped sinusoid fitted to the clean signal in F2Fsignal of S1. Original signal in blue and fitted signal in red. Fitted curve following the clean signal (top) and missing the noise at later time instances (bottom)

In Fig. 15 a damped sine curve (in red) is illustrated versus the original signal (in blue). The general notation of a damped sine wave is:

$$y(t) = Ae^{-\gamma t}\sin(2\pi f t) \tag{5}$$

where A is amplitude in mV, γ is decay constant in the reciprocal of time units of the x-axis, and f is frequency in Hz.

The optimal parameters for the clean signal are set to A=-8.5 mV, γ=1/3, *and* f=0.1 MHz for the purpose of this research. At the start of the shear wave the synthetic and the real signal coincide quite well, whereas time evolves the difference becomes greater. Probably reflection induced noise is causing the oscillations later in time. A single damped sine wave cannot match the original signal, therefore noise components are introduced.

2.4.2 Noise components

Noise components were added to the clean signal to make the signal statistically more realistic. In Fig. 13 it is shown that the F2F signal has a wide range of frequency components. The same applies to the ultrasonic signals from rocks. Based upon the frequency spectra of the rock samples the main frequency components per sample length were selected. In total there were 22 frequency components selected. Those were gathered in a histogram which indicated the main frequency components. This histogram is shown in Fig. 16.

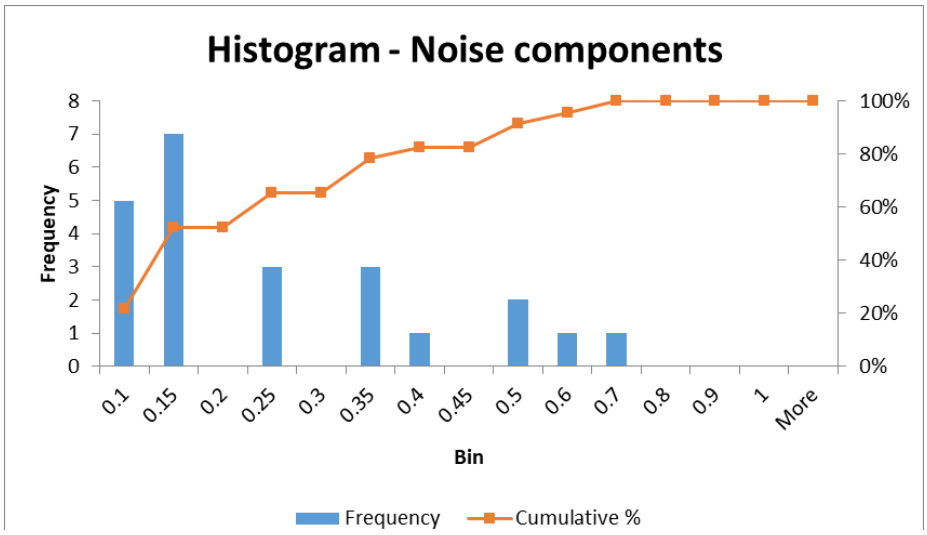


Fig. 16: Histogram of noise components selected for synthetic signal production. On the x-axis bin range is expressed in MHz. On the left-hand-side the frequency of appearance of a frequency is plotted and on the right-hand-side the cumulative percentage of the frequencies is plotted.

On the x-axis the bin range is defined in MHz. The six largest bins were selected as noise components. However, six different noise components are not sufficient to build synthetic signals in tenfold. Also, the noise components are never exactly the same, there is always a slight deviation in either frequency, amplitude and/or phase. Therefore, every selected noise components out of the six noise components has eight variations. This concludes to a total of 48 noise components.

For the synthetic signal production a random selection from the 48 noise components was added to the clean signal. The shape of the noise components is also a damped sinusoid. The onset of the noise components was chosen to start at the first time instance. The onset of the clean signal depends upon the sample length. Breaking down, this means that the longer the sample the later the clean signal onset and the more obvious its arrival time becomes due to the damping of the noise signals.

The number of noise components was varied from 1-12 and eventually set to a random number in the sequence 6-8. Applying the AIC picking method to 35 signals with 6-8 noise components per signal resulted in a partially correct and partially incorrect estimation of the shear wave onset. That was the goal of the synthetic signals, because it enabled the evaluation of the effect of the filters on formerly correctly estimated onset and formerly incorrectly estimated onsets. The short sample lengths (14 mm and 20 mm diameter) were almost all incorrectly estimated, due to the small influence of damping on the noise components during the shear wave onset. The large sample sizes (60 mm and 100 mm diameter) had very accurate estimations, because the noise components were already damped strongly making the shear wave arrival obvious. Interesting were the medium sample sizes (30 mm, 42 mm, and 50 mm diameter), here the ratio correct/incorrect onset estimations was roughly 50/50, so the influence of filters could be tested well on those signals. Fig. 17 shows a synthetic signal of length 30 mm and Fig. 18 shows a real signal of 30 mm diameter. Fig. 19 and Fig. 20 show the onset arrival on a short window of a synthetic and real signal, respectively, from which the difference in onset becomes clear.

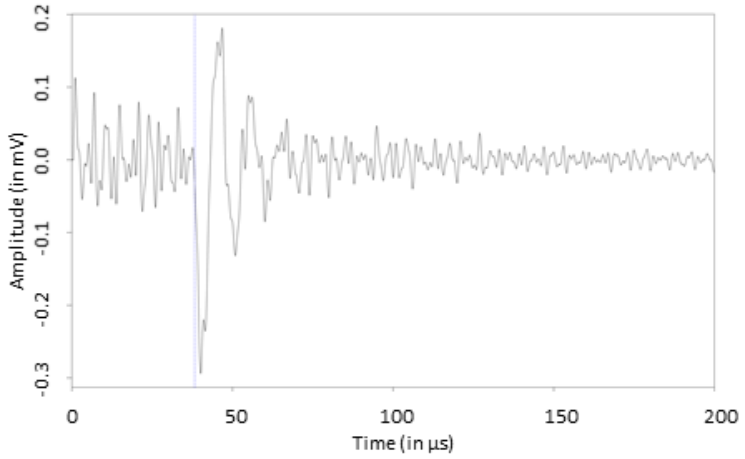


Fig. 17: Synthetic S-wave signal of 30 mm diameter sample

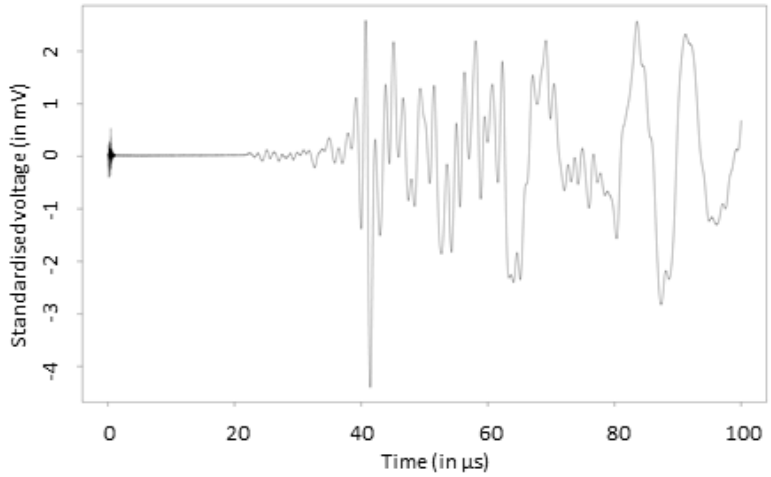


Fig. 18: Real S-wave signal of 30 mm diameter sample

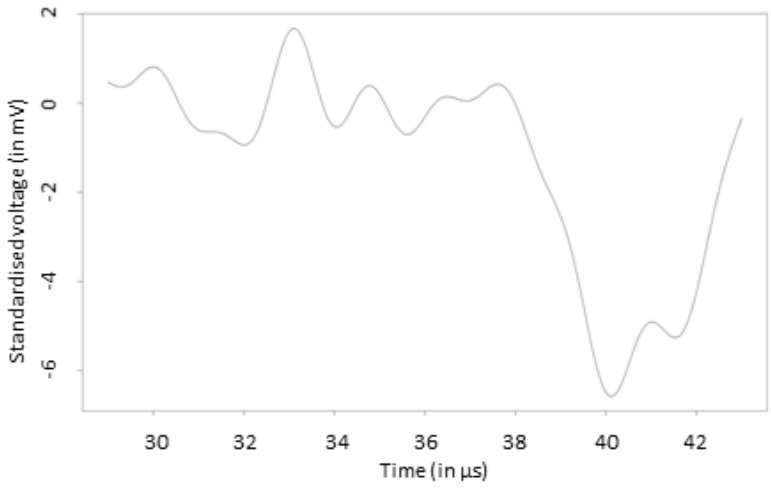


Fig. 19: Synthetic S-wave signal of 30 mm sample zoomed in around the onset

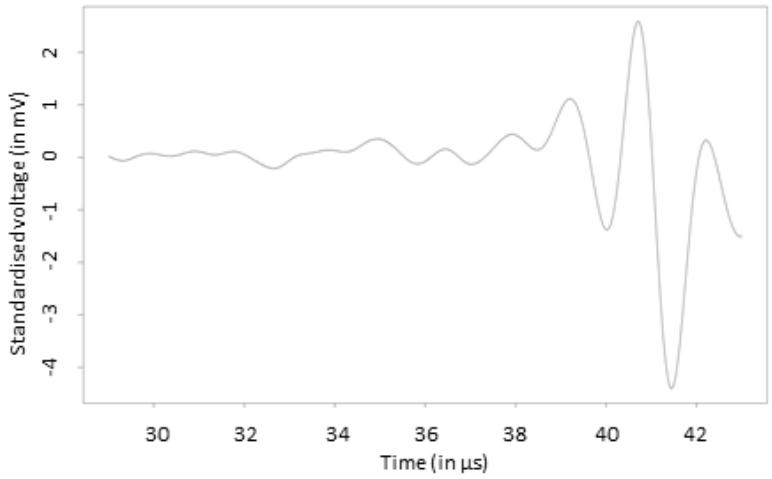


Fig. 20: Real S-wave signal of 30 mm sample zoomed in around the onset

3. Signal processing methods

This section contains all the signal processing methods that were considered for application on the synthetic and real ultrasonic data. The filters are categorised per field of application. Four main fields were researched, those are stated in the name of the first level subsections. The considered filters in those fields are the names of the second level subsections. The functionality of every filter is addressed followed by an explanation of its applicability to ultrasonic signals.

3.1 Statistical signal processing

3.1.1 Kalman filter

The Kalman filter is a live state predictor that predicts the current state of a system, x_k, on the previous state, x_{k-1}, and on related uncertainties (Kalman, 1960). It is logically reasoned to be a computationally light filter, because it only remembers the previous state of a system to predict the current one. The Kalman filter is mainly used in real-time estimation of moving object.

The Kalman filter takes into account the current state of a system, which is described by a Gaussian distribution, the measurable external influences that affect the state of the system, and, external uncertainties that affect the system that are not directly measurable, but that are accounted for by a Gaussian distribution. Due to combining multiple Gaussian distributions the overlapping provides a new Gaussian distribution with a lower standard deviation and therefore more accurate predictions. This is explained in the next paragraphs supported by simple mathematical notations (Kalman, 1960; Brown & Hwang, 2012).

The state variable of the current state is denoted by x_{k-1}. This variable can be for example, temperature, velocity, height, position, amplitude, etc. Also, multiple variables can be stored in the state vector. The state variables in the state vector are measured at every time instance and they are known value. However, in many cases the measured values follow a Gaussian distribution function. Therefore, the state vector contains the mean value of this Gaussian distribution, but the all values in the function are used for the prediction of the *best estimate.*

$$\hat{x}_{k-1} = \begin{bmatrix} Var1 \\ Var2 \end{bmatrix}$$
$$P_{k-1} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \quad (6)$$

where $\hat{x}_{k-1}$ is the *best estimate* of the current state, *Var1* and *Var2* are state variables one and two, respectively, P_{k-1} is the current state covariance matrix of the covariance between *Var1* and *Var2*, and Σ_{xx} is the covariance between the variables denoted by the subscript.

The next step is to predict the next state x_k by using the current state and its covariance matrix. A prediction matrix is defined, denoted by F_k, that uses physical relationship between the current and next time step of the chosen parameters.

$$\hat{x}_k = F_k * \hat{x}_{k-1} \quad (7)$$

where F_k is the prediction matrix.

So, the next state is estimated based on the prediction matrix. The covariance matrix of the current state is also modified by the same prediction matrix by:

$$P_k = F_k * P_{k-1} * F_k^T \quad (8)$$

where P_k is the next state covariance matrix and F_k is the prediction matrix.

At this stage the next state estimate and its covariance matrix are generated. There are external influences acting upon the system as well. Those are factored into the estimation.

$$\hat{x}_k = F_k * \hat{x}_{k-1} + B_k * \mu_k \quad (9)$$

where B_k is the control matrix and μ_k is the control vector. The control matrix contains the modification factors that are applied on the control vector to eventually change the state variables.

The last step is to add the external uncertainty that cannot be measured. For example, the velocity and position of a drone are measured, but due to wind influences the velocity parameter is not the only parameter influencing the position of the drone. The new position is a distribution around the next state position which is already a distribution due to imperfect measurement accuracy. The extra distribution caused by external influences that are not measured are denoted by Q_k. This influences the covariance matrix of the next state by:

$$P_k = F_k * P_{k-1} * F_k^T + Q_k \quad (10)$$

After consideration it was concluded that the Kalman filter does not apply to ultrasonic laboratory signals measured on rocks. Those signals do not benefit from real live state prediction by taken into account uncertainties of the measurement device, because that would not filter out the noise. Therefore, the Kalman filter does not contribute to the goal of this thesis.

3.1.2 Kolmogorov-Zurbenko Adaptive

The Kolmogorov-Zurbenko (KZ) filter is a moving average filter proposed by A.N. Kolmogorov and defined by I. Zurbenko (Zurbenko, 1986). The moving average filter has two clearly interpretable parameters which make its application easy. The first parameter is the moving average window defined by *m*. The second is the iteration parameter defined by *k*. The nature of the filter, namely a moving average one, makes it especially applicable in missing data environments (Yang & Zurbenko, 2010).

The input of the filter is a real-valued time series. The moving average window progresses per time unit over the time series and produces a new time signal. The iteration parameter *k* represents the number of iterations. The first iteration is defined in (11).

$$KZ_{m,k=1}[X(t)] = \sum_{s=-(m-1)/2}^{(m-1)/2} X(t+s) * \frac{1}{m} \quad (11)$$

Where *X(t)* is real-valued time series, *m* is the moving average window, and *k* is the iteration. The output $KZ_{m,k=1}[X(t)]$ is a real-valued times series and is used for the next iteration. The second iteration is defined in (12).

$$KZ_{m,k=2}[X(t)] = \sum_{s=-(m-1)/2}^{(m-1)/2} KZ_{m,k=1}[X(t+s)] * \frac{1}{m}$$
$$= \sum_{s=-2(m-1)/2}^{2(m-1)/2} X(t+s) * a_s^{m,k=2} \quad (12)$$

Where $a_s^{m,k=2}$ is a Chebyshev polynomial.

This sequence continues until the last iteration is completed. The eventual outcome remains a real-valued time signal. The filter is expected to be applicable to ultrasonic laboratory signals and to have a smoothing effect. Low amplitude noise will be average to zero quicker than the high amplitude onset. Therefore, potentially it smoothens out the first noise and leave the S-wave onset as the first visible peak in the time-domain. Iterating too often might decrease the sharpness of the onset, which could lead to imprecise estimations of the onset.

There are multiple variants based on the standard KZ filter. The Kolmogorov-Zurbenko adaptive (KZA) filter is a variation on the KZ filter that aims to detect a breakpoint in the mean. The effect of this filter could be very positive if it manages to connect a difference in mean with the onset. However, it could also be possible that arrival of the target signal does not cause a breakpoint in the mean or that there are multiple breakpoints. Concluding, the KZ and KZA are both considered as applicable.

3.2 Radar

The acronym RADAR stands for ‘radio detection and ranging’. The main purposes of radar is detecting stationary and moving objects. In a radar system a transmitter emits a radio signal, which is an electromagnetic (EM) wave, in a predetermined direction (Translation Bureau, 2013; McGraw-Hill & Parker, 2002). If the wave encounters an object part of the wave is reflected back to the receiver. The time between transmission and reception of a signal is called the two-way-travel-time. The distance between object and receiver can now be calculated via the EM wave velocity in air and the two-way travel time. See Fig. 21 for a schematic explanation of the radar principle.

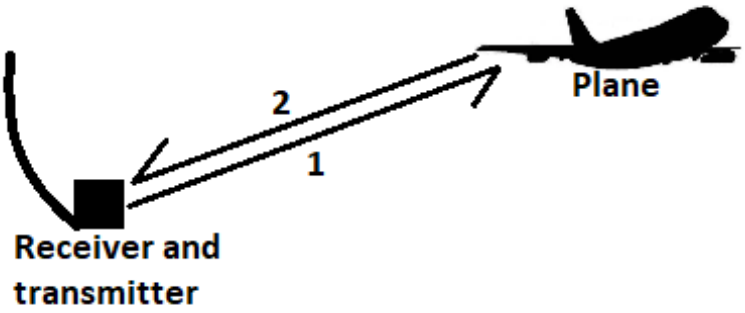


Fig. 21: Principle of radar detection (Created by author, idea from (Kirlf, 2019))

Objects of high electric conductivity reflect EM waves well. This means that also rain, wet ground and the sea reflect EM waves. They might appear as noise when they are not the target object and are called clutter (Golbon-Haghighi, et al., 2016; Richards, 2014). Clutter can be counteracted by Pulse-Doppler processing, which is the first radar signal processing technique considered for application on ultrasonic laboratory S-waves. The second signal processing technique considered in this section is the constant false alarm rate (CFAR), which is based on the assumption that there is a constant rate of false alarms disturbing the receiver (Rohling, 1983; Richards, 2014).

3.2.1 Pulse-Doppler processing

Clutter, as stated in the previous section, is the reflection of a radio wave on usually large objects with a natural background that do not have a single reflection but many small ones. For example, wet ground, sea, precipitation, birds swarms, and sand storms. Target objects could be located in clutter, which makes them difficult to detect. If there is a difference in velocity between clutter and target object the frequency of the emitted EM wave will be altered by both object differently (Richards, 2014). This is called the Doppler-Effect (O'Donnel, 2009; Scensor, 1973).

Pulse-Doppler is signal processing technique applied to the frequency domain. A pulse with a certain frequency is transmitted. When two object are hiding in the same direction, for simplicity's sake let's say exactly towards the transmitter, than the faster moving object creates a frequency higher than the slower moving object and the slower moving object creates a frequency higher than the frequency of the original signal. In the frequency domain the objects can be distinguished.

In Fig. 22 the Doppler effect is schematically visualized. The pink circle represents a moving object in direction of the orange arrow. The frequency of the reflected wave in direction of the moving object is higher than the reflected wave in opposite direction. Rock samples do not contain moving object and therefore this method will not be applicable to ultrasonic laboratory measurements.

