Damage Mechanisms in Composite Materials

Real-time Clustering and Classification of Acoustic Emission Signals

M. Taams

Technische Universiteit Delft

TU Delft
Delft University of Technology

Challenge the future

Damage Mechanisms in Composite Materials

Real-time Clustering and Classification of Acoustic Emission Signals

by

Maurice Taams

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
to be defended publicly on Wednesday July 15, 2020 at 9:30 AM.

Supervisor:	Dr. D. Zarouchas,	TU Delft
Thesis committee:	Dr. R.M. Groves ,	TU Delft
	Dr. M. Saeedifar ,	TU Delft
	Dr. J.J.E. Teuwen ,	TU Delft

This thesis is confidential and cannot be made public until July 15, 2020.

Preface

This thesis is original, unpublished, independent work by the author, M. Taams. It has been written to fulfill the graduation requirements of the Aerospace Structures & Materials Master program at the Delft University of Technology. I was engaged in researching and writing this thesis from October 2019 to June 2020.

I would like to thank my supervisor D. Zarouchas for introducing me to the fascinating field of Acoustic Emission and guiding me through the process of conducting research in this field. I would also like to thank A. Broer-Reinoso Rondinel for helping me with conducting the experiments and giving advice in the early stages of the project. Finally, a particular note of thanks to my fiancée Saskia, who has been a mental support throughout my educational career.

I hope that this research will be of help to others and open doors to new research.

M. Taams
Delft, June 2020

Contents

Executive Summary

A lot of studies have been done on Acoustic Emission (AE) covering a wide range of materials and applications. Within a structure that is under loading, AE has proven to be useful in determining the type, location and accumulation of damage within a material. The general goal is to use AE for in-service monitoring of structurally loaded parts. However, more research is necessary before the method can be employed in real-time applications.

The objective of this project is to contribute to the development of algorithms that classify failure modes real time in composites using AE. Composite material specimens are loaded under tension while recording AE, after which the signals that are recorded are used to create an independent damage mode signature. This leads to the ability to classify damage modes per time increment.

Mechanical tests are performed on specimens while recording AE signals. Carbon fiber reinforced polymer specimen with unidirectional 90°layup are loaded in tension up to failure to create a fingerprint of the matrix cracking damage mechanism. The same material with a crossply layup is loaded in tension under quasi static and fatigue loading. The fingerprint is then used to separate signals coming from this type of damage and other damage mechanisms. Next to the basic parameters of these signals, additional features are generated by processing the waveform of each signal. The wavelet transform is used to find frequency related features that characterize each signal. Machine learning algorithms are developed and used to cluster and classify the AE Signals.

Overall, the real-time clustering algorithms have proven to be successful in clustering and classifying incoming AE signals in an efficient way. The system was able to correlate incoming data to matrix cracking data, within a very limited time frame. With room for optimization the proposed methods seem fit to be applied in real-time applications. Before this can be done however, more testing and validation of these methods is required.

The most valuable addition to this study would be a proper validation of the clustering results. Regarding improvement of the system, the main bottleneck of the proposed algorithms is loading large datafiles. This can be solved by reading data in batches, but requires further development of the algorithms.

Abbreviations

Abbreviation	Explanation
AE	Acoustic Emission
AF	Average frequency
CP	Crossply
CFRP	Carbon fiber reinforced polymer
DF	Decay frequency
FFT	Fast Fourier Transform
ICD	intra-cluster distance
IQR	Interquartile range
IPCA	Incremental Principal Component Analysis
MAD	Median absolute deviation
MF	Median freqeuncy
NDE	Non-destructive evaluation
PCA	Principal component analysis
PF	Peak frequency
RF	Rise frequency
SNR	Signal to noise ratio
SVD	Singular Value Decomposition
UD	Unidirectional
WP	Wavelet Packet

List of Symbols

Symbol	**Unit**	**Description**
A	[dB]	Amplitude
a	[-]	Wavelet scaling factor
b	[-]	Wavelet shifting factor
c	[-]	Wavelet coefficient
C	[-]	Cluster coarsening factor
D	[s]	Duration
E	[eu]	Energy
R	[s]	Rise time
R_{noise}	[-]	Noise term
$\tilde{R}$	[-]	Cluster refining factor
T	[s]	Time
V	[m/s]	Wave velocity
μ	[-]	Statistical mean
μ_k	[-]	Cluster k centroid
δ	[-]	Evolutionary Intra-cluster distance
γ	[-]	Point-wise centroid drift vector
σ	[-]	Statistical variance

List of Figures

List of Tables

1
Introduction

The use of carbon fiber reinforced polymer (CFRP) is becoming more and more popular in the aircraft industry, mainly due to their high strength-to-weight ratio. However, due to the inherent inhomogeneity of the CFRP structures, damage develops in the early stages of its service life and accumulates continuously [1]. Therefore, monitoring the condition of the structure is essential to predict and prevent failure of parts. For this purpose, the non-destructive evaluation (NDE) method Acoustic Emission (AE) is most suitable. Contrary to other NDE methods, AE does not require an external stimulus. Instead it relies on the elastic waves that are created when there is a sudden release of strain energy, due to damage [2].

Testing with AE started as early as 85 years ago [3]. A lot of studies have been done since, covering a wide range of materials and applications. This non-destructive evaluation method has proven to be useful in determining the type, location and accumulation of damage within a structure [4–6]. Since AE is measured at the moment of damage development and progression, it is very suitable for real-time damage monitoring.

In previous studies, it is common that AE signals are clustered with an unsupervised learning method called K-means [7, 8]. These clusters can then in turn be correlated to damage mechanisms. However, there are two major downsides to this method when it comes to applying them in real time. The first is that the method has an iterative nature, causing it to be computationally expensive. The second downside is that the number of clusters have to be specified beforehand. For these reasons there is a need for a clustering algorithm that is fit for real-time applications. A first setup for a real-time clustering algorithm was proposed by Pomponi and Vinograov [9]. Here, a method is developed, which modifies the conventional k-means, making it more efficient and require less input by being data-driven. One of the main advantages of this method is that there is no need to specify the number of clusters beforehand. However, more research is necessary on the performance of this method before the method can be employed in real-time applications.

1.1. Objective of the research

The research objective and questions are based on the gap between current applications of AE and possible real-time applications. The research objective is to contribute to the development of algorithms that classify failure modes real time in composites using AE by testing composite specimens under tension while recording AE, after which the signals that are recorded are used to create independent damage mode signatures that lead to the ability to classify each failure mode per time increment. From the objective a number of questions can be generated, which are defined as follows:

1. Which methods exist to process AE data in real time using Wavelets?
   (a) What Wavelet types/parameters characterizes an AE signal?
   (b) What is the time-wise performance of processing methods?
2. What parameters characterize the AE signals?

  (a) In what way do the signals of matrix cracking differ from other signals?
  (b) How do the AE signals of matrix cracking change over time?
  (c) How does the layup of a composite specimen influence the AE signal?

3. Which methods exist to cluster and classify AE in real time?
   (a) What is the performance of clustering and classification algorithms?
   (b) What is the time-wise performance of the algorithms?

1.2. Methodology

In order to answer these research questions, several experiments are performed and a set of algorithms is created to process and analyze the data. The experimental work is all be performed at the lab of the Aerospace Engineering Faculty at TU Delft. A limitation of the current set-up is that the AE data is not recorded in real time. It is possible that the installation is adapted specifically for this task, but it is not a necessity for the purpose of this research. Instead, the data is simulated to stream continuously.

Two types of composite specimens are made: one with a unidirectional (UD) 90°layup and the other with a crossply (CP) layup. Five UD specimens will be tested under quasi static loading until failure. Signals from these tests will function as a signature for the matrix cracking damage mechanism. Five CP specimens are also loaded until failure in quasi static tests and another five specimens are loaded in fatigue.

An algorithmic framework is designed that processes the AE signals and clusters the data. Both the commonly used K-means as well as the real-time variant of this method are applied for the analysis. The matrix cracking signature is used to identify one of the clusters and is associated with the matrix cracking damage mechanism. Cluster evaluation metrics are used to assess the quality of each model. These metrics are also used to determine how similar a cluster and the signature data are.

1.3. Structure of the report

The structure of the report is as follows: the background information on AE is found in chapter 2. Then, theory about the processing of signals using the wavelet transform is outlined in chapter 3. After that, chapter 4 explains the algorithms required to cluster and classify the AE data. The experiment procedure and raw results are shown in chapter 5. The results of the experiments and analysis are described in chapter 6. Finally, conclusions are drawn and recommendations are described in chapter 7.

2
Background

This chapter covers the background information necessary to understand the origin and working of AE and the damage mechanisms of a composite structure. The concept of AE is explained in section 2.1 and the damage mechanisms are described in section 2.2. This information can in turn help with connecting the AE signals to their corresponding damage mechanisms. The chapter is summarized with some concluding remarks in section 2.3.

2.1. Acoustic Emission

AE is the spontaneous elastic energy release by materials when they experience deformation or fracture. This rapid energy release generates transient elastic waves from localized sources within a material.

2.1.1. Sources of AE

AE can be caused by many different types of deformation and fracture. In composite materials is generally caused by three damage mechanisms which are matrix cracking, delamination and the fracture of fibers. Those damage mechanisms are the characteristic reactions when the material is loaded. Besides deformation and fracture, there are also other mechanisms that are detectable with AE sensing. Some examples are leaks, friction and solidification. Such examples are referred to as secondary sources to separate these from the AE signals caused by mechanical deformation of loaded materials.

Physically, sound is the vibration of molecules through a medium. This phenomenon is not directional, but instead it propagates into all directions taking the form of a sphere. A transducer located near an AE source can often detect the event. In case multiple transducers are used, there is also a possibility to retrace the location of the damage. The potential to locate the damage as it occurs is a unique feature of AE testing. Contrary, other methods generally need more information as to where the damage might be and what orientation it has to successfully assess the damage [2].

2.1.2. Measuring AE

AE sensors are mounted to the surface of a material to sense the motion coming from AE signals and transform this motion into a voltage signal which can be read and processed. The sensitivity of the piezoelectric parts in the transducers varies with frequency and is often largest in the range of 0.1 to 2.0 MHz. This means that only frequency components in this range are observed from the AE signals. Hence, also the static surface strain is not measured. If the elastic wave contains enough frequency components within the range of the transducer, the mechanical motion is converted to an electrical potential. Whether a potential is created depends on the sensitivity of the transducer. As long as the voltage created is higher then the threshold, the signal is picked up by the transducer.

The physical nature of the source of the AE signal will determine whether it is detectable or not. It dictates the energy of every spectral element in the elastic waves. Another factor that determines the detectability of a signal is the time for the material to regain equilibrium before the crack has grown to a significant size. If the time taken is short enough, the signal is not detected. For the waves to be detectable, the rise time has to be in the range of one over the transducer bandwidth. This is when the frequency elements of the waves are in the detectable range.

A voltage time signal is the output of the transducer. This output however, depends considerably on the characteristics of the transducer. Therefore, it is important to take this in to account when trying to reproduce AE tests results for example. For AE to be a successful testing method the transducers have to have similar characteristics or the processing method has to work in a generic way to deal with the dissimilarities. AE testing is in most cases performed with transducers that use piezoelectric elements for converting motion to an electric signal. These elements are connected to the surface of the material such that the motion propagates into it. Due to the piezoelectric qualities, the dynamic strain is converted to a potential that can be tracked over time.

The motion of the material surface due to AE consists of three components: one component is normal to the surface and the two other components are orthogonal and tangential to the surface. Different waves are detectable by the transducers because all of the waves that travel at longitudinal, shear and rayleigh speeds contain a normal component. A typical frequency range for the processing of AE signals is between 30kHz and 1MHz [10]. However, for some specific applications there might also be relevant information in the lower frequency range, at sounds audible for humans, below 20kHz. In such cases, a conventional microphone can be used.

As the frequency increases, the attenuation of the motion increases as well at a quick rate. For materials with higher attenuation like composite materials, it might be crucial to include some of the lower frequencies to have the capability to sense the AE signals. In the opposite case, when a material has a lower attenuation, the AE signals are likely to contain more information at higher frequencies, with the benefit of having less noise in this range. The piezoelectric elements of the transducers can be made to size in order to select a relevant frequency range [11].

2.1.3. Noise

A common issue in any nondestructive method is the presence of noise. The AE method is no exception as the AE signals can carry unrelated or interfering signals that have no relevance.

Continuous or intermittent signals can interfere with the AE signal either coming from within the part or from an external source. The examination of noise affecting the AE signals is an integral part of the AE testing method. The sources should be identified and then eliminated altogether if possible. One measure to reduce the noise is to set a frequency threshold. A threshold of 50kHz or above can be used to diminish the amplitude of noise, as it is lower in this range [12]. A narrow and high frequency band can also reduce the noise, however the issue here is that useful parts of the AE signal also gets lost. The same happens when reducing the instrument gain or increase the frequency detection threshold.

Noise from electrical or mechanical sources can be estimated by investigating the signal noise when the whole system is at rest, thereby only measuring signals from the environment. It is important to take note of any machinery that may be running during the AE test. Noise will affect some parameters of the AE signal while other remain unchanged. It can have an influence on the peak amplitude, but noise spikes for example will not change parameters like the root mean square voltage. When the purpose of the AE test is not to detect mechanical failure, it is important to be aware that any movement of mechanical parts can generate unwanted signals when it is in contact with the part. Damaged bearings in roller assemblies is an example of this type of noise.

One way to identify mechanical noise is to assess the frequency and rise time of the AE signal. Compared to signals from actual damage, the frequency is low as well as the rise time. The AE burst signals caused by cracks in general have a rise time from threshold to peak of less than 25 μs if the transducer is located close to the source. However, frequency discrimination is usually more reliable. Before performing AE tests, background noise needs to be investigated to see if they create false AE signals and to see the effect on outcome of the test. The noise of electrical equipment and any operating machinery has to be either physically removed or filtered in the data analysis [11].

2.1.4. AE signal

A typical waveform caused by an AE event is shown in fig. 2.1, in the voltage time domain. For most AE applications, a voltage threshold is used in order to only measure the relevant signals and not noise. The definition of the rise time is the time where the signal passes the threshold to the point where the signal reaches its maximum. The duration is taken as the point where the signal first passes the threshold until the point where the voltage is lower than the threshold. The energy is calculated as the area under the signal curve that is above the threshold. The counts is how many times the signal has exceeded the threshold.

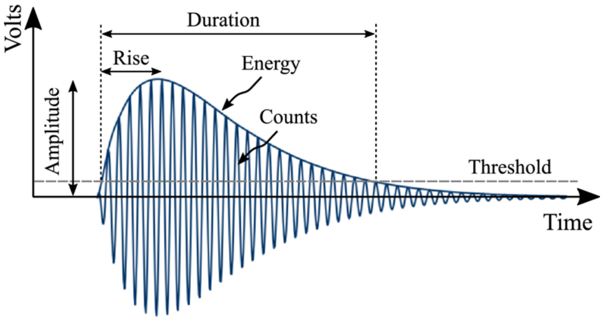


Figure 2.1: General AE signal [13]

2.2. Damage mechanisms

For the scale of the structures that will be analyzed, damage mechanisms on a meso-level are of interest. In fiber reinforced composite materials the damage mechanisms that occur in tensile loading can be grouped into three different categories: the matrix can crack, the plies can delaminate and the fibers can break.

2.2.1. Matrix cracking

Matrix cracking is a significant type of damage since it can lead to the development of other damage mechanisms. A local cluster of fibers that break or delamination of plies could be a consequence. The preceding mechanism of matrix cracking is the debonding of a fiber with the matrix. When matrix cracking occurs in a 90 °ply it is generally called transverse cracking. The process of a formation of a transverse crack is shown in fig. 2.2. Since the interface is in most cases weaker than the fiber or matrix, debonding occurs first, as depicted in fig. 2.2(a). If this occurs at multiple fibers, they can connect to form a macroscopic crack, as seen in fig. 2.2(b). An actual microscopic picture of a transverse crack is shown in fig. 2.2(c). The cracks generally form along the thickness direction of the ply or set of plies that have the same orientation. The initiation is often at the free edge of the part, which is where the stress state is different than anywhere else [14].

The transverse cracks would eventually lead to failure in a UD transverse laminate by detachement. Also, fiber failure is unlikely in this type of laminate since the applied stress will never be higher than the matrix failure stress [15]. Damage developed in one ply may influence the damage growth in other plies.

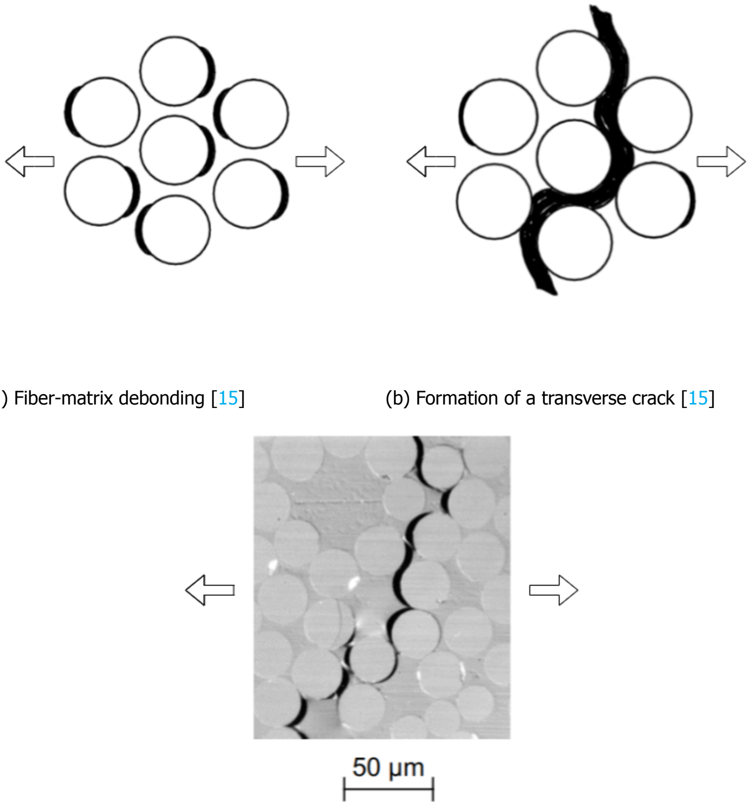
2.2.2. Delamination

The difference in elastic properties between neighboring plies is what causes delamination. As mentioned before, matrix cracking is in most cases an essential precursor for this damage mechanism to take place. Figure 2.3 shows that a delamination has initiated from a transverse crack tip at the interface of 0°/90°plies. As the delaminations become larger, the stress state will also change with it in the transverse ply. Delamination influences the growth of transverse cracking as well as the reverse, leading to stress concentrations. This could in turn lead to the breaking of fibers.

2.2.3. Fiber breaking

The fiber reinforcement in composite materials is made from bundles of fibers. The size of bundles will be different and they will also inhibit defects either from manufacturing of the fibers themselves or from the composite part. This leads to them having a different failure strain. The diversity of strengths will lead to isolated fiber fracture at different values of applied strain. Even at moderate loads, there will be fibers breaking that that are on the lower end of the strength spectrum. This in turn can lead to failure in adjacent fibers, which eventually leads to failure of the entire part.

Here again matrix cracking can be a precursor for the damage mechanism. The absence of matrix cohesion leads to stress concentrations in the fibers, leading to local fiber breaking. Two different scenarios of individual fibers breaking are visualized in fig. 2.4, which depend on the efficiency of the interface. In case there is a strong interface, the matrix will crack from the low strength fiber breaks. This matrix crack grows in perpendicular direction to the fiber and load. The crack will reach adjacent



(a) Fiber-matrix debonding [15]

(b) Formation of a transverse crack [15]

(c) Microscopic image of a transverse crack [15]

Figure 2.2: Evolution of a matrix crack

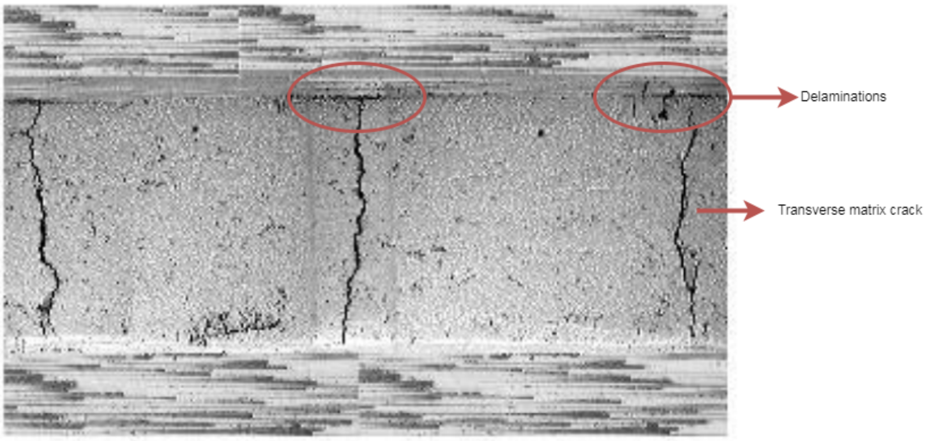
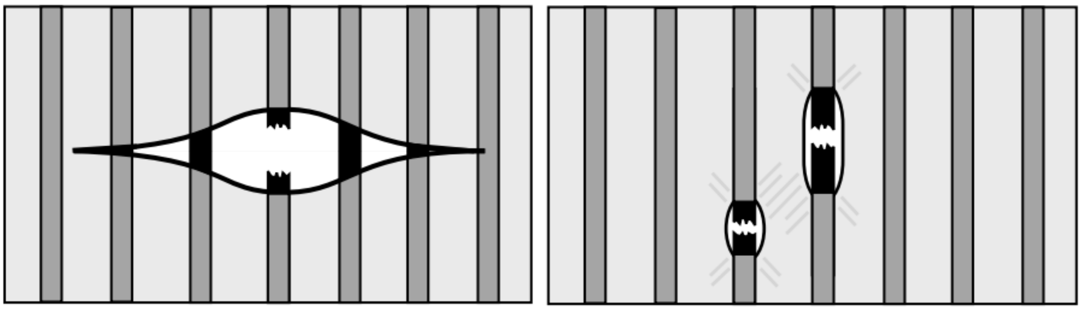


Figure 2.3: Delamination [16]

fibers and continues growing around them, also called bridging which is shown in fig. 2.4(a). This case is however rare given a high fiber volume fraction, high matrix bond strength and ductility. In the second case in fig. 2.4(b) the debonding extends along the fiber direction and as the matrix yields, the local loads get redistributed to the fibers, causing them to fail.



(a) Fiber breaking due to bridging [15]

(b) Fiber breaking by debonding and matrix yielding [15]

Figure 2.4: Propagation of cracks from a single fiber break

2.3. Conclusion

In this chapter the concept of AE was described in detail. The different sources of AE that will occur in the experiments that follow are outlined. Whether a signal can be picked up by the sensors, depends on the sensor specifications and the intensity of the AE signal. This points to a general issue of AE, which is that the analysis is influenced by the sensor type and this has to be taken into consideration when comparing results to other studies. Another important aspect is the influence of external sources. Testing equipment and surrounding sounds can provide noise to the signals, which has to be filtered correctly. Basic signal features were introduced, however, there is much more information that can be extracted by further analyzing the waveform of each AE signal. Additional feature extraction that can prove to be characteristic for the damage mechanisms and filtering of AE signals are therefore discussed in the next chapter.

3

Signal processing using Wavelets

In this chapter the signal processing of AE signals will be discussed. More specifically, how wavelets are used to reconstruct the signals obtained from AE testing. The significance of Wavelets is stated in section 3.1. The concept is then introduced in section 3.2. Lastly, some concluding remarks are stated in section 3.4.

3.1. Introduction

During AE testing there is often a large amount of background noise. Understanding and eliminating these noise sources is necessary in order to identify the original signal. The types and sources of noise have been discussed elaborately in section 2.1.3. Due to the presence of these noises it is difficult to correctly interpret the AE signature. It is therefore of utmost importance to eliminate or reduce the noise in the signal. There are various methods to reduce noise like using filters or decreasing the gain of the signal. However, lowering the gain might lead to signals getting filtered out completely, which is not favorable. With Fast Fourier Transform (FFT) one could filter out noise, but an even better way to reconstruct the original signal is to use a signal processing method called the Wavelet Transform. This method has been successfully applied to de-noise AE data [17]. With the Wavelet Transform one can decompose a signal into different frequency bands. Frequency bands that contain useful information are kept whereas frequency bands that contain noise can be disregarded. The main advantage of this method compared to Fast Fourier Transform (FFT) is that it gives the frequency content at any time t, whereas FFT can not tell at what time frequencies were present. Another benefit is that a certain wavelet can be chosen which shape corresponds to the phenomena that is being investigated. In this way, more of the actual shape of the signal will be retained when filters are applied.

Over the past years a series of developments and applications have been created using the wavelet analysis. It has been used among other things to describe complex algebraic equations and to analyze empirical continuous data that was gathered from several signal types and ranges of resolution. So far, the most widespread application of the wavelet transform is data compression. Other examples include signal processing, pattern recognition, numerical analysis and image processing. Wavelets allow the decomposition of complex information such as music, speech, images and patterns at different positions and scales [18].

3.2. Wavelets Transform

In this section the mathematical background of the Wavelet Transform will be explained and the different types of wavelets will be described to figure out what is most suitable for this research.

3.2.1. Description

Wavelets are a family of functions created by translating and scaling a master function referred to as the "mother wavelet", $\Psi(t)$. The functions are defined by:

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\Psi(\frac{t-b}{a}),\ a,b \in \mathbb{R}, a \neq 0 \tag{3.1}$$

where a is the scaling parameter that determines to what degree the wavelet is compressed and b is the translation variable that determines at which time the wavelet is placed. If $|a| > 1$, the wavelet compresses to a smaller version of the mother wavelet which corresponds primarily to larger frequencies. Contrarily, when $|a| < 1$, the wavelet has an increased width in time which is conforms to lower frequencies. So the time-width of wavelets is adapted to their frequencies. As the scale a becomes smaller, the time domain resolution gets finer whereas the resolution of the frequency domain becomes coarser.

A function $f(t)$ can be reconstructed using wavelets, provided they form an orthonormal basis, by means of the following equation:

$$f(x) = \sum_{m,n=-\infty}^{\infty} \tilde{f}(m,n)\Psi_{m,n}(t) \tag{3.2}$$

where $\Psi m, n(t)$ are the wavelets given by:

$$\Psi_{m,n}(t) = a_0^{-m/2}\Psi(a_0^{-m}t - nb_0) \tag{3.3}$$

with $a_0 \neq 0, 1$ and $b_0 \neq 0$, and $\tilde{f}$ is the discrete wavelet transform:

$$\tilde{f}(m,n) = \int_{-\infty}^{\infty} f(t)\overline{\Psi}_{m,n}dt \tag{3.4}$$

However, it is not guaranteed that the original function, $f(t)$ can in all cases be recomposed from the wavelet coefficients. It would be possible if the mesh of the discrete lattice is infinitesimally small, but for meshes that are coarser the coefficients might not hold the necessary information to determine $f(t)$. There are certain values for the lattice parameter (m, n) where it would be possible to reconstruct a numerically stable equation [18].

3.2.2. Wavelet Analysis

One could distringuish between two different types of wavelets: the continuous wavelet transform (CWT) and the discrete wavelet transform (DWT). They differ in the way that they operate, namely at a different set of scales and positions. The CWT is able to operate at every scale. However, due to the fact that CWT is computationally very expensive, an upper bound is determined. In addition, the analyzing wavelet in CWT is shifted in a smooth way across the whole domain of the data that is considered. In order to get a reduction in the computing power needed, the wavelet coefficients can be determined on a smaller group of scales. That is refered to as the DWT.

Principally, the wavelets can be split in to two types: orthogonal and nonorthogonal wavelets. The generic term $wavelet\ function$ refers to one of the two types of wavelets. An orthogonal set of wavelets is referred to as a wavelet basis, whereas a nonorthogonal wavelet set is called a wavelet frame. DWT makes use of a wavelet basis, while frames are employed in both DWT and CWT. As time passed, a considerable amount of wavelet functions appeared in literature. The different type of wavelets can be matched with the shape of the signal that is to be transformed. In case the wavelet family is close to the actual signal, it can be selected for use. Figure 3.1 gives a few different types of wavelets. The Gaussian and the Mexican hat wavelets are defined by an explicit mathematical expression and therefore only used in continuous decomposition. The rest of the wavelets that are shown are merely an estimate of a continuous wavelet created from a set of the original filter points. In general, the DWT is applied to compress a set of data in case the signal is sampled, whereas the CWT is used to analyze signals [19].

The CWT operates at a continuous set of scales and positions. Also, during the analysis of the function, the wavelet is shifted in a smooth manner across the whole domain. The wavelets can be either real or also have imaginary components. Generally, the real part can detect a sharp transition in the signal, whereas the complex wavelets are used to distinguish between the components related to amplitude or phase. The result of the CWT analysis is a number of wavelet coefficients, where the scale and position defined the coefficient.

The wavelet with the appropriate scale and shift is multiplied with the corresponding wavelet coefficient, which results into the original signal. The wavelet coefficients can be used for data operations due to the fact that the original signal can is represented by a wavelet expansion.