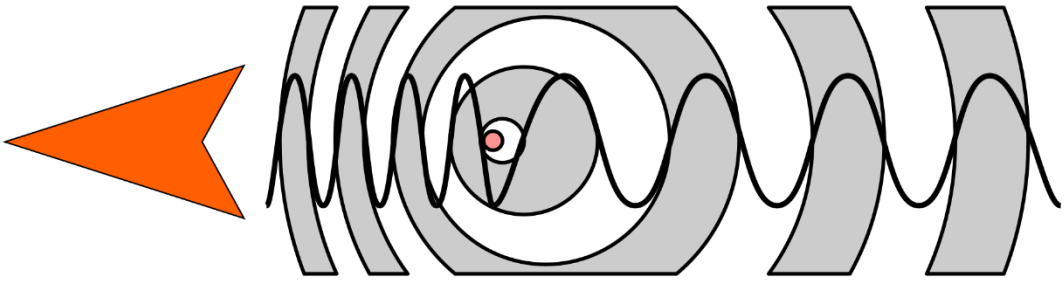


Fig. 22: Doppler effect. Moving target indicated by pink circle. Direction of the moving target indicated by the orange arrow. Sound emitted in same direction as movement has higher frequency than sound emitted in the pposite direction. (Tkarcher, 2006)

3.2.2 Constant false alarm rate

The strength of a received signal is strongly related to the electrical conductivity of the medium the wave reflects upon. Water surfaces and metal objects are good reflectors. When an EM wave hits an object the wave scatters.

A transmitted wave encounters many object and the receivers receive many reflections. Not all reflections have a significant meaning in radar detection. The reflections of unimportant objects should be denied and the reflections of target object should be raised.

An amplitude filter only shows signals with an amplitude higher than a certain amplitude threshold. If this threshold is set too low than too many targets are detected of which many are unimportant. Those are called false alarms. If the threshold is set too high fewer false alarms are detected, but at the expense of not noticing targets. In radar the background noise due to clutter and interference changes through time and space and therefore the threshold should be changing throughout those dimensions. This type of adaptive filter is called the constant false alarm rate (Rohling, 1983). Its name refers to the assumption that there is a constant rate of false alarms coming in represented by a Gaussian distribution. In ultrasonic laboratory measurements the amplitude of the amplitude signal is always equal and the background noise is independent of time therefore a fixed threshold can be used. In this case a simple amplitude filter will be sufficient.

3.3 Biomedical engineering

Biomedical devices massively incorporate digital signal processing techniques. Devices that measure muscular contractions, for example those from the hart or lungs or devices that measure brain activity or the pH-level of blood over time (Tranquillo, 2014).

For example (Tranquillo, 2014), an electromyogram is recorded from muscles on the torso. Its main purpose is to evaluate muscle contraction during breathing. The muscles contract approximately 20 times per minute. However, the electrodes measure voltage in general and therefore also pick up other signals from the body. A beating heart is normally the main component of noise in these types of measurements. It beats at a rate of approximately 70 Hz. This means that the time domain contains two main signals. Transforming this signal to the frequency domain results in a clear distinction between the two main frequencies. To

filter out one of the frequencies a frequency filter is commonly used in biomedical devices. According to (Tranquillo, 2014), two common frequency filters used in biomedical engineering are the Butterworth filter and the Chebyshev type 1 filter. Those are discussed in the next two subsections.

3.3.1 Butterworth filter

Butterworth filter is a frequency domain filter designed by Stephen Butterworth (Butterworth, 1930). The frequency response of the filter is characterised by a maximally flat pass band. The order of the filter determines the steepness of the roll-off and the amount of poles present in the filter (Panagos, 2014). A frequency filter cannot cut-off directly all frequencies above or below the cut-off frequencies. The frequencies that are to be eliminated are 'rolled off' gradually. An ideal filter has a 90˚ roll-off and zero poles. However, the steepness of the roll-off comes with the price of poles. This will be shown in the first two subsections. Butterworth filters are standardly designed as low-pass filters, but can be modified to a high-pass, band-pass or band-stop filters. Scaling of a normalised filter to a filter that can be applied to a specific frequency spectrum is derived in the third subsection.

The frequency response of a Butterworth filter is defined by:

$$|H(j\omega)| = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}} \tag{13}$$

Where $H(j\omega)$ is the frequency response, ω_c is the filter cutoff frequency and n is the order of the filter.

In order to apply the Butterworth filter to a frequency domain that belongs to an ultrasonic signal it is first shown what the influence of the order is on the frequency response, because the order should be defined before application of the Butterworth filter. The code for Butterworth filtering already exists in R and requires the filter order as input. To represent the relationship between order and poles the normalised Butterworth filter equation is first derived. A normalization of the filter states that the cut-off frequency is equal to one and therefore the (13) becomes:

$$|H(j\omega)| = \frac{1}{\sqrt{1 + \omega^{2n}}} \tag{14}$$

The frequency respons of n^{th} order normalised Butterworth filter is shown in the figure below. The ideal 'brick wall' response is approached by increasing the order of the filter. The higher the filter order the steeper the roll-off of the frequency response. The ideal 'brick wall' response is ideal but cannot be reached, because there the order must be a finite number.

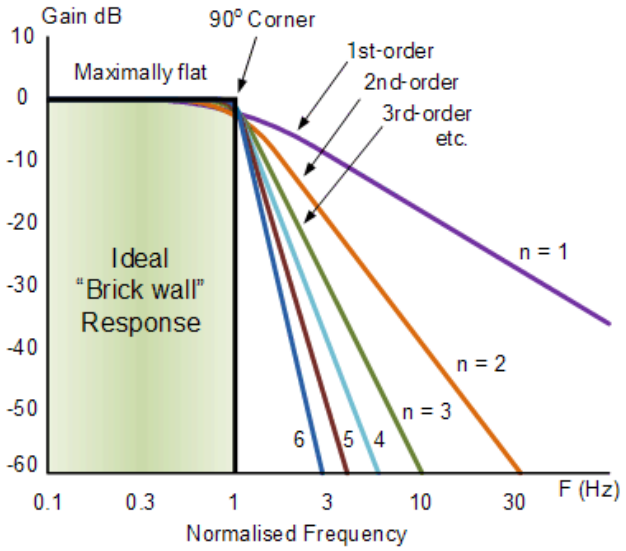


Fig. 23: Progression of the frequency response of a Butterworth filter with increasing order (Storr, 2014)

The relation between filter order and poles will be established in this section. The transfer function of the normalized Butterworth filter respons is denoted as *H(s)* where $s=\sigma+j\omega$, so the frequency response *H(jω)* can be obtained from *H(s)* by evaluating it *as* $s=j\omega$ *(σ=0).* And vice versa, *H(s)* can be obtained from *H(jω) by* $\omega=s/j$. Resulting in:

$$|H(j\omega)|^2 = H(j\omega)H(-j\omega)$$

$$= H(s)H(-s)$$

$$= \frac{1}{1+(s/j)^{2n}} \tag{15}$$

From this formula the poles of the filter can be determined. Whenever $(s/j)^{2n}=-1$ the denominator equals zero and the response blows up. Rewriting $(s/j)^{2n}=-1$ to $s^{2n}=-(j)^{2n}$ and substituting for $-1=e^{j\pi(2k-1)}$ and $j=e^{j\pi/2}$ results in:

$$s^{2n} = e^{j\pi(2k-1+n)} \tag{16}$$

both sides of the equation yield:

$$s_k = e^{\frac{j\pi}{2n}(2k+n-1)}$$

$$= \cos\left(\frac{\pi}{2n}(2k+n-1)\right) + jsin\left(\frac{\pi}{2n}(2k+n-1)\right) \tag{17}$$

$$for\ k = 1, 2, \dots, 2n$$

The poles can now be determined. Fig. 24 represents the poles of a Butterworth filter of order *n* in the s-plane. The s-plane is a mathematical domain where the x-axis is real and the y-axis is imaginary in order to plot complex numbers (Calderon, 2011).

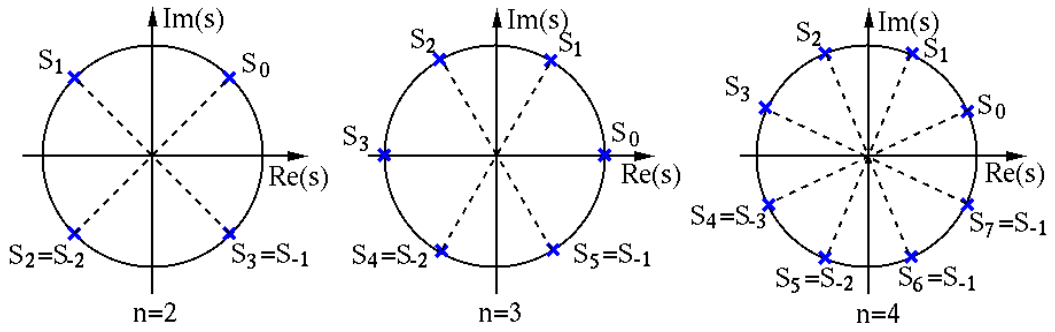


Fig. 24: Pole locations of n-th order Butterworth filter (Wang, 2019)

The poles in the left hand plane correspond to H(s). Those poles belong to a stable and causal filter and correspond to *k=1, 2, …, n*. The poles in the right hand plane correspond to *H(-s)*.

The transfer function of the normalised Butterworth filter is now developed. It can be written as:

$$H_n(s) = \frac{1}{(s - s_1)(s - s_2) \dots (s - s_n)} \tag{18}$$

It is now shown that increasing the order of the Butterworth filter is directly proportional to the number of poles. The higher the order of the filter, the more poles disturb the frequency response. It is therefore necessary to find the correct balance between the required filter order and the amount of poles present in the filter.

In most filter designs the cut-off frequency is not equal to 1, but is rather related to noise frequencies. Therefore, the filter needs to be scaled. In general there are four parameters that need to be determined to design a filter. The pass-band gain and pass-band frequency and the stop-band gain and stop-band frequency.

The units of gain are dB and the units of ω are rad/s. To define the gain at a certain frequency ω_x the (19) is used:

$$G_x = 20 \log_{10}|H(j\omega_x)|$$

$$= 20 \log_{10} \left(\frac{1}{\sqrt{1 + \left(\frac{\omega_x}{\omega_c}\right)^{2n}}} \right)$$

$$= 0 - 20 \log_{10} \sqrt{1 + \left(\frac{\omega_x}{\omega_c}\right)^{2n}}$$

$$= -10 \log_{10} \left[1 + \left(\frac{\omega_x}{\omega_c}\right)^{2n} \right] \tag{19}$$

Filling in (19) for the gain at the pass-band and stop-band frequencies, ω_p and ω_s results in:

$$G_{p,dB} = -10\log_{10}\left[1 + \left(\frac{\omega_p}{\omega_c}\right)^{2n}\right] \quad (20)$$

and

$$G_{s,dB} = -10\log_{10}\left[1 + \left(\frac{\omega_s}{\omega_c}\right)^{2n}\right] \quad (21)$$

which can be rewritten to a form where the pass-and stop-band frequencies are divided by the cut-off frequency.

$$\left(\frac{\omega_p}{\omega_c}\right)^{2n} = 10^{-G_{p,dB}/10} - 1 \quad (22)$$

and

$$\left(\frac{\omega_s}{\omega_c}\right)^{2n} = 10^{-G_{s,dB}/10} - 1 \quad (23)$$

Dividing (22) by (23) yields:

$$\left(\frac{\omega_s}{\omega_p}\right)^{2n} = \frac{10^{-G_{s,dB}/10} - 1}{10^{-G_{p,dB}/10} - 1} \quad (24)$$

(24) can be solved for the filter order *n*.

$$n = \frac{\log_{10}\left[(10^{-G_{s,dB}/10} - 1)/(10^{-G_{p,dB}/10} - 1)\right]}{2\log_{10}(\omega_s/\omega_p)} \quad (25)$$

(25) connects the filter order to the stopband and passband frequency and stopband and passband gain. This allows to define four parameters and obtain the fifth by filling in the equations.

3.3.2 Chebyshev filter

A Chebyshev filter is a digital low-pass filter, similarly to a Butterworth filter. The advantage of a Chebyshev filter is its sharper transition between pass-band and stop-band (Elprocus, 2016). The steep transition is made possible by introducing equiripple in the pass-band. The effect of ripple itself in the pass-band is negative, because it alters the power of the wanted frequencies, which would be ideally flat. However, the absolute error of the filter becomes smaller and its execution speed increases. See Fig. 25 for the difference in equiripple in Chebyshev type I and II filters. The trade-off made in a Chebyshev filter is between the steepness of the roll-off and the amplitude of the ripple. The higher the ripple factor the steeper the roll-off, but also the less stable the function becomes (Elprocus, 2016; Butzer & Jongmans, 1999).

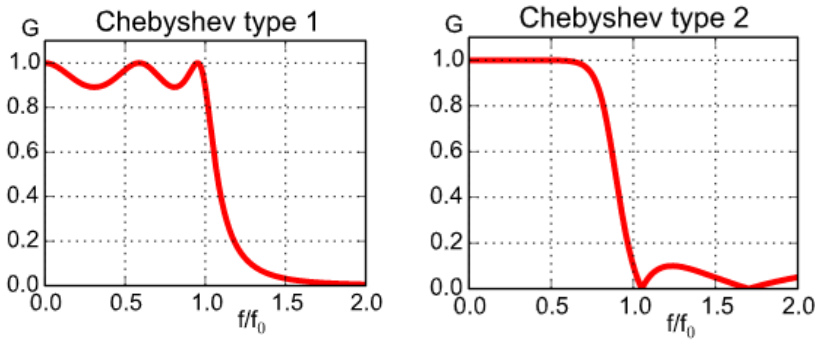


Fig. 25: Normalised Chebyshev type I and II frequency response curve (Geek3, 2016)

There are two types of Chebyshev filters. Type 1 filters have equiripple in the pass-band, this is the most commonly used type of Chebyshev filter. Type 2 includes equiripple in the stop-band. This type of filter is seldom used, because it includes unwanted frequencies. This section is therefore focused on Chebyshev type 1 filters.

The transfer function of a Chebyshev type 1 filter is very similar to the one of a Butterworth filter. The difference is within the introduction of equiripple, denoted by ϵ. The transfer functions and mathematical derivations are derived by (Elprocus, 2016). The transfer function is:

$$|H(j\omega)| = \frac{1}{\sqrt{1 + \varepsilon^2 T_n^2\left(\frac{\omega}{\omega_c}\right)}} \tag{26}$$

where ϵ is the ripple factor, ω_c is the cutoff frequency and T_n is a Chebyshev polynomial of order n.

The poles of this filter can be determined by normalizing the transfer function, substituting ω for $-js$ (using the equality: $s=j\omega$), and then setting the denominator equal to zero. Implementing the trigonometric definition of $-js=cos(\Theta)$ the equality becomes:

$$1 + \varepsilon^2 T_n^2(\cos(\theta)) = 0 \tag{27}$$

The Chebyshev polynomial, T_n, follows the rule $T_n{*}cos(\Theta)=cos(n\Theta)$ which rearranges the equation to:

$$1 + \varepsilon^2 \cos^2 n\theta = 0 \tag{28}$$

Rewriting for Θ:

$$\theta = \frac{1}{n}\arccos\left(\frac{\pm j}{\varepsilon}\right) + \frac{m\pi}{n} \tag{29}$$

where n is the filter order, and m is the integer index.

$$s_{pm} = j\cos(\theta) \tag{30}$$

$$= j\cos\left(\frac{1}{n}\arccos\left(\frac{\pm j}{\varepsilon}\right)+\frac{m\pi}{n}\right)$$

This can be rewritten to explicit complex form as:

$$s_{pm}^{\pm} = \pm\sinh\left(\frac{1}{n}\operatorname{arsinh}\left(\frac{1}{\varepsilon}\right)\right)\sin(\theta_m) + j\cosh\left(\frac{1}{n}\operatorname{arsinh}\left(\frac{1}{\varepsilon}\right)\right)\cos(\theta_m) \quad (31)$$

(31) implies that all poles of a Chebyshev filter are located on an ellipse in the s-plane. The poles of an 8th order Chebyshev type I filter are represented by the white dots in Fig. 26**.**

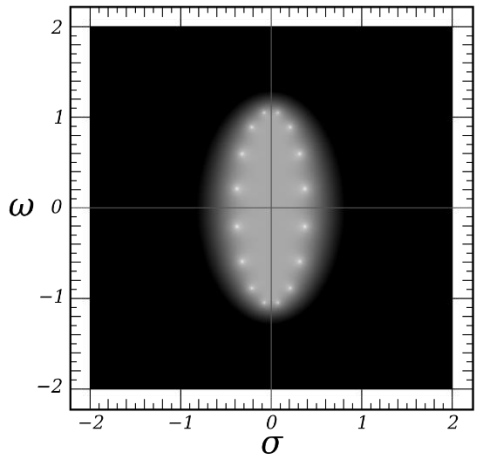


Fig. 26: Pole locations of an 8th order Chebyshev filter (InductiveLoad, 2009)

3.4 Speech processing

There are two main reasons for speech processing. The first objective is to model speech in order to minimise the amount of transferable data. A model of a signal describes a signal in a few parameters instead of a series of data points over time. If only the characteristics of a data series can be transmitted such as the frequency, phase and amplitude it would save on the amount of data transmittance. Secondly, and more important in the scope of this research, is the removal of noise, which is commonly referred to as speech enhancement. There are several methods developed for this purpose, spectral subtraction, adaptive noise cancelling, Wiener filtering and time domain harmonic scaling are the ones discussed in this section (Deller, et al., 2000).

3.4.1 Spectral subtraction

The first method of discussion is spectral subtraction. Frequencies with a large contribution to the overall signal have high power in the frequency domain. This domain is obtained using the Fourier transform or, in practice, the Fast Fourier Transform (fft). The principle of spectral subtraction (Boll, 1979; Deller, et al., 2000) is described in the next section, supported by equations (Loizou, 2013; Lyons, 2012).

Let *y*, *d* and *n* be the signal, noise, and clean speech, respectively. Then:

$$y(n) = s(n) + d(n) \tag{32}$$

By assuming that noise, *d*, is an uncorrelated process and speech and, hence, the signal are stationary it follows that:

$$\Gamma_y(\omega) = \Gamma_s(\omega) + \Gamma_d(\omega) \tag{33}$$

where $\Gamma_x(\omega)$ is the power spectrum of the subscripted process.

By estimating the power density spectrum (PDS) of the noise, $\hat{\Gamma}_d(\omega)$, it is possible to estimate the cleaned speech signal by:

$$\hat{\Gamma}_s(\omega) = \Gamma_y(\omega) - \hat{\Gamma}_d(\omega) \tag{34}$$

This states that the method would work if speech was stationary throughout the complete signal. What characterises speech are the short-time intervals on which signals are created and the short periods of silence within the signals. Those silent periods are key for spectral subtraction. The goal is to model the noise from the silent periods, and subtract its power spectrum from the noisy signal. This is called single-channel filtering. Another method is double-channel filtering where a separate receiver is used to record noise, so that a clean noise spectrum can be measured which is then subtracted from the noisy speech signal that occurred simultaneously.

The ultrasonic signals that are measured in the laboratory are comparable to single-channel measurements and the only possibility to implement spectral subtraction is to characterise the noisy period before the S-wave onset. This power spectrum will then be subtracted from the signal's power spectrum. It is assumed here that the noise is stationary throughout the complete signal, which is an not true. From Fig. 10 it can be derived that the amplitude of and tangent to the noise change constantly. Those changes are not periodically prove that the noise is not stationary.

However, if the noise is measured shortly before the S-wave onset in a small window then local stationarity can be assumed. The spectral subtraction method is now rewritten as:

$$f_y(n;m) = f_s(n;m) + f_d(n;m) \tag{35}$$

where $f_y(n;m)=y(n)w(m-n)$ in which $w(m-n)$ represents the time window.

Transforming this equation to PDS notation.

$$\hat{\Gamma}_s(\omega;m) = \Gamma_y(\omega;m) - \hat{\Gamma}_d(\omega;m) \tag{36}$$

where $\hat{\Gamma}_x(\omega; m)$ is the estimated spectrum of *x* with window size *m*.