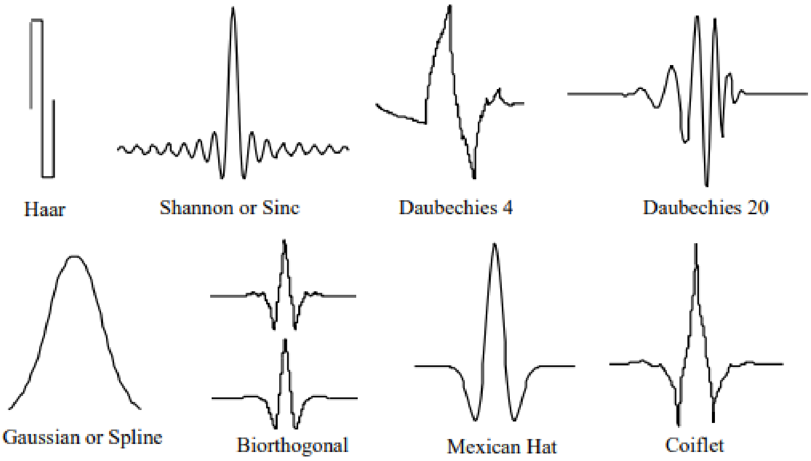


Figure 3.1: Some examples of types of Wavelets [20]

Calculating the wavelet coefficients at each and every scale for the DWT requires a lot of computational effort. But, a smarter way is to choose a subset of scales and translation based on the powers of 2. This will lead to a more efficient analysis that has the same accuracy. The basis wavelet functions from eq. 3.3 are then usually in the form[19]:

$$\Psi_{m,n}(t) = 2^{-m/2}\Psi(2^{-m}t - n) \tag{3.5}$$

3.3. Wavelet processing

Python has become very popular over the last few years due to its user friendliness and wide range of applications. This section will look into the signal processing using Wavelets in Python. By deconstructing a signal the noise can be eliminated and meaningful information is kept by reconstructing with the remaining parts. Characteristic features have to be extracted from the remaining parts that can be used for classification of the signals. However, before any wavelet processing, it is important to select the right mother wavelet that will represent the signal.

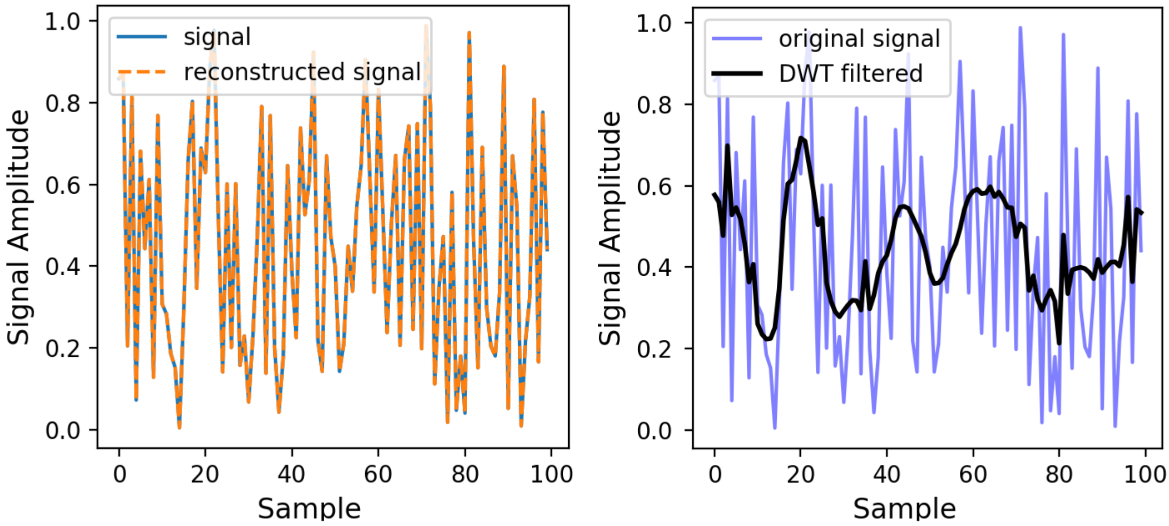
3.3.1. Reconstructing a signal in Python

Two ways can be used in Python to deconstruct and reconstruct a signal. Deconstruction can be done by applying the DWT or by a wave decomposition method [21], both available in the PyWavelet library. Then the signal can be reconstructed using the inverse DWT or a wave reconstruction method [22].

Figure 3.2(a) shows the reconstruction of a signal using the DWT with the Daubechies 4 wavelet. With this method it is possible to leave out some of the higher frequencies. Figure 3.2(b) shows the reconstructed signal where an arbitrary threshold of 0.63 was applied. This means that when a wavelet coefficient is lower than 0.63, it is set to 0. This causes the high frequency noise to be removed from the signal. This way of processing is also possible with other methods, however, the advantage of using Wavelets lies in the many wavelet shapes that are possible. A certain wavelet can be chosen that have the shape characteristic corresponding to the phenomena that is being investigated [23].

3.3.2. Signal denoising and thresholding

The recommendation for a threshold for an automatic inspection system is around 1.6 to 2.0 times the average level of the noise. If that is the case then in order to detect a defect the signal to noise ratio (SNR) must exceed a certain value. The application of the wavelet transfrom can increase the SNR, but only to a limited extend. One way to increase the SNR is by pruning the wavelet transforms. This is when a certain range of the wavelet transform $W_s(a,b)$ is limited to a range of $a \in (a_1, a_2)$, and values outside of this range are set to zero. This is comparable with low or band-pass filtering in FFT. What is different is that $W_s(a,b)$ is the correlation between the daughter wavelet $h_{a,b}(t)$ and the signal $y(t)$, and by eliminating some of the contributions to $W_s(a,b)$ the results are pruned. The eliminated parts



(a) Reconstructed signal

(b) Filtered signal

Figure 3.2: The reconstructed wavelet signal and a filtered wavelet signal

are likely originating from noise. In mathematical terms the process can be written as:

$$W_F(a,b) = \begin{cases} 0 & \text{for } a < a_1 \\ W_F(a,b) & \text{for } a_1 < a < a_2 \\ 0 & \text{for } a > a_2 \end{cases} \tag{3.6}$$

Another way to reduce the noise is by thresholding the wavelet transform. With this method, all wavelet coefficients that have a magnitude less than a specified threshold are set to zero, and the other coefficients reconstruct the original signal:

$$W_F(a,b) = \begin{cases} 0 & \text{for } |W_s(a,b)| < \lambda \\ sign(|W_F(a,b)|)\dot{(}|W_F(a,b)| - \lambda) & \text{for } |W_s(a,b)| > \lambda \end{cases} \tag{3.7}$$

where λ is defined as the threshold. The process results in local changes in the time-scale domain. It works optimally for white noise, but could be unsuccessful when the noise is correlated. Important information of the signal can be lost in case the noise has a high correlation with one of the daughter wavelets $h_{a,b}(t)$. To prevent this from happening, a level dependent soft threshold can be used. The threshold can then be written as a function of the dilation a. By utilizing known information on the noise characteristics a function can be created: $\lambda = \lambda(a)$ [24].

3.3.3. DWT classification

The idea behind DWT signal classification is to split a signal into different frequency bands. If the types of signals have different frequency characteristics, the difference will show up in one or more of the frequency bands. The features are generated from each sub-band, which are then is used as an input for a classifier, which can differentiate between the various types of signals. A process of decomposing a signal is shown in figure 3.3. Here, each time the approximation coefficients are based on the lower level of the coefficients. Another type of tree can be seen in figure 3.4, that is used in [25–27]. In this structure each level of approximation and detail split up into two branches, like a pyramid structure. For the features one could think of using statistical data from the detail coefficients like: variance, standard deviation and root mean square value.

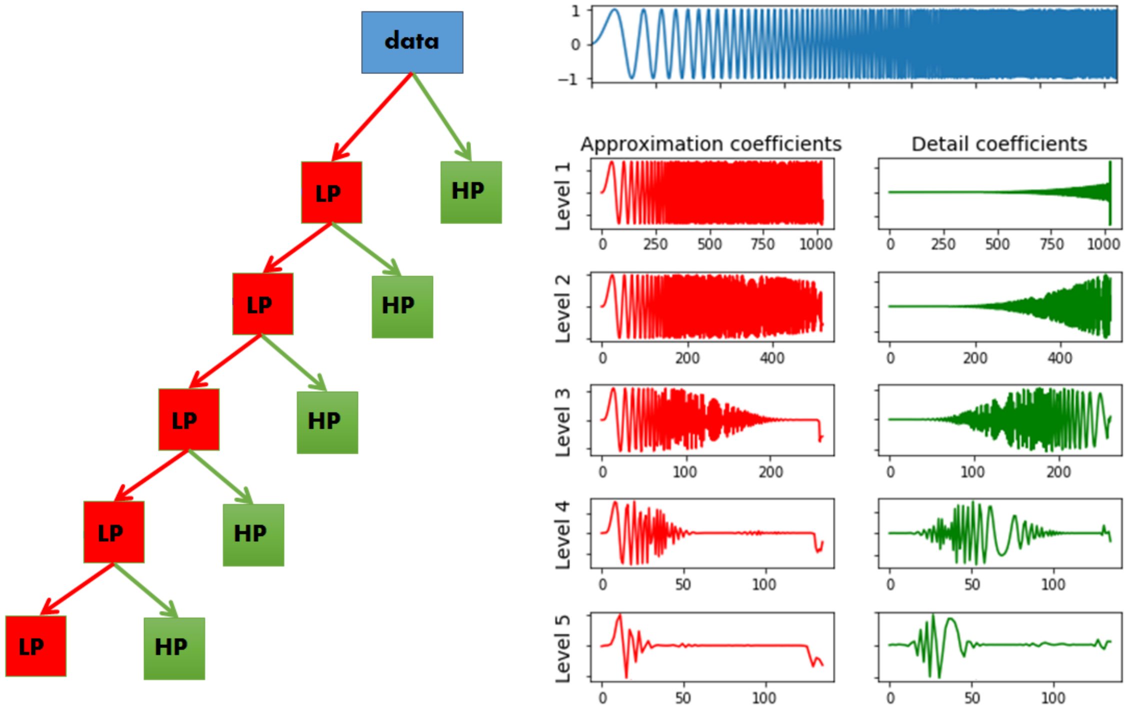


Figure 3.3: Decomposition of frequency sub-bands [23]

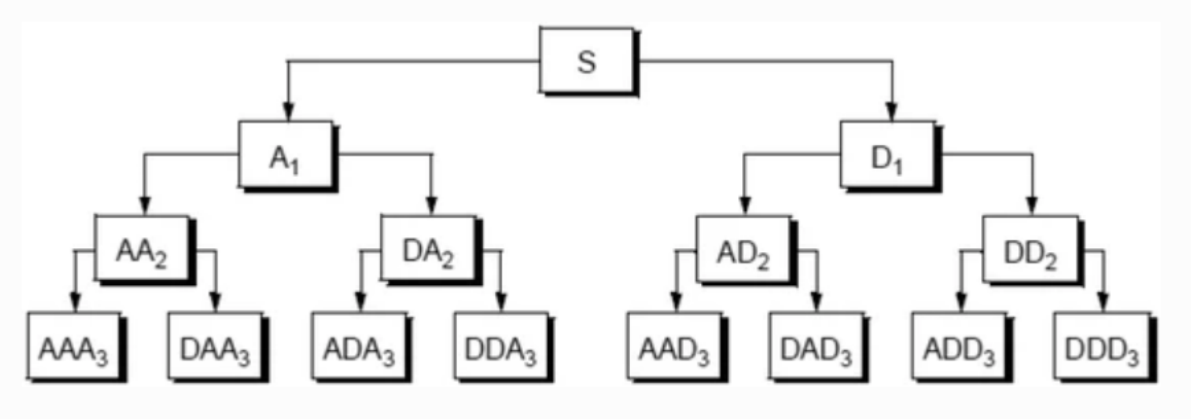


Figure 3.4: Decomposition of frequency sub-bands into a pyramid structure [28]

3.3.4. Wavelet transform features

A. Marec et al. [29] used both DWT and CWT analysis to extract features that were used to cluster and classify the AE signals generated from GFRP damage mechanisms. The type of wavelet that is used for the analysis is the Daubechies with 10 vanishing moments. Apart from the duration of the signal, the sum and maximum square modili value of the coefficients from the CWT are incorporated. Also, the maximum value of the squared coefficients from DWT for detail levels D3-D5 is taken. Principal Component Analysis (PCA) is then used to reduce the dimensionality of the data and visualize the data. From a subset of data it is shown that by using fuzzy C-means clustering, the events from matrix cracking and fiber-matrix debonding are clearly separable.

A slightly different way of processing the AE signals was proposed by T.H. Loutas et al. [30]. AE signals from quasi-static tensile tests of GFRP materials is analysed using the wavelet transform. The wavelet that is used it the Daubechies 20 wavelet. To this point there has been no explicit way to determine what the optimal wavelet and number of decompositions are for a certain application. There are no specific rules to choose one family of wavelets over another. The difference between the wavelets as well as their characteristics have to be known to make a justified choice. However for the number of detail levels there are mathematical criteria that determines whether a specified number of levels is acceptable of more levels are needed. This criteria are called the entropy criteria and can be used to determine the optimal number of levels [31] for a signal. However, when the data consists of a large number of signals, criteria for one waveform might not satisfy the criteria for another waveform.

The number of levels used in this study was set to 6, after a some systematic trials. It was found that using less levels was not sufficient, and using more than 6 was redundant. There are a number of reasons why the Daubechies family was chosen as mother wavelet for the AE signals. It has biorthogonal wavelets and is sufficiently regular, but not symmetric. Families with similar characteristics could also be used like symlets and coiflets which leads to only slightly different results. The energy from the different detail levels was then used to analyse the signals. The energy can be calculated as:

$$E^{(i)}(t) = \sum_{\tau=t_0}^{t} \left(f^{(i)}(\tau)\right)^2 \tag{3.8}$$

where the i stands for the number of the level.

3.3.5. Mother wavelet selection

The decomposition of a signal into wavelet levels results into the vectors of the detail and approximation coefficients. The more levels that are taken, the lower the frequency of the components become. Since the frequency becomes lower for higher levels, less coefficients are needed to convey the information from the lower frequencies. Fast Fourier Transform could be used to analyze the frequency content of the wavelet levels in more depth. To assess the performance of a mother wavelet, the Shannon entropy can operate as a cost function [31] . The outcome of the Shannon entropy evaluation is a probability distribution of the signal energy of each vector of coefficients. The value of this function will be big when the coefficients of the vector are about the same in magnitude and the energy of the signal is evenly distributed. This can then be seen as the noise of the signal. The entropy value is small in the case where there are a few dominant coefficients. The Shannon entropy of the coefficient vector cm, n at level m, located at position n and of size N, is given as follows:

$$Entropy(c_{m,n}) = -\sum_{i=1}^{N} p_i log(p_i) \tag{3.9}$$

with p_i being the energy probability distribution which is defined as:

$$p_i = \frac{|c_{m,n}(i)|^2}{||c_{m,n}||^2} \tag{3.10}$$

To make sure that the vector of coefficients describes a large amount of the original signal, the signal energy has to be maximized. This can be done by:

$$E_s(c_{m,n}) = \sum_{i=1}^{N} |c_{m,n}(i)|^2 \tag{3.11}$$

which is equivalent to eq. (3.11). The ratio of the energy and entropy of the coefficient vector can tell which mother wavelet extracts the most relevant components of the original signal. In order to find the best performing mother wavelet, the following ration has to be maximized[32] :

$$Energy - to - entropy(c_{m,n}) = \frac{E_s(c_{m,n})}{Entropy(c_{m,n})}. \tag{3.12}$$

Another evaluation metric that is used to select a mother wavelet is the computational effort required to use a specific mother wavelet for the analysis of a signal. Since the purpose is to process a great amount of AE signals in real-time, the computational effort should be minimal.

3.4. Conclusion

This chapter covered processing AE signals with the Wavelet Transform. It is safe to say that the Discrete Wavelet Transform is most suitable for the purpose of this analysis, due to its efficiency and relevant feature extraction in previous studies. In order to discover which wavelet type is best for processing the signals, a selection method has been outlined. With the appropriate wavelet type, the AE signals can be decomposed into different frequency bands. There are two common ways of doing this: in a tree-like structure or in a pyramid-like structure. From the decompositions, the energy can be calculated which then represents the energy content for that specific frequency band. The energies will be used in distinguishing the AE signals coming from different damage sources. Now that so many features have been collected from the AE signals, it is time to figure out how to cluster this data and correlate the clusters to damage mechanisms.

4

Clustering and classification

In this chapter the algorithms necessary to cluster and classify the AE data are described. The significance of this process is explained in section 4.1. In section 4.2 dimensionality reduction of data by means of the Principal Component Analysis is described. The K-means method and two real-time variations of this method are outlined in section 4.3. Lastly, in order to assess the clustering performance and evaluate clusters, different evaluation metrics are required, which are described in section 4.4.

4.1. Introduction

Once the AE signals are processed with the methods described in chapter 3, the goal is to cluster them into groups and relate them to the different damage mechanisms shown in section 2.2. Per signal, over 20 characteristic features are collected. Clustering data that has so many dimensions can lead to issues such as small isolated clusters. Also, distance metrics tend to not work properly for such cases. This issue is also referred to as the ”curse of dimensionality” [33]. For that reason the data is reduced to two dimensions with a method called Principal Component Analysis (PCA). The reduced data can then be clustered with the K-means method and real-time variants of this method. The performance of these methods can be assessed with cluster evaluation metrics. This is mainly done based on the separation between clusters and variance within individual clusters. The same metrics can be used to evaluate similarity of a cluster with known data and therefore act as classification metrics.

4.2. Principal Component Analysis

PCA can be applied to data with the purpose to reduce the dimensionality of the data. By projecting the data onto a subspace of a 2D dimension for example, the data can be visualized much better. In the case of streaming data, there is the option to use Incremental Principal Component Analysis (IPCA).

4.2.1. Conventional PCA

For the analysis a dataset X is considered consisting of n data points and d features:

$$\mathbf{X} = \begin{bmatrix} x_1^t \\ x_2^t \\ \vdots \\ x_n^t \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^d \\ x_2^1 & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \vdots & \vdots \\ x_n^1 & x_n^2 & \dots & x_n^d \end{bmatrix} \tag{4.1}$$

It is important for this data to be standardized, which means the mean equals to 0 and the standard deviation equals to 1. The reason for this is that the different features in most cases have different units, which means the effect of the feature on the outcome is proportional to the magnitude of the unit. Not applying standardization would significantly reduce the clustering algorithm's performance.

After this, the covariance matrix for the data can be determined:

$$Cov(\mathbf{X}) = E[\mathbf{X}\mathbf{X}^T] = \begin{bmatrix} C_{11} & C_{12} & \dots & C_{1d} \\ C_{21} & C_{22} & \dots & C_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ C_{d1} & C_{d2} & \dots & C_{dd} \end{bmatrix} \tag{4.2}$$

which is a symmetric matrix of size $(d \times d)$ where the component C_{kl} represents the covariance between variables x^k and x^l. An orthogonal basis can be created using the eigenvalues λ_k and eigenvectors e_k, which are solutions to:

$$Cov(\mathbf{X})e_k = \lambda_k e_k, \quad k = 1, 2, \dots, d \tag{4.3}$$

A sorted orthogonal basis is constructed where the first eigenvector has the direction of the biggest variances in the data. That means that the directions where $\mathbf{X}$ contains the most amount of energy are determined. The data can then be represented by a number of dimensions lower than d, by selecting a number of basis vectors from the orthogonal basis. A matrix $\mathbf{A_K}$ with dimension (d×K) which consists of the first K, eigenvectors can be used to transform $\mathbf{X}$:

$$\mathbf{Y} = \mathbf{X}\mathbf{A_K} \tag{4.4}$$

where $\mathbf{Y}$ are the coordinates in the reduced system found by the eigenvectors [29].

4.2.2. Incremental Principal Component Analysis

The purpose of Incremental Principal Component Analysis is to incrementally update the eigenvectors of the covariance matrix. The covariance matrix is simply the mulitplication of the singular value decomposition (SVD) with its transpose. Given a $d \times n$ data matrix $\mathbf{A}$ for which the SVD is already calculated:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T \tag{4.5}$$

A new set of data $\mathbf{B}$ of shape $d \times m$ is added to $\mathbf{A}$ and the SVD has to be computed efficiently:

$$[\mathbf{A} \quad \mathbf{B}] = \mathbf{U}'\Sigma'\mathbf{V}'^{\mathbf{T}} \tag{4.6}$$

Defining $\overline{\mathbf{B}}$ as the orthogonal component to $\mathbf{U}$, the concatenation of $\mathbf{A}$ and $\mathbf{B}$ can be defined as:

$$[\mathbf{A} \quad \mathbf{B}] = [\mathbf{U} \quad \overline{\mathbf{B}}] = \begin{bmatrix} \Sigma & \mathbf{U}^T\mathbf{B} \\ 0 & \overline{\mathbf{B}}^T\mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{V}^T & 0 \\ 0 & \mathbf{I} \end{bmatrix} \tag{4.7}$$

Now define $\mathbf{R} = \begin{bmatrix} \Sigma & \mathbf{U}^T\mathbf{B} \\ 0 & \overline{\mathbf{B}}^T\mathbf{B} \end{bmatrix}$. This is a square matrix which has size $k+m$, where k is defined as the amount of singular values in Σ. Now the SVD of $\mathbf{R}$ can be determined as $\mathbf{R} = \overline{\mathbf{U}}\,\overline{\Sigma}\,\overline{\mathbf{V}}^T$. The SVD of $[\mathbf{A} \quad \mathbf{B}]$ can then be written as:

$$[\mathbf{A} \quad \mathbf{B}] = ([\mathbf{U} \quad \overline{\mathbf{B}}]\overline{\mathbf{U}})\,\overline{\Sigma}\,(\overline{\mathbf{V}}^T \begin{bmatrix} \mathbf{V}^T & 0 \\ 0 & \mathbf{I} \end{bmatrix}) \tag{4.8}$$

The only matrices of interest are $\mathbf{U}'$ and Σ' and $\mathbf{V}'$. Therefore the incremental PCA boils down to the following steps, given $\mathbf{U}$ and Σ from the SVD of $\mathbf{A}$:

1. Take the QR decomposition of $[\mathbf{U}\,\Sigma\,\mathbf{B}]$: $[\mathbf{U} \quad \overline{\mathbf{B}}]\mathbf{R} = [\mathbf{U}\,\Sigma\,\mathbf{B}]$, to get $\overline{\mathbf{B}}$ and $\mathbf{R}$
2. Calculate the SVD of $\mathbf{R}$: $\mathbf{R} = \overline{\mathbf{U}}\,\overline{\Sigma}\,\overline{\mathbf{V}}^T$
3. $\mathbf{U}' = [\mathbf{U} \quad \overline{\mathbf{B}}]\overline{\mathbf{U}}$

Now the updated covariance matrix can be determined and the specified number of vectors can be selected [34].

4.3. Clustering algorithms

To cluster the AE data, the K-means method is used, which is common for AE signal clustering. In order to cluster data real-time, the Incremental K-means method is utilized. There also exists a more advanced version that evaluates the clusters as they develop, called the Adaptive Sequential K-means.

4.3.1. K-means

In k-means clustering method there is a set of n data points in d-dimensional space R^d and an amount of centroids k and the problem is to determine a set of k points in R^d which minimizes the mean squared distance from every data point to the closest centroid. The method falls within the general class that is called variance based clustering. A local minimal solution is found for the k-means problem by employing a simple set of iterations. This method outputs the data points related to a cluster and the centroid of each cluster. This can be used to label new data in the same clusters. K-means is used for applications like pattern clustering and file compression [35][36]. Given a set of n d-dimensional vectors:

$$\begin{aligned} X_1 &= (x_{11}, x_{11}, \dots, x_{1d}) \\ X_2 &= (x_{21}, x_{22}, \dots, x_{2d}) \\ &\vdots \\ X_n &= (x_{n1}, x_{n2}, \dots, x_{nd}) \end{aligned}$$

The goal is to partition the vectors $X_1, X_2, \dots, X_n$ into K clusters $C_1, C_2, \dots, C_k$. The k-means algorithm finds the locations $\mu_i (i = 1, \dots, K)$ of the clusters that have the minimum distance from the vectors to the cluster centroids. The following cost function F is minimised:

$$F = \sum_{i=1}^{k} \sum_{X_j \in C_i} ||X_j - \mu_i||^2 \tag{4.9}$$

where μ_i is the mean of the points in C_i. The steps of the k-means algorithm are as follows:

1. Split the n vectors into K subsets
2. Calculate the centroid of each cluster in the current partition. The centroid is computed as: $\mu_i = \frac{1}{|c_i|} \sum_{j \in c_i} \underline{x}_j, \forall i$
3. Allocate each vector to the nearest cluster centroid.
4. Repeat the process from step 2 until convergence is reached.

The algorithm converges when the centroids do not change anymore. In order to assign a point to a specific cluster, the distance between them needs to be known. This can be done in several ways but the most common one is by using the Euclidean distance. The Euclidean between two vectors $x1$ and $x2$ is calculated as:

$$distance(x1, x2) = \sqrt{(x_{11} - x_{21})^2 + (x_{12} - x_{22})^2 + \dots + (x_{1d} - x_{2d})^2} \tag{4.10}$$

Convergence of the algorithm does not guarantee a global optimum for the problem. However, a local optimum is always found [37].

4.3.2. Sequential K-means

The conventional k-means method is involves iterative process which makes it unsuitable for applications where data is accumulated with time. An alternative to this problem is the sequential k-means method. This algorithm deals with streaming data in an efficient way. The steps are as follows[38]:

1. Make initial guesses for the centroids μ_1, μ_2, ... μ_i.
2. Set the counts $n_1, n_2, \dots n_i$ of the elements to zero

3. Obtain the incoming vector of dataset x
4. If μ_i is nearest to x: incerment n_i and update $\mu_i = \mu_i + \frac{1}{n_i}(x - \mu_i)$
5. In case there is no more data terminate, otherwise go back to step 3.

This method can be further developed into an algorithm that functions in AE applications.

4.3.3. Adaptive Sequential K-means

An extended version of the sequential k-means is proposed by E. Pomponi and A. Vinograov [38], which they refer to as the adaptive sequential k-means. For this algorithm to work, it is essential to have knowledge of the noise level and its statistical properties. It is recommended that the noise is recorded for a significant time to get a complete representation of the noise. The assumption is made the noise has a Gaussian distribution, which is used to make a lower boundary estimation of the distance between the clusters. Each cluster k_i contains the following parameters:

- Mean m_i
- The amount of elements n_i
- Evolutionary intra-cluster distance (ICD): $\delta_{k_i} = [\delta_1, \delta_2, ..., \delta_{n_i}]$ where $\delta_j = D(x_j, m_j)$
- Point-wise centroid drift vector: $\gamma_{k_i} = [\gamma_1, \gamma_2, ..., \gamma_{n_i}]$ where $\gamma_1 = 0$ and $\gamma_{i,j\neq 1} = D(m_j, m_{j-1})$

where D(-,-) is a distance function like the Euclidean distance. The algorithm distinguishes itself from the conventional sequential k-means by implementing an evolutionary intra-cluster distance parameter δ_i. This parameter is used to assess the homogeneity and separability of a cluster and holds information about the distances between points within the cluster. The point-wise centroid drift vector γ_{k_i} functions as weighting coefficients for the ICD parameters, δ_{k_i}. Two extra parameters are introduced: C for coarsening and R for refinement. The parameter C decides when the merge of two clusters happens and the parameter R when there is a creation of an additional cluster. Both parameters depend on the distance between centroids of the clusters [39]. In order to find the right choice for C and R, the concept of natural clusters is introduced [40]. The idea is to discover a new cluster as it appears without having knowledge about the number of clusters that will exist. The natural cluster is defined as a selection of similar features while having no significant structure in a specific measure. The parameters C and R are estimated directly from the compactness and distribution in the feature space.

The process starts by evaluating the noise, which is done by means of the distance function D(-,-) in the feature domain. Firstly, the distance between each datapoint $D(y_i, y_j)$ is calculated. The mean, $\bar{D}$ and standard deviation σ_{yy} are then taken to determine the noise term $\tilde{R}$:

$$R_{noise} = \bar{D} + \beta\sigma_{yy} \tag{4.11}$$

where β is a value such that $0 \leq \beta \leq 3$. In case the noise is not available the value of $R_{noise} = 0$. If the noise members are assumed to be normally distributed, the value of β becomes 3. The value of R is calculated as:

$$R = max(\tilde{R}, R_{noise}) \tag{4.12}$$

where the value of $\tilde{R}$ is:

$$\tilde{R} = \frac{1}{2}\underset{ij}{max}D(m_i, m_j) \tag{4.13}$$

The $\tilde{R}$ is used to increase the number of clusters. On the contrary, the parameter C balances the process by controlling the merging process of the clusters. As new events occur, it is possible that they cause overlap between two clusters. The value of C adapts with each new event and relies on the structure of the clusters. If two clusters have a large overlap, it is likely that they belong to the same class, and they should merge. To evaluate the overlap, the intra-cluster distance w_{k_i} and the standard deviation $\sigma^*_{k_i}$ are introduced:

$$w_{k_i} = \delta^*_{k_i} max\big(0, (1 - \frac{\gamma_{k_i}}{max\delta_{k_i} - min\delta_{k_i}})\big) \tag{4.14}$$

$$w^*_{k_i} = mean(w_{k_i}) \tag{4.15}$$

$$\sigma^*_{k_i} = \sqrt{\frac{mad(w_{k_i})}{0.6745}} \tag{4.16}$$

Under the assumption that the data points have a normal distribution around the centroid, the overlap between centroids can be evaluated with the cluster diameters and weighted intra-cluster distance $w^*_{k_i}$ plus $\alpha\sigma^*_{k_i}$, where α is a constant value between 2 and 3 according to what kind of estimator is used in eq. (4.15) and eq. (4.16). The coarsening parameter C can be written as follows:

$$\begin{gathered} C_i = max(w^*_{k_i} + \alpha\sigma^*_{k_i}, \tilde{R}) \\ C_j = max(w^*_{k_j} + \alpha\sigma^*_{k_j}, \tilde{R}) \\ C_{ij} = C_{ji} = max(C_i, C_j) \end{gathered} \tag{4.17}$$

Two clusters k_i and k_j are merged if the distance between them is smaller than C_{ij}:

$$D(m_i, m_j) \leq C_{ij} \tag{4.18}$$

When merging clusters k_i and k_j, the means m_i and m_j merge and the number of events add up as follows:

$$m_{merged} = \frac{n_i}{n_i + n_j} m_i + \frac{n_j}{n_i + n_j} m_j, \qquad n_{merged} = n_i + n_j \tag{4.19}$$

Additionally, the ICD values and point-wise centroid drift vectors are concatenated as shown:

$$\begin{gathered} \delta_{merged} = append(\delta_{k_i}, \delta_{k_j}) \\ \gamma_{merged} = append(\gamma_{k_i}, \gamma_{k_j}) \end{gathered} \tag{4.20}$$

The steps for the adaptive sequential k-means algorithm now are:

1. Fix a lower bound R_{noise} for R.
2. For the first event, create a new cluster k_1 with $m_1 = x_1$ and assign 0 to δ_1 and γ_1.
3. When a new event, x, occurs calculate R as in eq. (4.12) if the distance, $\tilde{d}$, between the closest mean, m_i, and the new event x is smaller than R:
   - assign x to that cluster k_i which belongs to m_i.
   - add $\tilde{d}$ to the ICD, δ_{k_i}.
   - update the centroid of the cluster $m_i = m_i + \frac{1}{n_i}(x - m_i)$.
   - calculate $D(n_{i_{updated}})$ and append it to γ_{k_i}

   else:

   - create a new cluster, k_j
   - assign x to the formed cluster
   - assign 0 to δ_j and γ_j
   - check if the clusters have to be merged and if so, apply eq. (4.20)
4. Repeat the process from step 3 until the test is finished

4.4. Cluster evaluation

In order to assess how well a clustering algorithm works, a performance index is used. This can be either an internal or external evaluation. External evaluation requires knowledge of the true classes of data points, making it unsuitable for AE use cases. Internal measures evaluate the intra-cluster similarity and low similarity between clusters [41]. Common internal evaluation indices are Davies-Bouldin index, Silhouette coefficient and Dunn index. These evaluation metrics can also serve another purpose, which is classification of the clusters themselves. The metrics assess the dissimilarity or similarity of clusters, depending on how one looks at it. A bad performance index between two clusters means that the clusters are very similar. Knowing this, a set of known data can be evaluated with each cluster individually where the cluster with the worst performance index is most likely related to the known data.