3.4.2 Wiener filter

The Wiener filter is a linear filter that aims to produce an optimal estimation of the clean speech signal by means of the mean square estimate (MSE). Equal representations are used for speech, noise, and noisy speech as in spectral subtraction just as the local stationarity assumption (Deller, et al., 2000; Wiener, 1964). The goal of Wiener filtering is to obtain a filter with impulse response $h^{\dagger}(n)$ such that the output is an estimator $\hat{s}(n)$ given an input $s(n)$ which minimizes (Deller, et al., 2000):

$$\xi = \mathcal{L}\{[s(n) - \hat{s}(n)]^2\} \tag{37}$$

Resulting in a frequency response of:

$$H(\omega) = \frac{\Gamma_s(\omega)}{\Gamma_s(\omega) + \Gamma_d(\omega)} \tag{38}$$

where $\Gamma_s(\omega)$ and $\Gamma_d(\omega)$ are the power spectra of the processes $\underline{s}$ and $\underline{d}$, respectively.

Just as in spectral subtraction, (38**)** cannot be applied yet, because it refers to complete signal and, hence, the stationarity requirement is not fulfilled. (38) has to be manipulated to a frame-by-frame basis by using the short-term power density spectra (stPDS).

$$H(\omega; m) = \frac{\hat{\Gamma}_s(\omega; m)}{\hat{\Gamma}_s(\omega; m) + \hat{\Gamma}_d(\omega; m)} \tag{39}$$

Besides the addition of the window, denoted by m, hats are applied. The hats indicate that the stPDS are estimators. Noise is estimated during silent periods in speech, but should be estimated before S-wave arrival in ultrasonic signals. This estimation is fairly easy for ultrasonic signals, because there is a clear period of solely noise. Estimation of the speech signal is more problematic, because it is always buried in the noise.

Since $H(\omega;m)$ is now obtained in (40**)** it is possible to estimate the short-term speech spectrum from the noisy speech spectrum from its time or frequency domain by:

$$S_s(\omega; m) = H(\omega; m) S_y(\omega; m) \tag{40}$$

This filter is a spectral magnitude modifier which uses the stPDS of noisy speech to estimate clean speech. It, however, does not change the phase resulting in a clean speech estimation with the noisy speech phase.

3.4.3 Adaptive comb filtering

Adaptive comb filtering is a commonly used method whenever a signal is degraded by an aperiodic noise process. Adaptive comb filtering can be applied in either the time or frequency domain. It is also often referred to as the addition of a delayed version of the signal to itself. In audio recordings (time-domain) or in closed spaces this can cause unwanted cancelling of the signal, but for noise reduction purposes it can be beneficial. Comb filtering can also be applied in the frequency domain. In this case the noisy signal should have multiple fundamental frequencies that follow a harmonic structure. The frequency spectrum of the signal would show evenly distributed peaks of fundamental frequencies and a white

noise band. By constructing a typical comb filter the spacing between the ‘teeth’ corresponds to the fundamental frequencies. As a result, the filter passes fundamental frequencies and rejects the noise between the peaks (Deller, et al., 2000; Nehorai & Porat, 1986; Lim, et al., 1978).

Although the noise in the ultrasonic measurements is due to refraction and scattering and, thus, has a periodic fundament, since it is derived from the original signal which is a sinusoid it is still possible to use a comb filter. The frequency spectra of the ultrasonic signals have multiple peaks which could correspond with adjusted ‘tooth’-spacing of a comb filter and, hence, have a noise reduced output.

3.4.4 Time domain harmonic scaling

Time domain harmonic scaling is a noise reduction method applied in the time-domain. The value of TDHS lies within its ability to mask background noise (Deller, et al., 2000). A signal with a clear fundamental frequency is divided into windows, each covering an integer amount of fundamental periods. The windows are scaled giving one side of the window a relatively high scaling factor and the opposite side a relatively low scaling factor having a linear degradation (Malah, 1979). The windows are either decimated (compressed) or interpolated (expanded). In Fig. 27 an example is shown of the compression cycle.

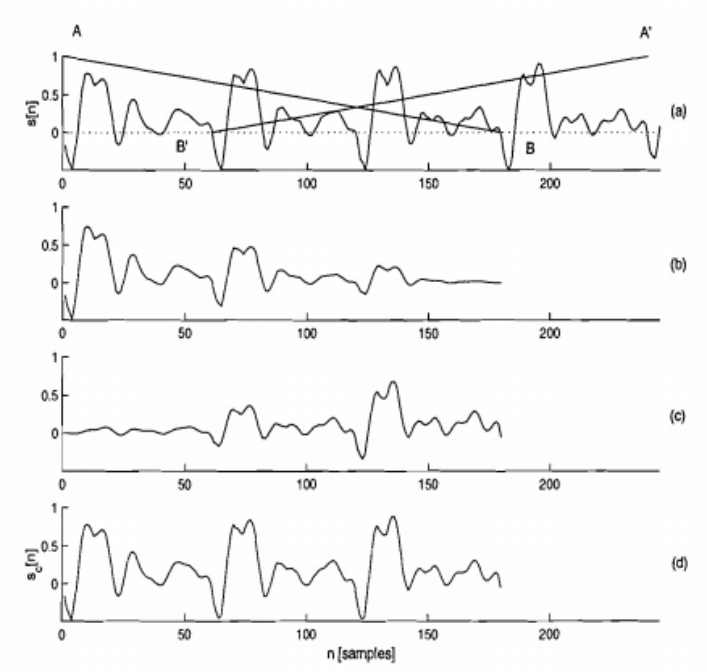


Fig. 27: Compression cycle of TDHS: (a) Original signal with the scaling factors, (b) Left hand scaled window, (c) Right hand scaled window, (d) sum of both. (N_p=60, α=0.75, N_c=180) (Burazerovic, 2000)

THDS is controlled by three parameters. N_p is the local pitch period, the length of the fundamental period. α is the time stretch factor meaning the measure of compression or expansion. If $\alpha<1$ compression takes place and if $\alpha>1$ expansion takes place. N_c is the cross-fade length referring to the window length of one scaling window. Fig. 10. shows that there

is no fundamental period which could be used for compression. Consequently, the local pitch period cannot be expressed and TDHS cannot be applied.

4. Results

In this section the methods for data generation, data evaluation and signal processing come together and provide results. Every filter is assigned its own subsection. Per filter the onset estimation and visual comparison are assigned a second level subsection as well. The results are listed comprehensively and briefly commented. The final subsection contains a summary of the results and a short comparison of the filters. A more detailed discussion about the results can be found in Section 5.

4.1 Kolmogorov-Zurbenko filter

4.1.1 Onset estimation

The averaging window of the Kolmogorov-Zurbenko filter was set to 1 µs and the number of iterations to 20. These settings produced the best results regarding the AIC onset for the KZ-filter. Next, the KZA filter was applied. The KZA smoothens the KZ results even more. The KZA is an adaptive filter which adapts to a changing mean. However, the ultrasonic time signals are standardised and have a zero mean. Therefore, the adaptive part does not have an effect on the ultrasonic signals. The KZA uses the time signal output of the KZ as its input. The optimal AIC results after KZA smoothing were obtained with a 3 µs averaging window and 20 iterations. R code related to KZ and KZA filtering is appended in Appendix 6 (10/31). The onset estimation of the KZ and the KZA filter presented in Appendix 7 (11/31)**.**

The AIC picking method estimates the onset more accurate after KZ and KZA filtering than before filtering. The onset in the synthetic data is characterised by a strong displacement in the amplitude. Smoothening over the set number of iterations using the defined averaging window the signal is smoothened enough to eliminate any abrupt displacements in the noise. Due to the large amplitude of the onset it remains visible after smoothing and allows the AIC method to pick the onset more accurately.

The smallest relative differences to the manual onset and the theoretical onset was obtained with a KZ averaging window of 1 µs and three iterations. The KZA was set to 1 µs and two iterations for optimal results. The results are plotted in in figure and Appendix 7 (11/31)**.**

From the figures it is clear that there is no big improvement in the relative differences to the manual and theoretical onset. The mean and standard deviations are calculated and denoted in Table 1.

Table 1: Mean and standard deviation of relative differences before and after KZ and KZA filtering

	Synthetic		Theoretical		Manual	
	Mean (%)	St. dev.	Mean (%)	St. dev.	Mean (%)	St. dev
KZ	7.3	5.3	11	11	7.8	7.7
KZA	4.9	1.9	13	10	8.9	7.7
No.filter	12	6.8	10	9.7	9.0	9.2

The KZ and KZA filter improve the mean and standard deviation of the relative difference to the theoretical onset. The accuracy of the onset estimation based on manual picking has decreased. The data obtained from the synthetic signal evaluation does not coincide with the findings in the real signals. It can be speculated that the noise periods in the real ultrasonic signals are to variable to smooth it out. Noise in 30 mm diameter samples has a significantly lower amplitude than 42 mm and 50 mm diameter samples. Smoothening is based on amplitude difference between noise and onset. If the amplitude of the noisy period is relatively high compared to the onset the KZ and KZA are unable to solely target rapid changes in the noise.

4.1.2 Visual comparison

In Appendix 8 (12/31) an overview of the signal classification per sample is listed. A summary of the visual classification on the processed and unprocessed real ultrasonic signals was made as well. Those results are listed in Table 2.

Table 2: Summary of the visual classification results before and after KZ and KZA filtering

	Synthetic data			Real data		
	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
KZ	0	11	19	5	10	15
KZA	0	0	30	0	2	28
No filter	24	6	0	15	11	4

After KZA filtering the signal always got worse. Due to smoothening it was manually not possible to point out any rapid changes in the signal. The amplitude still indicated a window in which the onset should be located, but the signal was too smooth to point out the exact time instance. KZ filtering gave in the majority of signals worse visual results too. For manual picking of the onset the KZ and KZA filters are bad performers. However, mathematically they allow for slightly better results compared to manual picking and slightly worse results compared to theoretical onset estimation.

4.2 Amplitude filter

4.2.1 Onset estimation

The onset in an ultrasonic signal can be recognised by a sudden change in the tangent paired with a strong amplitude displacement. An amplitude filter targets the latter by setting a threshold for the amplitude. Optimal results were obtained with a threshold of 50 % of the maximum amplitude. All the data points below that threshold were set to zero. The onset was estimated based on the resulting signal. That is a piecewise signal and therefore the onset was chosen at the first time instance this threshold was exceeded. R code related to amplitude can be found in Appendix 9 (13/31). The onset estimations per sample for the real and synthetic data are plotted in Appendix 10 (14/31).

The results show that amplitude is a clear indicator for the onset regarding the synthetic data and the real data. Except for one value the relative differences to the true onset are smaller than 5 % reflecting a relatively low standard deviation. The optimal results for the real data were obtained with an amplitude threshold of 33 % of the maximum amplitude. An amplitude filtered signal increases the accuracy and precision of the AIC results with respect to both manual and theoretical onsets. The mean and standard deviation decrease significantly. See Table 3.

Table 3: Mean and standard deviation of relative differences before and after amplitude filtering

	Synthetic		Theoretical		Manual	
	Mean (%)	St. dev.	Mean (%)	St. dev.	Mean (%)	St. dev
Amplitude	2.4	2.8	6.0	5.9	7.4	8.7
No filter	12	6.8	9.0	9.2	10	9.7

The mean of the relative differences has decreased by 3.0 p.p. on the theoretical onset case and by 2.6 p.p. on the manual onset case. This is increase in precision of roughly 25 %. Also the standard deviation has decreased 1 p.p. for the manual case and 3.3 p.p. for the theoretical case. This proves that amplitude filtering provides better results even though the signal loses information.

4.2.2 Visual comparison

In Appendix 11 (15/31) an overview of the signal classification per synthetic sample is listed. Even though the AIC results have improved significantly, due to the loss of information in the time domain it is not possible to recover the exact onset. The onset is always within a zero-part of the time signal. The only indication is the first exceeding of the threshold. Therefore all signals were assigned to class two. The real ultrasonic signals were also visually classified. The results are also shown in Appendix 11 (15/31)**.** The summary of the visual classifications is shown in Table 4.

Table 4: Summary of the visual classification results before and after amplitude filtering

	Synthetic data			Real data		
	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
Amplitude	0	30	0	1	18	11
No filter	24	6	0	15	11	4

The real signals are, in contrast to the synthetic signals, assigned to all classes instead of to class two only. If the first amplitude exceeding coincided with the manual or theoretical onset, the signals was assigned to class one. If the first amplitude exceeding deviated more than one percent from the manual or theoretical onset, but if the peak was still part of the original shear wave signal it was assigned to class two. All other cases were assigned class three. The majority of the signals has decreased in the visual picking ease. This can be connected to the loss of information in the time domain.

4.3 Butterworth filter

4.3.1 Onset estimation

Four types of Butterworth filters are applied, the low-pass, high-pass, band-pass and band-stop. The cut-off frequency of the low-pass filter was set to 1.5 times the dominant frequency which was measured in the frequency spectrum. The high pass filter was set to 0.55 times the dominant frequency. The first cut-off frequency of band-pass filter was set to 0.7 times the dominant frequency and the second cut-off was set to 2 times the dominant frequency. The band-stop filter's lower and upper cut-off frequencies were set to 0.55 and 1.2 times the dominant frequency, respectively. Those settings were found by frequency spectrum analysis and trial and error. They provided near optimal results regarding the AIC onset estimation. R code related to Butterworth filtering is appended in Appendix 12 (16/31). Of every Butterworth filter type a scatterplot was generated with respect to the synthetic data, see Appendix 13 (17/31)**.**

The Butterworth filters decreased the mean of the relative difference slightly. This was calculated based on the results represented in the scatterplots of the Butterworth filters. The means and standard deviations are shown in Table 5.

The results obtained from the real ultrasonic data are also plotted in Appendix 13 (17/31)**.** The cut-off frequencies of the high pass and low-pass filter were both set the dominant frequency. The band-stop and band-pass filter's cut-off frequencies provided optimal results with a bandwidth from 0.7 up to 2 times the dominant frequency. The mean and standard deviations of the synthetic and real data are summarised in Table 5.

Table 5: Mean and standard deviation of relative differences before and after Butterworth filtering

	Synthetic		Theoretical		Manual	
	Mean (%)	St. dev	Mean (%)	St. dev	Mean (%)	St. dev
Butter.PB	10	5.1	12	8.0	16	10
Butter.SB	10	7.3	6.0	6.5	9.1	9.3
Butter.HP	11	6.3	8.0	7.0	10	8.5
Butter.LP	11	5.9	14	8.2	18	9.1
No.filter	12	6.8	9.0	9.2	10	9.7

The standard deviation also slightly decreased for the low-pass, high-pass, and pass-band filter. The stop-band filter increased in standard deviation. This can be connected to the resulting frequency spectra of the signals. The former three filters maintain the dominant frequency after filtering whereas the stop-band filter filters out the dominant frequency. It can be speculated that the dominant frequency is a requirement for the AIC to be accurate. On the contrary, the difference in standard deviation between the Butterworth stopband and No-filter case is only 0.5 p.p. which also shows that even after eliminating the dominant frequency a time signal was reproduced that allowed for onset estimation by AIC even better than when the dominant frequency is maintained. This is an interesting point of discussion.

4.3.2 Visual comparison

In Appendix 14 (20/31) a visual classification of all synthetic samples before and after Butterworth filtering is listed. Signals after Butterworth high-pass processing were mostly assigned to class one. The signals after Butterworth low-pass and band-stop were mainly assigned to class two and three. The band-pass signals were roughly divided over class one and two. A short overview of the results is given in Table 6.

Table 6. Summary of the visual classification results before and after Butterworth filtering

	Synthetic			Real		
	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
Butterworth HP	25	5	0	25	5	0
Butterworth BP	11	16	3	11	16	3
Butterworth BS	5	6	19	5	6	19
Butterworth LP	2	15	13	2	15	13
No filter	24	6	0	24	6	0

These results allow for speculation that frequency components below the dominant frequency do not have a strong impact on the manual picking ease of the signal. Filtering out the dominant frequency or the frequencies above the dominant frequency provides time signals that decreased the manual picking ease.

Combining the onset estimation results and the visual classification results showed that high-pass filtering improved the visual picking ease of the signal and increased the accuracy of the AIC picking method. However, band-pass, band-stop, and low-pass filters performed better with respect to onset estimation by AIC rather than manual picking.

4.4 Chebyshev1 filter

4.4.1 Onset estimation

The Chebyshev type 1 filter was applied as a low-pass, high-pass, band-pass and band-stop filter. The Chebyshev filters were designed using a pass-cut-off-frequency (F_{pc}) and stop-cut-off frequency (F_{cs}). F_{pc} is the frequency where the roll-off starts and F_{sc} is the frequency where the roll-off stops. This was required by the function for Chebyshev filtering in R. The equiripple was set to 0.5 dB and the gain to 29 dB for the synthetic and real data.

The Chebyshev1 filter settings applied to the synthetic data were as follows: The F_{pc} and F_{sc} were set to 0.3 and 1 times the dominant frequency, respectively, for the low-pass variant. The high-pass variant had a F_{pc} of 1 and a F_{sc} of 0.7 times the dominant frequency. The Chebyshev1 band-pass filter had a passband interval from 1 times the dominant frequency until 1.4 times the dominant frequency. The filter settings were optimised by trial and error with respect to the AIC mean and standard deviation. The low-pass, high-pass and band-pass filter provided useful results. By combining the results of the low-pass and high-pass filter in the roll-off area it was tried to approach the results of a band-stop filter. The results of this experiment were not useful for evaluation. R code related to the Chebyshev filter is appended

in Appendix 15 (21/31). In Appendix 16 (22/31) the results of the synthetic data onset estimation before and after Chebyshev1 are plotted per sample.

The Chebyshev1 low-pass parameters applied to the real data were 0.3 times the dominant frequency and 0.7 times the dominant frequency for the F_{sc} and F_{pc}, respectively. The high-pass filter provided optimal results with a F_{sc} of 0.5 times the dominant frequency and a F_{pc} of 1 times the dominant frequency. The band-pass filter was reconstructed by applying the Chebyshev1 high-pass filter over the output of the Chebyshev1 low-pass filter. The low-pass filter was set to F_{pc}=4 and F_{sc}=5 times the dominant frequency. The high-pass filter was set to F_{sc}=1 and F_{pc}=1.2 times the dominant frequency. The results of the three filter types applied to the real data are shown in Appendix 16 (22/31)**.** A summary of the results of the synthetic and real data is shown in Table 7. They are expressed in the mean and standard deviation of the relative differences.

Table 7: Mean and standard deviation of relative differences before and after Chebyshev1 filtering

	Synthetic		Manual		Theoretical	
	Mean (%)	St. dev	Mean (%)	St. dev	Mean (%)	St. dev
Cheby1 LP	7.4	1.8	19	3.4	20	8.7
Cheby1 HP	11	6.3	6.8	6.1	8.0	8.8
Cheby1 BP	11	7.0	12	5.6	13	8.8
No filter	12	6.8	9.0	9.2	10	9.7

All filters have a positive influence on the synthetic data. The low-pass filter performs best in the synthetic case, but worst in both real data cases. It can be speculated that the AIC does not function on a signal with a low number of frequency components in the same range. This can be enlightened from the other side as well. The AIC requires, besides the dominant frequency, also frequencies that are related to noise of much higher frequency, because that is what creates the rapid change in tangent in the time domain. When the frequencies are too close together the change in tangent in the time signal might become unrecognisable.

4.4.2 Visual classification

In Appendix 17 (25/31) the visual classifications of all synthetic and real signals before and after Chebyshev1 filtering are listed. The results reflect that none of the Chebyshev1 filters improved the visual classification of the synthetic signals. In the majority of cases the visual classification became worse.

The visual classification of the real ultrasonic signals reflects exactly the same results as the synthetic data. In general the time signals decreased a lot regarding the manual picking ease. It can, therefore, be stated that in the synthetic nor real signal case none of the three filters improved the manual picking ease. A brief overview of the results shown in Table 8 supports that statement.

Table 8: Summary of the visual classification results before and after Chebyshev1 filtering

	Synthetic data			Real data		
	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
Cheby1HP	2	2	26	10	9	11
Cheby1 LP	1	8	21	0	0	30
Cheby1 BP	0	1	29	0	4	26
No filter	24	6	0	15	11	4

4.5 Spectral subtraction

4.5.1 Onset estimation

Two parameters were determined for spectral subtraction. The first one was the frame length expressed in microseconds, the second one was the number of frames used for the noise average. The frame length was varied from 0.5 µs to 10 µs and the number of frames for averaging from 1 to 10. The best synthetic data results were obtained when the noise was averaged over 5 frames of 1 µs per frame. R code related to spectral subtraction is appended in Appendix 18 (26/31). The onset estimation results are visualised in Appendix 19 (27/31)**.** The number of estimations above 15 % relative difference decreased and the number of estimation below 5 % relative difference increased, especially with the 30 mm and 50 mm diameter samples.

The optimal results for the real data onset estimation were obtained using 5 frames for noise averaging of 2 µs length per frame. The results clearly show more AIC onsets picked at a relative difference higher than 20 % after spectral subtraction compared to before spectral subtraction with respect to the real data. The same applies to the theoretical onset case. In theory, spectral subtraction could perform better than it did here. However, due to a standard frame length and an equal number of frames for every sample it does not always capture a complete noise period. In Table 9 the mean and standard deviations of all cases after spectral subtraction are summarised.

Table 9: Mean and standard deviation of relative difference before and after spectral subtraction

	Synthetic		Manual		Theoretical	
	Mean (%)	St. dev	Mean (%)	St. dev	Mean (%)	St. dev
Spectral sub.	7.5	6.5	19	14	18	17
No filter	12	6.8	9.0	9.2	10	9.7

The synthetic data has 6-8 noise components which start at the first time instance of the data series. Therefore, the noise period was fairly easy to capture and hence, the synthetic results improved by spectral subtraction. In the real data cases the noise components started at different unknown time instances and therefore it became difficult to capture a complete noise period with a standard frame length and number of frames for noise averaging.

4.5.2 Visual classification

A visual classification of the processed synthetic signals is given in Appendix 20 (28/31). There is one case where spectral subtraction provided a visually better time signal to extract the onset from than before filtering. In the rest of the cases the time signals remained in the same class or got worse. The reconstructed time signal were never continuous. This was caused by the magnitude of the average noise spectrum. This magnitude is subtracted from every frame in the signal. In any two adjacent frames different frequencies and amplitudes were present and, therefore, they all have a different phase spectra. Subtraction of the average noise spectrum adjacent frames to be end up with non-connecting magnitude and resulted in a discontinuous reconstruction of the time frame.

A visual classification was also produced regarding the real signals, see Appendix 20 (28/31). Except for a single sample spectral subtraction could not improve the visual classification of the real data. The reason for this is identical to the one for the synthetic data. Discontinuous time signals make it more difficult to recognise the onset. The results of the visual comparison of synthetic and real signals are summarised in Table 10.

Table 10: Visual classification overview of spectral subtraction

	Synthetic data			Real data		
	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
Spectral sub.	9	11	10	6	8	16
No filter	24	6	0	15	11	4

4.6 Comb filter

4.6.1 Onset estimation

The comb filter adds a delayed version of a time signal to itself. The delay in units was set to 3 µs for the synthetic data. This value was estimated based on the distance between the local maxima and minima in the noise in the time domain. It was tried to shift a local maximum exactly over a nearby minimum in order to cancel out the noise period. The distance was visually determined and optimized by trial and error. R code related to comb filtering is appended in Appendix 21 (29/31). The results are shown in Appendix 22 (30/31).

The synthetic onset could be estimated accurately after comb filtering. Most relative differences were within a 2.5 % range and 2 out of 30 samples were higher than 10 % relative difference. This could point out uniformity in the noise period in the synthetic data, which would be a reasonable assumption since it was created with 6-8 noise components only.

For the real data the optimal time shift for the comb filter was also set by trial and error. AIC results obtained from a time shift of 0.75 µs, 1 µs, and 2 µs were evaluated, but not did differ significantly. It was chosen that optimal results were obtained with a 0.75 µs time shift regarding the mean and relative difference. Those results are also plotted in Appendix 22 (30/31)

More onsets were estimated above the 20 % line in the manual case after filtering than before. The same applied to the theoretical case. The reason for this could be the non-uniformity of the real data noise. There were tens of frequency components which all start at different times, in contrast with the synthetic data where all noise started at the first time instance. A generalised time shift does not work for all signals. In Table 11 the comb filter statistics are listed.

Table 11: Mean and standard deviation of the relative differences before and after comb filtering

	Synthetic		Manual		Theoretical	
	Mean (%)	St. dev	Mean (%)	St. dev	Mean (%)	St. dev
Comb	2.3	4.4	11	8.4	13	12
No filter	12	6.8	9.0	9.2	10	9.7

4.6.2 Visual classification

The signals after comb filtering were visually classified. The results of the synthetic signal classification are shown in Appendix 23 (31/31)**.** Comb filtered signals only rarely scored a three in the visual classification system. The important part of the signal, the onset, remained visible most of the time and was located at the same time instance as before filtering.

Real signals after comb filtering often appeared as good as before filtering. The noise periods looked more uniform, amplitudes got smaller, and the number of small oscillations increased. If the onset could be manually estimated with decent certainty before processing than it would be very likely that the onsets were to be manually determined with decent certainty after processing. A summary of the visual evaluation results is denoted in Table 12.

Table 12: Summary of the visual classification results before and after comb filtering

	Synthetic data			Real data		
	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
Comb	17	11	2	10	15	5
No filter	24	6	0	15	11	4

Referring to the classification criteria of class one the onset should be clearly recognisable and no uncertainty can exist between two points in the time domain further away than 1 % of their time instance. The second class was defined as signals where expertise is required to pick between two possible onsets which are further apart than 1 % of their time instance. Table 12 shows that the onset of a comb filtered signal is almost as likely to be found as in an original signal, but more signals require expertise. Two samples in the synthetic data and one sample in the real data were assigned to class three more than before filtering. Concluding that the comb filter still provides acceptable time domain signals.

4.7 Summary of results

4.7.1 Onset estimation

The mean relative difference and standard deviation were calculated for every filter. The filters were ranked with respect to mean relative difference. The standard deviation was taken into account when the means of two or more filters were closer to each other than 1 %.

In Fig. 28 the mean and standard deviation of the relative differences to the true onset before filtering are highlighted with a dashed line and a shaded area, respectively. All filters have a positive effect on the synthetic signals regarding the onset estimation by AIC. The ranking of the filter with respect to the synthetic data can be found in Table 13.

Table 14 and Fig 29 show that, in contrast with the synthetic data results, not all filters performed better than the no-filter case. Six filters have a lower mean and standard deviation.

Table 15 and Fig 30 show that four signal processing techniques improve the mean and standard deviation of the real signals with respect to the manual onset. That is two filters less than the theoretical case.

Table 13: Ranking of signal processing techniques with respect to the true onset

Rank	Filter type	Mean (%)	St. dev (p.p.)
1	Amplitude	2.4	2.8
2	Comb	2.3	4.4
3	KZA	4.9	1.9
4	Cheby1.LP	7.4	1.8
5	KZ	7.3	5.3
6	Cheby1.HP	7.2	5.4
7	Spectral.sub.	7.5	6.5
8	Butter.PB	10	5.1
9	Butter.SB	10	7.3
10	Butter.HP	11	6.3
11	Butter.LP	11	5.9
12	Cheby1.PB	11	7.0
13	No.filter	12	6.8

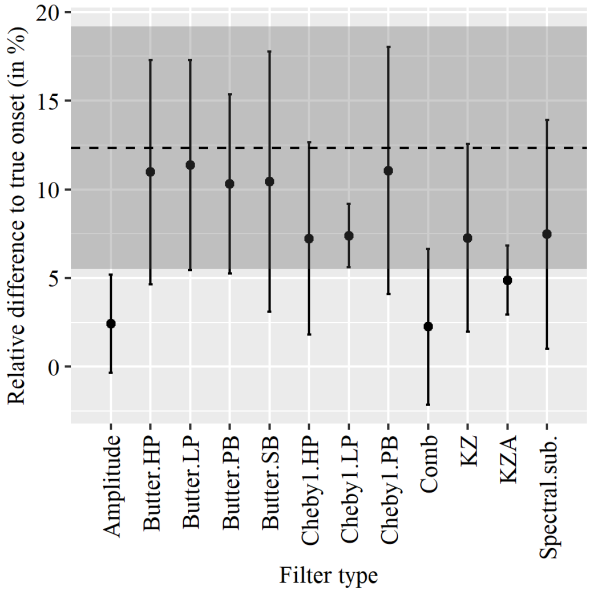


Fig. 28:Mean and standard deviation of the synthetic signals per filter. Dashed line and shaded area are mean and standard deviation of no-filter case, respectively.

Table 14: Ranking of signal processing techniques with respect to the manual onset

Rank	Filter type	Mean (%)	St.dev p.p
1	Amplitude	6.0	5.9
2	Butter SB	6.0	6.5
3	Cheby1 HP	6.8	6.1
4	Butter HP	8.0	7.0
5	KZ	7.8	7.7
6	KZA	8.9	7.7
7	No filter	9.0	9.2
8	Cheby1 PB	12	5.6
9	Butter PB	12	8.0
10	Comb	11	8.4
11	Butter LP	14	8.2
12	Cheby1 LP	19	3.4
13	Spectral sub.	19	14

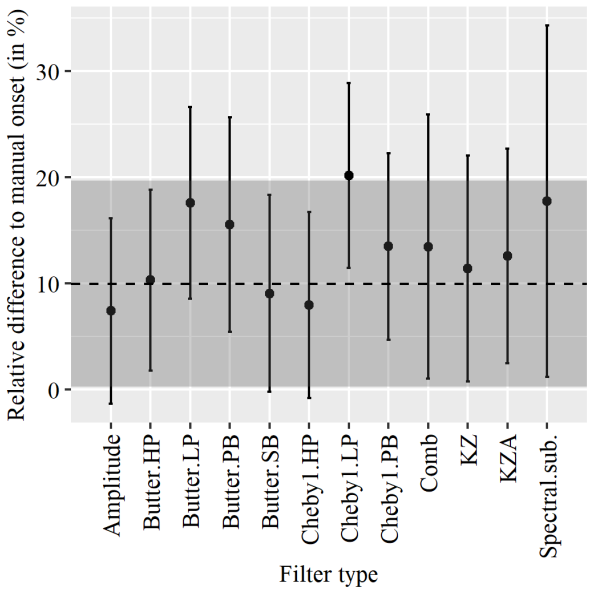


Fig. 29: Mean and standard deviation of the real signals per filter with respect to the manual onset. Dashed line and shaded area are mean and standard deviation of no-filter case, respectively.

Table 15: Ranking of signal processing techniques with respect to the theoretical onset

Rank	Filter type	Mean (%)	St. dev p.p.
1	Amplitude	7.4	8.7
2	Cheby1 HP	8.0	8.8
3	Butter SB	9.1	9.3
4	Butter HP	10	8.5
5	No filter	10	9.7
6	KZ	11	11
7	KZA	13	10
8	Cheby1 PB	13	8.8
9	Comb	13	12
10	Butter PB	16	10
11	Spectral sub.	18	17
12	Butter LP	18	9.1
13	Cheby1 LP	20	8.7

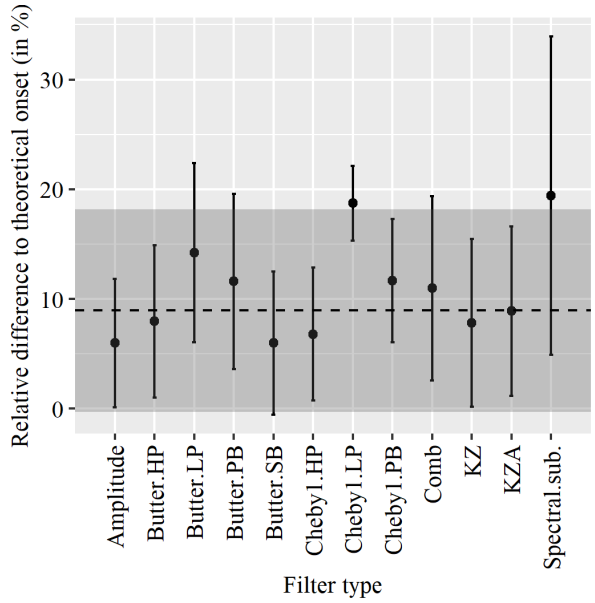


Fig. 30: Mean and standard deviation of the real signals per filter with respect to the theoretical onset. Dashed line and shaded area are mean and standard deviation of no-filter case, respectively.

All filters provided better onset estimation results with respect to mean and standard deviation than the no-filter case regarding the synthetic data. The synthetic data was created to verify the performance of the filters which could then be reflected in the filters applied to the real ultrasonic data. In the filter performance with respect to the theoretical onset there

are six filters ranked higher than the no-filter case. In the manual onset case there are four. This is a sign that the synthetic data is either not 100 % representative or that the picking methods based on the theoretical and manual onset were inaccurate and cause the deviations. The former would be more plausible and of higher impact than the latter, because the simplicity of the synthetic signals made signal enhancement by filter parameter generalisation easier. Due to the complexity of the real signals the generalised filter parameters did often deteriorate the signal, because every signal is unique, especially between sample sizes signals are not alike. The theoretical relationship between P- and S-waves was mathematically proven (see Section 2.2.1) and the manual picking was carried out by two professionals and one novice, which in both cases could lead to small deviations in the filter performance, but not as large as they are shown in the results.

4.7.2 Visual comparison

In order to rank the filters on visual classification results two ranking systems were considered. The first system was based on the fact that expertise could lead to correct estimations of class one and two. The number of filters in those classes were added up and the highest number was assigned rank one. When two filters obtained an equal score the filter with the largest number of signals in class one would be ranked higher. The second system was purely based on the number of filters per class individually. The filters were ordered based on the number of signals in class one. Signals in class two would be decisive if an equal score would be obtained in class one. Neither of the systems is wrong. It was chosen to use the first ranking system for evaluation.

Visual comparison indicates that only a Butterworth high-pass filter improves the manual picking ease of the synthetic signals, see Table 16. All other filters deteriorate the time domain signal. The visual comparison test of the real data shows the same behaviour as the synthetic data results, see Table 17. Only a Butterworth high-pass filter could improve the manual picking ease of the signals. All other filters have a negative effect on the manual picking ease.

Table 16: Ranking of filters with respect to the visual classification results of the real data

Rank	Filter type	Class 1	Class 2	Class 3
1	Butterworth HP	25	5	0
2	No filter	24	6	0
3	Amplitude	0	30	0
4	Comb	17	11	2
5	Butterworth BP	11	16	3
6	Spectral sub.	9	11	10
7	Butterworth LP	2	15	13
8	Butterworth BS	5	6	19
9	KZ	0	11	19
10	Cheby1 LP	1	8	21
11	Cheby1HP	2	2	26
12	Cheby1 BP	0	1	29
13	KZA	0	0	30

Table 17: Ranking of filters with respect to the visual classification results of the real data

Rank	Filter type	Class 1	Class 2	Class 3
1	Butterworth HP	17	9	4
2	No filter	15	11	4
3	Comb	10	15	5
4	Butterworth BS	12	9	9
5	Cheby1 HP	10	9	11
6	Amplitude	1	18	11
7	KZ	5	10	15
8	Spectral sub.	6	8	16
9	Cheby1 BP	0	4	26
10	Butterworth BP	0	4	26
11	KZA	0	2	28
12	Butterworth LP	0	0	30
13	Cheby1 LP	0	0	30

5. Discussion

5.1 Interpretation and implications

Four main findings are interpreted from the results. The first main finding indicates that an amplitude filter decreases the mean and standard deviation of the relative differences to the S-wave onset the most of all filters. The results prove that solely looking for a change in amplitude, disregarding variance, gives a better approximations of the S-wave onset than traditional onset estimation by AIC.

The second important finding is that a high-pass filter of Chebyshev1 or Butterworth increases the performance of the AIC picking method. It is speculated that if the dominant frequency indeed contains information about the initial shear wave signal the frequencies just below the dominant frequency make the onset less recognizable for the AIC picking method. And, that frequencies higher than the dominant frequency are required, because they make a change in tangent in the time domain signal visible.

The third main finding is the correlation between the performance of the Chebyshev type 1 filter and the Butterworth filter, based on the real data results. The high-pass, low-pass, and band-pass variants of both filters appear as couples in the ranking, concluding that the steeper roll-off of the Chebyshev filter or the maximally flat pass band of the Butterworth filter does not have an advantage over the other. It could also imply that the Chebyshev band-stop filter would provide better results than the no-filter case, because the Butterworth band-stop filter provides better results than the no-filter case. On the contrary, it can also be speculated that the results will be approximately the same as the Butterworth band-stop filter and are, therefore, not necessarily worth testing.

The fourth and most unexpected result is the performance of the Butterworth band-stop filter. Even though the dominant frequency is filtered out the mean and standard deviation of the relative difference decreased. This implies that the dominant frequency does not contain relevant information about the target signal. If that is true then the fundament of this thesis is endangered, because all frequency filter parameters are automated with respect to the dominant frequency.

5.2 Limitations

5.2.1 Subsetting the data

A subset of the synthetic and real data was taken. The subsets only contained 30 mm, 42 mm and 50 mm diameter samples. The effectiveness of the filters was not tested on 14 mm, 20 mm, 60 mm, and 100 mm diameter samples. Time signals appear different per sample length with respect to noise, amplitude, and onset. Therefore, the filter ranking might not be generalizable for all sample length.

5.2.2 Accuracy of the manual onset

The manual pick was performed by three individuals whom independently picked the S-wave onsets. The estimated onsets were in most cases aligned between the three individuals, but in multiple samples two out of three individuals picked the same onset and the third one picked a different one. Sometimes there were even three different picks. It was decided to average the onsets per sample. For the cases where the three picks were nearly equal this gave representative values, but for the cases where three different picks were chosen it does not take into account which picks are correct and incorrect. The average value from those signals is not representative. Eliminating picks based on one's interpretation would bias the results, therefore it was chosen to average them. The liability of the manual onset dataset is controversial.

5.2.3 Filter parameter optimization

The automation of most filters is not fully optimized. The KZ and KZA filter have a set averaging window and number of iterations. The filter would potentially perform better if the number of iterations is adaptable to amplitude of the noise. Plausibly, low amplitude noise does not need an equal number of iterations as high amplitude noise. The amplitude of the noise increases as sample size increases and, therefore, the results of the KZ and KZA could be improve if the parameters were sample specific.

Spectral subtraction was expected to produce better results, because it aims to capture exactly one noise period. The biggest problem encountered in spectral subtraction was capturing noise right before the onset, because the onset is unknown. It was, therefore, chosen to start the noise selection from the first time instance any amplitude displacement larger than 0.01 mV occurred. The selected noise might not have been sufficiently representative for noise at the onset. A solution for this problem was not found, but an improvement of noise selection could be made by using the theoretical relationship between P- and S-waves for an estimation of the S-wave onset. Noise could be captured right before the theoretical S-wave onset. This must be executed with caution to avoid capturing the S-wave onset in the noise selection. The second problem related to spectral subtraction is the discontinuous reproduced signal. Discontinuity makes it difficult to visually determine the onset and it causes jumps in the variance which is bad automated onset picking by AIC

The last filter that could produce better results after automation optimization is the comb filter. The time shift is now set to a specific time based on visual judgement and trial and error. When local maxima and minima are not coinciding the noise has no chance to cancel out. A time shift based on local maxima and minima would most likely increase the performance of the comb filter. An attempt to implement this was made, but the selected local optima were not the correct ones and therefore the automation gave bad results.