Davies-Bouldin index

The Davies-Bouldin index is an internal metric to evaluate clustering results. The distance between clusters as well as the average distance datapoints have to the centroid of the cluster are used to evaluate the clusters. The index is written as follows:

$$DB = \frac{1}{N}\sum_{i=1}^{N} \underset{j\neq i}{max} \frac{S_i + S_j}{M_{i,j}} \tag{4.21}$$

with S_i being a measure of scatter within the cluster:

$$S_i = \left(\frac{1}{T_i}\sum_{j=1}^{T_i} |X_j - A_i|^p\right)^{\frac{1}{p}} \tag{4.22}$$

where A_i is the centroid of the cluster i with size T_i and p is 2 for a Euclidean distance function and $M_{i,j}$ is the separation between cluster k_i and k_j given by:

$$M_{i,j} = \left(\sum_{k=1}^{N} |a_{ki} - a_{kj}|^p\right)^{\frac{1}{p}} \tag{4.23}$$

where $a_k i$ is the kth element of the vector a_i, the centroid of the cluster i. A lower DB value means a better clustering performance [42].

Silhouette coefficient

The silhouette coefficient is determined by a set of silhouette values. This value represents how similar a point is when comparing it to its cluster in relation to the other clusters. For each data point i in a cluster a sillhouette value is defined as:

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{4.24}$$

where $a(i)$ is the mean distance between i and the remaining data points of that cluster:

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i,j) \tag{4.25}$$

and $b(i)$ is the smallest mean distance of i to all of the points in all the other clusters:

$$b(i) = \underset{k\neq i}{min} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j) \tag{4.26}$$

The silhouette coefficient is then given by:

$$SC = \underset{k}{max}\ \tilde{s}(k) \tag{4.27}$$

which can range from -1 to 1 [43]. An average silhouette value larger than 0.5 means that the data is partitioned reasonably, where 1 is the maximum attainable value, indicating a great model. An average silhouette value smaller than 0.2 indicates that there is no cluster structure at all [44].

Dunn index

The Dunn index gives an indication on the density and separation of the clusters. In this evaluation the minimum distance between two clusters and the maximum cluster diameter, $D(K_l)$, is used to score the clustering result:

$$DU = \min_{1 \leq i \leq k} \Big(\min_{1 \leq j \leq k, i \neq j} \big(\frac{D(m_i, m_j)}{\max\limits_{1 \leq l \leq k} D(K_l)} \big) \Big) \tag{4.28}$$

A high Dunn index value is related to a well performing clustering algorithm [45].

5
Experimental Campaign

This chapter covers the methods that are used to obtain the data that is used for further processing and analysis. First, the specimens have to be dimensionalized and produced, which is described in section 5.1. The setup of the test is and the type of tests that will be done are explained in section 5.2 and the AE setup is discussed in section 5.3. Finally, the raw test results will be shown in section 5.4 which are discussed briefly in section 5.5.

5.1. Production

For any mechanical test and machine, there exist guidelines that explain recommended specimen dimensions. The way that these specimens are manufactured, can also affect the way the test progresses and therefore also have an impact on the results.

5.1.1. Specimens

The specimens are to be tested under quasi-static tension loading and tension-tension fatigue loading. The testing standard for quasi-static tensile and tension-tension fatigue testing are D3039 and D3479, respectively. The recommended geometry for the test specimens are as specified in table 5.1.

Table 5.1: Recommended geometry for D3039 testing

Length [mm]	Width [mm]	Thickness [mm]
250	25	2

5.1.2. Manufacturing

The composite laminate specimens were manufactured with HexPly 6376 prepreg material [46]. First, two laminates were created. One with layup $[90]_{16}$ and the other with layup $[0_2/90_4]_s$. Individual sheets were cut to size using a Gerber cutting table. Then the layup was done in a clean room. During the layup every three layers, the stack of plies was debulked. After layup and debulking was completed, the laminates were vacuum bagged as shown in fig. 5.1. The assembly was then put into the autoclave and went through the autoclave process as described in the datasheet provided by the manufacturer of the prepreg.

Specimens were then created by roughly cutting the laminate into the recommended size. The specimens that were used for testing and their corresponding label and dimensions are shown in table 5.2. Apart from the thickness of the crossply specimens, the dimensions are roughly the same as what is recommended. Previous experiments had already been done with this type of crossply specimen which did not lead to complications.

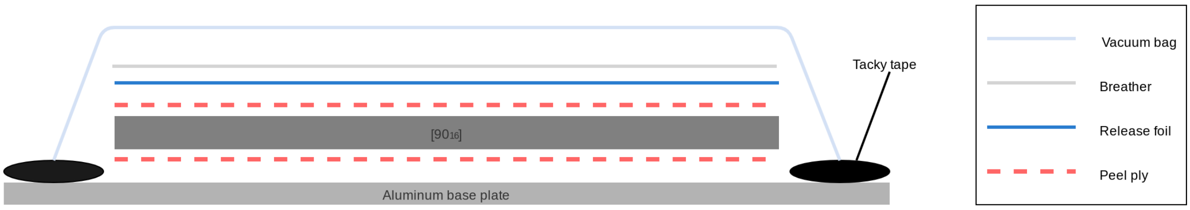


Figure 5.1: Vacuum bag layup

Table 5.2: The specimens used for testing and their dimensions

Specimen label	Type of test	Length [mm]	Width [mm]	Thickness [mm]
UD1	Static	250.1	24.7	2.0
UD2	Static	250.1	25.2	2.0
UD3	Static	250.1	25.1	2.0
UD4	Static	250.1	25.2	2.0
UD5	Static	250.1	25.2	2.0
CP3	Static	250.1	25.1	1.5
CP4	Static	250.1	25.1	1.5
CP5	Static	250.1	25.1	1.5
CP6	Static	250.1	25.1	1.5
CP7	Static	250.1	25.1	1.5
CP8	Fatigue	250.1	25.1	1.5
CP9	Fatigue	250.1	25.0	1.5
CP10	Fatigue	250.1	25.1	1.5
CP13	Fatigue	250.1	24.6	1.5
CP14	Fatigue	250.1	25.0	1.5

5.2. Test set-up

Several specimens are tested in a quasi static tensile test, and a number of other specimens in a tension-tension fatigue test. For the static tests the specimens are loaded until failure. However, the area of interest is in the initialisation and evolution of damage. The final stage of loading might be disregarded when processing the data as this could contain a lot of superimposed signals.

5.2.1. Quasi static tensile test

The quasi static tensile tests were carried out using a $60kN$ tensile testing machine. An overview of the test set-up is shown in fig. 5.2. The specimens are loaded with the standard strain rate of 0.01 per minute as specified in the test standard. Two AE sensors VS900-M from Vallen Systeme are placed on the specimen which are connected by Ultrasonic Couplant from Magnaflux. The signals of the AE sensors are amplified and fed to a data acquisition device which sends the data to the computer. The signals were amplified with $34dB$ and only acquired when the amplitude exceeded $40dB$. A camera in combination with a light source, takes a picture every second. The images are also transferred to the computer.

5.2.2. Tension-tension fatigue test

The tension-tension fatigue test was carried out using the same $60kN$ tensile machine. For this test a similar setup as for the quasi static test is used, but some additional parameters have to be considered. Also, the camera settings change with this test. Here, every 500th cycle a picture is taken and sent to the pc. The specimens were loaded with a frequency of $f = 5Hz$, stress ratio $R = 0.1$ and a maximum load of about 70% of the ultimate load. It is at this load where matrix cracking is initiated [47]. For the cross-ply specimens this is a maximum load of $F_{max,crossply} = 27kN$. The crossply specimens are loaded for a total of 200,000 cycles. This is deemed sufficient for the goal of this test, which is tracking the evolution of damage in a realistic load case. The testing parameters are summarized in table 5.3.

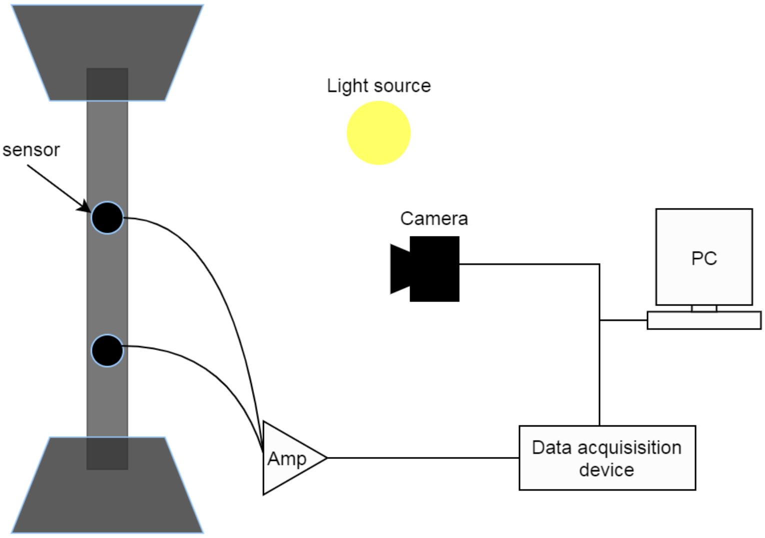


Figure 5.2: Test setup overview

Table 5.3: Fatigue test parameters

Layup	Frequency [Hz]	Minimum Load [kN]	Maximum Load [kN]	Number of cycles
$[0_2/90_4]_s$	5	1.90	19.0	200,000

5.3. Acoustic emission setup

The data acquisition device that is used during the test is called the Vallen Systeme AMSY-6. From the same manufacturer VS900-M AE sensors are used. This is a passive piezoelectric AE sensor which has frequency range from 100 to 900 kHz. Two sensors are placed on the specimen with Ultrasonic Couplant from Magnaflux and fixed with clamps, as visible in fig. 5.3. For each test the gap distance between the sensors d is measured to be $10cm$ to allow for source localization. Also, the sensors are checked to see if they still work properly before each test with pencil lead breaks. With the pencil lead break, the wave velocity through the specimen can also be determined by considereing the difference in time of arrival at the sensors and the distance d between them. The signal of the sensors are preamplified using the Vallen Systeme preamplifier AEP4H with a gain of $34dB$ and then picked up by the data acquisition device if the amplitude of the signal is equal to or larger than $40dB$. The acquisition device is connected to a PC with Vallen Systeme software that records and visualizes incoming data. For every AE signal basic parameters like the amplitude, rise time, energy, duration and the number of counts are recorded. Next to that, the waveforms of each signal are saved to the PC for all of the static tests. For the fatigue tests, the waveforms are only stored for 3 tests as the data file grows drastically in size, in the order of gigabytes for a single test.

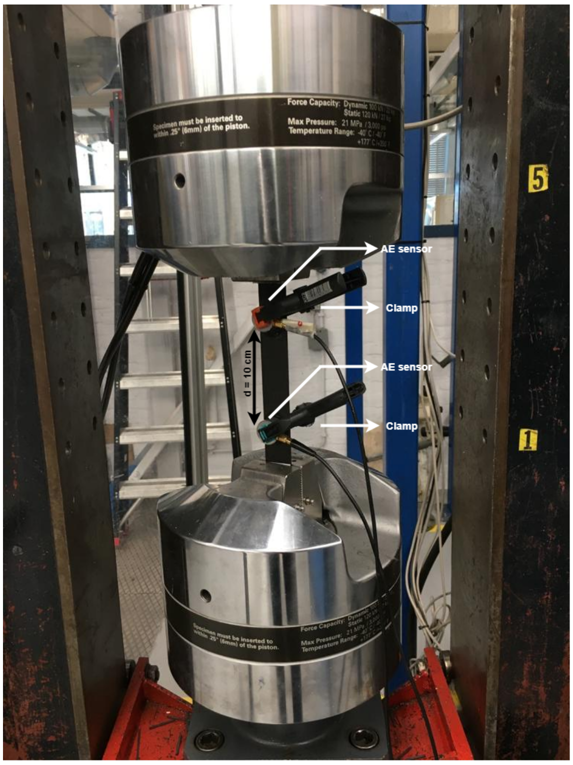


Figure 5.3: Test setup

5.4. Test results

The results from the static and fatigue test will be shown in this section. After that a decision can be made on which specimens are fit for further analysis and which are not.

5.4.1. Static test results

The results of the UD static tests for UD1-UD4 are shown in fig. 5.4. It is apparent that there are very limited AE signals for these tests. During the tests it was observed that all specimens fractured in one location in a straight line in the direction of the 90°layers, as is visible in fig. 5.7(a). So the damage development could be localized until failure which yields a limited amount of AE signals. With some specimens it is also the case that one sensor measured significantly more AE signals. This is possible since the damage can develop close to one sensor, causing the signal strength to be below the threshold at the other sensor. The failure load and strain for some specimens is shown in table 5.4. In particular the UD specimens failed at a very low strain, due to one complete transverse matrix crack being fatal. For the CP specimens, the failure load has a positive correlation with the amount of AE signals measured.

Table 5.4: Maximum load and strain for eight of the tested specimens

Specimen	Failure Load [kN]	Failure Strain
UD1	3.5	0.006
UD2	3.6	0.006
UD3	3.2	0.005
UD4	3.0	0.005
CP3	24.8	0.013
CP4	29.2	0.015
CP5	28.6	0.015
CP6	25.4	0.013

In fig. 5.5 the results of CP3-CP6 of the static tests are shown. Relative to the plots of the UD specimen, many AE signals are detected. A lot more damage is inflicted in this type of specimen before it fails i.e. matrix cracking, delamination and fiber breakage are will all occur. The specimen after failure is shown in fig. 5.7(b), after which the damage mechanism are identified in fig. 5.7(c).

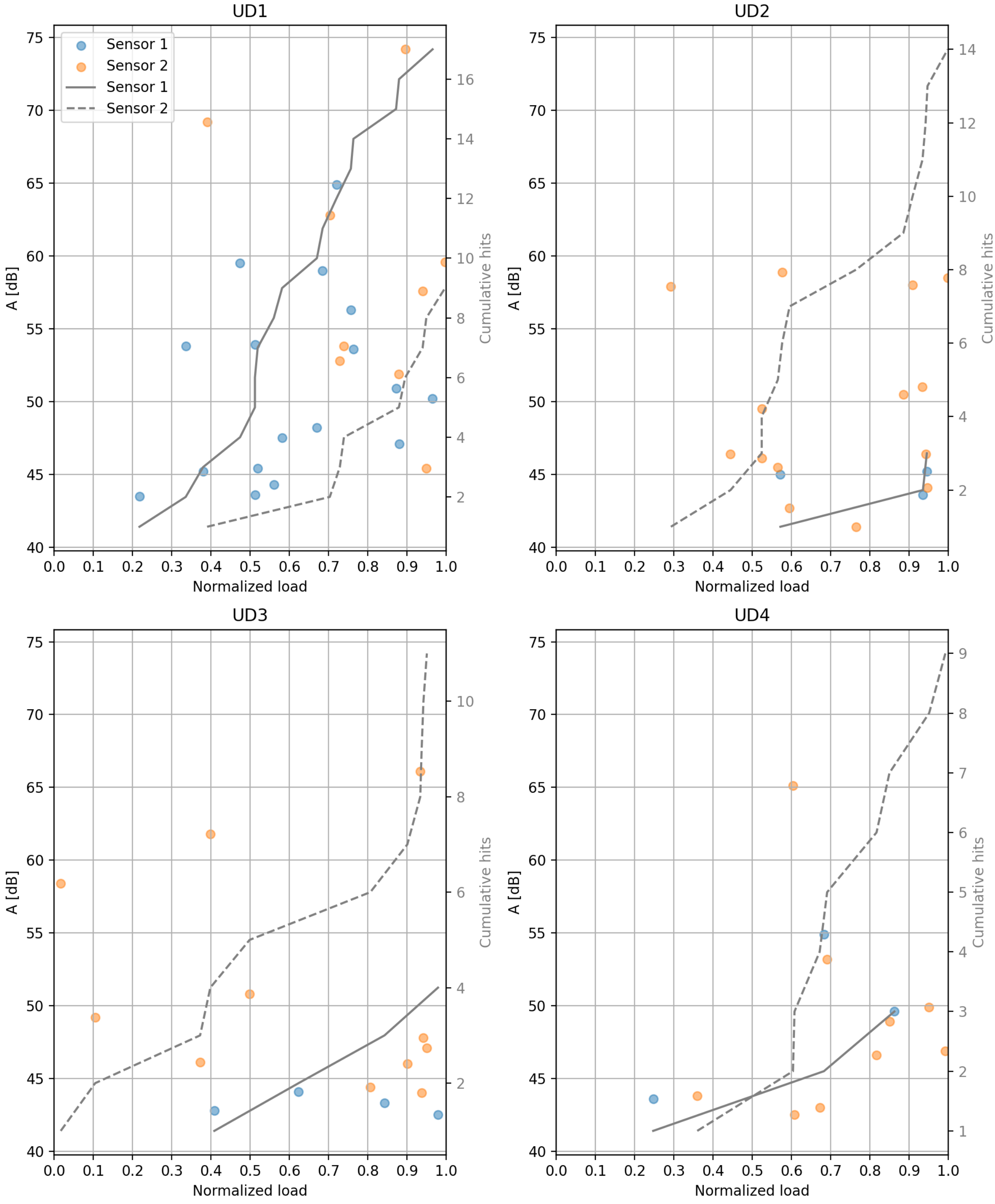


Figure 5.4: UD1-UD4 static test results

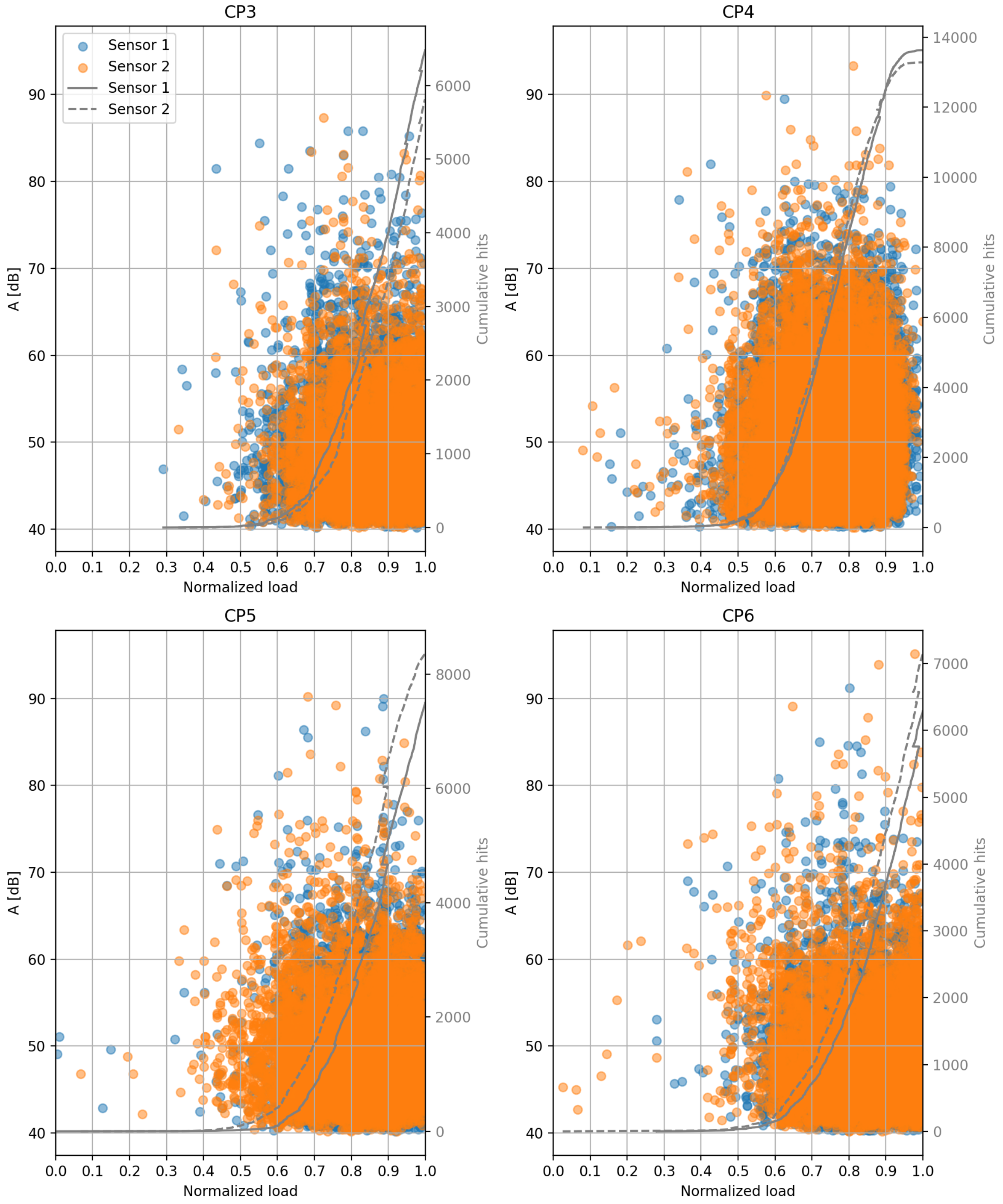


Figure 5.5: CP static test results

5.4.2. Fatigue test results

The AE signals recorded during the two fatigue tests CP8 and CP14 are plotted in fig. 5.6. An immediate observation can be made here which also points out an issue for using AE in fatigue tests. There is a tremendous amount of signals so it is unlikely that all of them contain useful information. Another interesting note is that the number of hits scale almost linearly with time. The resulting amount of signals for a single fatigue test of 200,000 cycles is over ten milion. Figure 5.7(d) shows a CP specimen which is not tested, with one that was tested under the conditions that were mentioned. Matrix cracks were expected after testing and were visible as indicated with the arrows. There were also some fibers that split at the edge of the specimen.

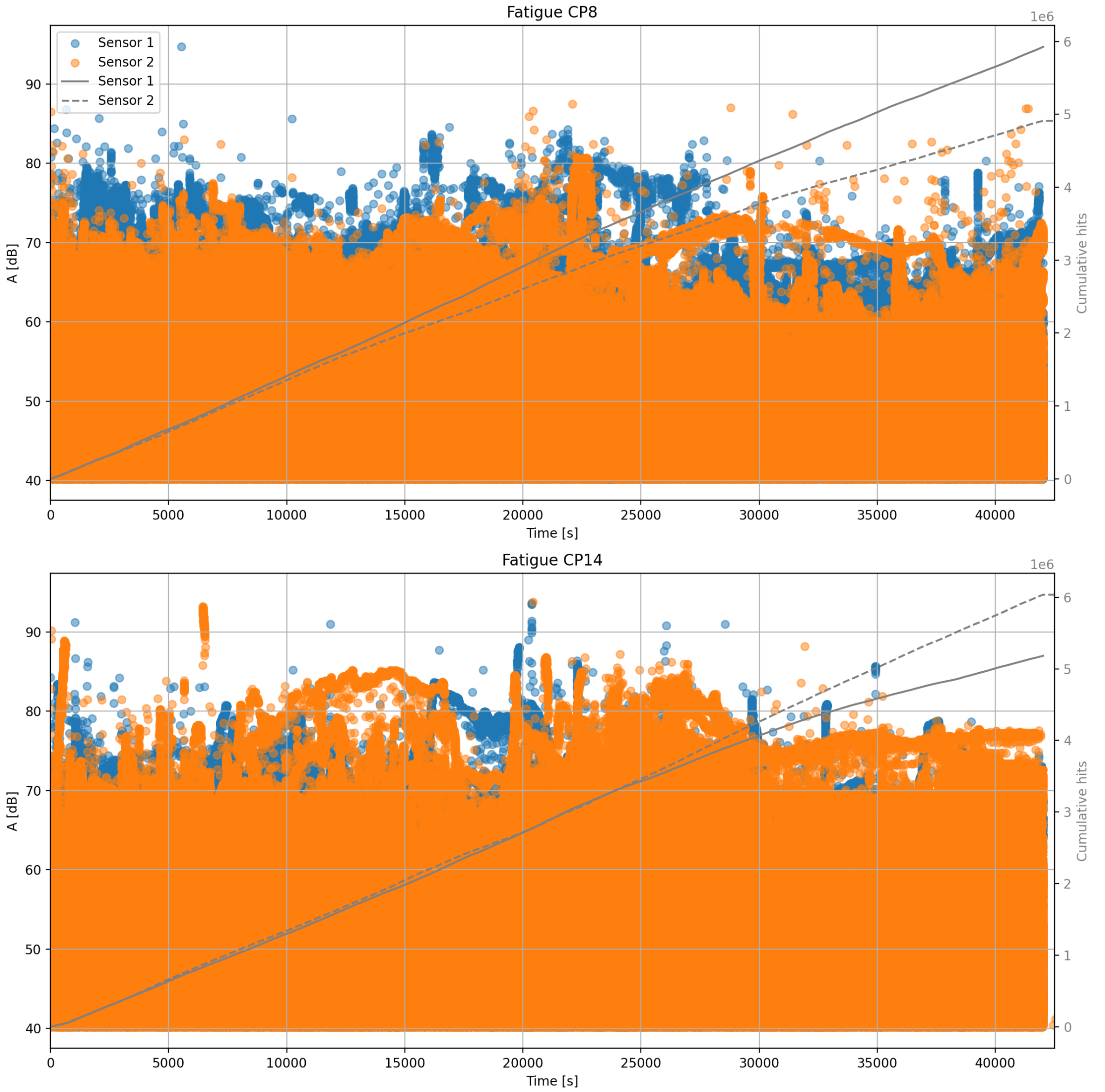
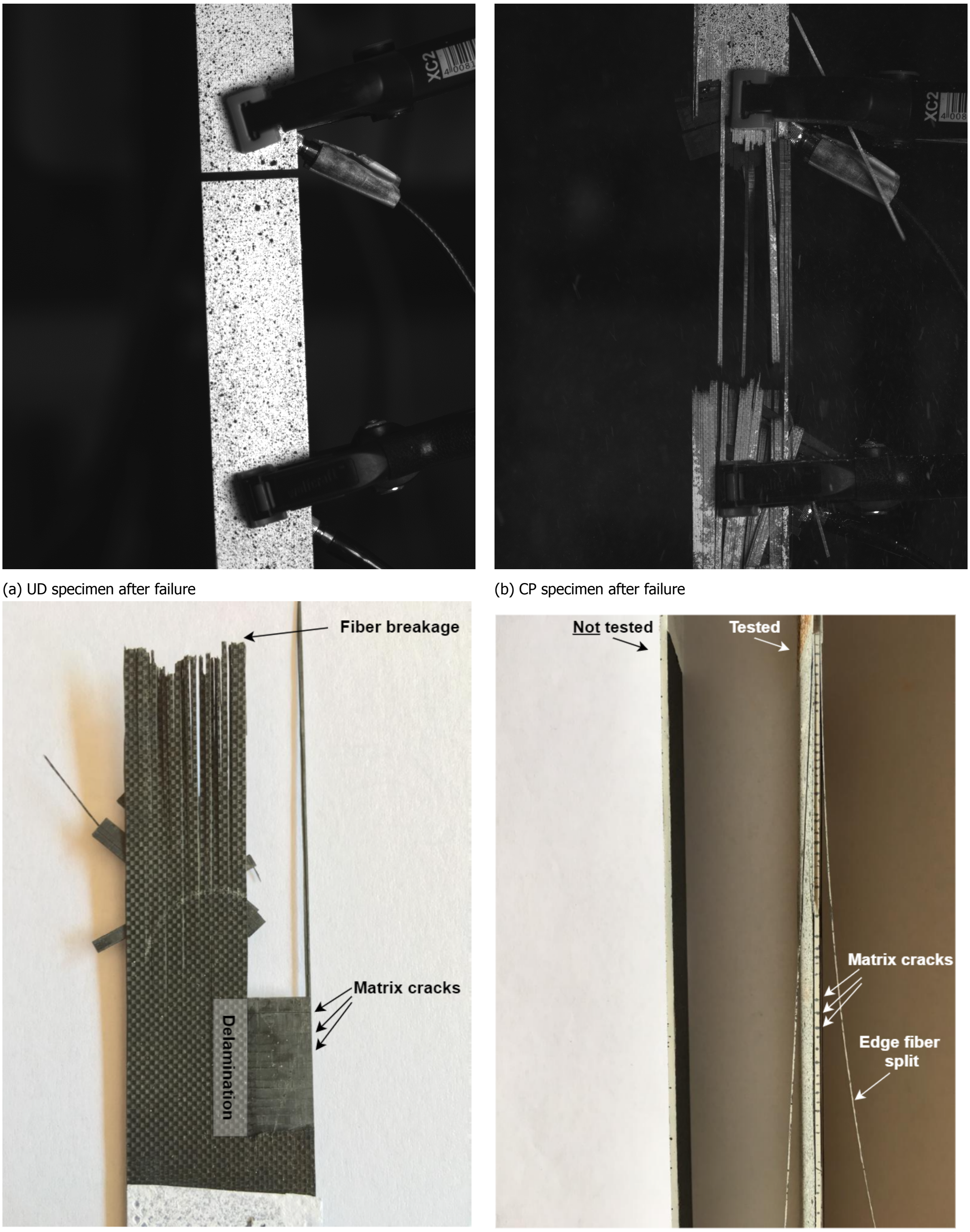


Figure 5.6: CP fatigue test results



(a) UD specimen after failure

(b) CP specimen after failure

(c) CP damage identification from a quasi static test

(d) CP fatigue specimen before and after testing (Side view)

Figure 5.7: Specimens after failure

5.5. Discussion

From the results it can be noted that the specimen type and type of test strongly influence the resulting AE signals. It was shown that their was localized matrix cracking failure in the UD specimen, which yielded only several signals per test.