5.3 Future research

5.3.1 Signal analysis

In being able to modify the ultrasonic signals in order to produce a new signal that contains enough information to detect the onset by automation and little enough information to reject noise a solid understanding of the system is required. In this case the system is a rock sample. Since only the in- and output are known and the system is unknown this future study could be referred to as an inverse problem (Tarantola, 1987). Research questions that could be asked include "*Which part of the frequency spectrum contains indispensable information?"*, *"What causes the initial dominant frequency of 1 MHz to decrease to roughly 100 kHz in all samples larger than 14 mm diameter?"*, *"How can a system be mathematically described?"*, and *"Is there a trend in the mathematical description of the system that uncovers essential information regarding the S-wave onset?"*. Further studies, which take these questions into account, will need to be undertaken to develop a complete understanding of the influence of the rock on the signal.

Frequency analysis was carried out in this studies to acquire information about the noise and target signal. The signals were cut right before the onset and a frequency spectrum of either part of the signal was calculated. The frequency spectra were compared in the hope that it would reveal information about the target signal. Despite the expectation the separate parts did not show useful differences.

5.3.2 Quality of synthetic data

In further investigations it might be possible to use a more sophisticated synthetic signal. Reflections and refractions could be represented by different noise onsets depending on the sample length and diameter. This would allow for more realistic testing of the processing techniques. Throughout this research it was chosen to start all noise components at $t=0$. Due to the damped characteristic of the noise component the onset was easy to detect in large samples where the onset appeared when all the noise was damped, but difficult to detect in short samples where the noise was not damped as much at onset arrival. Further research would require deep analysis of reflections and refractions and possibly numeric modelling of reflections and refractions in a rock sample.

6. Conclusions

The aim of this study was to evaluate signal processing techniques with respect to ultrasonic shear wave signals in rocks. The overarching objective was to improve the precision and accuracy of onset estimation by the AIC picking criterion. The main research question to answer was: "*How do signal processing methods influence the ultrasonic shear wave onset estimations?"* This question is answered by: *"Of all methods tested during this study the amplitude filter, Butterworth high-pass filter, Chebyshev type 1 high-pass filter, and Butterworth stop-band filter positively influence the shear wave onset estimation by AIC."*

The filters parameters are not fully optimized and could possibly improve the results even more. Also, the understanding that the dominant frequency contains essential information about the onset is called into question and should be further investigated.

The generalisability of the results are subject to a certain limitations, namely, the processing techniques were only tested on 30 mm, 42 mm, and 50 mm diameter samples. Also, the synthetic data was not fully representative. Notwithstanding these limitations , the study suggests that there is potential benefit in the application of signal processing techniques to ultrasonic shear wave signals measured on rocks in order to detect the S-wave onset.

Bibliography

Agarwal, A. & Lang, J., 2005. *Foundations of Analog and Digital Electronic Circuits.* Massachusetts: Morgan Kaufmann.

Akaike, H., 2nd International Symposium on Information Theory. *Information theory and an extension of the maximum likelihood principle.* Budapest, Akademiai Ki à do.

Alejano, L., Arzùa, J., Castro-Filgueira, U. & Kiuru, R., 2018. *Scale Effect of Intact Olkiluoto Gneissic Rocks Through Uniaxial Compressive Testing and Geophysical Measurements,* Olkiluoto: Posiva.

Boll, S., 1979. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing* , 27(2), pp. 113-120.

Brown, R. & Hwang, P., 2012. *Introduction to Random Signals and Applied Kalman Filtering.* 4th red. U.S.A.: John Wiley and Sons Ltd..

Burazerovic, D., 2000. *Time-scale modification for speech coding.* Eindhoven: Technische Universiteit Eindhoven.

Butterworth, S., 1930. On the Theory of Filter Amplifiers. *Experimental Wireless and the Wireless Engineer,* pp. 536-541.

Butzer, P. & Jongmans, F., 1999. P.L. Chebyshev - A guide to his Life and Work. *Journal of Approximation Theory,* Volume 96, pp. 111-138.

Calderon, D., 2011. *ConneXions.* [Online] Available at: http://pilot.cnxproject.org/content/collection/col10064/latest/module/m34855/latest [Geopend 27 8 2019].

Cavanaugh, J. E. & Neath, A., 2019. The Akaike Information Criterion: Background, derivation, properties, application, interpretation, and refinements.. *Wiley Interdisciplinary Reviews: Computational Statistics,* e1460(3), p. 11.

Deller, J. R. J., Hansen, J. H. & Proakis, J. G., 2000. *Discrete-time processing of speech signals.* New York: Macmillan Publishing Co..

Elprocus, 2016. *Different type of Chebyshev filters with calculations.* [Online] Available at: https://www.elprocus.com/types-of-chebyshev-filters/ [Geopend 20 April 2019].

Garnero, E., 2010. *Garnero_Splits.* [Illustraties].

Geek3, 2016. *Frequency response curves of four linear analog filters: Butterworth filter, Chebyshev filter of type 1 and 2 and Elliptic filter, each one as 5th order filter.*. [Illustraties].

Gercek, H., 2007. Poisson's ratio values for rocks. *International Journal of Rock Mechanics and Mining Sciences,* pp. 1-13.

Golbon-Haghighi, M.-H., Zhang, G., Li, Y. & Doviak, R. J., 2016. Detection of Ground Clutter from Weather Radar Using a Dual-Polarization and Dual-Scan Method. *Amtosphere,* 7(6), p. 83.

Gray, R. M. & Davisson, L. D., 2004. *An introduction to Statistical Signal Processing.* Cambridge: Cambridge University Press.

Hayes, M. H., 1996. *Statistical digital signal processing and modelling.* New York: John Wiley.

InductiveLoad, 2009. *Wikimedia.* [Online] Available at: https://commons.wikimedia.org/wiki/File:Chebyshev_Type_I_Filter_s-Plane_Response_(8th_Order).svg#/media/File:Chebyshev_Type_I_Filter_s-Plane_Response_(8th_Order).svg
[Geopend 15 May 2019].

Jackson, L. B., 1991. *Signals, sytems, and transforms.* United States of America: Addison-Wesley Publishing Company, Inc..

Kahraman, S., 2002. The effects of fracture roughness on P-wave velocity.. *Engineering Geology,* 3-4(63), pp. 347-350.

Kaiwen, X. & Wei, Y., 2015. Dynamic rock tests using split Hopkinson bar system - A review. *Journal of Rock Mechanics and Geotechnical Engineering,* pp. 27-59.

Kalman, R., 1960. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering,* 82(1), pp. 35-45.

Langenberg, K.-J., Marklein, R. & Mayer, K., 2012. *Ultrasonic Nondestructive Testing of Materials - Theoretical Foundations.* Boca Raton: CRC Press.

Lim, J., Oppenheim, A. & Braida, L., 1978. Evaluation of an adaptive comb filtering method for enhancing speech degraded by white noise addition. *IEEE Transactions on Acoustics, Speech, and Signal Processing,* 26(4), pp. 354-358.

Loizou, P. C., 2013. *Speech Enhancement: Theory and Practice, Second Edition.* 2e red. Boca Raton: CRC Press.

Lyons, J., 2012. *Speech Enhancement tutorial: Spectral Subraction.* [Online] Available at: http://practicalcryptography.com/miscellaneous/machine-learning/tutorial-spectral-subraction/
[Geopend 27 8 2019].

Malah, D., 1979. Time-domain algorithms for harmonic bandwidth reduction and time scaling of speech signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing,* 27(2), pp. 121-133.

Mathews, K., Hoek, E., Wyllie, D. & Stewart, S., 1981. Prediction of stable excavation spans at depths below 1000m in hard rock mines. *Tech Report.*

McGraw-Hill & Parker, S. P., 2002. *McGraw-Hill Disctionary of Scientific and Technical Terms.* 6e red. New York: McGraw-Hill Education.

Naser, A. A.-S., 2004. Effects of testing methods and conditions on the elastic properties of limestone rock. *Engineering Geology,* pp. 139-156.

Nehorai, A. & Porat, B., 1986. Adaptive comb filtering for harmonic signal enhancement. *IEEE Transactions on Acoustics, Speech, and Signal Processing,* 34(5), pp. 1124-1138.

O'Donnel, D. R. M., 2009. Radar Systems Engineering. *IEEE Aerospace and Electronic Systems Society,* Issue University of Newhampshire.

Panagos, D. A., 2014. *Discrete-time signals and systems - Butterworth filter design review.* [Online]
Available at: https://www.adampanagos.org/dt-signals-and-systems

Poisson, S., 1831. Mémoires sur la propagation du mouvement dans les milieux élastiques. *MéMoires de l'Académie des Sciences de l'Institut de France,* pp. 549-605.

Richards, M. A., 2014. *Fundamentals of Radar Signal Processing.* 2e red. U.S.A.: McGraw-Hill Professional Engineering.

Rohling, H., 1983. Radar CFAR Thresholding in Clutter and Multiple Target Situations. *IEEE Transactions on Aerospace and Electronic Systems,* 19(4), pp. 608-621.

Sabbagh, J., Vreven, J. & Leloup, G., 2002. Dynamic and static moduli of elasticity of resin-based materials. *Dental materials,* pp. 64-71.

Scensor, D., 1973. The generalized doppler effect and applications. *Journal of the Franklin Institute,* 295(2), pp. 103-116.

Schoenberg, M. & Sayers, C., 1995. Seismic anisotropy of fractured rock. *GEOPHYSICS,* pp. 204-211.

Sedlak, P., Hirose, Y., Enoki, M. & Sikula, J., 2008. Arrival time detection in thin multilayer plaets on the basis of the Akaike Information Criterion. *Acoustic Emission,* pp. 182-188.

St-Onge, A., 2011. Akaike information criterion applied to detecting first arrival times on microseismic data. *SEG Technical Program Expanded Abstracts 2011,* pp. 1658-1662.

Storr, W., 2014. *Butterworth filter design.* [Online] Available at: https://www.electronics-tutorials.ws/filter/filter_8.html [Geopend 15 May 2019].

Tarantola, A., 1987. *Inverse Problem Theory.* Amsterdam: Elsevier Science B.V..

The Gale Enceclopedia of Science, 2002. *Ultrasonics.* [Online] Available at: https://www.encyclopedia.com/science-and-technology/physics/physics/ultrasonics [Geopend 25 8 2019].

Tkarcher, 2006. *Change of Wavelength by the motion of the source.* [Illustraties] (CC BY-SA 3.0).

Tranquillo, J. V., 2014. *Biomedical Signals and Systems.* University of Cennecticut: Morgan & Claypool publishers.

Translation Bureau, 2013. *Public Works and Government Services Canada,* Canada: Government of Canada.

Wang, R., 2019. *Butterworth filters.* [Online] Available at: http://fourier.eng.hmc.edu/e84/lectures/ActiveFilters/node6.html [Geopend 15 05 2019].

Wiener, N., 1964. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications.* 1 red. Massachussets: M.I.T. Press.

Yang, W. & Zurbenko, I., 2010. Kolmogorov-Zurbenko filters. *Wiley Interdisciplinary Reviews: Computational Statistics,* 2(3), pp. 340-351.

Yilmaz, Ö., 2001. *Seismic data analysis.* Tulsa: Society of Exploration Geophycist.

Zölzer, U., 1998. *Digital Audio Signal Processing.* Chicester: Wiley and Sons.

Zurbenko, I., 1986. *The Spectral Analysis of Time Series.* 1e red. New York: Elsevier Notrth Holland.

Appendices

Appendix 1 (1/31)

Code for AIC picking method in R (R. Kiuru, 2017)

```
AIC <- function(pat, tmin, tmax){
  #
  # List stack files with the desired pattern in file name
  #
  list.files (path = './out/Processed', pattern = pat) -> filelist
  #
  # Create directory paths
  #
  file.path('./out') -> outputdir
  file.path('./out/Picks') -> pickdir
  file.path('./out/Plots') -> plotdir
  #
  # Check for the folders, create if not there
  #
  if (!dir.exists(outputdir)) dir.create(outputdir)
  if (!dir.exists(pickdir)) dir.create(pickdir)
  if (!dir.exists(plotdir)) dir.create(plotdir)
  #


  for (n in 1:length(filelist)){
    #
    # Read the file into R
    #
    file.path('./out/Processed',
              basename(filelist[n])) -> procpath
    read.table(file=procpath, sep="\t",
               stringsAsFactors=FALSE, check.names = FALSE) -> temp
    assign(paste(substr(basename(filelist[n]), 1,
                        nchar(basename(filelist[n])) - 4)), temp)


    #
    # Find the rows for time window extent
    #
    for (i in 1:nrow(temp)){
      if (temp[i,1] >= tmin) temp[i,1] -> tmin
      if (temp[i,1] >= tmin) i -> tminrow
      if (temp[i,1] >= tmin) {break}
    }
    #
    for (i in 1:nrow(temp)){
      if (temp[i,1] >= tmax) temp[i,1] -> tmax
      if (temp[i,1] >= tmax) i -> tmaxrow
      if (temp[i,1] >= tmax) {break}
    }
    #
    # Extract the data from the window
    #
    temp[tminrow:tmaxrow,1:2] -> vardata
    NULL -> rownames(vardata)
    temp[tminrow:tmaxrow,1:2] -> var1
    NULL -> rownames(var1)
    temp[tminrow:tmaxrow,1:2] -> var2
    NULL -> rownames(var2)
    temp[tminrow:tmaxrow,1:2] -> AIC
    NULL -> rownames(AIC)
    #
```

```
    # Calculate first variance
    #
    for (i in 1:nrow(var1)){
      (log(var(vardata[1:i,2]))) -> var1[i,2]
    }
    #
    # Calculate second variance
    #
    for (i in 1:nrow(var2)){
      (log(var(vardata[(i+1):nrow(vardata),2]))) -> var2[i,2]
    }
    #
    # Calculate AIC
    #
    for (i in 1:nrow(AIC)){
      (i*var1[i,2] + (i-nrow(AIC)-1)*var2[i,2]) -> AIC[i,2]
    }
    #
    # Find the minimum of the AIC function
    #
    for (i in 2:nrow(AIC)){
      if (AIC[i,2] <= min(AIC[,2], na.rm=TRUE)) i -> AIC_min_time
      if (AIC[i,2] <= min(AIC[,2], na.rm=TRUE)) {break}
    }
    #
    # Find the corresponding pick time
    #
    AIC[AIC_min_time,1] -> AIC_pick_time
    #


    #
    # Plot the data and the picks
    #
    png(filename=plotpath1, width = 6000, height = 4000, units ="px", res = 300, pointsize = 18)
    plot(vardata, main = paste(substr(basename(filelist[n]), 1, nchar(basename(filelist[n])) - 15),
                               "window", sep = "_"), xlab = "Time (us)",
         ylab = "Standardised voltage", type = "l")
    dev.off()
    #
    png(filename=plotpath2, width = 6000, height = 4000, units ="px", res = 300, pointsize = 18)
    plot(var1, main = paste(substr(basename(filelist[n]), 1, nchar(basename(filelist[n])) - 15),
                            "var1", sep = "_"), xlab = "Time (us)",
         ylab = "First variance", type = "l")
    dev.off()
    #
    png(filename=plotpath3, width = 6000, height = 4000, units ="px", res = 300, pointsize = 18)
    plot(var2, main = paste(substr(basename(filelist[n]), 1, nchar(basename(filelist[n])) - 15),
                            "var2", sep = "_"), xlab = "Time (us)",
         ylab = "Second variance", type = "l")
    dev.off()
    #
    png(filename=plotpath4, width = 6000, height = 4000, units ="px", res = 300, pointsize = 18)
    plot(AIC, main = paste(substr(basename(filelist[n]), 1, nchar(basename(filelist[n])) - 15),
                           "AIC", sep="_"), sub = paste("AIC pick:", AIC_pick_time, sep=" "),
         xlab = "Time (us)", ylab = "AIC", type = "l")
    abline(v=AIC_pick_time, col='blue')
    dev.off()
    #


    png(filename=plotpath5, width = 6000, height = 4000, units ="px", res = 300, pointsize = 18)
    plot(temp, main = substr(basename(filelist[n]), 1, nchar(basename(filelist[n])) - 15),
         sub = paste("AIC pick:", AIC_pick_time, sep=" "), xlab = "Time (us)",
         ylab = "Standardised voltage", type = "l")
    # plot(datasource, main = substr(basename(filelist[n]), 1, nchar(basename(filelist[n])) - 10),
    # sub = paste("AIC pick:", AIC_pick_time, "  ", "Manual pick:", manual_pick, sep=" "),
    # xlab = "Time (us)", ylab = "Standardised voltage", type = "l")
    abline(v=AIC_pick_time, col='blue')
    # abline(v=manual_pick, col='red')
    dev.off()
    #
    # Save the pick values
    #
    write.table(AIC_pick_time, file=pickpath, sep="\t")
    #
  }
  #
}
#
```

Appendix 2 (3/31)

Code for cumulative distribution function in R

```
cfd=function(){
  #
  # Create file path
  #
  file.path('./out/KS') -> ksdir
  if (!dir.exists(ksdir)) dir.create(ksdir)
  #
  # Load data
  #
  list.files(path = './out/Processed', pattern="freq.txt") -> filelist
  noise=grep("Noise_freq.txt", filelist)
  kza=grep("KZA_freq.txt", filelist)
  filelist=filelist[-c(noise, kza)]
  #
  setwd("./out/Processed")
  for (j in filelist){
    dat=read.table(j, header=TRUE, sep="\t")
    #
    # Create data frame in frequency
    #
    freq=dat[,1]
    amp_ft=abs(dat[,2])
    df=data.frame(freq=freq, amp=amp_ft)
    #
    # Subset frequency data frame


    #
    fmin=0
    fmax=1
    for (i in 1:nrow(df)){
      if (df[i,1] >= fmin) df[i,1] -> fmin
      if (df[i,1] >= fmin) i -> fminrow
      if (df[i,1] >= fmin) {break}
    }
    #
    for (i in 1:nrow(df)){
      if (df[i,1] >= fmax) df[i,1] -> fmax
      if (df[i,1] >= fmax) i -> fmaxrow
      if (df[i,1] >= fmax) {break}
    }
    #
    df=df[fminrow:fmaxrow,]
    rownames(df)<- NULL
    #
    # Define bins
    #
    df$bins <- cut(df$freq, breaks=seq(0,max(df[,1]),length.out=100))
    #
    # Define new dataframe based on bins
    #
    df2 <- df %>%
      group_by(bins) %>%
      summarise(avgamp=mean(amp), medfreq=median(freq))
    #
    # Create plot for checking purposes only


    #
    medfreq=df2[["medfreq"]]
    avgamp=df2[["avgamp"]]
    plot(medfreq,avgamp, type="l")
    #
    # Define the actual Cumulative frequency distribution
    #
    diff= as.vector(diff(as.matrix(medfreq)))
    diff= c(diff[1],diff)
    perbox=diff*avgamp
    cumsum=cumsum(perbox)
    tot=sum(perbox)
    cfd=cumsum/tot
    plot(cfd)
    #
    # Save cfd
    #
    file.path('../KS', paste(substr(basename(j), 1, nchar(basename(j)) - 9),
                             "cfd.txt", sep="_")) -> cfdpath
    write.table(cfd, cfdpath, sep="\t")
  }

}
```

Appendix 3 (4/31)