In the CP specimens there were over 10,000 AE signals per test due to the onset of delamination and fiber breakage. These damage mechanisms were identified by further inspecting the specimen. It begs the question of how representative the signals from the UD tests can be for identifying signals from the CP tests, since there relatively few. This has to be taken into consideration when analyzing the results.

In two of the CP fatigue tests, there were close to 2,000,000 AE signals measured, which is significantly more than in the static tests. There are two main causes for this effect: the damage evolution is more gradual in the fatigue test and the grips of the test machine also cause AE signals due to friction. Also, after testing the specimen showed clear indications of matrix cracking, which was expected. The main issue regarding the fatigue tests was that with waveform analysis of each signal, the datafiles tend to get very large.

6 Results

This chapter will cover the results from the work that is done on the data of the experiments that were shown in chapter 5. Firstly, the different features of the data are analyzed to establish which are useful for characterizing the signals in section 6.1. With the selected features, the data is clustered with the k-means method and the resulting clusters are compared with matrix cracking signals, which is described in section 6.2. After that, in section 6.3, the same is done with two real-time variations of the k-means method. The performance of all the clustering methods are compared in section 6.4. To see how the matrix cracking signals change as damage progresses, the evolution of these signals is shown in section 6.5.

6.1. Acoustic Emission features

For each signal recorded during the tests, some basic characteristic parameters are stored which are: amplitude, rise time, energy, duration and counts. From these basic parameters, additional aggregations of these features can be generated. The decay time for example can be calculated by subtracting the rise time from the total duration of the signal. These features could lead to better characterization of the signals. Next to this, the waveform of each signal is measured. With methods like the Fast Fourier Transform, various frequency components of the signal can be calculated. All of these parameters with the corresponding symbol, unit and theoretical limits are shown in table 6.1.

Table 6.1: Basic AE signal features

Descriptors	**Symbol**	**Unit**	**Lower limit**	**Upper limit**
Amplitude	A	dB	40	100
Counts	C	-	2	1000
Duration	D	μs	0	1000
Energy	E	eu	0	1000000
Rise time	R	μs	0.1	200
Rise time/duration	R/D	-	0	0.2
Duration/Amplitude	D/A	μs / dB	0	25
Decay time	D-R	μs	0	1000
Rise angle	A/R	dB / μs	0.1	1000
Decay angle	A/(D-R)	dB / μs	0.02	100
Rise time/Decay time	R/(D-R)	-	0	400
Relative energy	E/A	eu/dB	0	25000
Peak frequency	PF	kHz	0	1000
Rise frequency	RF	kHz	0	1000
Decay frequency	DF	kHz	0	1000
Average frequency	AF	kHz	0	1000
Median frequency	MF	kHz	0	1000

This waveform can also be processed further with the wavelet transform to create more features that can characterize the signal. There are two ways to decompose the signal. One is in a tree like structure and the other is with a pyramid structure, as shown in section 3.3.3. This is also referred to as wavelet packet decomposition. Both options are explored to figure out which components are relevant. For each level of the wavelet transform, the energy is calculated. These values are then normalized to find the percentage of energy content per component. These wavelet transform parameters with the corresponding symbol, unit and theoretical limits are shown in table 6.2.

Table 6.2: Wavelet transform features

Descriptors	**Symbol**	**Unit**	**Lower limit**	**Upper limit**
Detail level 1 energy	D1	%	0	1.0
Detail level 2 energy	D2	%	0	1.0
Detail level 3 energy	D3	%	0	1.0
Detail level 4 energy	D4	%	0	1.0
Detail level 5 energy	D5	%	0	1.0
Wavelet packet aaa energy	WP1	%	0	1.0
Wavelet packet aad energy	WP2	%	0	1.0
Wavelet packet ada energy	WP3	%	0	1.0
Wavelet packet add energy	WP4	%	0	1.0
Wavelet packet daa energy	WP5	%	0	1.0
Wavelet packet dad energy	WP6	%	0	1.0
Wavelet packet dda energy	WP7	%	0	1.0
Wavelet packet ddd energy	WP8	%	0	1.0

6.1.1. Mother wavelet selection

In order to justify the choice of the mother wavelet, a trial with two evaluation metrics is performed. A matrix cracking signal from a static test is taken to do wavelet transformations on. The signal is transformed with varying wavelet types from different families: Coiflets (coif), Daubechies (db), Biorthogonal (bior) and Symlets (sym). This is done up to detail level 3 since at this level is where most of the energy in matrix cracking signals is. The coefficients of this detail level are used to calculate the energy and the entropy, using equations eq. (3.11) and eq. (3.12), relatively. The goal would be to maximize the energy to entropy ratio and minimize the time that is needed for computation. Since a single transform takes less than a millisecond and such a time measurement can be inaccurate, the process is repeated a thousand times. The resulting time and energy to entropy ratio are shown in fig. 6.9. From analyzing the figure it can be noted that the Biorthogonal wavelets do not reach the same energy to entropy ratio as the other families and thus performs worse. The Coiflets wavelets reaches a high performance but at the cost of a relatively high computation time. The Symlet wavelets all require relatively low computation time, while having a similar performance to the Daubechies family. Keeping within the time range of the Symlet wavelets, there are two Daubechies wavelets that stand out. These local optimums are 'db10' and 'db17'. The priority for time-wise performance leads to the selection of the 'db10', while sacrificing what could be a small increase in performance.

6.1.2. Pyramid detail levels

The result of a matrix cracking signal decomposition in a pyramid structure is shown in fig. 6.1. The first detail level shows the components that have the highest frequency. As the detail level becomes higher, the frequency range of the components decrease. From reading the y-axis of the plots, it can be noted that most of the energy is found in detail levels 2 and 3. The sum of these detail levels should, with some discrepancy, equal the original signal.

For each detail level a FFT analysis is done to better understand what frequency range of each resulting signal is. The result of this is shown in fig. 6.2. Theoretically, with each subsequent level, the frequency range drops with a factor of 2 [48]. The sampling frequency is 2MHz, which indicates that the first detail level should contain components up to 1000kHz, which is the case. Any subsequent level should contain frequency components up to half that of the previous level, which is more or less true for all of them. The largest value found in any of the frequency plots is in detail level 3, just below 200 kHz at about 180 kHz. This is in line with the study from P.J. Groot et al. [12].

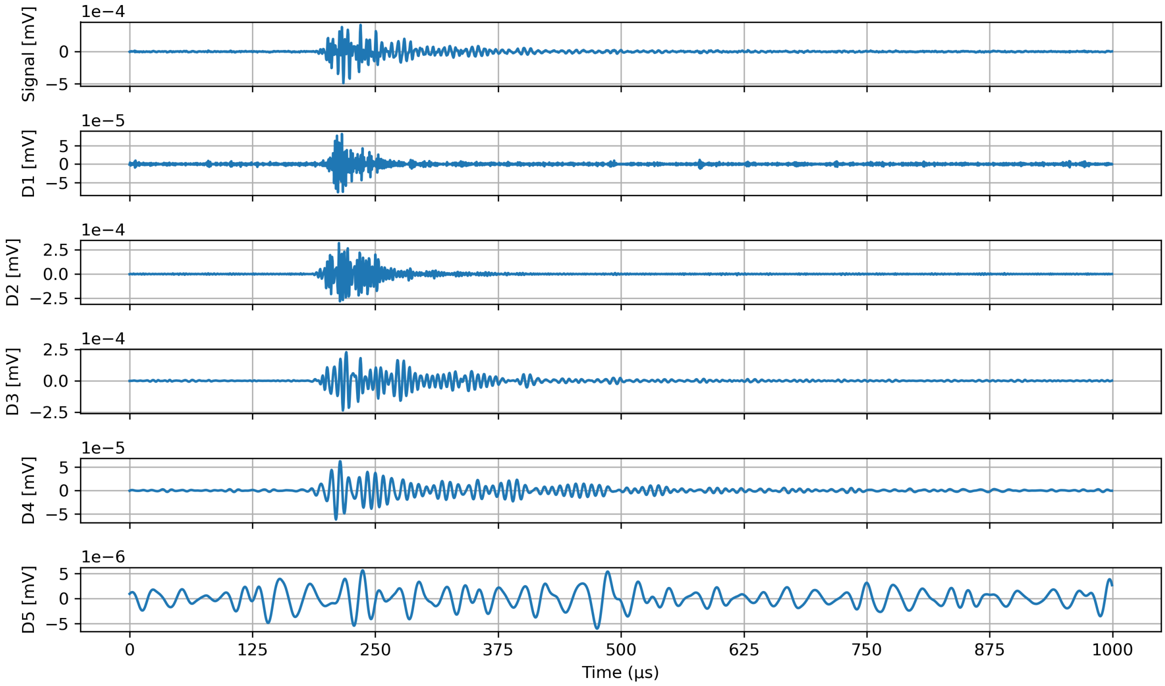


Figure 6.1: Pyramid wavelet decomposition of a matrix cracking signal

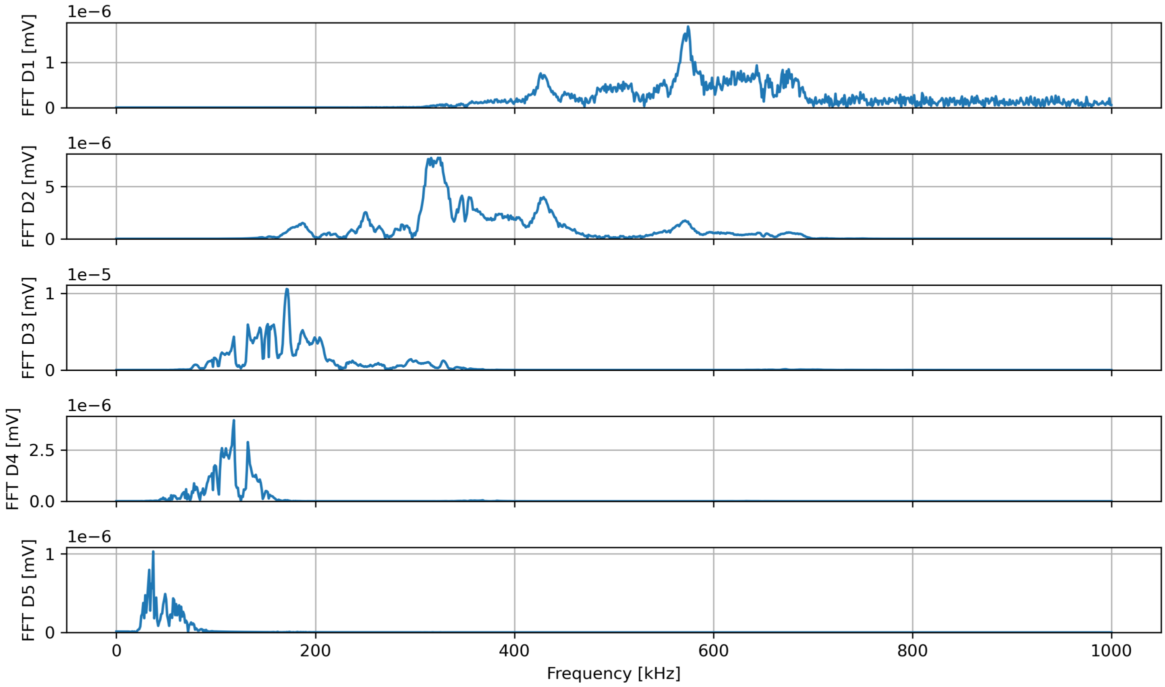


Figure 6.2: FFT of the pyramid detail levels

6.1.3. Wavelet packet decomposition

The same signal is now analyzed with wavelets using a subband tree structure. This is done up to three levels which leads to a total of eight resulting components. Similarly, if these components are added up, this should result in a reconstruction of the original signal. The eight components are shown in fig. 6.3. Also for these components an FFT analysis is performed, which is shown in fig. 6.4. Each

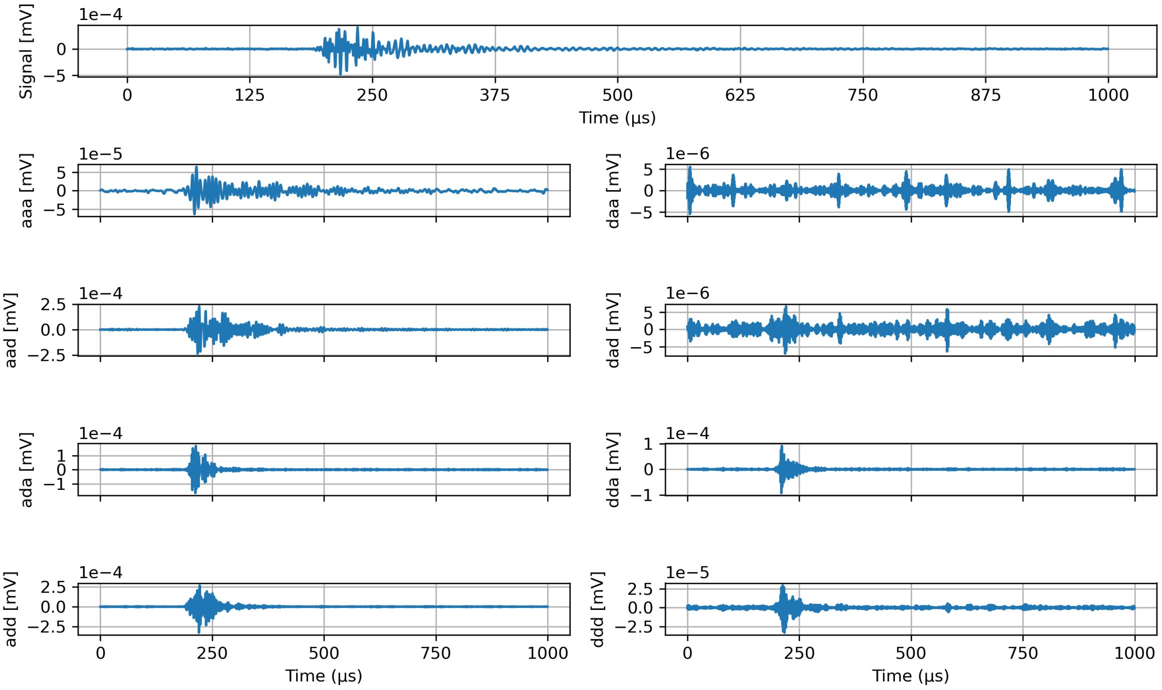


Figure 6.3: Tree wavelet decomposition of matrix cracking signal

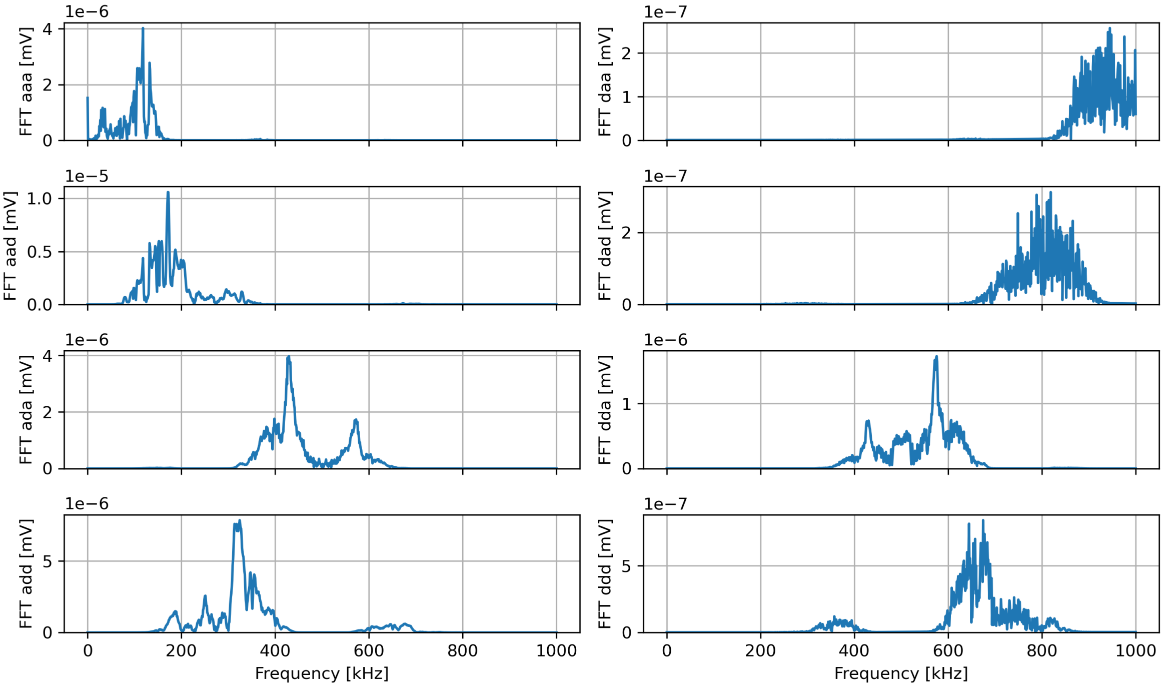


Figure 6.4: FFT of the tree detail levels

wavelet packet theoretically contains an octave of the frequency contents. Whether this approximation holds, depends on how frequency-localized the wavelet is. From the figure it is quite clear that this is not the case since there is overlap between the different packets. However, the peaks of the packets do take place at distinct frequencies. The peak frequency value range of each packet will be evaluated upon with the actual data that is used for clustering. From looking at the y-axis of the components, it can be noted that the largest peak is found in the detail level 'aad'. This is again at a frequency at about 180 kHz.

6.1.4. Feature selection

The feature selection consists of two parts: part one is to find which features have the most variance and the second part is to figure out which features are correlated. Features with a large variance provide a larger spread of the data and are therefore more likely to contain useful information when distinguishing clusters. If there is a pair of correlated features, it does not help to incorporate both of them in the analysis, since they contain the same information. The complete selection process is shown in the shape of a flowchart in fig. 6.5. This process will be done for the basic AE features measured in table 6.1, for the data of CP5. One of the two groups of wavelet transform features will form a basis feature set for the cluster analyses that will be done, since they have been shown to work for this purpose in literature [25]. There are other methods for feature selection that incorporate all features. A whole study could be devoted to this, but the suggested approach is simply efficient while still covering a lot of the features in the analysis. Since the features have different units the variance of the raw data can not be compared. That would lead to features having the largest numbers to have the biggest variance, like the energy of the signal. For this reason the feature data is first scaled to a domain of $[0, 1]$. Three measures of dispersion are then calculated from the scaled data: variance, median absolute deviation and the interquartile range. The variance is defined as follows [49]:

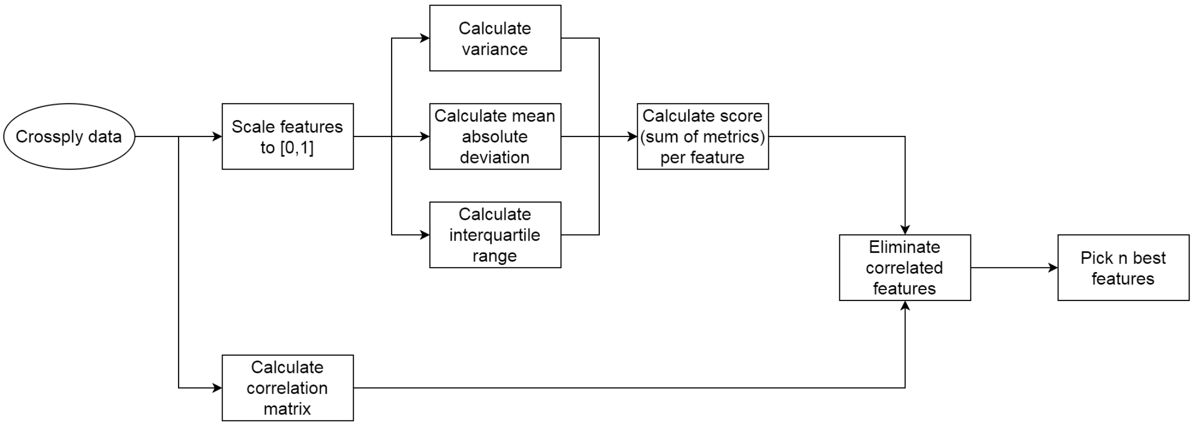

Crossply data
Scale features to [0,1]
Calculate variance
Calculate mean absolute deviation
Calculate interquartile range
Calculate score (sum of metrics) per feature
Eliminate correlated features
Pick n best features
Calculate correlation matrix

Figure 6.5: Feature selection flowchart

$$Var(X) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2 \tag{6.1}$$

The median absolute deviation (MAD) is simply the median value of the difference of each point with the mean:

$$MAD(X) = median(x_i - \overline{x}) \tag{6.2}$$

and the interquartile range (IQR) is the distance between the 25th percentile and the 75th percentile, where a percentile is defined as the value below which a given percentage of data points is found. The values of these metrics are summed up and functioned as a score per feature, where a high score is considered a more relevant feature. The IQR will in most cases have the highest value from the three metrics, so it weighs the most in the final score. This is considered beneficial since the IQR is least susceptible to outliers. The scores of each feature from the basic features for the data of CP5 are shown in fig. 6.6. From the figure it is obvious that there are three features that stand out from the rest: median frequency, the ratio of rise time over duration and the ratio of the rise time over the decay time. In theory, these variables would lead to the largest spread of the data and therefore lead to a relatively high the clustering performance. Whether this also holds in practice will be presented in the next section. The figure also shows that the features that incorporate the energy of the signal have an extremely low score. The reason for this has to do with the theoretical upper limit of the feature. If there is a single high value in the original data, it significantly increases the scaling down factor. This in turn leads to a decrease in all of the evaluation metrics. What is also interesting is that the rise time itself performs poorly, but when it is used in a ratio with duration it functions better. This can be explained by higher rise times having higher durations, causing there to be less outliers. Histograms of the raw feature data can be found in appendix A. Here the percentage of occurrence of data is plotted over the feature range.

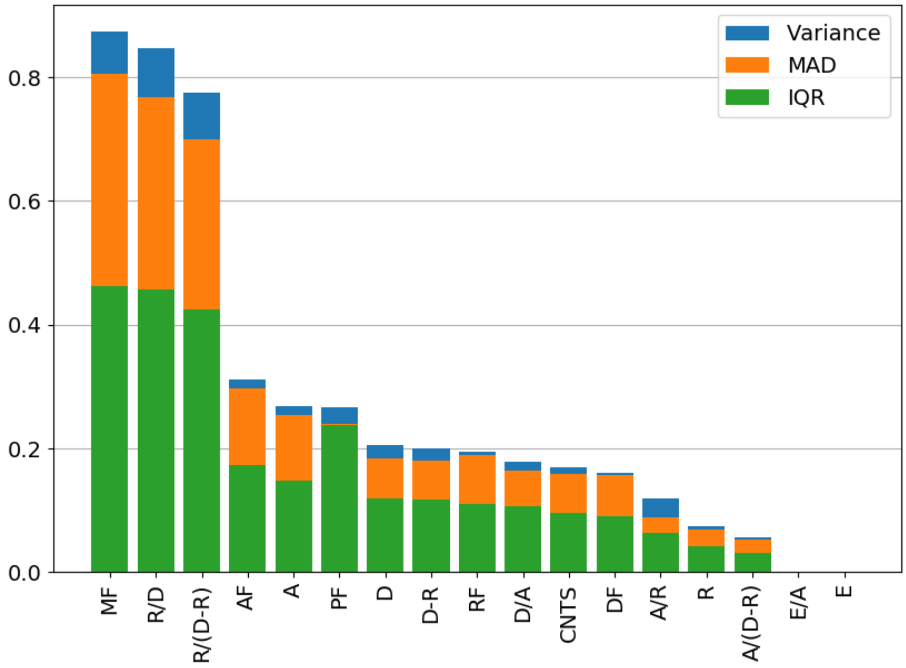


Figure 6.6: Feature variance metrics: Variance, Median absolute deviation (MAD) and Interquartile range (IQR).

The second part of feature selection consists of finding correlated features and eliminating them. The correlation between two features x and y is measured with the Pearson correlation coefficient, r_{xy}, which is given by [50]:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{6.3}$$

The correlation matrix with the basic features and wavelet packet energy percentages is shown in fig. 6.7. The values range from -1 to 1, where 1 means there is an exact positive correlation and -1 means that there is a exact negative correlation between the pair of features. The diagonal values are always equal to 1 since a feature is perfectly correlated to itself. If two features are correlated to a certain extent, the one with the lowest score can be disregarded since it provides similar information but causes a lower spread of the data.

In order to eliminate features a threshold is set to determine which features are correlated to such an extend that they can be disregarded. The value for the threshold is set to 0.8, as this indicates a strong relation between two features [51]. The correlation matrix can then be reduced to what is shown in fig. 6.8. From this it is clear that the amplitude and number of counts of a signal are strongly correlated. This makes sense as the vibrations are damped gradually, so a higher amplitude leads to more crossings of the amplitude threshold. Following the same reasoning, one would also expect a strong correlation between the amplitude and duration of the signal. This is the case, but the correlation is not high enough to reach the threshold. Many of the other correlations are obvious since the pairs contain the same features. Finally, the reason that the decay frequency and the average frequency correlate so much is because the decay time is the largest part of the signal time, so the average frequency of the whole signal will always be similar to the frequency of a large part of the signal. Now for each correlated pair, the feature with the lowest score is disregarded for any future analysis. The following features are eliminated:

- E
- CNTS
- D/A
- D-R
- R/(D-R)
- DF

What is left is an ordered list of features with the first one being the most useful for performing cluster analysis:

1. MF
2. R/D
3. AF
4. A
5. PF
6. D
7. RF
8. A/R
9. R
10. A/(D-R)
11. E/A

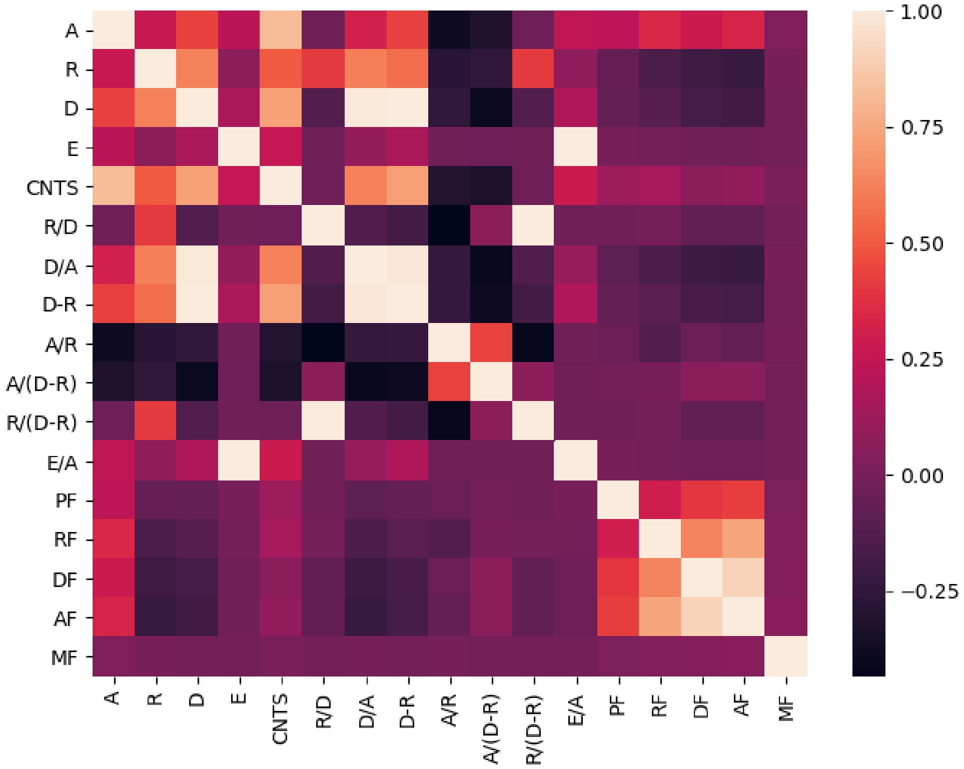


Figure 6.7: Correlation matrix

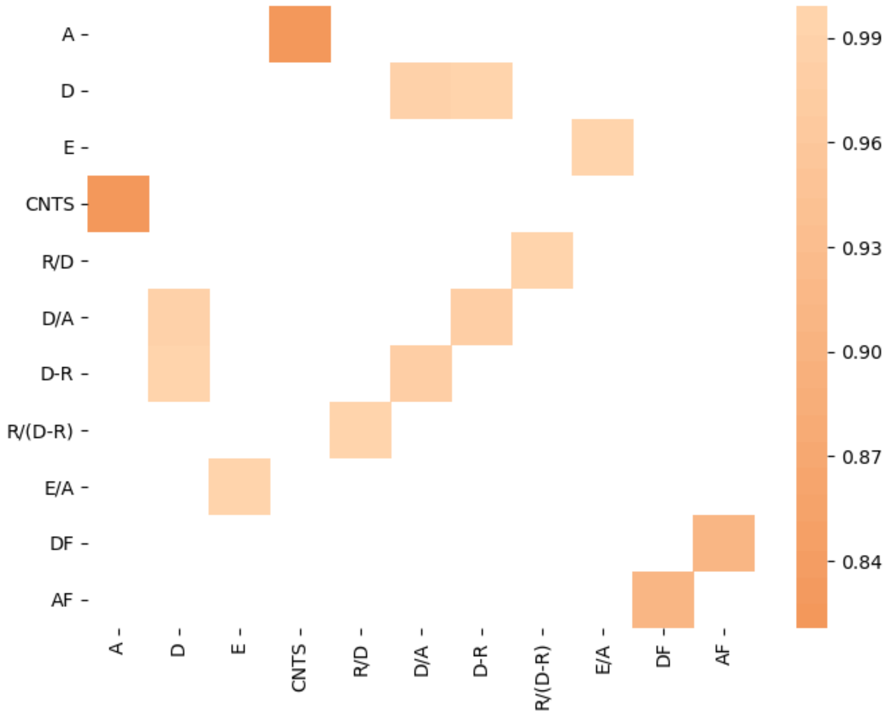


Figure 6.8: Reduced correlation matrix with a threshold of 0.8

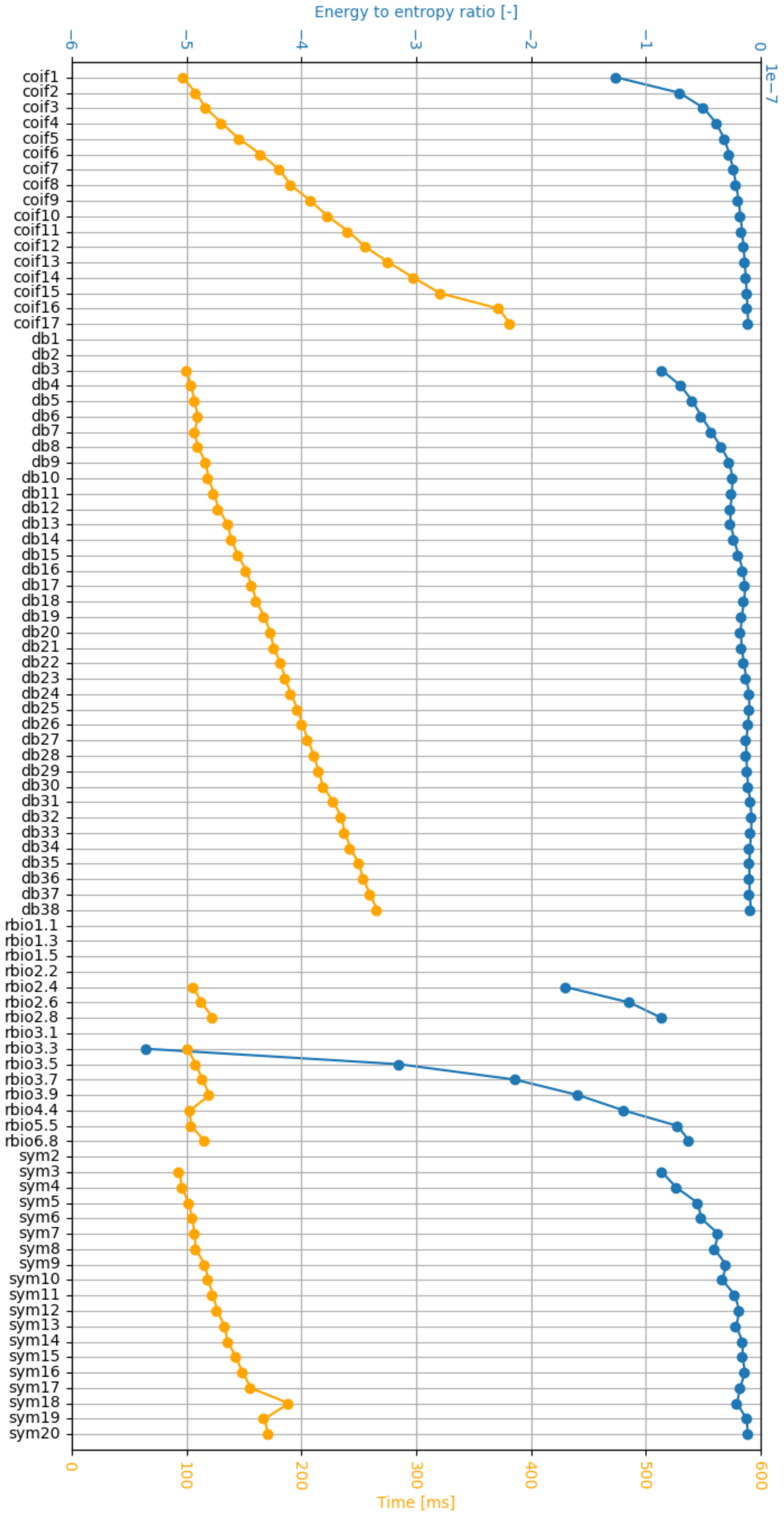


Figure 6.9: Performance of discrete wavelet types

6.2. Cluster analysis

For the clustering analysis is was deemed sufficient to use data from five UD specimens and data from five CP specimens. For the more in depth analysis CP5 is used since it contained a lot of signals without being an outlier. In the analysis the CP data will be clustered and the UD data will act as a fingerprint in order to identify one of the clusters. Before any clustering analysis, the data from the UD and CP tests have to be processed appropriately. In the first step, the basic descriptors are combined to compute additional features, which are shown in the second block of table 6.1. Then the waveform can be used to calculate the frequencies of interest and the wavelet transform components. The energy of each wavelet component is then normalized to give the percentage of energy content per component. Once the features are processed the data is standardized, which means that the data is scaled to have a mean of 0 and a standard deviation of 1. This can be done by applying the following equation to each datapoint:

$$x_{standardized} = \frac{x - \mu}{\sigma} \tag{6.4}$$

where μ is the mean and σ is the standard deviation of the data. The purpose of this step is to make sure each feature contributes equally to further steps in the clustering process. The final step for the data processing is to reduce the dimensionality of the data. The data is reduced to two dimensions, in order to not run into issues with having high dimensional data and to be able to visualize the results. Skipping dimensionality reduction would lead to a lot of small isolated clusters, simply because most distance metrics do not work well in high dimensional space. The mathematical process is described in detail in section 4.2.1.

The K-means method is applied to cluster the processed CP data. To evaluate the performance, several evaluation metrics are used for the clustering performance as well as for correlating the clusters to the UD signals. These metrics for clustering performance are: the Silhouette coefficient, the Davies-Bouldin index and the Dunn index. The silhouette coefficient and the Davies-Bouldin index are also used to indicate how well each cluster relates to the UD signals. All of the evaluation metrics are described mathematically in section 4.4. It is important here to keep in mind that the ranges of the evaluation metrics are different and the comparison will therefore be somewhat empirical. The outcome of the process is three scores from different cluster performance metrics and two scores for each cluster indicating how similar it is to the UD data. The entire clustering process is visualized in a flowchart as shown in fig. 6.10. In order to select the best features for cluster separation, trials are performed with different sets of features. The wavelet packet energies form a basis for the analyses as they perform well as a stand alone group and contain more information than the wavelet detail levels about the frequency range of the signals, especially for the upper range of the spectrum. This is also shown in the wavelet feature histogram, presented in fig. A.4 and fig. A.5. The peak frequency range for each wavelet feature is also mentioned there. It can also be concluded that WP5 does not provide any useful information as it is close to 0 for all data and is there for eliminated. For each trial, the clustering process is repeated while adding one variable at a time. The variables are added in order of the ranking stated in section 6.1.4.

Two variations of the clustering process are performed: one where the data is standardized based on the mean and variance of all data and the other where the data is standardized based on the mean and variance of the UD data only. The latter is where eq. (6.4) simply becomes:

$$x_{standardized} = \frac{x - \mu_{UD}}{\sigma_{UD}} \tag{6.5}$$

The case with standardization based on UD data is incorporated to be able to compare results with the real-time clustering process. When going to the real-time clustering process, the mean and variance are constantly changing causing a lot of fluctuations in the output. This is solved by having a precomputed mean and variance. The adapted standardization will also affect subsequent steps of the clustering process. It could be the case that feature data from the CP test with the applied mean and variance falls outside of the domain of the standardized UD data. If that is the case, the particular feature is weighted heavier in the PCA, as the variance becomes larger. In other words, features from the CP test that differ from the UD test weigh more in the clustering process. This is in fact beneficial for the purpose of the analysis, which is the tracking of matrix cracking.

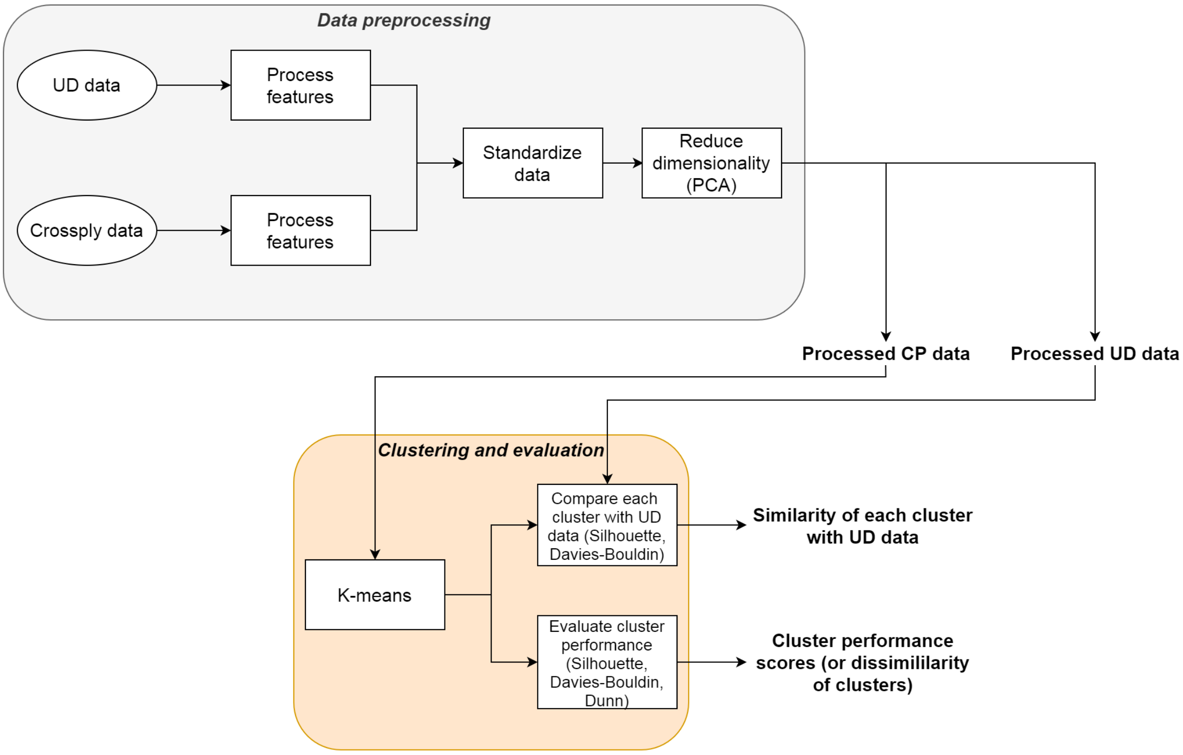

Data preprocessing
UD data
Process features
Crossply data
Process features
Standardize data
Reduce dimensionality (PCA)
Processed CP data
Processed UD data
Clustering and evaluation
K-means
Compare each cluster with UD data (Silhouette, Davies-Bouldin)
Evaluate cluster performance (Silhouette, Davies-Bouldin, Dunn)
Similarity of each cluster with UD data
Cluster performance scores (or dissimililarity of clusters)

Figure 6.10: Clustering process

6.2.1. Standardization with all data

A total number of 12 trials are done in order to find the optimal set of variables for the clustering analysis. The clustering performance is tracked, as well as the time taken for the analysis, since computation time plays an important role in the the real-time clustering process. Also, the process is repeated several times to check for any fluctuations in the computation time. The Silhouette coefficient and the time taken for each feature group trial is shown in fig. 6.11. The other evaluation metrics are taken into account as well but Silhouette is shown here since the coefficient has a limited domain of $[-1, 1]$ and is therefore easiest to interpret. From the figure it can be seen that the cluster performance is best when 4 variables are added to the wavelet packet energy features. The time that is taken for the analysis shows no clear pattern, except for a jump at the addition of 7 variables. Also, the axes show that both metrics show no striking changes in the performance when adding variables. Still, for the more in depth analysis the best performing set of variables is taken: WP1-WP4, WP6-WP8, MF, R/D, AF and A.

Since the k-means method requires the number of clusters as input, the processed CP5 data with the selected feature set is clustered with the k-means method for a number of clusters ranging from 2 to 6. For each cluster count the result is evaluated with the Silhouette coefficient, the Dunn index and the Davies-Bouldin index. The outcome is shown in fig. 6.12. In this figure, the top plot contains the Silhouette coefficient and the Dunn index, where a high value is preferable. For the Davies-Bouldin index in the bottom plot, a low value is preferable. The Dunn index shows that six clusters is best, whereas the other metrics give the best score to a number of two clusters. Giving all metrics the same weight, the number of clusters is kept at two for further analysis. The clustering is shown in the left plot of fig. 6.13. The right plot shows the K-means clustering result as well as the processed UD data. Visually, cluster 2 is very similar to the UD data. There are only two datapoints from the UD data that overlap with cluster 1. Another thing to note is that the number of datapoints in cluster 2 is about twice as much as the number of points in cluster 1, indicating cluster 2 is the dominant damage mechanism present. The same result is plotted with corresponding histograms on the axes in fig. 6.14. This gives more insight in how the data is distributed. Unfortunately, from the histograms there is no clear distinction between the clusters. The orange cluster also does not have a visible mean on the x-axis.

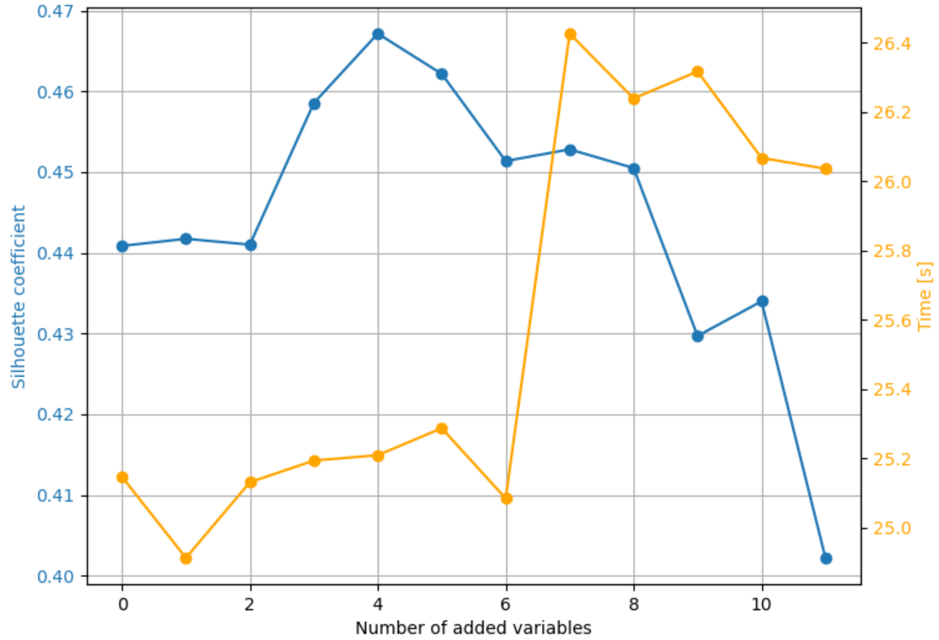


Figure 6.11: Cluster and time-wise performance of different feature groups

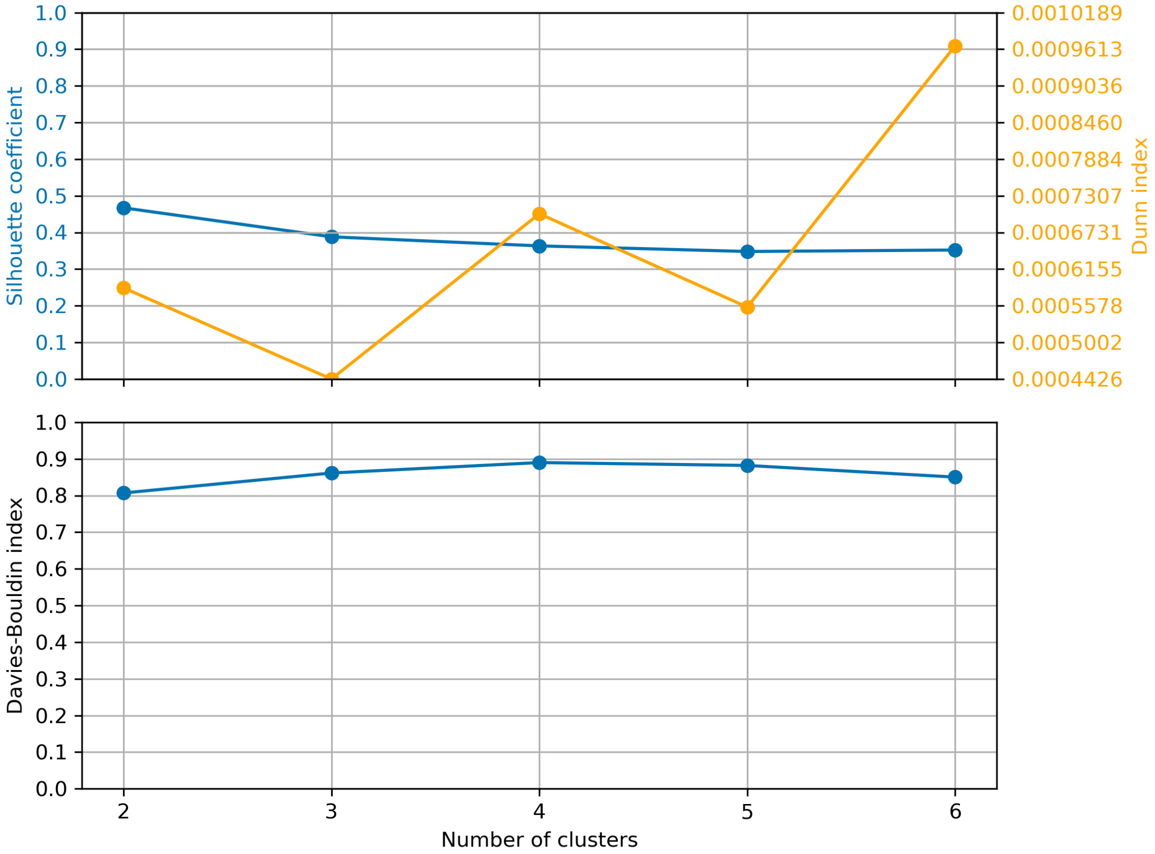


Figure 6.12: Cluster evaluation per cluster count

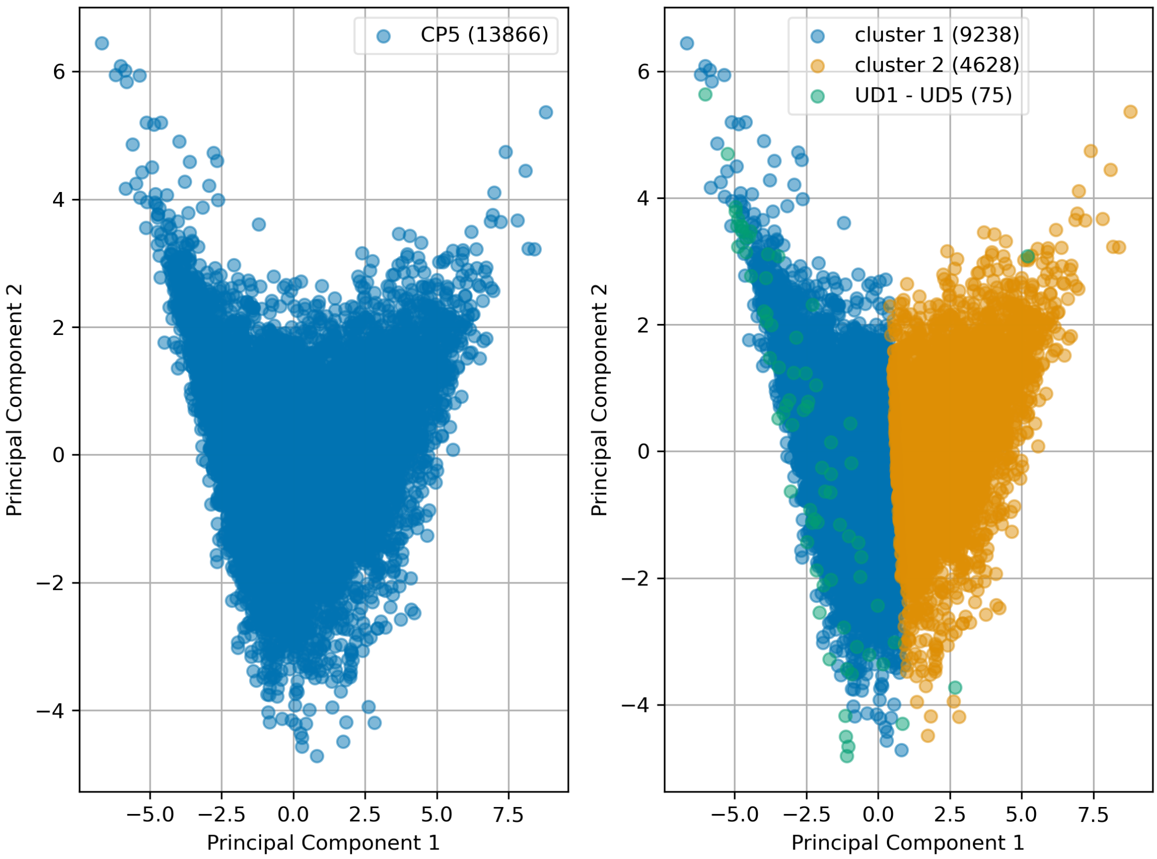


Figure 6.13: K-means result on CP5 data

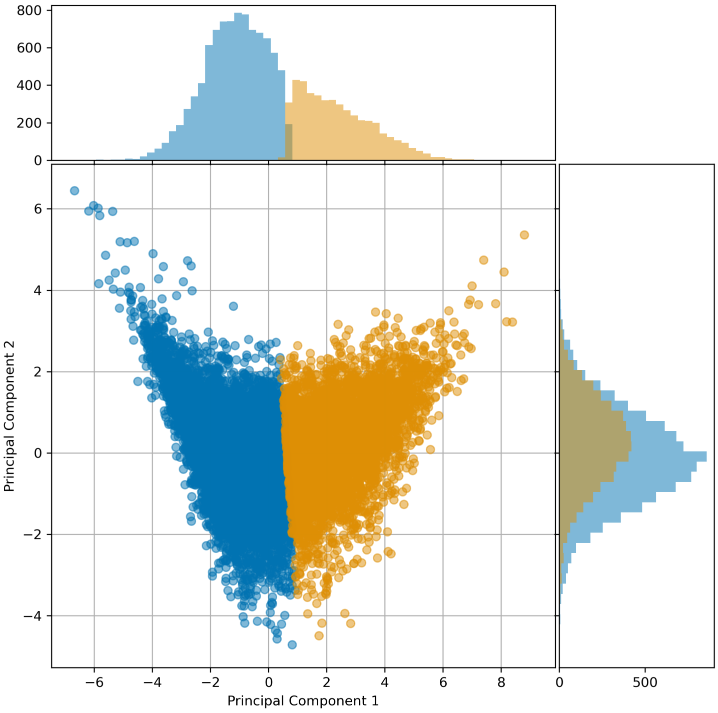


Figure 6.14: Histogram on the K-means result

The exact same process is done for other CP specimen, to check the consistency of the method. The evaluation metrics for cluster performance and the correlation of clusters with the matrix cracking data is shown in table 6.3. The figures for these results are shown in fig. B.1 - fig. B.4. The silhouette index is pretty much consistent across the tests. The same hold for the Davies-Bouldin index, although they both give a bad score to CP4, which has a lot of datapoints relative to the other specimens. The Dunn index seems to be somewhat influenced by the number of datapoints as well. For the Silhouette index for the matrix/cluster correlation, a low value is preferred. The optimal case would be a silhouette value of 0, or even negative, which indicates that the cluster and matrix cracking data are very similar. For all specimens there is a clear distinction between the Silhouette indices. The difference in values range from 0.37 to 0.51. The Silhouette index from the cluster analysis is in most cases similar to the minimum value of the matrix cracking/cluster correlation. This would mean that cluster 1 and cluster 2 are just as similar as cluster 1 and the matrix cracking data, even though this is visually clearly not the case. In that regard the Davies-Bouldin index proves to be a better metric. There is a clear distinction between the Davies-Bouldin index for the cluster analysis and the maximum index between the clusters and the matrix cracking data. CP3 for example has a Davies-Bouldin index of 0.86 and the index for cluster 2 and the UD data are 1.01. This indicates that cluster 1 and 2, are distinct about as much as cluster 2 and the UD data. Cluster 1 and the UD data are clearly more similar with a index of 4.75.

Table 6.3: Cluster analysis for CP3-CP7

		Cluster analysis				**UD data/cluster correlation**	
	N points	**N clusters**	**Silhouette**	**Davies-Bouldin**	**Dunn**	**Silhouette indices**	**Davies-Bouldin indices**
CP3	9574	2	0.44	0.86	0.002568	• 0.37 • 0.57	• 4.75 • 1.01
CP4	26743	2	0.41	0.93	0.000452	• 0.51 • 0.65	• 0.94 • 1.74
CP5	13866	2	0.47	0.81	0.000586	• 0.41 • 0.62	• 3.28 • 0.89
CP6	10504	2	0.44	0.89	0.00205	• 0.41 • 0.58	• 3.42 • 0.99
CP7	11947	2	0.46	0.82	0.000431	• 0.42 • 0.61	• 3.65 • 0.94

The clustered data is remapped to the four basic parameters: amplitude, duration, rise time and counts in fig. 6.15. Two main things can be observed in these plots. The scatter of cluster 1 in the Amplitude vs Counts plot is a bit steeper, meaning that a signal of similar amplitude is likely to have more counts in cluster 1. The second point is that cluster 2 seems to have a more restricted rise time. With this figure it also becomes apparent that it is very difficult to separate the clusters based only on basic parameters, given that using wavelet transform features is an appropriate way to cluster the data.

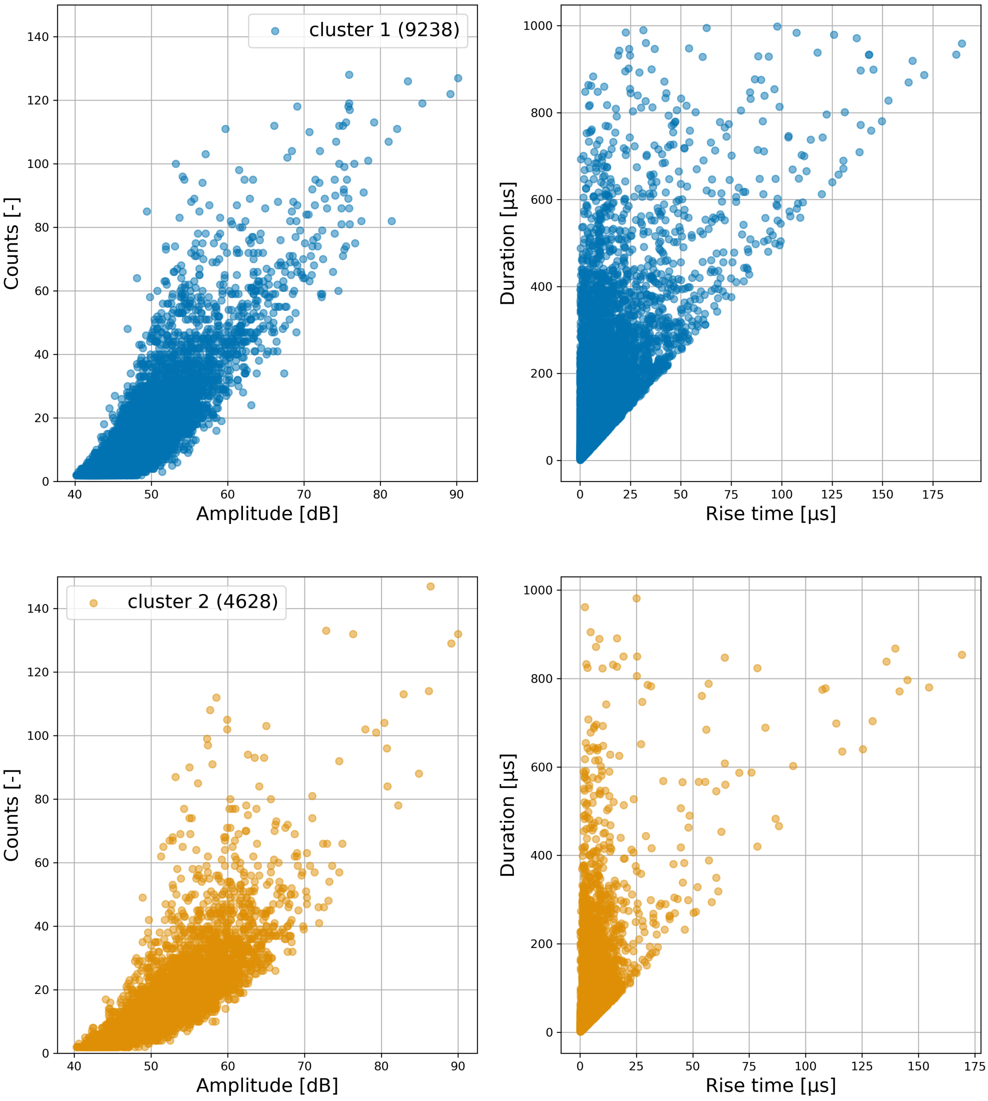


Figure 6.15: Remapping to basic AE features

6.2.2. Standardization UD data

Similarly to the standardization of CP data, for this case also a total number of 12 trials are done in order to find the optimal set of variables for the clustering analysis. The clustering performance is tracked, as well as the time taken for the analysis. Again, the process is repeated several times to check for any fluctuations in the computation time. The Silhouette coefficient and the time taken for each feature group trial is shown in fig. 6.16. This figure is clearly different from the one where all data was standardized. The time required for the analysis does not progress logically. The axis range is also quite small so apparently adding variables does not increase the time. The cluster performance is steady with the exception of the last point. There is a peak at five added variables, but it is not considered worth the potential extra time and memory required. So, for the more in depth analysis for this case the more convenient set of variables is taken: WP1-WP4 and WP6-WP8.