Code for KS-testing in R

```
ks=function(){
  setwd("./out/KS")
  filelist=list.files(pattern="cfd")
  clean_excl=grep("Clean_cfd.txt", filelist)
  synth_excl=grep("Synthetic_cfd.txt", filelist)
  filelist_filt=filelist[-c(clean_excl, synth_excl)]
  #
  clean=read.table("Clean_cfd.txt", sep="\t")
  clean=clean[,1]
  synthetic=read.table("Synthetic_cfd.txt", sep="\t")
  synthetic=synthetic[,1]
  #
  # Create empty matrix for data storage
  #
  column.names=c("Statistic", "P.value", "Method")
  row.names=c(basename(filelist_filt))
  dimensions=c(length(filelist_filt), length(column.names))
  output=array(c(0,0), dim=dimensions, dimnames=list(row.names, column.names))
  #
  # Perform Ks-tests: clean vs filtered


  #
  j=0
  for (i in filelist_filt){
    j=1+j
    filt=read.table(i, sep="\t", header=TRUE)
    filt=filt[,1]
    result=ks.test(clean, filt, alternative="two.sided")
    output[j,1]<-result[["statistic"]]
    output[j,2]<-result[["p.value"]]
    output[j,3]<-result[["method"]]
  }
  write.table(output, file="Clean_vs_filtered.txt", sep="\t", row.names=TRUE)
  #
  # Create empty matrix again
  #
  column.names=c("Statistic", "P.value", "Method")
  row.names=c(basename(filelist_filt))
  dimensions=c(length(filelist_filt), length(column.names))
  output=array(c(0,0), dim=dimensions, dimnames=list(row.names, column.names))
  #
  # Perform Ks-test: Synthetic vs filtered
  #
  for (i in filelist_filt){
    filt=read.table(i, sep="\t", header=TRUE)
    filt=filt[,1]
    result=ks.test(synthetic, filt, alternative="two.sided")
    output[j,1]<-result[["statistic"]]
    output[j,2]<-result[["p.value"]]
    output[j,3]<-result[["method"]]
  }
  write.table(output, file="Synthetic_vs_filtered.txt", sep="\t", row.names=TRUE)


  #
  # Perform ks-test: synthetic vs clean
  #
  result=ks.test(clean, synthetic, alternative="two.sided")
  df=data.frame(Statistic=result[["statistic"]],
                P_value=result[["p.value"]], Method=result[["method"]])
  write.table(df, file="Clean_vs_synth.txt", sep="\t", row.names=TRUE)
  #
  #
}
```

Appendix 4 (5/31)

Code for onset accumulation in R

```
# Load Theoretical shear wave value and shear wave estimations by AIC without processing
#
setwd("C:/Users/Victor Meijs/Desktop/GAGS/Chapters - Thesis")
man_picks=read.table("Manual_picks.txt", sep="\t", header=TRUE, stringsAsFactors=FALSE)
man_picks=man_picks[,-5:-16]
table<-man_picks[,-1]
rownames(table)<-man_picks[,1]
table$Mean <- ave(table$Ludo, table$Risto, table$Jordi)
names=rownames(table)
#
pmat=matrix(0,30,3)
for (i in seq(1,nrow(table),1)){
  id=names[i]
  type=substr(id, 11, nchar(id))
  if (type == "S1"){
    F2F_S1=12.79326279
    diam=substr(id, 4, nchar(id)-7)
    sample_length=2.5*as.numeric(diam)
    pmat[i,2]=round((sample_length/(table[i,4]-F2F_S1)), 2)
  } else{
    F2F_S2=13.670215
    diam=substr(id, 4, nchar(id)-7)
    sample_length=2.5*as.numeric(diam)
    pmat[i,2]=round((sample_length/(table[i,4]-F2F_S2)), 2)
  }
  pmat[i,1]<-id
}

# Add S-values
#
parent.folder="C:/Users/Victor Meijs/Desktop/GAGS/Ultrasonic data - Subset"
sub.folders=list.dirs(parent.folder, recursive=TRUE)
sub.folders=grep("_S", sub.folders, value=TRUE)
sub.folders=grep("Picks", sub.folders, value=TRUE)
#
j=0
for (i in sub.folders){
  setwd(i)
  j=j+1
  filelist=basename(list.files(i))
  filename=grep("_stack_pick", filelist, value=TRUE)
  pick=as.numeric(read.table(filename), stringsAsFactors=FALSE, sep="\t")
  waveform=gsub(".*[_]([^.]+)[_].*", "\\1", i)
  if (waveform=="S1"){
    F2F_S1=12.79326279
    diam=gsub(".*[-]([^_]+)[-].*", "\\1", i)
    sample_length=2.5*as.numeric(diam)
    s1vel=round(sample_length/(pick-F2F_S1), 2)
    pmat[j, 3]<- s1vel
  } else{
    F2F_S2=13.670215
    diam=gsub(".*[-]([^_]+)[-].*", "\\1", i)
    sample_length=2.5*as.numeric(diam)
    s2vel=round(sample_length/(pick-F2F_S2), 2)
    pmat[j, 3]<- s2vel
  }
}


#
new1=as.numeric(pmat[,2])
new2=as.numeric(pmat[,3])
new=cbind(new1,new2)
colnames(new)<-c("Manual", "no filter")

#
# Load all the onset estimations after filtering
#
parent.folder = "C:/Users/Victor Meijs/Desktop/GAGS/Ultrasonic data - Subset"
sub.folders = list.dirs(parent.folder, recursive=FALSE)
sub.folders = grep("_S", sub.folders, value=TRUE)
column.names=c("Butter LP", "Butter HP", "Butter SB", "Butter PB", "Cheby1 LP",
               "Cheby1 HP", "Cheby1 PB", "Spectral sub.", "Amplitude", "Comb", "KZ", "KZA")
row.names=c(basename(sub.folders))
dimensions=c(length(sub.folders), length(column.names))
output_ex=array(c(0,0), dim=dimensions, dimnames=list(row.names, column.names))
#
```

```
# Create absolute time table
#
j=0
for (i in sub.folders){
  setwd(i)
  j=1+j
  butterlow=round(as.numeric(read.table("out/Picks/Butter_pick.txt",
                                        stringsAsFactors=FALSE, sep="\t")), 2)
  butterhigh=round(as.numeric(read.table("out/Picks/Butterfrom2_pick.txt",
                                         stringsAsFactors=FALSE, sep="\t")), 2)
  butterstop=round(as.numeric(read.table("out/Picks/Butterstop2_pick.txt",
                                         stringsAsFactors=FALSE,sep="\t")), 2)
  butterpass=round(as.numeric(read.table("out/Picks/Butterpass2_pick.txt",
                                         stringsAsFactors=FALSE,sep="\t")), 2)
  cheby1low=round(as.numeric(read.table("out/Picks/Cheby1low_pick.txt",
                                        stringsAsFactors=FALSE,sep="\t")), 2)
  cheby1high=round(as.numeric(read.table("out/Picks/Cheby1high_pick.txt",
                                         stringsAsFactors=FALSE,sep="\t")), 2)
  cheby1pass=round(as.numeric(read.table("out/Picks/Cheby1pass_pick.txt",
                                         stringsAsFactors=FALSE,sep="\t")), 2)
  spectral=round(as.numeric(read.table("out/Picks/Spectral_pick.txt",
                                       stringsAsFactors=FALSE,sep="\t")), 2)
  amplitude=round(as.numeric(read.table("out/Picks/Amplitude_pick.txt",
                                        stringsAsFactors=FALSE, sep="\t")), 2)
  comb=round(as.numeric(read.table("out/Picks/Comb2_pick.txt",
                                   stringsAsFactors=FALSE,sep="\t")), 2)
  kz=round(as.numeric(read.table("out/Picks/KZ_pick.txt",
                                 stringsAsFactors=FALSE, sep="\t")), 2)
  kza=round(as.numeric(read.table("out/Picks/KZA_pick.txt",
                                  stringsAsFactors=FALSE, sep="\t")), 2)
  #

  #
  output_ex[j,1] <- butterlow
  output_ex[j,2] <- butterhigh
  output_ex[j,3] <- butterstop
  output_ex[j,4] <- butterpass
  output_ex[j,5] <- cheby1low
  output_ex[j,6] <- cheby1high
  output_ex[j,7] <- cheby1pass
  output_ex[j,8] <- spectral
  output_ex[j,9] <- amplitude
  output_ex[j,10] <- comb
  output_ex[j,11] <- kz
  output_ex[j,12] <- kza
  #
  waveform= waveform=gsub(".*[_]([^.]+)[_].*", "\\1", i)
  if (waveform == "S1"){
    F2F_S1=12.79326279
    diam=gsub(".*[-]([^_]+)[-].*", "\\1", i)
    sample_length=2.5*as.numeric(diam)
    output_ex[j,]=round(sample_length/(output_ex[j,]-F2F_S1), 2)
  } else{
    F2F_S2=13.670215
    diam=gsub(".*[-]([^_]+)[-].*", "\\1", i)
    sample_length=2.5*as.numeric(diam)
    output_ex[j,]=round(sample_length/(output_ex[j,]-F2F_S2), 2)
  }
}
output_ex=data.frame(output_ex)
#
# Combine the theoretical value and the processed values
#
output_ex=cbind(new, output_ex)
# output_try<-as.numeric(as.character(output_try))
```

```
# Create relative time difference table
#
row.names=rownames(output_ex)
col.names=colnames(output_ex)
dimensions=c(length(row.names), length(col.names))
output_rel=array(c(0,0), dim=dimensions, dimnames=list(row.names, col.names))
output_rel[,1] <- output_ex[,1]
output_rel[,2] <- round(abs((output_ex[,1] - output_ex[,2])) / output_ex[,1]*100, 2)
output_rel[,3] <- round(abs((output_ex[,1]-output_ex[,3]))/output_ex[,1]*100, 2)
output_rel[,4] <- round(abs((output_ex[,1]-output_ex[,4]))/output_ex[,1]*100, 2)
output_rel[,5] <- round(abs((output_ex[,1]-output_ex[,5]))/output_ex[,1]*100, 2)
output_rel[,6] <- round(abs((output_ex[,1]-output_ex[,6]))/output_ex[,1]*100, 2)
output_rel[,7] <- round(abs((output_ex[,1]-output_ex[,7]))/output_ex[,1]*100, 2)
output_rel[,8] <- round(abs((output_ex[,1]-output_ex[,8]))/output_ex[,1]*100, 2)
output_rel[,9] <- round(abs((output_ex[,1]-output_ex[,9]))/output_ex[,1]*100, 2)
output_rel[,10] <- round(abs((output_ex[,1]-output_ex[,10]))/output_ex[,1]*100, 2)
output_rel[,11] <- round(abs((output_ex[,1]-output_ex[,11]))/output_ex[,1]*100, 2)
output_rel[,12] <- round(abs((output_ex[,1]-output_ex[,12]))/output_ex[,1]*100, 2)
output_rel[,13] <- round(abs((output_ex[,1]-output_ex[,13]))/output_ex[,1]*100, 2)
output_rel[,14] <- round(abs((output_ex[,1]-output_ex[,14]))/output_ex[,1]*100, 2)
output_rel=data.frame(output_rel)
#
#
stats_rel=t(output_rel)
stats_rel=stats_rel[-1,]
st_dev=round(apply(stats_rel,1, sd, na.rm = TRUE),2)
means=round(rowMeans(stats_rel, na.rm = TRUE, dims = 1),2)
stats_rel=data.frame(filter=rownames(stats_rel), mean=means, st_dev=st_dev)
nofilt_mean=stats_rel[1,2]
nofilt_dev=stats_rel[1,3]
stats_rel=stats_rel[-1,]
```

Appendix 5 (8/31)

Code for synthetic signal production in R

```
# Generator creates a specified amount of synthetic signals
#
generator=function(nr_of_samples){
  #
  time=seq.int(0, 200, 0.008) # Define tiem series
  sample_dm=c(30,42,50) # Define sample diamater
  #
  # Start iteration for number of samples per sample diameter
  #
  for (p in 1:length(nr_of_samples)){
    for (j in 1:length(sample_dm)){
      #
      # Create directory paths
      #
      file.path(paste0("./SD_", sample_dm[j], "_9910"), sep="") -> outputid
      file.path(paste0("./SD_", sample_dm[j], "_9910"), "out") -> outputfolder
      file.path(paste0("./SD_", sample_dm[j], "_9910"), "out/Parameters") -> outputsignalpar
      file.path(paste0("./SD_", sample_dm[j], "_9910"), "out/Plots") -> outputsignalplots
      file.path(paste0("./SD_", sample_dm[j], "_9910"), "out/Processed") -> outputsignaldata
      #
      # Check for the folders, create if not there
      #
      if (!dir.exists(outputid)) dir.create(outputid)
      if (!dir.exists(outputfolder)) dir.create(outputfolder)
      if (!dir.exists(outputsignalpar)) dir.create(outputsignalpar)
      if (!dir.exists(outputsignalplots)) dir.create(outputsignalplots)
      if (!dir.exists(outputsignaldata)) dir.create(outputsignaldata)
      #
      # Set signal parameters

      #
      L=sample_dm[j]
      F2F= 13.3
      v_av= 3.2
      LD= 2.5
      onset=runif(1, 0.9*((L*LD/v_av)+F2F), 1.1*((L*LD/v_av)+F2F))
      df=runif(1,0.09,0.11)
      SW=function(t){
        ifelse(t>=onset, -0.33*exp(-1/12*(t-onset))*sin(2*pi*(t-onset)*df), 0)}
      #
      # Select random noise components
      #
      nr_of_ncs=sample(6:8, size=1, replace=F)
      noise_picks=noise_comps[sample(nrow(noise_comps), nr_of_ncs), ]
      #
      # Save the noise and clean signal parameters in text files
      #
      file.path(outputsignalpar, "Noise parameters.txt") -> filepath
      write.table(noise_picks, file=filepath, sep="\t")
      file.path(outputsignalpar, "Number of noise components.txt") -> filepath
      write.table(nr_of_ncs, file=filepath, sep="\t")
      #
      # Define piecewise function for the noise components including the delay
      #
      noise_a=function(t, noise_onset){
        ifelse(t>=(onset-(noise_onset)),
               (noise_picks[i,2]/(nr_of_ncs))*
                 exp((-1/100)*(time-(onset-(noise_onset))))*
                 sin(2*pi*noise_picks[i,1]
                     *(time-(onset-(noise_onset)))+noise_picks[i,3]), 0)}
      #
```

```
      # Define base for noise and create empty plot
      #
      total_noise=0
      plot(1, type="l", xlab="Time in us", ylab="Amplitude",
           xaxs="i", main="Seperate noise components")
      #
      # Start iteration of random noise components
      #
      for (i in 1:nr_of_ncs){
        noise_onset=runif(1, onset*0.98, onset*0.99)
        noise=noise_a(time, noise_onset)
        assign(paste("noise", i, sep="_"), noise)
        total_noise=total_noise+noise
        lines(time, noise, col=i)
      }
      #
      # Create data series
      #
      S_wave=SW(time)
      S_wave=smth(S_wave, method="gaussian", window=0.01)
      S_wave[is.na(S_wave)] <- 0
      S_wave=S_wave+rnorm(length(time), mean = 0, sd=0.0001)
      #
      # Save the compensated onset
      #
      file.path(outputsignalpar, "Onset.txt") -> filepath
      write.table(onset, file=filepath, sep="\t")
      #
      synth_sig=S_wave + total_noise
      #


      # Normalise the synthetic signal
      #
      synthesizer=synth_sig
      #
      for (i in 1:length(synth_sig)){
        ((synth_sig[i]-mean(synth_sig))/(sd(synth_sig))) -> synthesizer[i]
      }
      #
      clean_ft=spec.fft(S_wave, time)
      synth_ft=spec.fft(synthesizer, time)
      noise_ft=spec.fft(total_noise, time)
      #
      # Create dataframes for time series
      #
      dft_SW=data.frame(Time_in_us=time, A=S_wave) # Clean shear wave data frame
      dft_synthsig=data.frame(Time_in_us=time, A=synth_sig) # Synthetic signal data frame
      dft_synth=data.frame(Time_in_us=time, A=synthesizer) # Synthetic signal data frame
      dft_noise=data.frame(Time_in_us=time, A=total_noise)
      #
      # Create data frame for frequency series
      dff_SW=data.frame(f=clean_ft[["fx"]], A=clean_ft[["A"]])
      dff_synth=data.frame(f=synth_ft[["fx"]], A=synth_ft[["A"]])
      dff_noise=data.frame(f=noise_ft[["fx"]], A=noise_ft[["A"]])
      #

              ..
      #
    }
  }
}
#
```

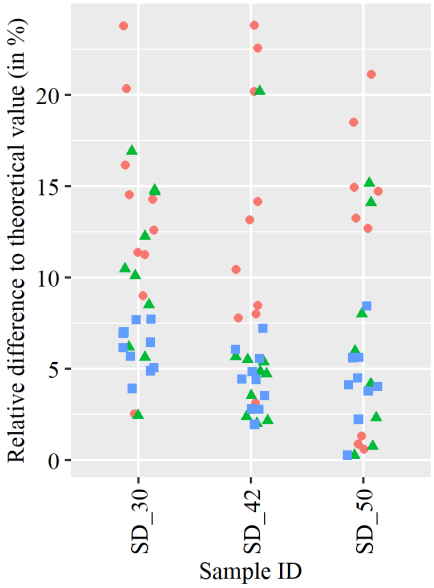
Appendix 6 (10/31)

Code for KZ and KZA filtering in R

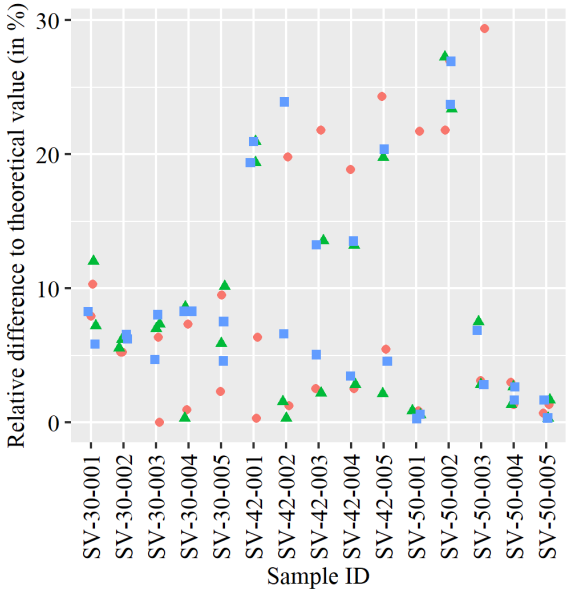
```
kza_mod=function(time, amp){
  #
  # Apply KZ-filter
  #
  amp_kz=kz(amp, m=125, k=3)
  #plot(amp, type="l")
  #lines(amp_kz, type="l", col="red")
  #
  # Apply KZA-filter
  #
  amp_kza=kza(amp, m=125, k=2, y=amp_kz)

  #
}
#
```

Appendix 7 (11/31)

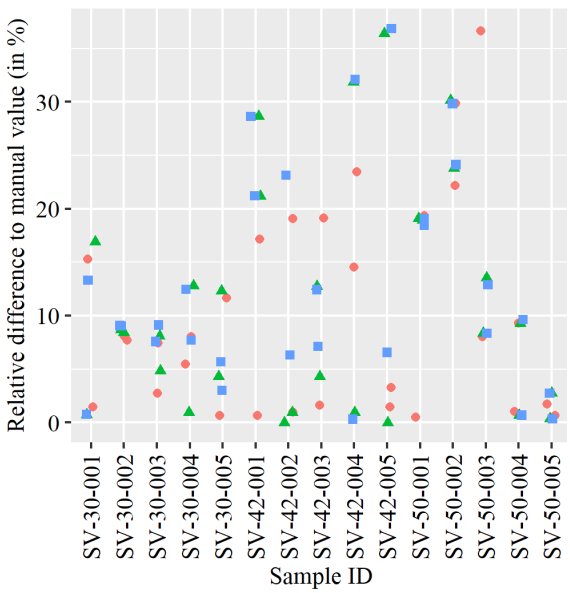
Overview of the onset per signal before and after KZ and KZA filtering



Red: No filter, Blue: KZ, Green: KZA



Red: No filter, Blue: KZ, Green: KZA



Red: No filter, Blue: KZ, Green: KZA

Appendix 8 (12/31)