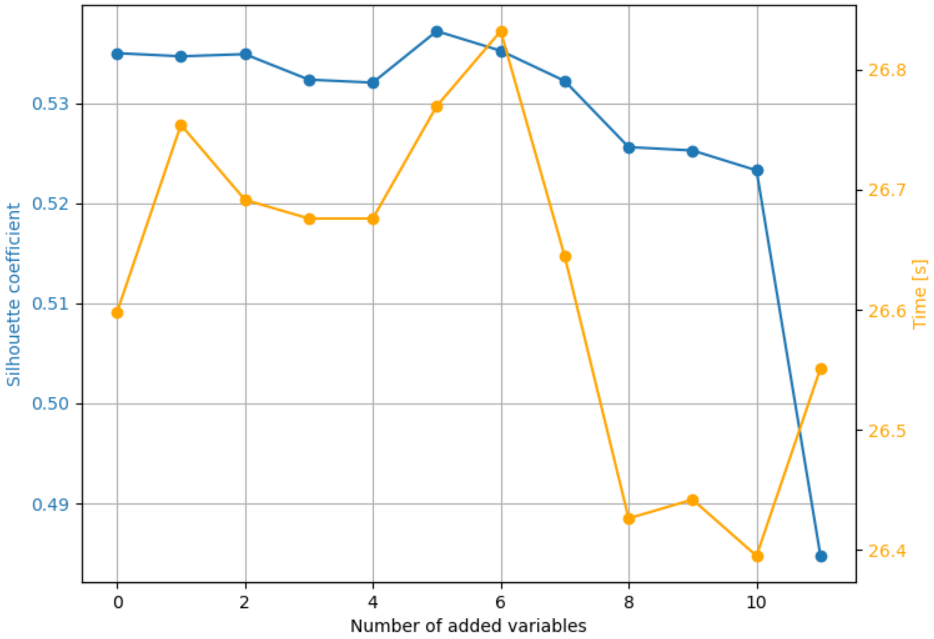


Figure 6.16: Cluster and time-wise performance of different feature groups with data standardized based on UD data

The selected feature set is again clustered with the k-means method for a number of clusters ranging from 2 to 6. For each cluster count the result is evaluated with the Silhouette coefficient, the Dunn index and the Davies-Bouldin index. The outcome is shown in fig. 6.17. For this case all metrics differ in the amount of clusters they score best. The Silhouette coefficient is highest for two clusters, whereas the Dunn index is highest for four clusters. The Davies-Bouldin index indicates the optimal amount of clusters is five. Judging from this the number of clusters is inconclusive. Since the value of the Silhouette is most easy to judge, this is taken as the decisive metric, thus the number of clusters is again set at two. Depending on the goal of the cluster analysis, one could argue for more clusters but that is not the case here. The result of the k-means with two clusters is shown in fig. 6.18. With the exception of one datapoint, the UD data and cluster 1 overlap fully. The number of datapoints that would be associated with matrix cracking would thus be almost 10000, which is a bit more than what was seen in the result where the data was standardized based on all data. This means that there are more signals similar to the UD signals, which is not shown in the case where the data is standardized based on all data. The histograms of the processed data is plotted in fig. 6.19. This plot is very similar to what was seen before. Based on the histograms one could not distinguish between two separate clusters.

The clustering process is repeated for other specimens, to see how consistent it is in table 6.4. The figures for these results are shown in fig. B.5-fig. B.8 .Here, there is one case where the optimal

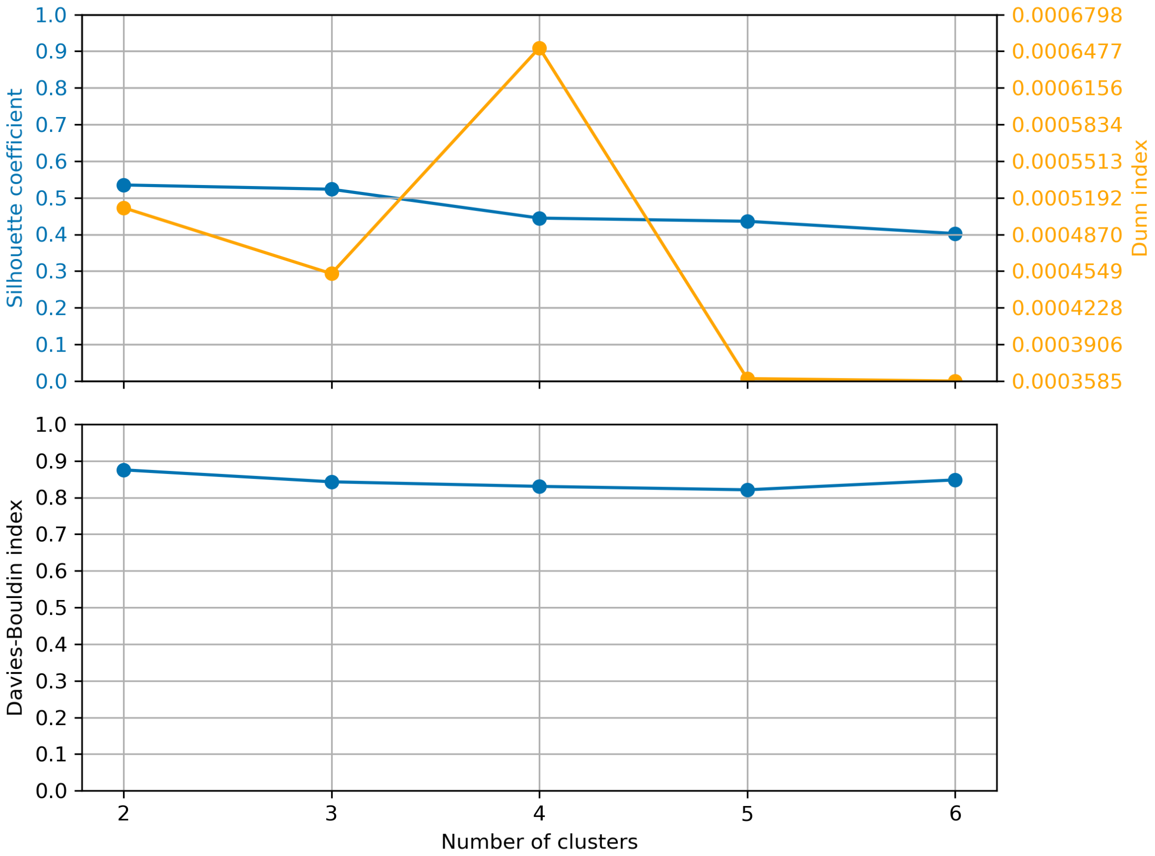


Figure 6.17: Cluster evaluation per cluster count with data standardized based on UD data

number of clusters is actually three. The Silhouette coefficient and the Davies-Bouldin are pretty much consistent with all tests. The Dunn shows along with the other metrics that CP4 has the worst performance. This specimen also lead to by far the most signals and separates itself from the rest in that regard. In general the minimum Silhouette indices for the similarity of the UD data and the clusters are very clear, varying from 0.06 to 0.13 and one exception of 0.35. These values indicate that the UD data and corresponding cluster are very similar. The Davies-Bouldin indices show the same pattern with a clear distinction of the clusters relation with the UD data.

Table 6.4: Cluster analysis for CP3-CP7, UD standardized data

		Cluster analysis				**UD data/Cluster correlation**	
	N points	**N clusters**	**Silhouette**	**Davies-Bouldin**	**Dunn**	**Silhouette indices**	**Davies-Bouldin indices**
CP3	9574	3	0.54	0.83	0.00154	• 0.54 • 0.13 • 0.54	• 0.54 • 2.00 • 0.53
CP4	26743	2	0.49	0.81	0.000140	• 0.54 • 0.35	• 0.69 • 1.79
CP5	13866	2	0.53	0.88	0.000510	• 0.13 • 0.40	• 1.52 • 0.64
CP6	10504	2	0.57	0.81	0.00092	• 0.07 • 0.45	• 1.37 • 0.57
CP7	11947	2	0.50	0.90	0.000391	• 0.06 • 0.42	• 1.39 • 0.60

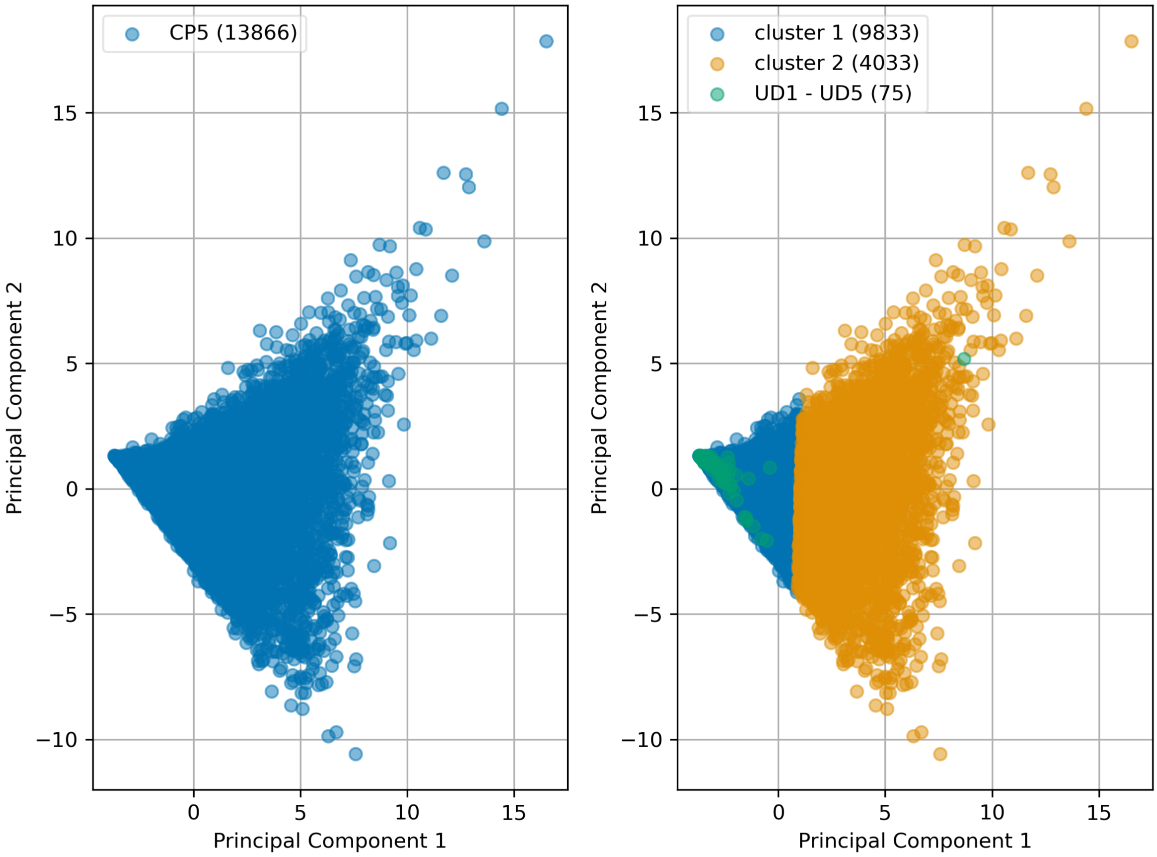


Figure 6.18: K-means result on CP5 data standardized based on the UD data

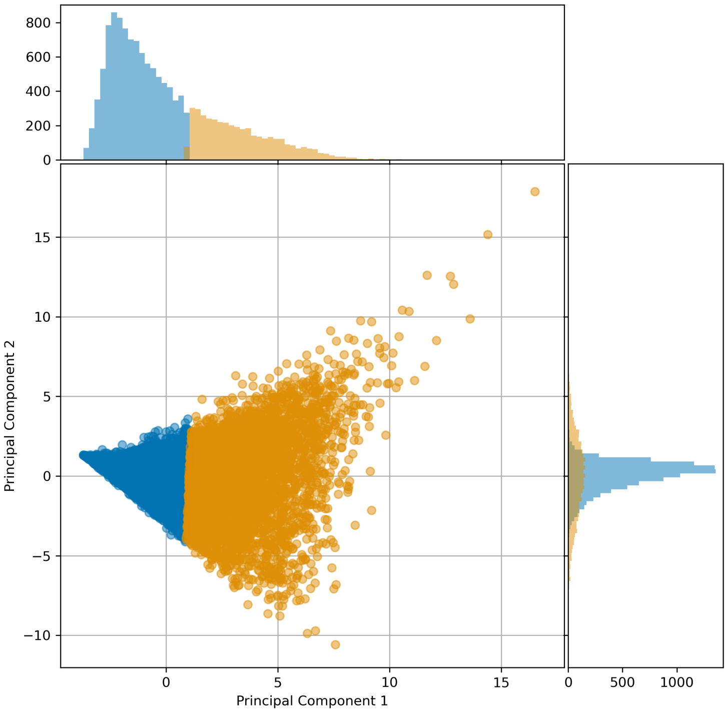


Figure 6.19: Histogram on the K-means result with data standardized based on UD data

The clustered data is remapped to the basic AE features in fig. 6.20. The plots are pretty similar to the one where the data was standardized based on all data. This is somewhat of a confirmation that the data allocated to the clusters is the same, whereas here even more data is assigned to cluster 1. Also in this case it still remains fairly impossible to distinguish the clusters based on the basic AE features only.

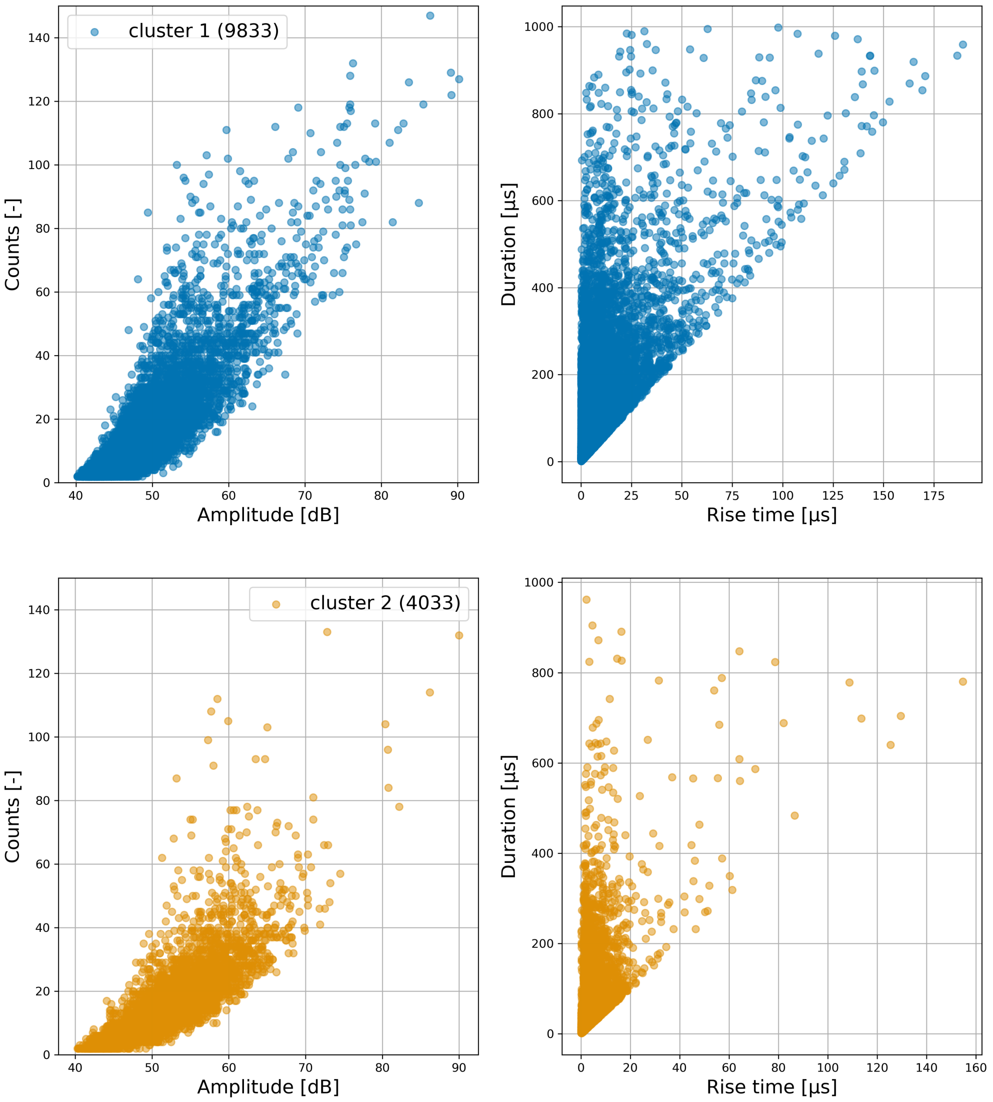


Figure 6.20: Remapping to basic AE features with standardization based on UD

6.3. Real-time damage evaluation

For the real-time analysis CP5 is again used from the static tests and CP14 from the fatigue tests. Unfortunately, CP14 was also the only fatigue specimen of which the data could be processed within a few days, due to the fact that the files were too large. The process of real-time clustering is quite similar to the post processing clustering process, with some modifications. The UD data is processed beforehand and functions as a fingerprint to identify the clusters that evolve with time. The most realistic simulation would be if AE signals come in and are processed only then. However, for ease of repeatability and analysis the features are processed beforehand. To understand what the effect will be on the time-wise performance of the algorithm can be deducted from fig. 6.9. It takes a little over $100ms$ to calculate wavelet features for a thousand signals. The computation time of the clustering process will be discussed however.

The processed features are standardized based on the UD data. This is to ensure a more stable analysis. A constantly changing mean and variance for standardization would require a lot of recalculation of datapoints. The dimensionality is then reduced to two dimensions with the Incremental Principal Component Analysis method, which is described in section 4.2.2. With the IPCA method one can update the covariance matrix and eigenvectors with a batch of data. For the quasi static tests the batch size is set to 1,000, meaning IPCA is applied with every 1000 datapoints. This process could be optimized by checking the change in eigenvectors and define a threshold after which the change is too small to keep updating the IPCA. Due to the large number of datapoints of the fatigue tests, the batch size is set to 5,000 data points and the IPCA is no longer updated after 20,000 data points. The reason for this stoppage is to prevent any issues with doing computations with large matrices of data. For clarity, IPCA returns a symmetric matrix of size $d \times d$, with d being the number of features. The two eigenvectors with the largest eigenvalues are selected giving a $d \times 2$ matrix. Multiplying data of size $n \times d$ with this matrix reduces the matrix size to $n \times 2$.

The processed and reduced CP data is then clustered incrementally in two ways: with the Incremental K-means and a more advanced method called Adaptive Sequential K-means or Advanced Incremental K-means. Here, Incremental K-means works like Adaptive Incremental K-means without merging of clusters. The workings of these methods is described in section 4.3.2 and section 4.3.3, respectively. The resulting clusters are then compared with the processed and scaled UD data, to figure out which one is most likely related to matrix cracking signals. The Davies-Bouldin index is used as an evaluation metric for the comparison of the clusters with the UD data. The outcome is the real-time tracking of matrix cracking as opposed to the other damage mechanisms: delamination and fiber breakage. The entire real-time clustering process is visualized in a flowchart as shown in fig. 6.21. The

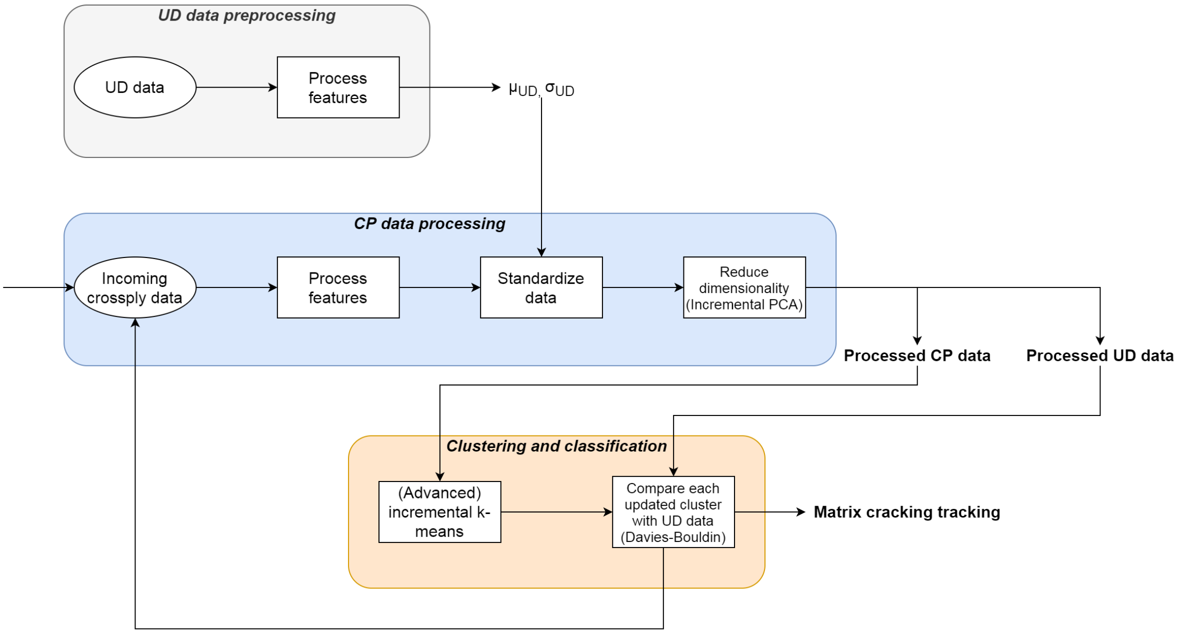

UD data preprocessing
UD data
Process features
μ_{UD}, σ_{UD}
CP data processing
Incoming crossply data
Process features
Standardize data
Reduce dimensionality (Incremental PCA)
Processed CP data
Processed UD data
Clustering and classification
(Advanced) incremental k-means
Compare each updated cluster with UD data (Davies-Bouldin)
Matrix cracking tracking

Figure 6.21: Real-time clustering process

real-time clustering analysis is done using the two stated methods on both CP tests and fatigue tests. Different stages during the test are captured and the result at that time will be displayed.

6.3.1. Incremental clustering

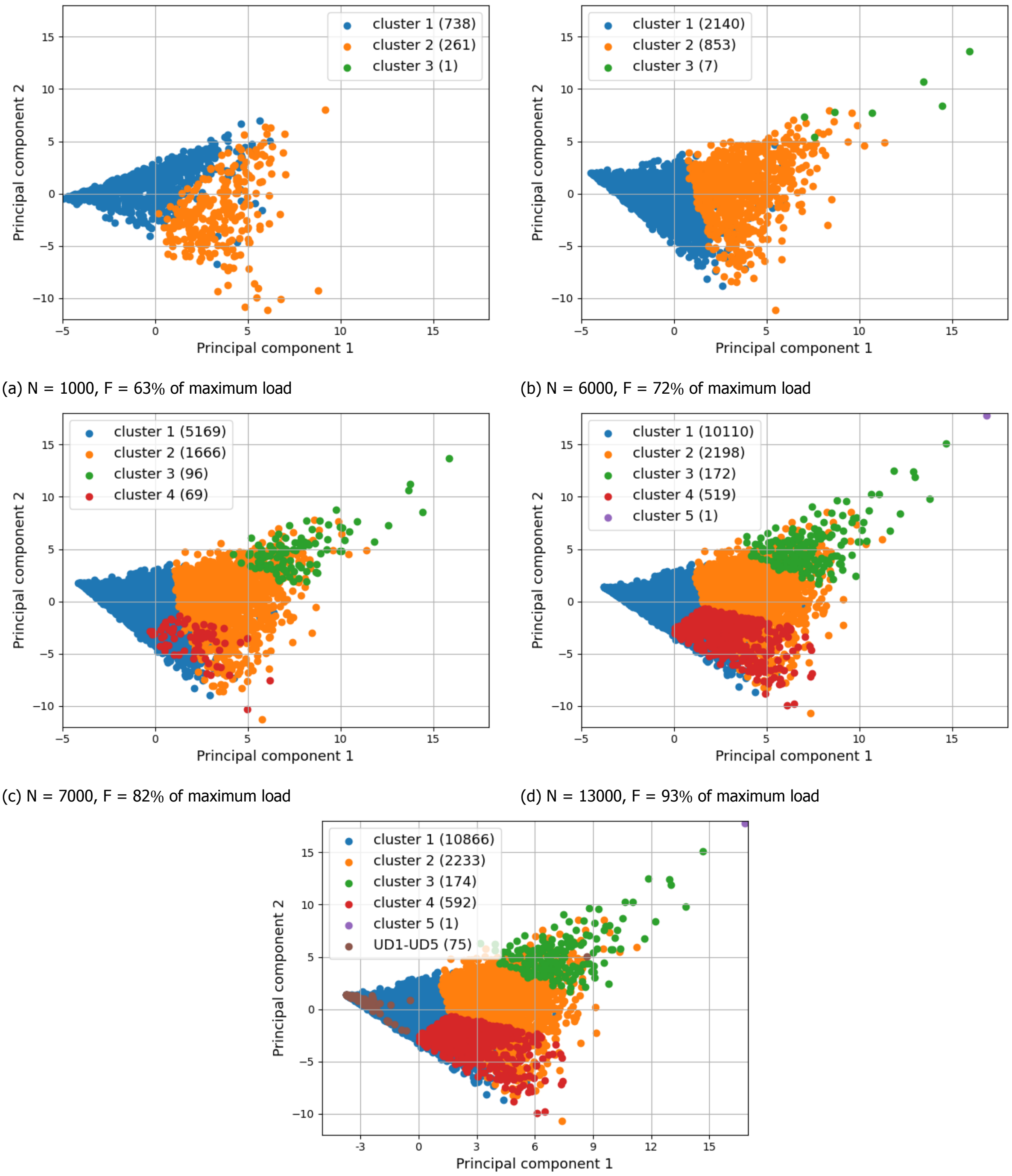
This is the most basic form of real-time clustering. It starts out with a single data point as a cluster. New datapoints are added to the cluster or new clusters are created based on the distance between the new datapoint and the clusters at that time. The minimal distance, d, at which a new cluster is created is based on the UD data, which is determined as in eq. (4.11) but repeated here:

$$d = \bar{D} + 3 \cdot \sigma_{yy}$$

where $\bar{D}$ is the mean value distance between the points and σ_{yy} is the standard deviation of these distances.

Quasi static: CP5

The incremental clustering result for CP5 is shown in fig. 6.22. A total of five stages are displayed here. The first plot, Figure 6.22(a), is when a reasonable amount of AE signal have been measured at 63% of the load. The next plot is after about a 10% increment of the load, fig. 6.22(b). In comparison to the first plot it seems to be rotated, which indicates that the eigenvectors from the covariance matrix have changed significantly. This is expected since the eigenvectors are likely to change in the early stages when there are not so many datapoints yet. In fig. 6.22(c) at 82% of the maximum load the total number of four clusters have emerged. Almost at the end of the test the clusters have taken shape in fig. 6.22(d). All clusters have a similar surface area, but are drastically different in count. The final figure, fig. 6.22(e), shows that cluster 1 visually matches quite well with the UD data. Whether this is actually true in a mathematical sense can be observed in fig. 6.23. Keeping in mind that a low Davies-Bouldin index indicates a good clustering performance, cluster 1 is distances itself from the other clusters. It is therefore quite clear that cluster 1 must be associated with the matrix cracking damage mechanism, since it has the highest values during the whole test. The Davies-Bouldin indices for cluster 2-4 are fairly consistent and low, thus they are less likely to be related to matrix cracking. Cluster 5 is disregarded here since it only has a single datapoint. The development of the clusters is plotted in fig. 6.24. Cluster 1 is clearly the most dominant one and is growing consistently in the second part of the test. Cluster 2 has also developed quite significantly towards the end of the test. Cluster 3 and 4 only come into play at the far end. Keeping in mind the damage mechanisms, it makes sense that cluster 1 develops during most part of the test, although it is expected that it would saturate at some point. Delamination could be a result of the matrix cracks, which could then be correlated to cluster 2. Fiber breakages can occur randomly, or at the end when the delamination has taken place and stress concentrations occur at the 0°layers.