Visual signal classification before and after KZ and KZA filtering

Diameter (in mm)	30			42			50		
Sample nr.	No filter	KZ	KZA	No filter	KZ	KZA	No filter	KZ	KZA
1	2	2	3	2	2	3	1	3	3
2	2	3	3	1	3	3	1	3	3
3	1	2	3	1	3	3	1	2	3
4	1	3	3	1	2	3	1	3	3
5	2	2	3	1	3	3	1	3	3
6	1	2	3	1	3	3	1	2	3
7	1	3	3	1	3	3	1	3	3
8	2	3	3	1	3	3	2	2	3
9	1	2	3	1	3	3	1	3	3
10	1	3	3	1	3	3	1	2	3

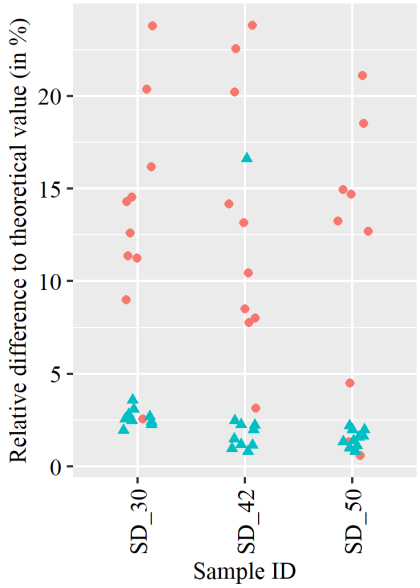
Appendix 9 (13/31)

Code for amplitude filtering in R

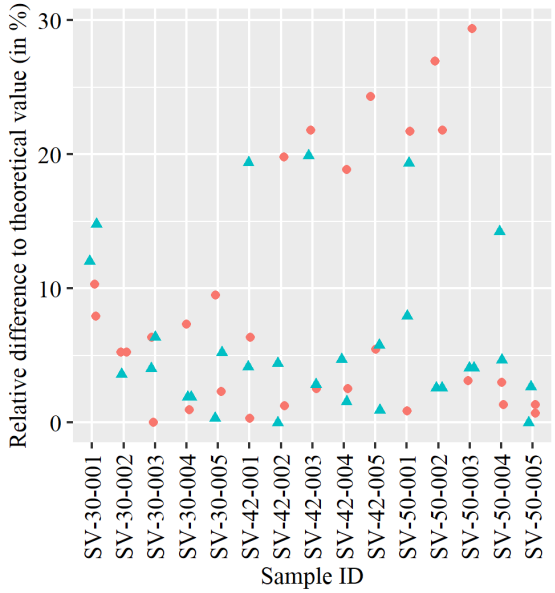
```
ampli_filter=function(time, amp){
  #
  # Create filtered data
  #
  amp_ft = spec.fft(amp,time)
  amp_af = as.vector(afilter(amp, f=1, threshold=33, plot= FALSE))
  t_thresh=min(which(abs(amp_af[1250:length(time)])>0.1))*0.008
  t_thresh=t_thresh+10
  amp_afft = spec.fft(amp_af, time)
  amp_af=amp_af+rnorm(length(amp_af), mean = 0, sd=0.0001)

  #
}
```

Appendix 10 (14/31)

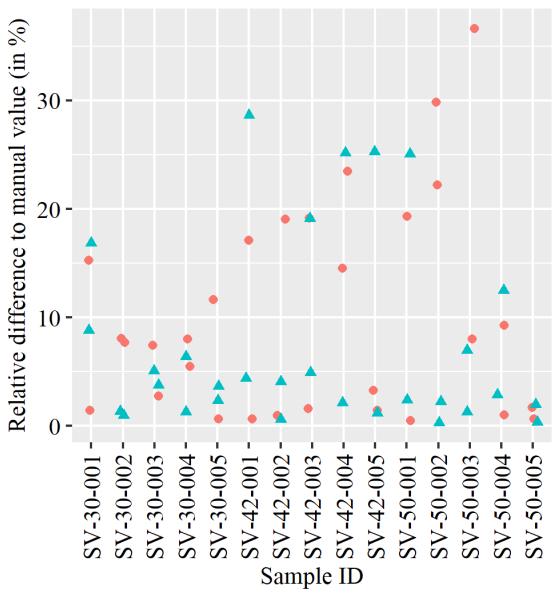
Overview of the onset per signal before and after amplitude filtering



Red: No filter, Blue Amplitude filter



Red: No filter, Blue Amplitude filter



Red: No filter, Blue Amplitude filter

Appendix 11 (15/31)

Visual signal classification before and after amplitude filtering

Diameter (in mm)	30		42		50	
Sample nr.	No filter	Amplitude	No filter	Amplitude	No filter	Amplitude
1	2	2	2	2	1	2
2	2	2	1	2	1	2
3	1	2	1	2	1	2
4	1	2	1	2	1	2
5	2	2	1	2	1	2
6	1	2	1	2	1	2
7	1	2	1	2	1	2
8	2	2	1	2	2	2
9	1	2	1	2	1	2
10	1	2	1	2	1	2

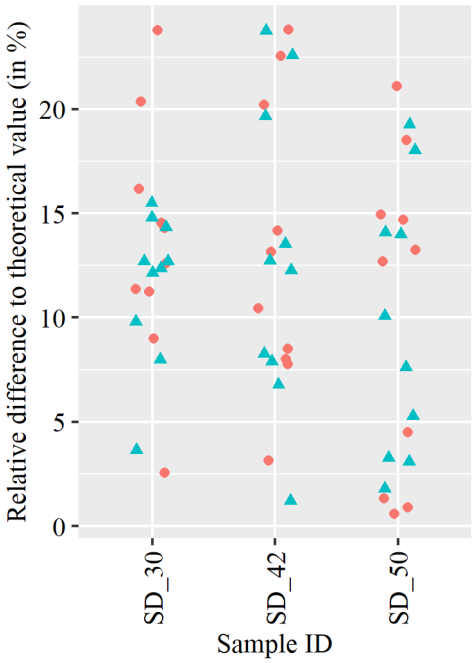
Appendix 12 (16/31)

Code for Butterworth filtering in R

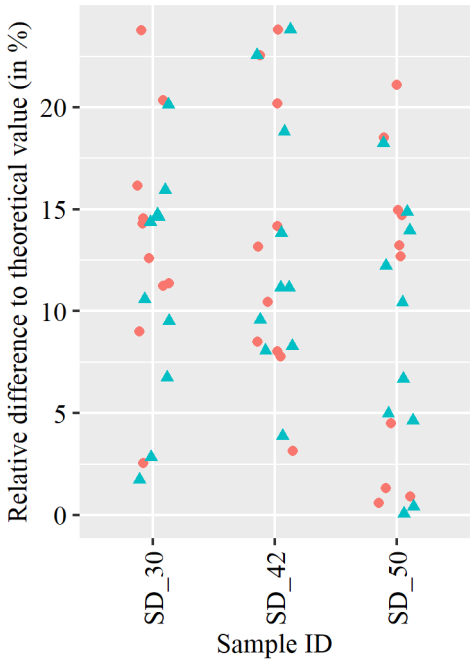
```
bw_filterlow=function(amp, time){
  #
  # Find dominant frequency
  #
  fft_or=spec.fft(amp, time)
  max(abs(fft_or[["A"]]))
  absA=abs(fft_or[["A"]])
  f=fft_or[["fx"]]
  df=data.frame(absA=absA[(length(absA)/2):length(absA)],
                f=f[(length(absA)/2):length(absA)])
  peaks=df[which.max(df$absA),]
  dom_freq=peaks[1,2]
  #
  # Filter the data
  #
  amp_bw=bwfilter(amp, f=100, n = 6, to=(dom_freq*1))
  layout(matrix(seq(2)))
  plot(amp_bw, type="l")
  plot(amp, type="l")
  fft_bw=spec.fft(amp_bw[,1], time)
  plot(fft_bw, type="l", xlim=c(0,1))
  plot(fft_or, type="l", xlim=c(0,1))
  freq=fft_bw[["fx"]]
  A=fft_bw[["A"]]
  dff_bw=data.frame(f=freq, A=A)
  dft_bw=data.frame(Time=time, A=amp_bw)


  #
}
```

Appendix 13 (17/31)

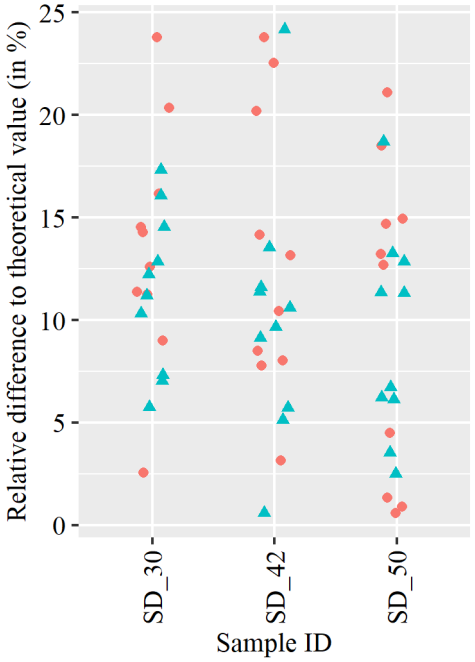
Overview of the onset per signal before and after Butterworth filtering



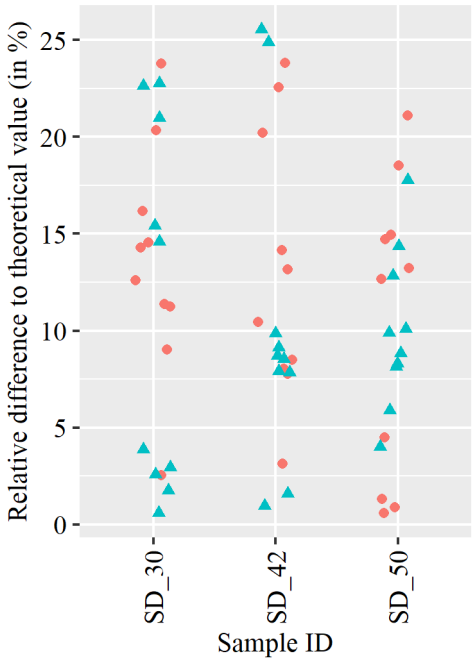
Red: No filter, Blue: Butterworth low-pass



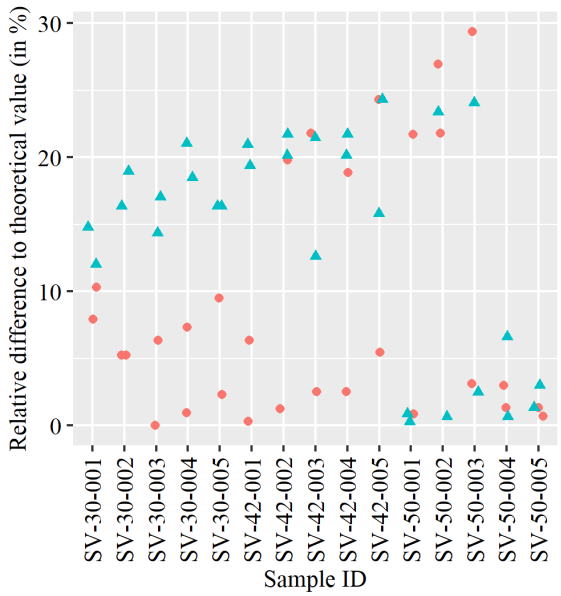
Red: No filter, Blue: Butterworth high-pass



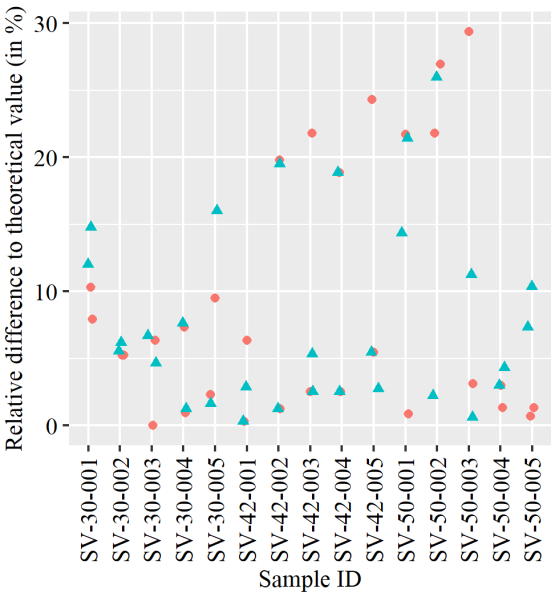
Red: No filter, Blue: Butterworth band-pass



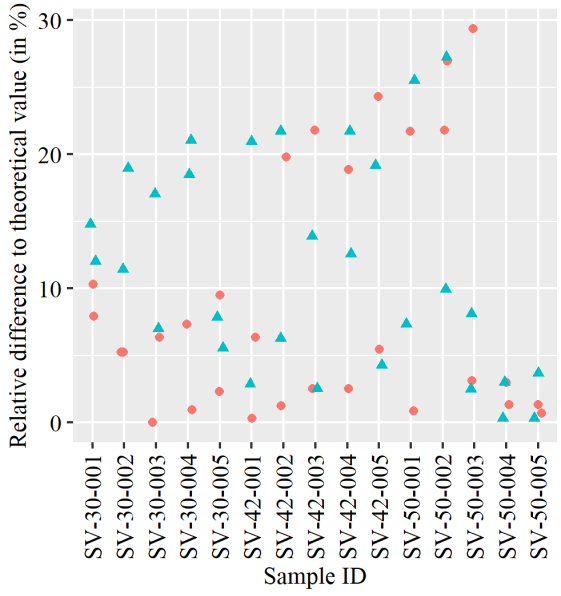
Red: No filter, Blue: Butterworth band-stop



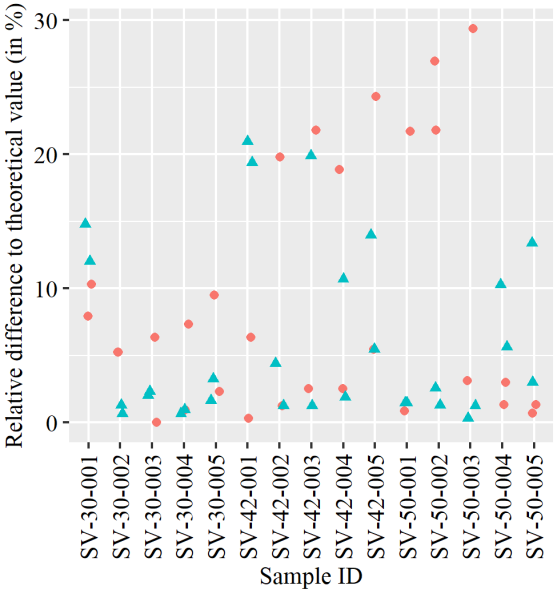
Red: No filter, Blue: Butterworth low-pass



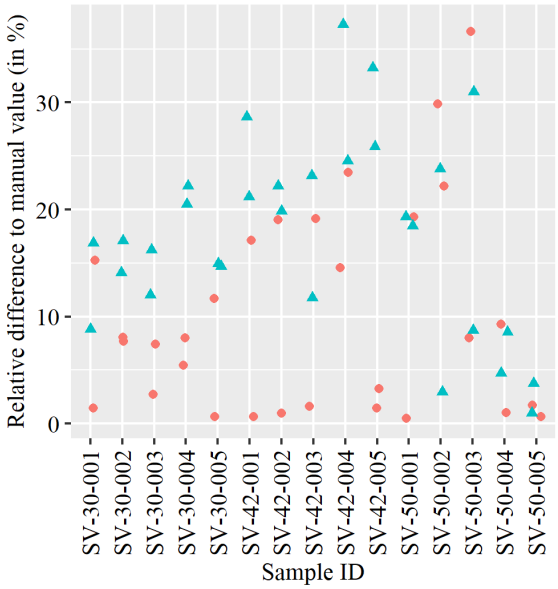
Red: No filter, Blue: Butterworth high-pass



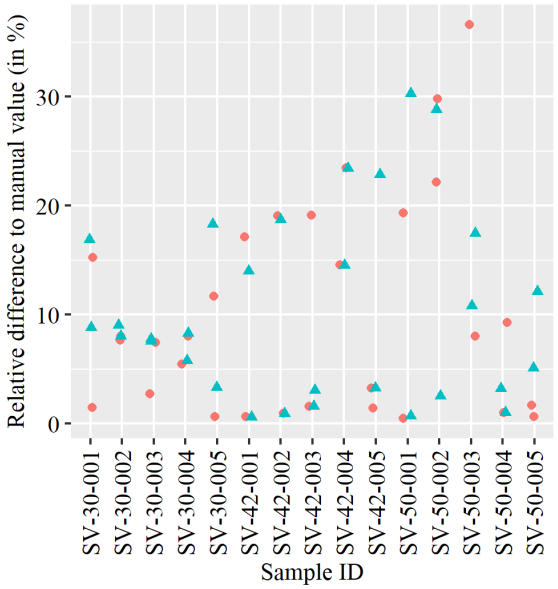
Red: No filter, Blue: Butterworth band-pass



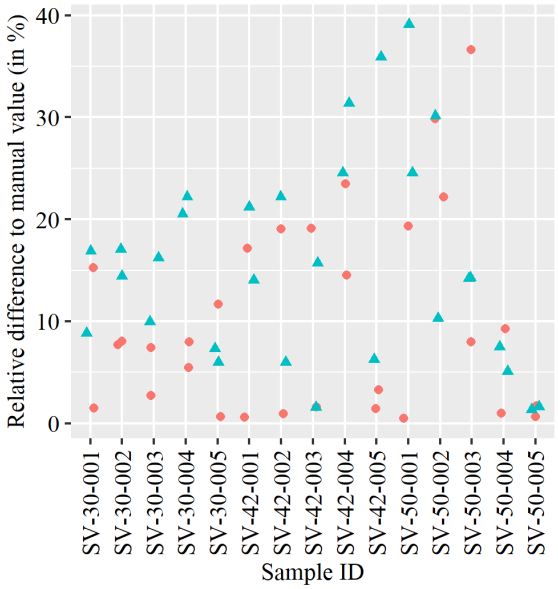
Red: No filter, Blue: Butterworth band-stop



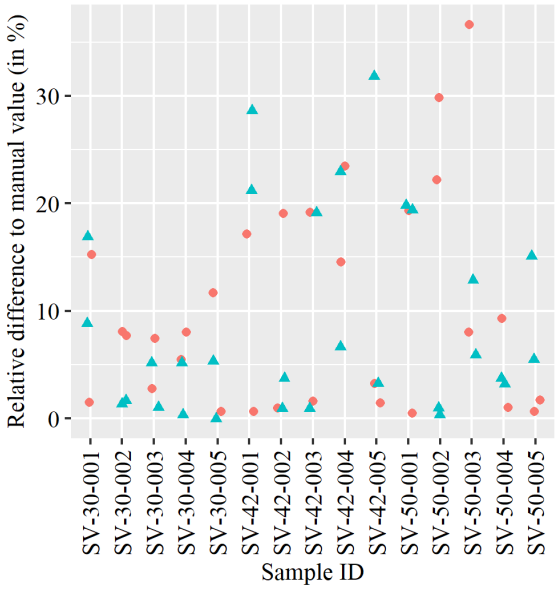
Red: No filter, Blue: Butterworth low-pass



Red: No filter, Blue: Butterworth high-pass



Red: No filter, Blue: Butterworth band-pass



Red: No filter, Blue: Butterworth band-stop

Appendix 14 (20/31)

Visual signal classification before and after Butterworth filtering

Diameter (in mm)	30				
Sample nr.	No filter	Low-pass	High-pass	Band-pass	Band-stop
1	2	2	2	1	3
2	2	1	1	3	3
3	1	1	1	1	1
4	1	2	1	2	2
5	2	2	2	2	3
6	1	3	1	1	3
7	1	2	1	2	1
8	2	3	2	2	2
9	1	2	1	2	3
10	1	3	1	2	3