(a) N = 1000, F = 63% of maximum load

(b) N = 6000, F = 72% of maximum load

(c) N = 7000, F = 82% of maximum load

(d) N = 13000, F = 93% of maximum load

(e) N = 13866, F = 95% of maximum load

Figure 6.22: Incremental clustering results from quasi static test CP5 at different stages

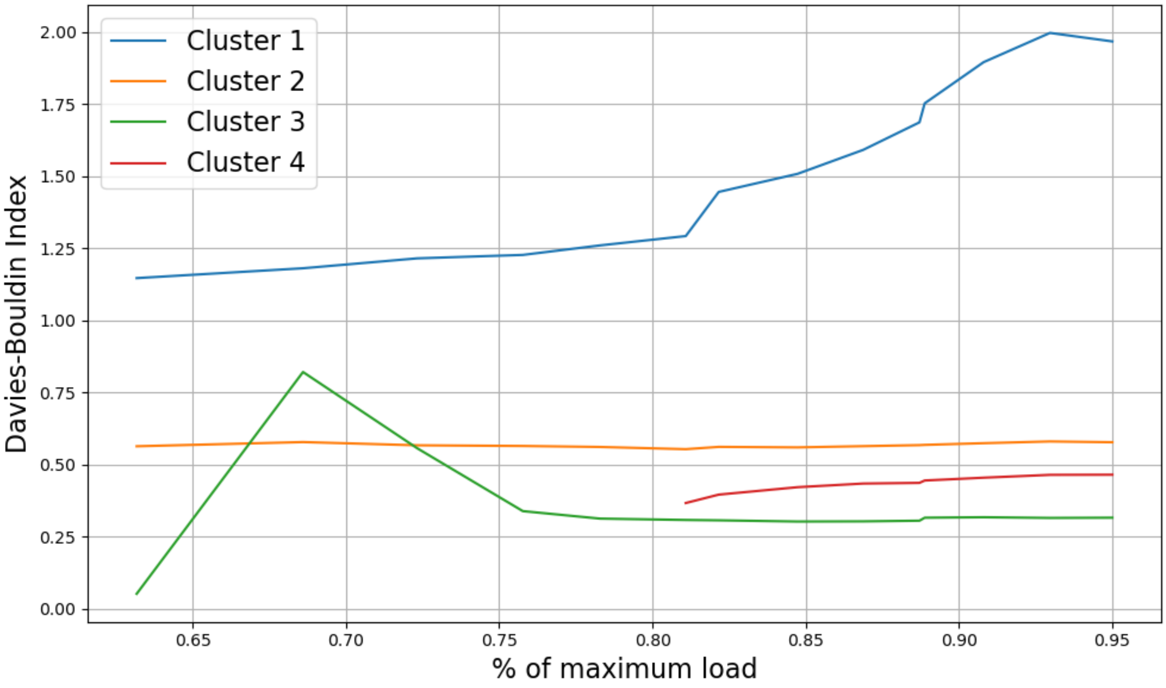


Figure 6.23: Davies-Bouldin indices for clusters resulting from incremental k-means on the static test CP5

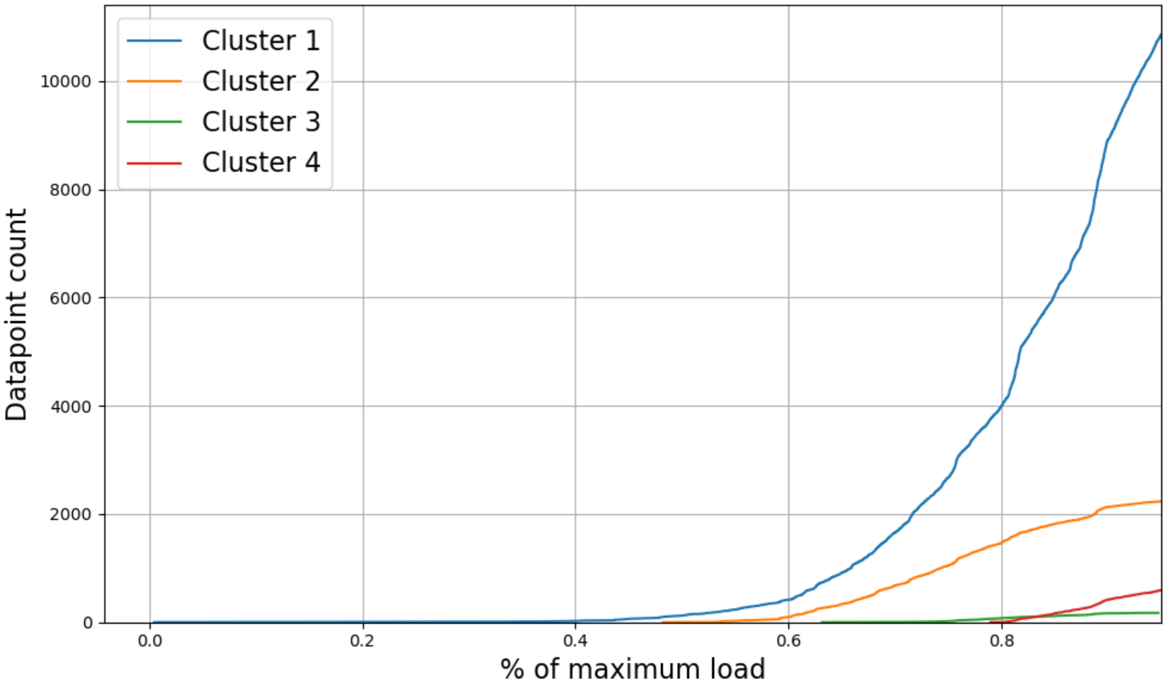
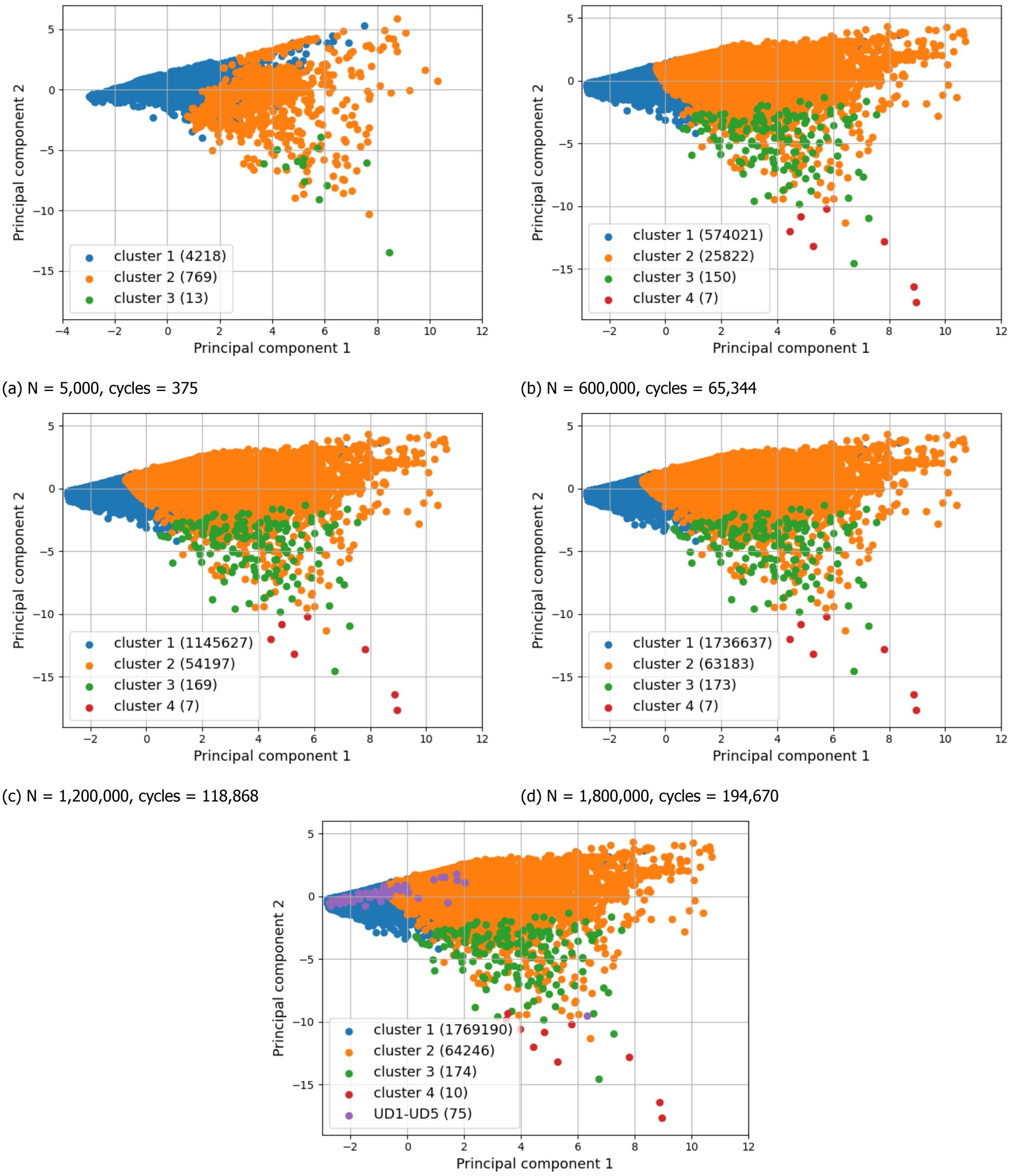


Figure 6.24: Cluster count for the incremental clustering result of CP5

Fatigue: CP14

Now the same incremental clustering process is applied to a fatigue test. A total of five stages have been plotted in fig. 6.25. It starts off with a plot of 5000 signals in fig. 6.25(a). Already three clusters have formed here, even though cluster 3 has only 13 datapoints. In the next plot, fig. 6.25(b), after about 65,000 cycles later, cluster 1 and cluster 2 have developed significantly. Also, another cluster has formed with very limited datapoints. The clusters progress in a similar way as visible in plots fig. 6.25(c) and fig. 6.25(d), with cluster 1 having by far the most datapoints. Cluster 2 also has a significant number of signals assigned to it, unlike cluster 3 and 4. Lastly, fig. 6.25(e) shows the result after a total of 200,000 cycles along with the UD data. Here, there is a visible overlap of the UD data on cluster 1 and 2. The same can be found in the cluster correlation with the UD data based on the Davies-Bouldin index, plotted in fig. 6.27. At the final stages, the indices for cluster 1 and cluster 2 start to converge, meaning both of them correlate with the matrix cracking. Since at the very end cluster 2 the similarity with the UD data increases, the mean of cluster 2 has to be moving towards the mean of the UD data. The development of all clusters is shown in fig. 6.26. From the plot it is clear that cluster 1 and 2 progress similarly, apart from at the beginning of the test. This could mean that they are related to the same damage mechanism, also considering that the specimen is loaded at about 70% of the maximum load, so it is very likely mainly matrix cracking will occur. It is also very important here to take into account that the UD data is from a quasi static test, whereas this specimen was loaded in fatigue. The fatigue loading causes a more gradual development of the matrix cracking damage mechanism, which can result in different AE signals.



(a) N = 5,000, cycles = 375

(b) N = 600,000, cycles = 65,344

(c) N = 1,200,000, cycles = 118,868

(d) N = 1,800,000, cycles = 194,670

(e) N = 1,833,620, cycles = 200,000

Figure 6.25: Incremental clustering results from CP14 fatigue test at different stages

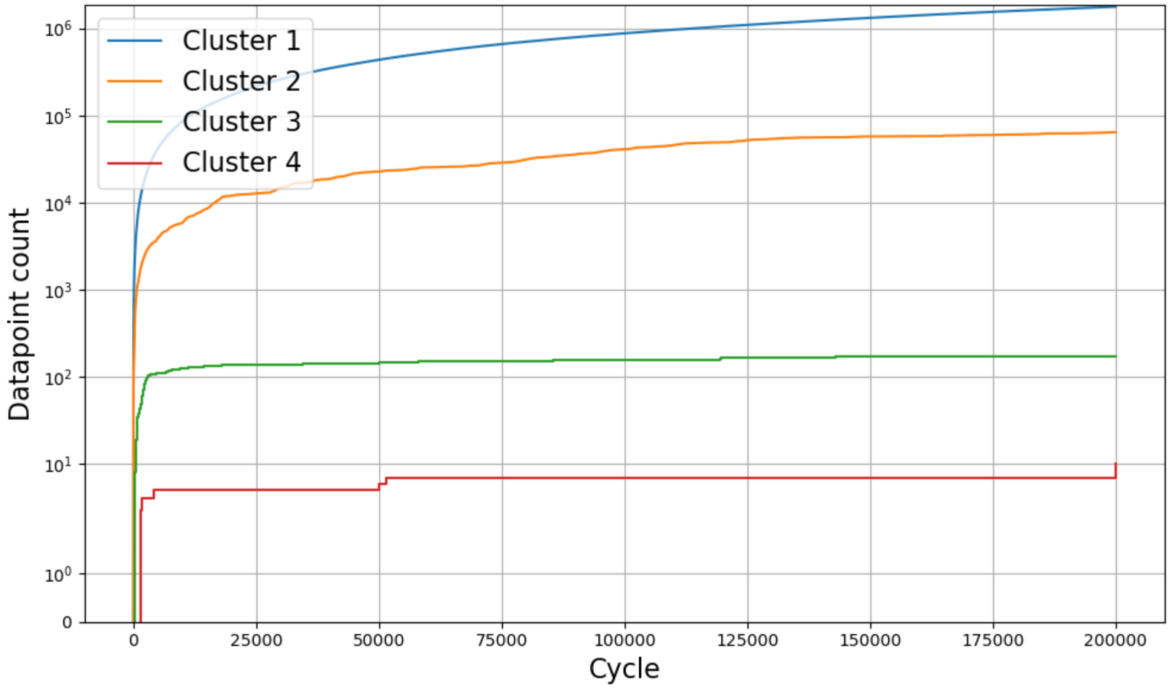


Figure 6.26: Cluster count for the incremental clustering result of fatigue test CP14

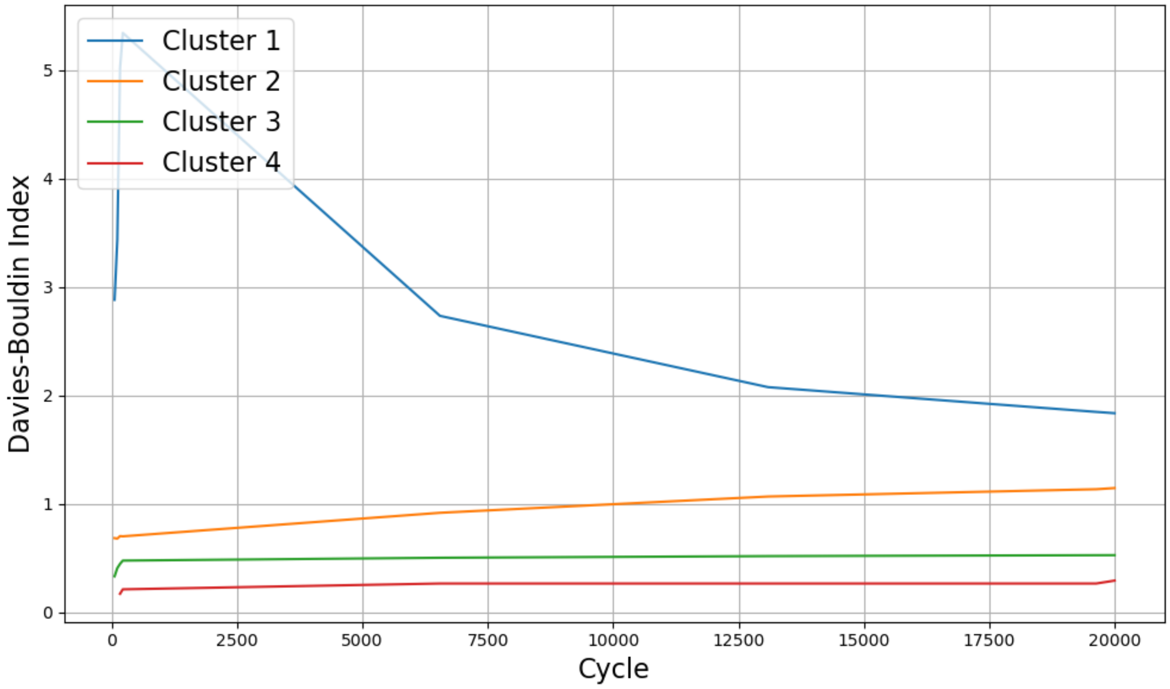


Figure 6.27: Davies-Bouldin indices for clusters resulting from incremental k-means on the fatigue test CP14

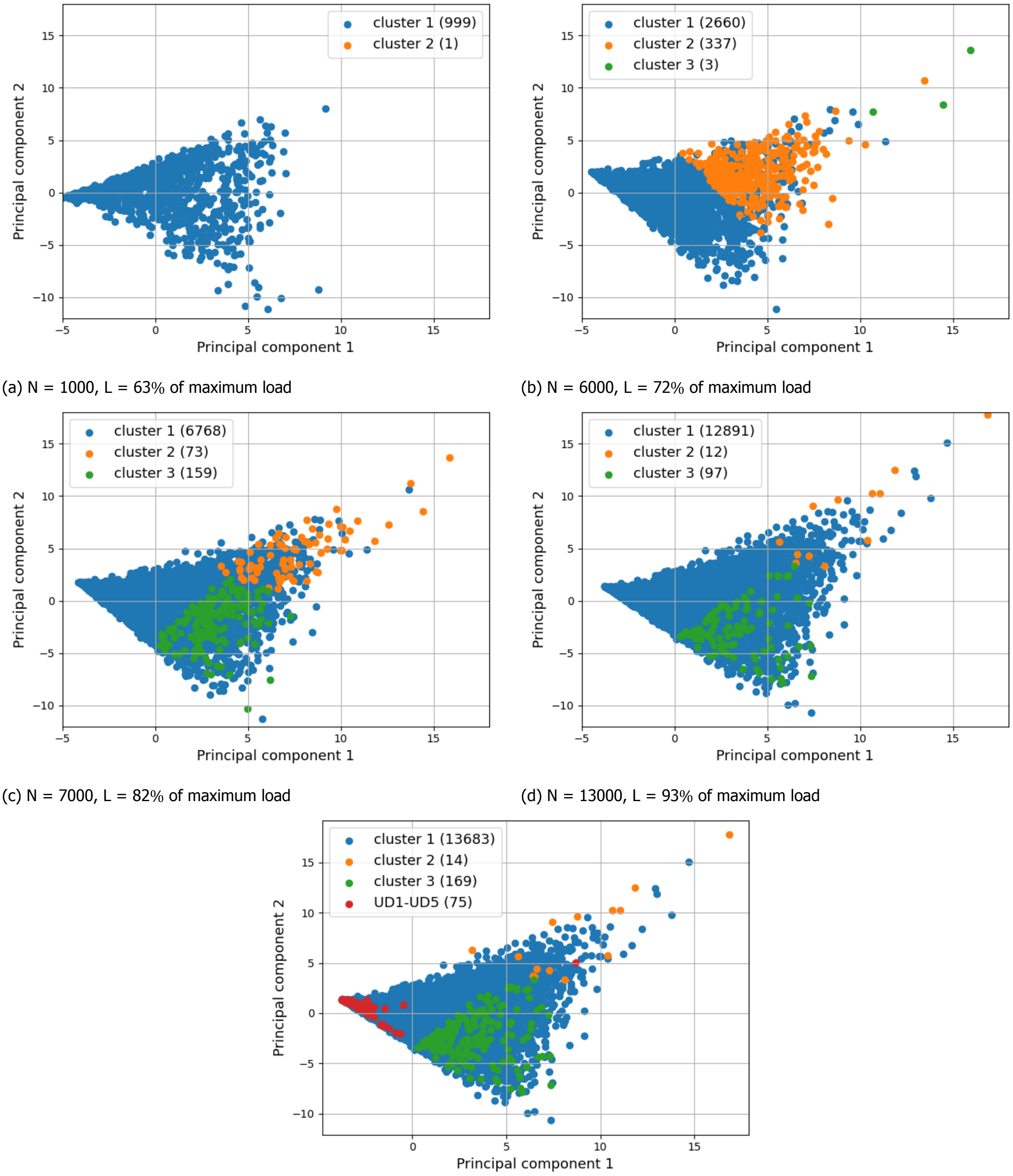
6.3.2. Adaptive incremental clustering

Now follows an analysis of a more advanced form of real-time clustering. Again, it starts out with a single data point as a cluster. New datapoints are added to the cluster or new clusters are created based on the distance between the new datapoint and the clusters at that time. The minimal distance, d, at which a new cluster is created is based on the UD data is calculated in the same way as with the basic incremental clustering algorithm as stated in the previous section. The difference here is that each time a new cluster is formed, the clusters at that point in time are evaluated. It is then decided whether clusters overlap too much and have to be merged or not. The mathematical details are described in section 4.3.3. The only variable that is to be set is α, which a number between 2 and 3 that will influence whether clusters are merged or not. By trials it was found that this value has very minimal impact on the results is therefore set to 2, which limits the merging of clusters.

The method will be applied to the CP test as well as to the fatigue test. The results are compared to the basic incremental clustering to see how it affects the clustering process.

Quasi static: CP5

The adaptive incremental clustering result for CP5 is shown in fig. 6.28. The same five stages are plotted again here as in the basic incremental clustering case. This analysis is very one sided, starting practically with a single cluster in fig. 6.28(a). If one compares the plot with fig. 6.22(a), it is clear that the clusters have merged into one, as they met the criteria for the merging process. New clusters 2 and 3 are formed as the load continues in fig. 6.28(b), but cluster 2 is merged and formed again as visible in fig. 6.28(c). The clustering keeps progressing in this manner with again a dominating cluster 1 in fig. 6.28(d). Finally the result is plot with the UD data as well in fig. 6.28(e). These results beg the question whether the algorithm is suitable for this kind of data, or whether changes need to be made in order for it to work. However, if the algorithm is assumed to work correctly, it indicates that for the most part there is only a single damage mechanism present. Any other damage mechanism represented by cluster 2 and 3 would only start to develop at the very end of the test, as can be seen in fig. 6.29. The similarity of the clusters to the UD data is plotted in fig. 6.30. It is again clear that cluster 2 and 3 are distant from the matrix cracking signals. Cluster 1 is most similar to the matrix cracking data, although the values are less, relative to the incremental clustering method (fig. 6.23).



(a) N = 1000, L = 63% of maximum load

(b) N = 6000, L = 72% of maximum load

(c) N = 7000, L = 82% of maximum load

(d) N = 13000, L = 93% of maximum load

(e) N = 13866, L = 95% of maximum load

Figure 6.28: Adaptive incremental clustering results from crossply quasi static test at different stages

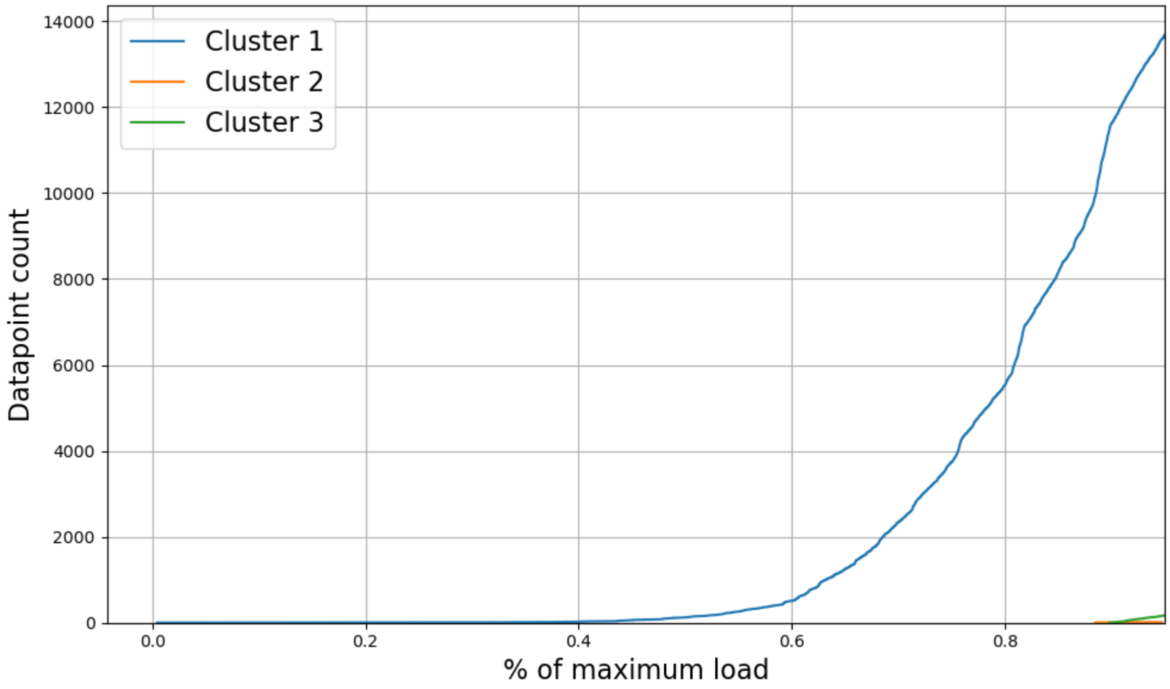


Figure 6.29: Cluster count for the adaptive incremental clustering result of CP5

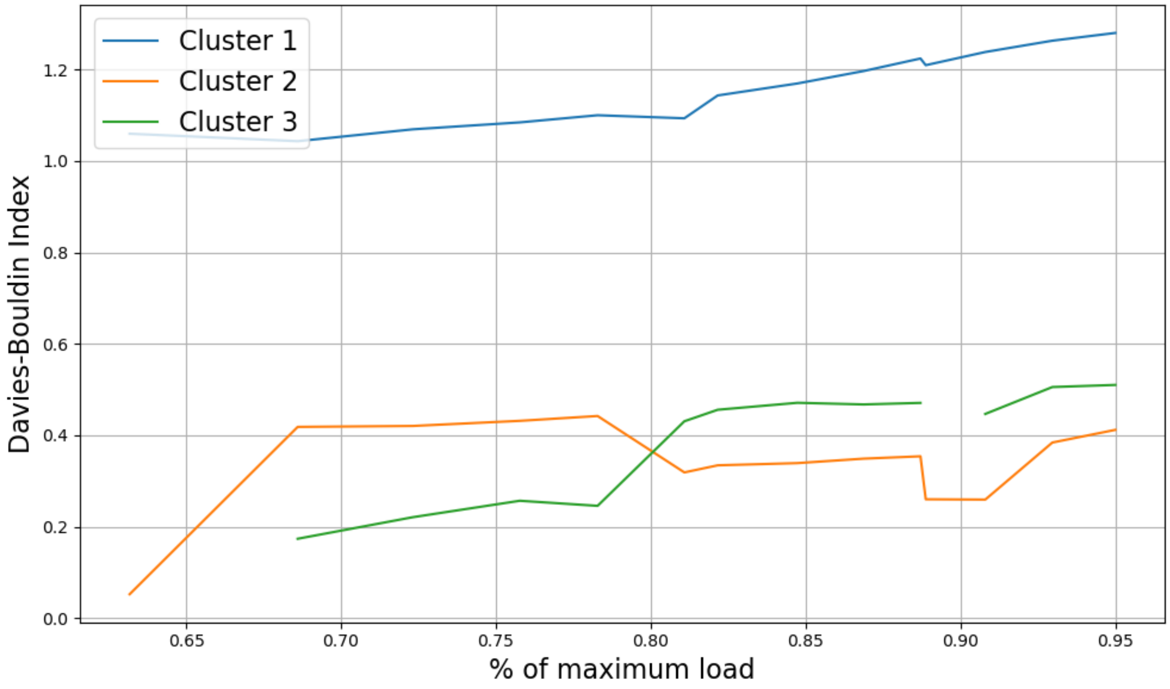
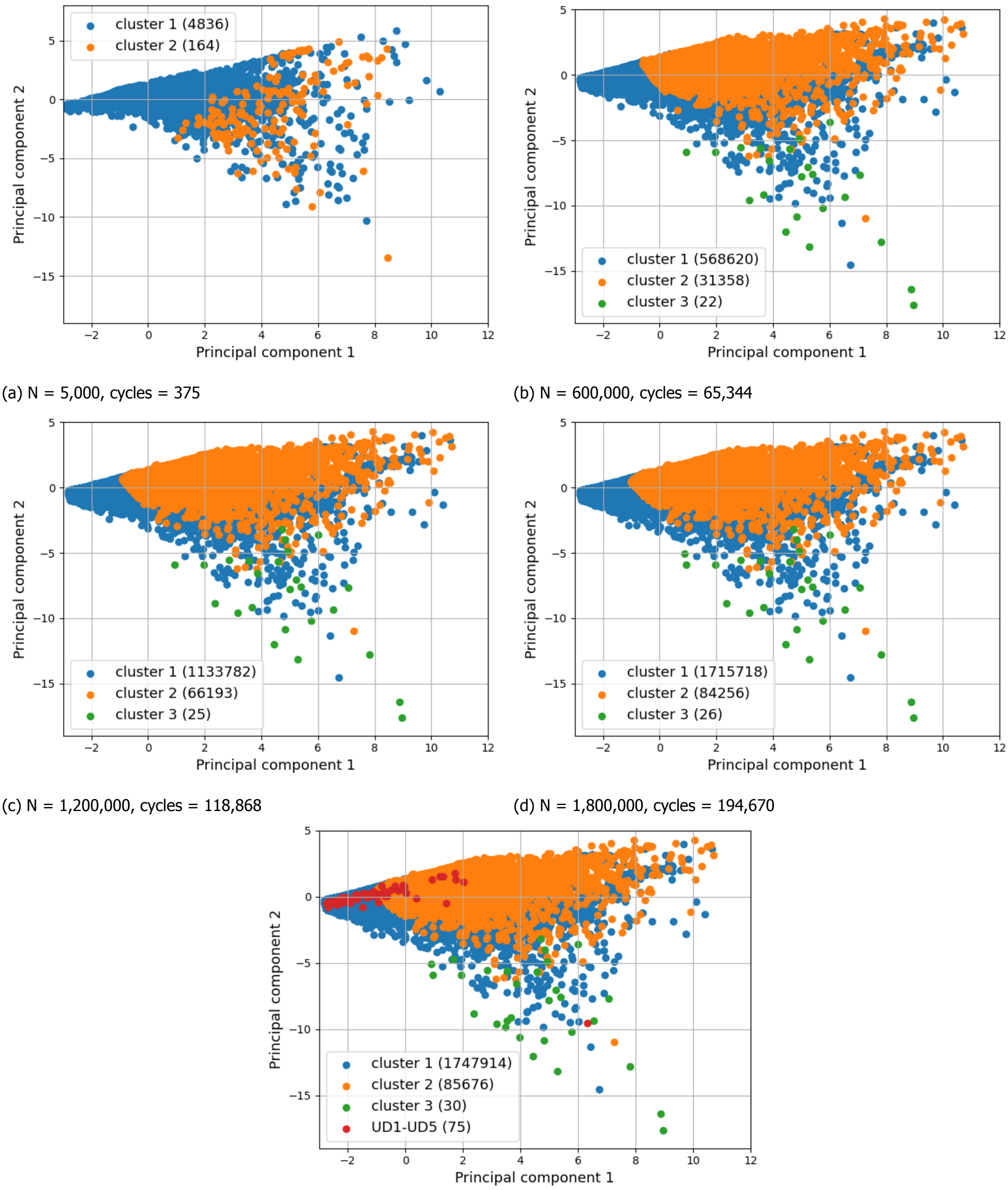


Figure 6.30: Davies-Bouldin indices for clusters resulting from adaptive incremental k-means on the static test CP5

Fatigue: CP14

The adaptive incremental clustering process has also been applied to a fatigue test in fig. 6.31. Contrary to the quasi static CP5 test, the adaptive clustering process for the fatigue test progresses similarly to the basic incremental clustering. At the first plot, fig. 6.31(a), there are two clusters present, contrary to the three clusters in fig. 6.25(a). These clusters keep progressing and a third has formed in fig. 6.25(b). Cluster 1 and cluster 2 keep growing in size as the test goes on, visible in fig. 6.25(c) and fig. 6.25(d). Finally, the result after 200,000 cycles is plot with the UD data as well in fig. 6.25(e). The difference with the basic incremental clustering is that more points have been allocated to cluster 2 and only three clusters have formed here. For the cluster correlation with the UD data a similar pattern is visible as was seen with the incremental k-means method. Cluster 1 and cluster 2 progress similarly during the test, apart from the beginning in fig. 6.32. Towards the end of the test the similarity of cluster 1 and cluster 2 with the UD data becomes more and more equal, which can be seen in fig. 6.33. This among other things, means that the means of the clusters have similar distances to the mean of the UD data. It is still possible to state that mainly matrix cracking has occurred and very little damage from delamination or fiber breakage. Cluster 3 could then be related to delamination or fiber breakage, or the datapoints from this cluster could even be considered as outliers.