Diameter (in mm)	42				
Sample nr.	No filter	Low-pass	High-pass	Band-pass	Band-stop
1	2	2	2	2	3
2	1	3	1	2	3
3	1	2	1	2	3
4	1	2	1	1	3
5	1	3	1	3	1
6	1	2	1	2	3
7	1	2	1	1	2
8	1	3	1	2	1
9	1	3	1	3	3
10	1	2	1	2	1

Diameter (in mm)	50				
Sample nr.	No filter	Low-pass	High-pass	Band-pass	Band-stop
1	1	3	1	1	3
2	1	3	1	1	3
3	1	2	2	1	3
4	1	3	1	2	2
5	1	3	1	2	2
6	1	2	1	1	3
7	1	3	1	2	3
8	2	2	1	2	3
9	1	3	1	1	3
10	1	2	1	1	2

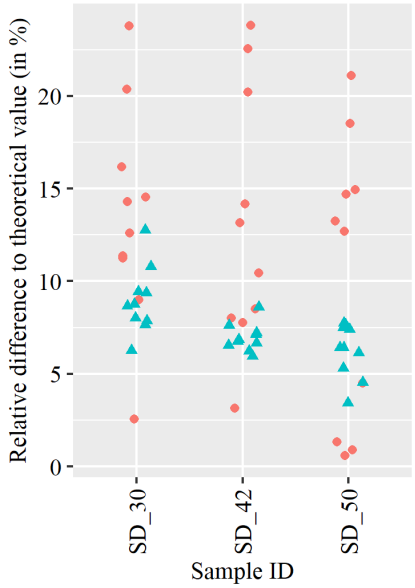
Appendix 15 (21/31)

Code for Chebyshev1 filtering in R

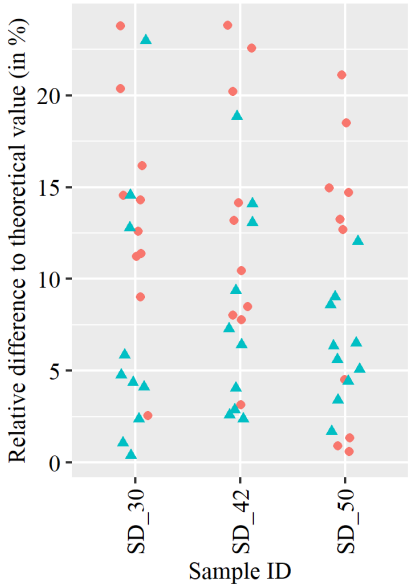
```
cheby_filterlow=function(amp, time){
  #
  # Find dominant frequency
  #
  fft_or=spec.fft(amp, time)
  max(abs(fft_or[["A"]]))
  absA=abs(fft_or[["A"]])
  f=fft_or[["fx"]]
  df=data.frame(absA=absA[(length(absA)/2):length(absA)],
                f=f[(length(absA)/2):length(absA)])
  peaks=df[which.max(df$absA),]
  dom_freq=peaks[1,2]
  #
  # Determine chebyshev parameters
  #
  Fs=125000000
  Fpc= dom_freq*1000000*0.3
  Fsc= dom_freq*1000000*0.7
  chord=cheb1ord(Wp=(Fpc/(Fs/2)),
                 Ws=(Fsc/(Fs/2)), 0.5, 29)
  df_chord=data.frame(Order=chord[["n"]], Cut_off=chord[["Wc"]],
                      Type=chord[["type"]], Ripple=chord[["Rp"]])
  #
  # Construct and apply Chebyshev filter
  #
  cf=cheby1(chord)
  amp_cf=as.numeric(filter(cf, amp))
  #
}
```

Appendix 16 (22/31)

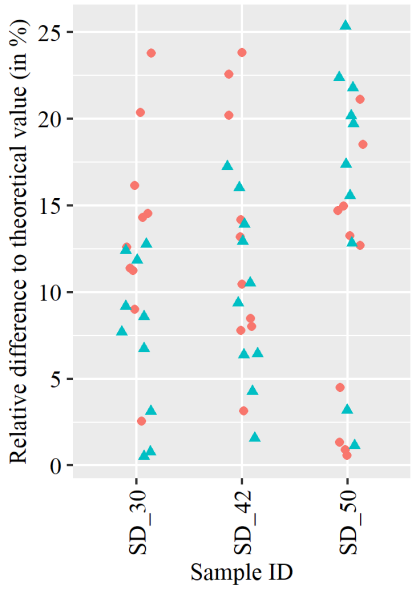
Overview of the onset per signal before and after Chebyshev filtering



Red: No filter, Blue: Chebyshev1 low-pass

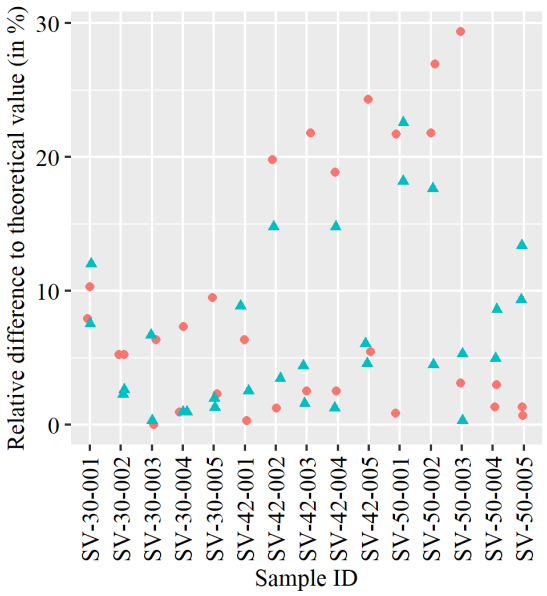


Red: No filter, Blue: Chebyshev1 high-pass

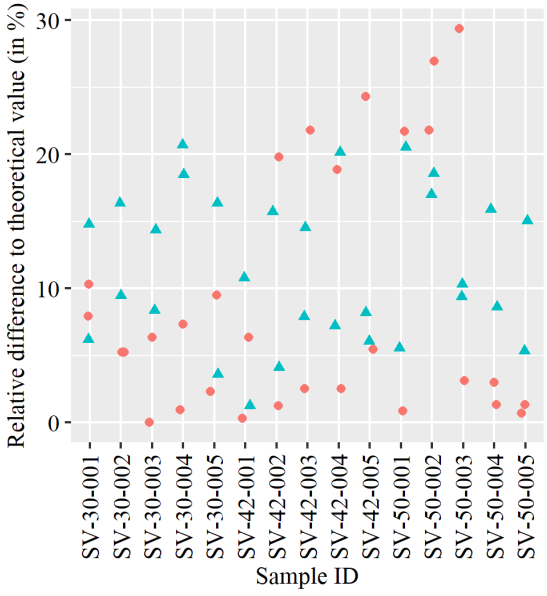


Red: No filter, Blue: Chebyshev1 band-pass

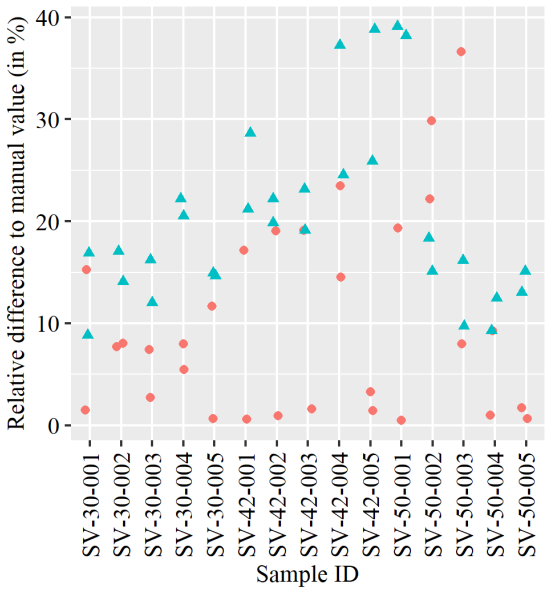
Red: No filter, Blue: Chebyshev1 low-pass



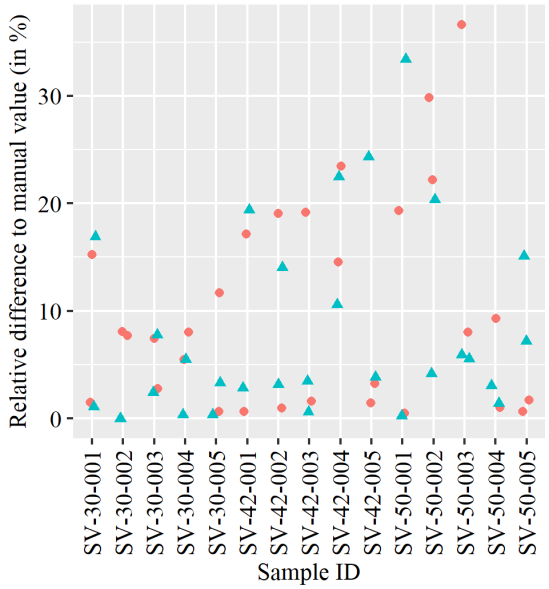
Red: No filter, Blue: Chebyshev1 high-pass



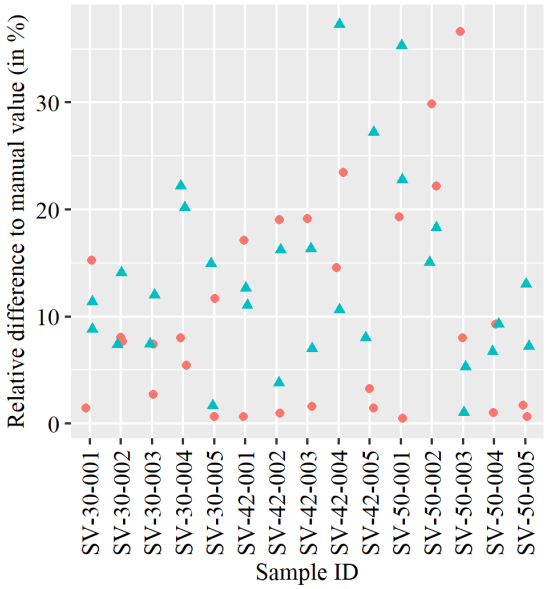
Red: No filter, Blue: Chebyshev1 band-pass



Red: No filter, Blue: Chebyshev1 low-pass



Red: No filter, Blue: Chebyshev1 high-pass



Red: No filter, Blue: Chebyshev1 band-pass

Appendix 17 (25/31)

Visual signal classification before and after Chebyshev1 filtering

Diameter (in mm)	30			
Sample nr.	No filter	Cheby1 HP	Cheby1 LP	Cheby1 BP
1	2	3	2	3
2	2	3	2	3
3	1	3	3	2
4	1	3	3	3
5	2	3	3	3
6	1	3	3	3
7	1	3	2	3
8	2	3	3	3
9	1	3	3	3
10	1	3	3	3

Diameter (in mm)	42			
Sample nr.	No filter	Cheby1 HP	Cheby1 LP	Cheby1 BP
1	2	3	3	3
2	1	3	2	3
3	1	3	3	3
4	1	1	2	3
5	1	3	3	3
6	1	3	2	3
7	1	1	3	3
8	1	3	3	3
9	1	3	3	3
10	1	3	3	3

Diameter (in mm)	50			
Sample nr.	No filter	Cheby1 HP	Cheby1 LP	Cheby1 BP
1	1	2	3	3
2	1	3	3	3
3	1	3	3	3
4	1	3	2	3
5	1	3	3	3
6	1	2	3	3
7	1	3	3	3
8	2	3	3	3
9	1	3	1	3
10	1	3	2	3

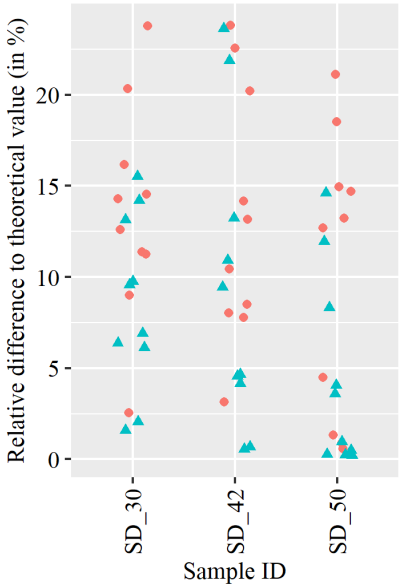
Appendix 18 (26/31)

Code for spectral subtraction in R

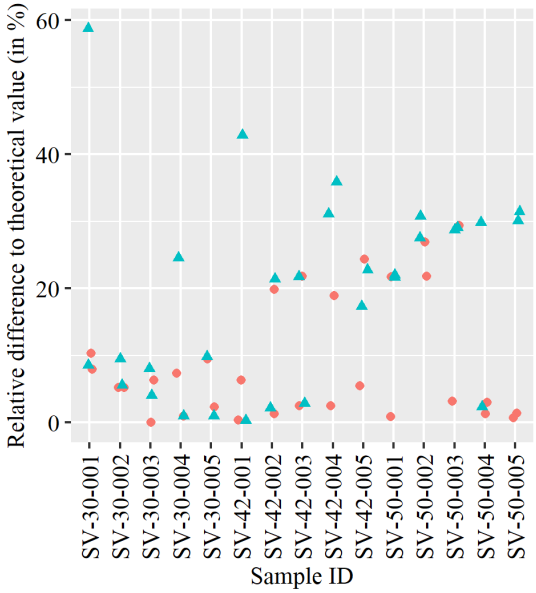
```
spec_sub=function(A, t, frames, int){
  #
  # Define int size
  #
  win=round(int/0.008)
  #
  # Calculate the fourier transform of every int
  #
  fft_frames=specgram(A, n=win, Fs=125,
                      window=rectangle.window(win), overlap=0)
  #
  # Calculate the magnitude and phase spectra
  #
  mag=abs(fft_frames[["S"]])
  phase=atan2(Im(fft_frames[["S"]]),Re(fft_frames[["S"]]))
  phase[is.nan(phase)] <- 0
  #
  # Find the first noise arrival
  #
  B=A[625:length(A)]
  noise_arr_frame=min(which(abs(B)>0.1))+(length(A)-length(B))
  noise_arr_time=noise_arr_frame*0.008
  start=round_any((noise_arr_time/int), 1, floor)
  #
  # Estimate the average magnitude of the noise
  noise_spec=0
  for (i in 1:length(frames)){
    noise_spec=noise_spec+mag[,start+i]
  }
  av_noise=noise_spec/length(frames)
  #


  # Subtract the noise magnitude from every frame
  #
  mag_est=mag-av_noise
  mag_est[mag_est < 0] = 0
  #
  # Combine new magnitude estimation with the original phase spectrum
  #
  j=complex(real=0, imaginary=1)
  enh_spec = mag_est*exp(j*phase)
  #
  # Inverse Fourier transform every time frame and store output
  # in vector to reconstruct the time series
  x=enh_spec
  vec=vector()
  for (i in 1:ncol(x)){
    inv=fft(x[,i], inverse=TRUE)/length(x[,i])
    inv=Re(inv)
    vec = c(vec,inv)
  }


  #
}
```

Appendix 19 (27/31)

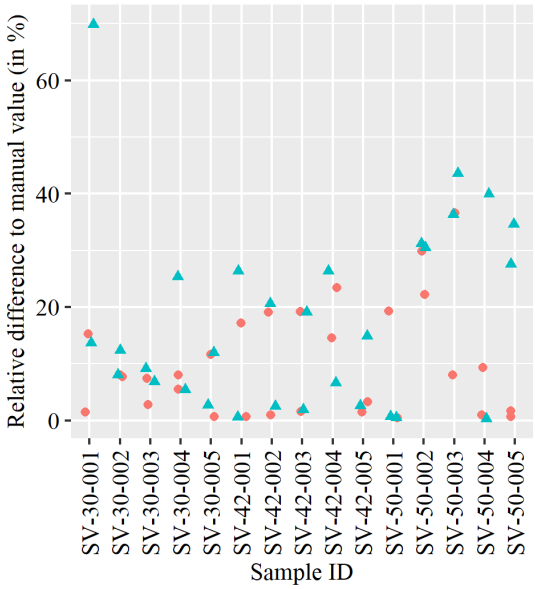
Overview of the onset per signal before and after spectral subtraction



Red: No filter, Blue: spectral subtraction



Red: No filter, Blue: spectral subtraction



Red: No filter, Blue: spectral subtraction

Appendix 20 (28/31)

Visual signal classification before and after spectral subtraction

Diameter (in mm)	30		42	
Sample nr.	No filter	Spectral sub.	No filter	Spectral sub.
1	2	3	2	1
2	2	2	1	3
3	1	2	1	3
4	1	1	1	3
5	2	3	1	3
6	1	2	1	2
7	1	1	1	1
8	2	2	1	2
9	1	3	1	1
10	1	3	1	1

Diameter (in mm)	50	
Sample nr.	No filter	Spectral sub.
1	1	1
2	1	2
3	1	2
4	1	3
5	1	1
6	1	2
7	1	1
8	2	3
9	1	2
10	1	2

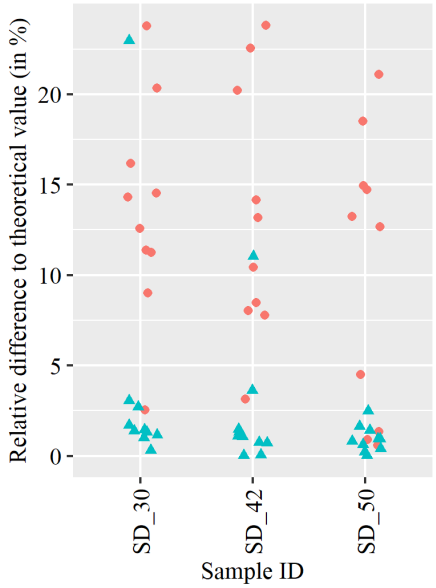
Appendix 21 (29/31)

Code for comb filtering in R

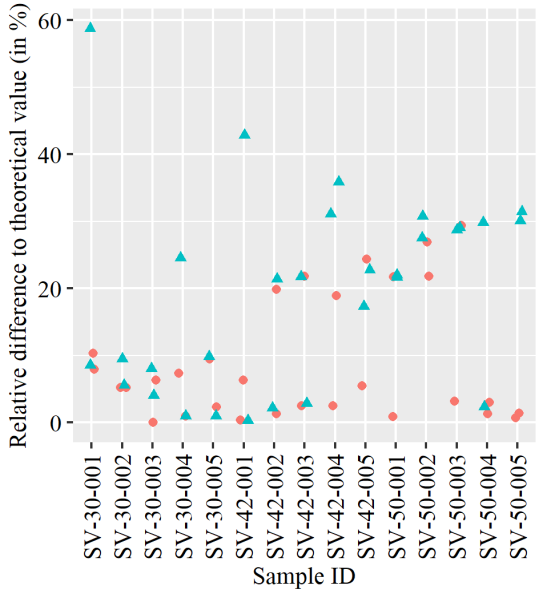
```
comb_filter=function(time, amp){
  #
  # Apply combfilter
  #
  combwave=as.numeric(combfilter(amp, f=125, alpha=1, K=90, units="samples"))
  amp_ft = spec.fft(amp,time)
  comb_ft = spec.fft(combwave, time)

  #
}
```

Appendix 22 (30/31)

Overview of the onset per signal before and after comb filtering



Red: No filter, Blue: comb filter



Red: No filter, Blue: comb filter

Red: No filter, Blue: comb filter

Appendix 23 (31/31)

Visual signal classification before and after comb filter

Diameter (in mm)	30		42		50	
Sample nr.	No filter	Comb	No filter	Comb	No filter	Comb
1	2	3	2	1	1	1
2	2	2	1	2	1	2
3	1	1	1	2	1	1
4	1	1	1	2	1	1
5	2	2	1	1	1	1
6	1	2	1	1	1	2
7	1	1	1	1	1	1
8	2	1	1	2	2	3
9	1	1	1	1	1	1
10	1	1	1	2	1	2