(a) N = 5,000, cycles = 375

(b) N = 600,000, cycles = 65,344

(c) N = 1,200,000, cycles = 118,868

(d) N = 1,800,000, cycles = 194,670

(e) N = 1,833,620, cycles = 200,000

Figure 6.31: Adaptive incremental clustering results from crossply fatigue test at different stages

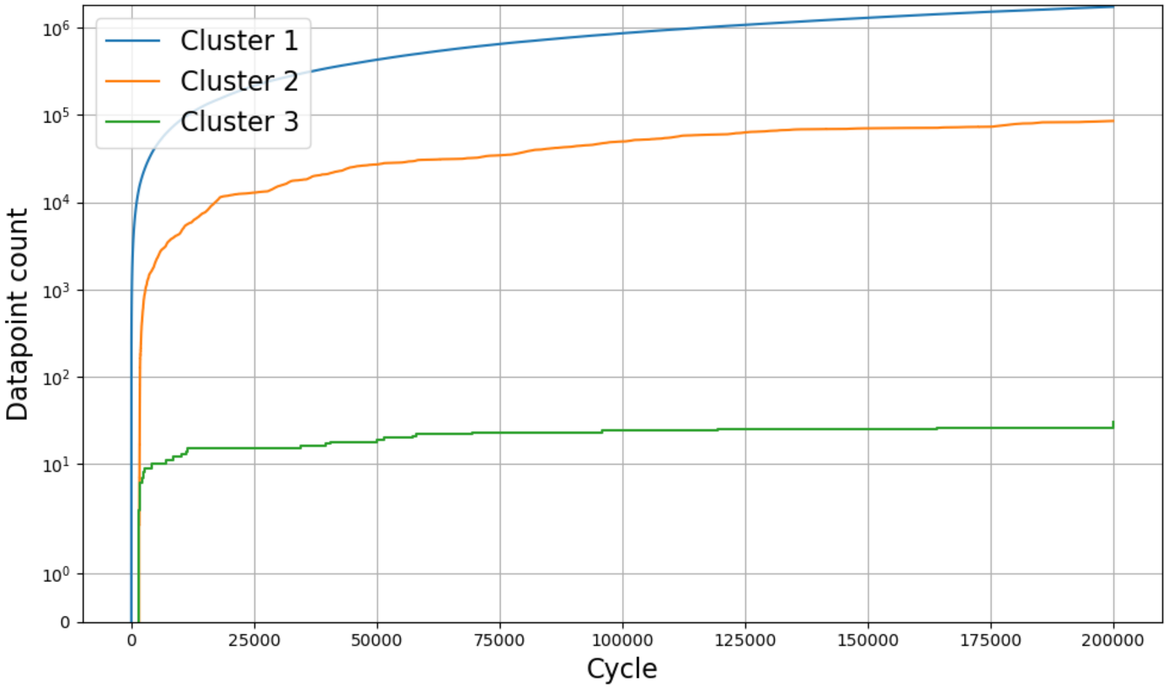


Figure 6.32: Cluster count for the adaptive incremental clustering result of fatigue test CP14

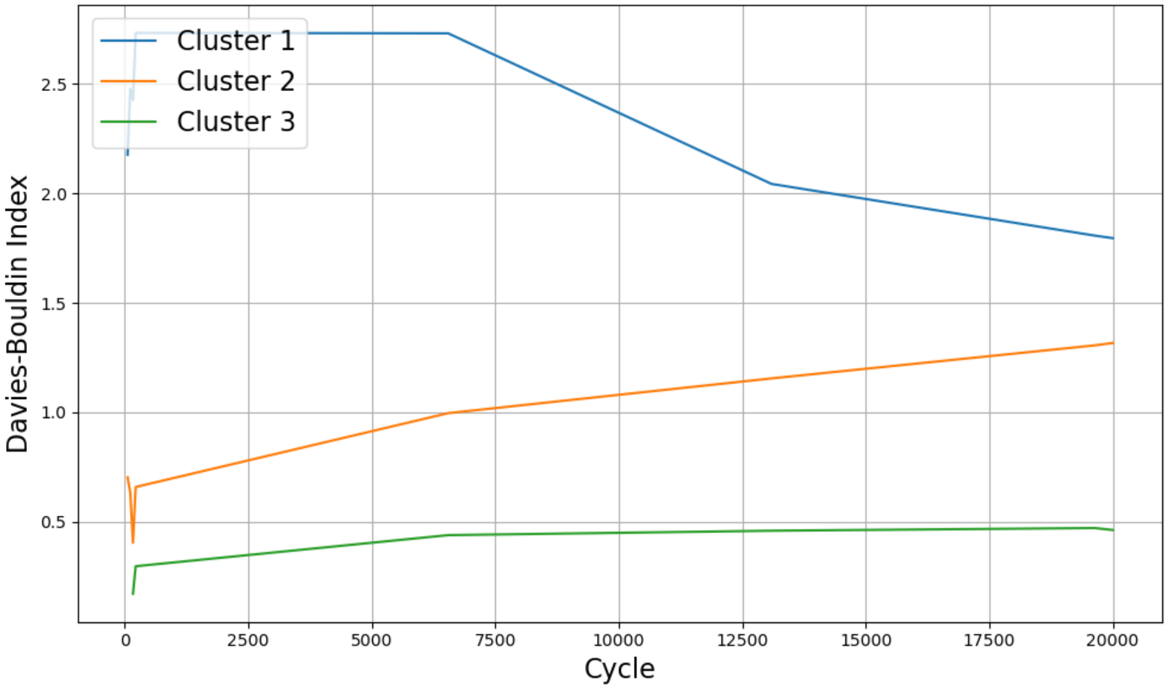


Figure 6.33: Davies-Bouldin indices for clusters resulting from adaptive incremental k-means on the fatigue test CP14

6.4. Comparison of Clustering Methods

In this section the results of the different clustering methods will be outlined and compared for the quasi static specimen and the fatigue specimen. The numbers of the resulting clusters and the evaluation of the similarity of the clusters with the matrix cracking data are analyzed. Also, the computation time required per datapoint will be presented. This, together with the feature processing time will give an indication of the time-wise performance of the whole process.

6.4.1. Quasi static test CP5

The cluster count, Davies-Bouldin indices and computation time required for each datapoint for K-means, Incremental K-means and Adaptive Incremental K-means used on the quasi static test CP5 is summarized in table 6.6. In this table it can be seen that K-means and Incremental K-means perform similarly in that they have a cluster 1 of comparable size that is related to matrix cracking. The Davies-Bouldin index is also more or less equal. The computation time for the K-means method is very short in this case. However, the K-means has time complexity $O(n^2)$, which means the time increases quadratically as the number of datapoints, n, increases. The Incremental K-means variants have a time complexity of $O(n)$, so the computation time scales linearly with the number of datapoints. So even though the K-means seems fast here, the computation time quickly adds up for larger datasets. The Adaptive Incremental K-means method did not perform well for this case. At different moments during the process, all datapoints were merged with cluster 1, causing this to be a very dominant cluster. The Davies-Bouldin index is also worse for this cluster, which makes sense since the mean of the cluster shifts away from the UD data as the cluster grows.

What follows is a comparison of the feature histograms per cluster. In these plots, the histograms are normalized so the integral of each histograms equals to 1. This is in order to be able to compare clusters with a large difference in datapoint count. The histograms of the K-means method are left out since they are already covered by the more diverse Incremental K-means result. In fig. 6.34 and fig. 6.35 the histogram per variable of the four clusters are shown. The biggest differences between cluster 1 and cluster 2 can be seen in WP2, WP4, WP7 and WP8. Cluster 1 contains generally more relative energy in the lower frequency packets WP2 and WP4 and cluster 2 contains more datapoints with most of the energy in WP7 and 8. So cluster 1 contains a lot energy in the frequency range 118-251kHz, which corresponds to matrix cracking damage. Cluster 3 and 4 are also unique in that they contain most of the energy in WP7 and WP8, respectively. The result for the Adaptive Incremental K-means, is less interesting since most of the datapoints are allocated to cluster 1. The histograms per feature for cluster 1-3 of the Adaptive Incremental K-means result is shown in fig. 6.36 and fig. 6.37. Cluster 1 again contains most energy in WP2 and WP4. Cluster 2 shows a lot of variation, due to the low amount of datapoints, but contains a lot of points that have a lot of energy from WP8. Cluster 3 is a bit more consistent and in general has most energy from WP7.

Table 6.5: The cluster count, Davies-Bouldin indices and computation time required for each datapoint for the different clustering methods on quasi static test CP5

Clustering method	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Davies-Bouldin indices**	**Computation time [hits/s]**
K-means	9833	4033	-	-	1. 1.52 2. 0.64	10069
Incremental K-means	10866	2233	174	592	1. 1.97 2. 0.58 3. 0.32 4. 0.47	1843
Adaptive Incremental K-means	13683	14	169	-	1. 1.28 2. 0.41 2. 0.51	1835

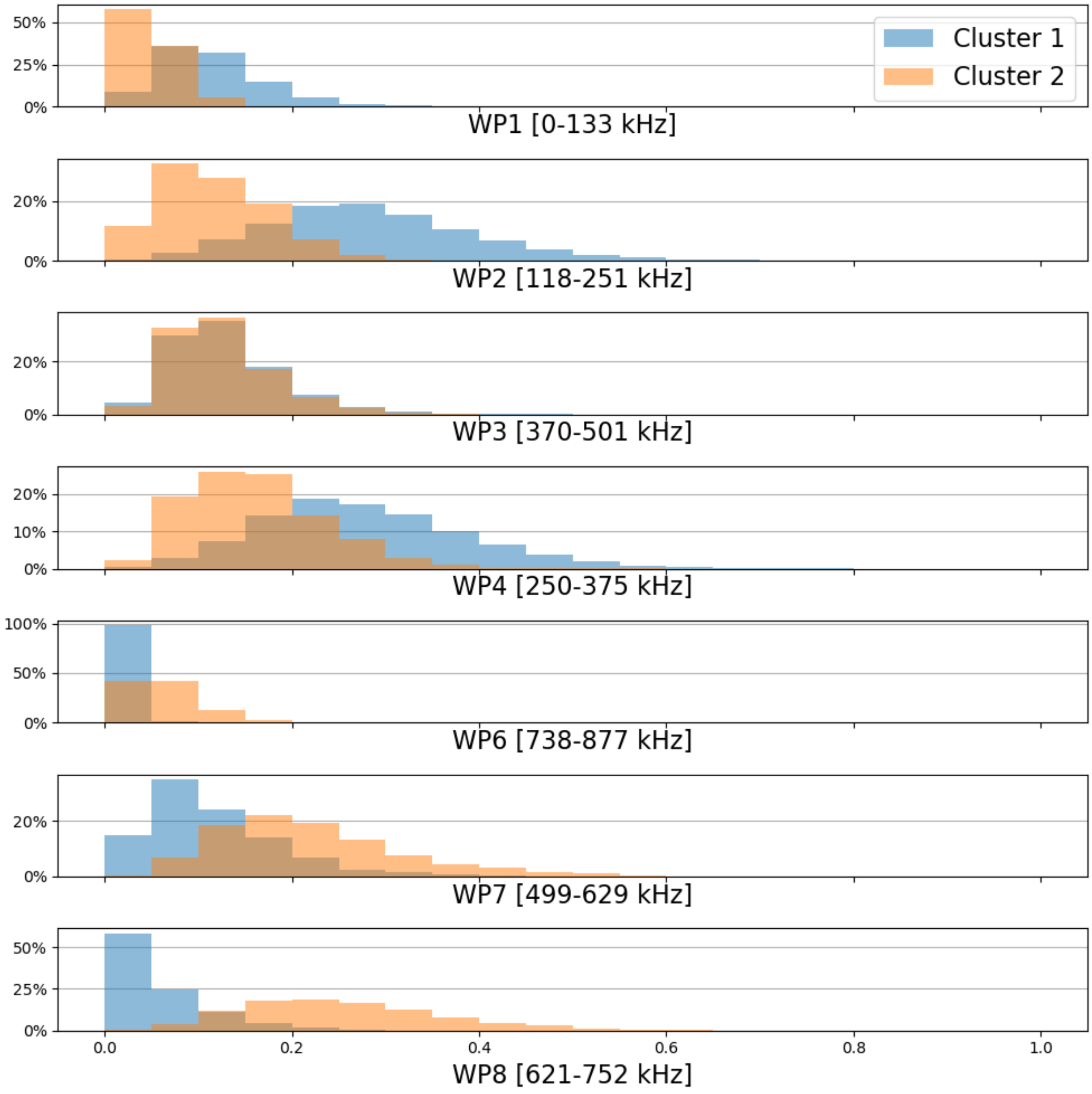


Figure 6.34: Feature histograms of the Incremental K-means on static test CP5: cluster 1 and cluster 2

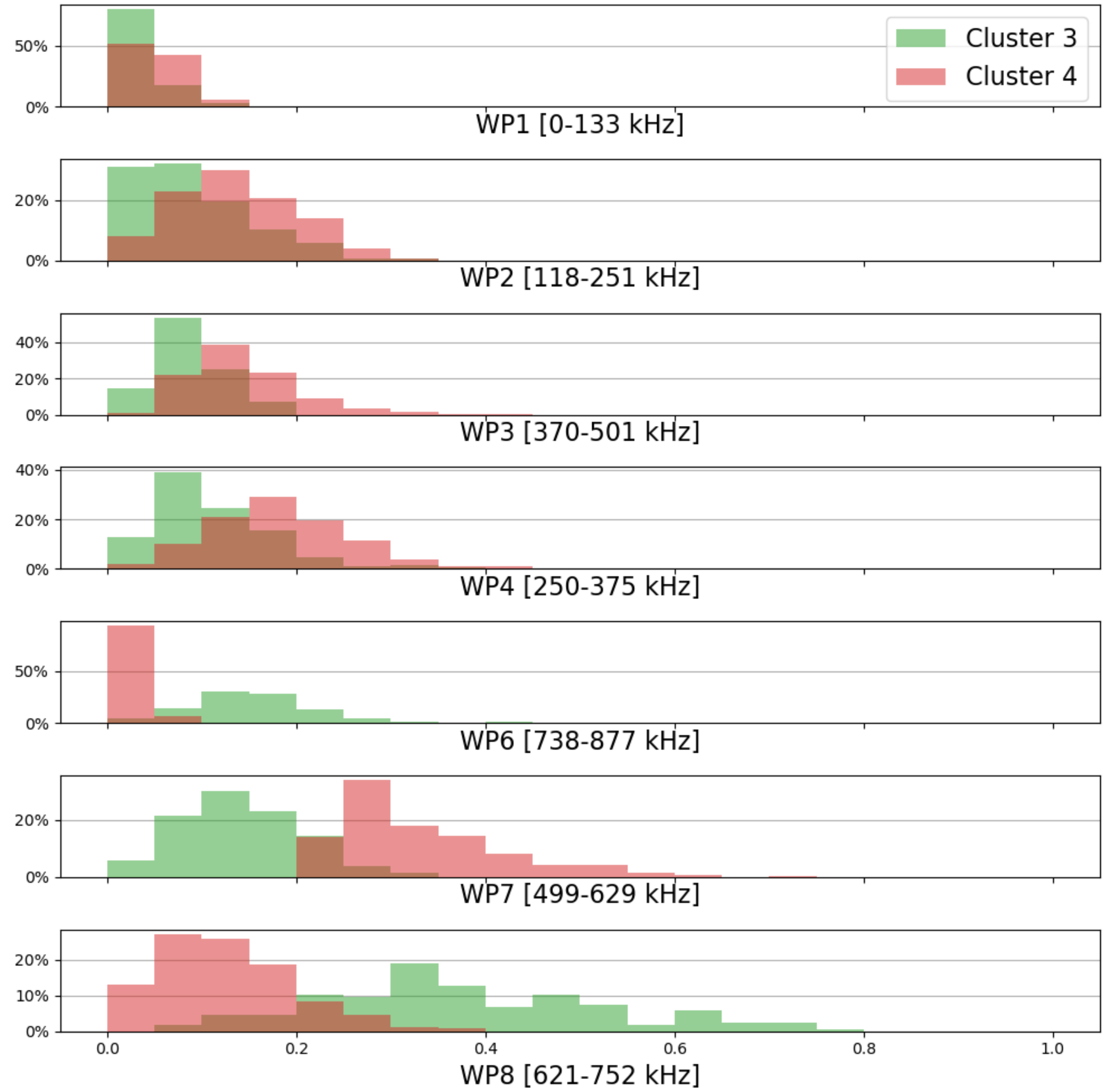


Figure 6.35: Feature histograms of the Incremental K-means on static test CP5: cluster 3 and cluster 4

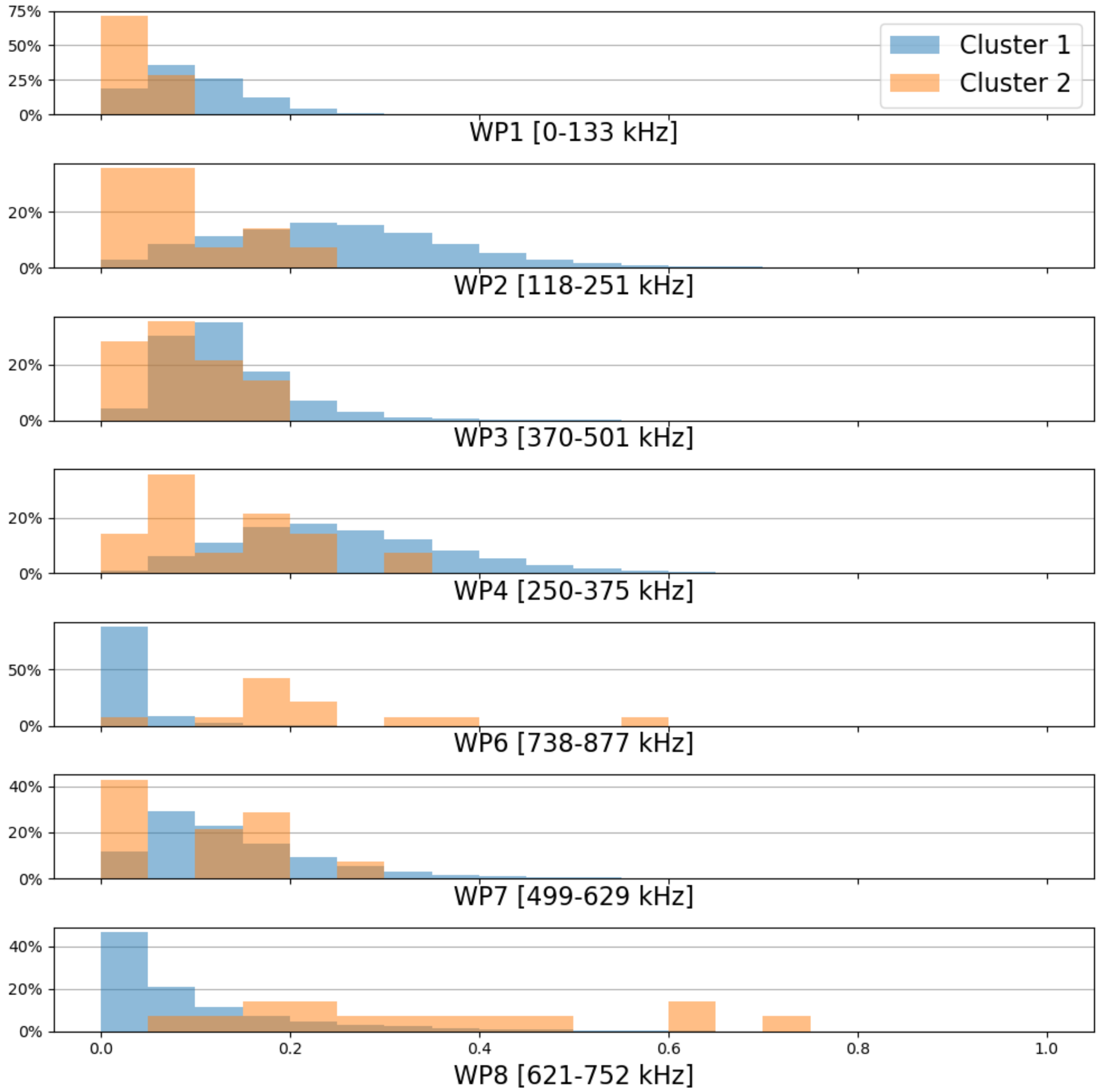


Figure 6.36: Feature histograms of the Adaptive Incremental K-means on static test CP5: cluster 1 and cluster 2

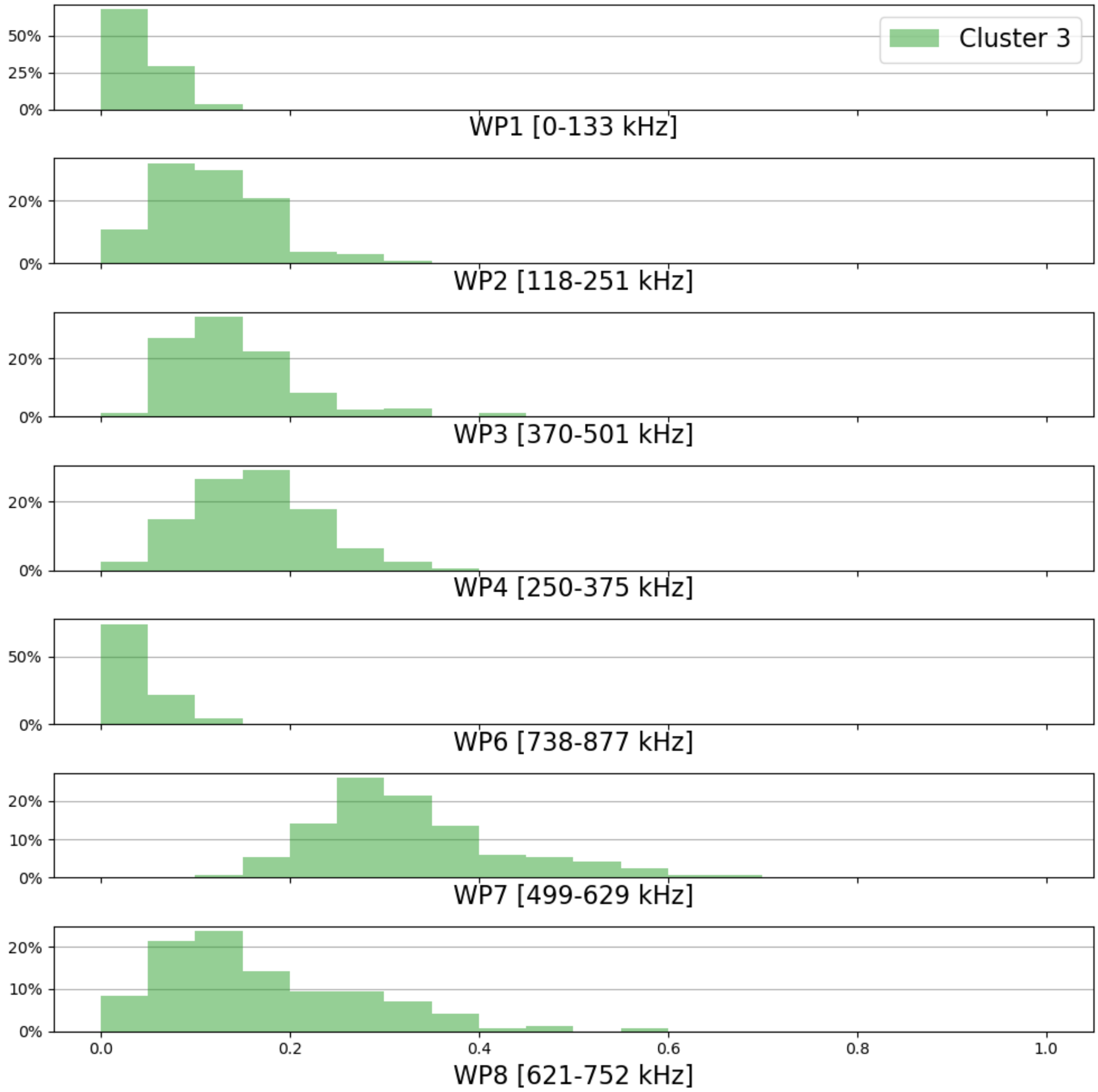


Figure 6.37: Feature histograms of the Adaptive Incremental K-means on static test CP5: cluster 3

6.4.2. Fatigue test CP14

The cluster count, Davies-Bouldin indices and computation time required for each datapoint for Incremental K-means and Adaptive Incremental K-means used on the fatigue test CP14 is summarized in table 6.6. Again, the K-means method is not suitable for fatigue test data, since the method has time complexity $O(n^2)$. This means the time increases quadratically as the number of datapoints, n, increases. The difference between the two methods is quite subtle in this case. Cluster 2 has been allocated more points with the Adaptive Incremental K-means and there are only three clusters with this method. What is more interesting is that the methods have a different cluster with the highest Davies-Bouldin index, even though the clusters did not change very much with respect to the two methods. So both clusters correlate well with the UD data and it is hard to make a clear distinction between. It could be that in reality both clusters originate from the matrix cracking damage mechanism.

The difference in computation time for the Incremental K-means and Adaptive Incremental K-means is small. The incremental K-means processes 52 datapoints per second, which means it takes 9,8 hours to cluster all data points. For the Adaptive Incremental K-means process this leads up to 10 hours. The test of 200,000 cycles at a frequency of 5 Hz, without taking pictures would have taken 11.1 hours. This means that in case the data comes in gradually, the algorithms can keep up with the incoming data. It has to be taken into account that this is however without the conversion of raw signal data into the wavelet packet features. On the other hand, there is still a lot of room for optimization in the algorithms. The analytical Python algorithm that was used to generate the results, could be written in program tailored to only do the clustering analysis, perhaps also in a more low level programming language like C++.

Since the number of datapoints per cluster are similar, the histograms will also likely look the same. The histograms for the wavelet packet features for the Incremental K-means and the Adaptive Incremental K-means are shown in fig. 6.38-fig. 6.39 and fig. 6.40-fig. 6.41, respectively. For cluster 1 and cluster 2 it is very difficult to spot any visual differences. Cluster 3 is also somewhat similar but differences can be spotted due to the low number of datapoints for the Adaptive Incremental K-means method. For cluster 1 and cluster 2 of both methods, the features have quite some overlap, but also significantly different ranges. Cluster 2 covers more area in the higher frequencies as visible in WP4 and WP7. It seems that cluster 2 is formed by matrix cracking and possible also delamination or fiber breakage. It is also important to keep in mind here that the size of the clusters was set with the UD data. It makes sense that for a different layup and more importantly, loading condition, the cluster size would also be different.

Table 6.6: The cluster count, Davies-Bouldin indices and computation time required for each datapoint for the different clustering methods on fatigue test CP14

Clustering method	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Davies-Bouldin indices**	**Computation time [hits/s]**
Incremental K-means	1769190	64246	174	10	1. 1.84 2. 1.15 2. 0.53 2. 0.29	52
Adaptive Incremental K-means	1747914	85676	30	-	1. 1.79 2. 1.32 2. 0.46	51

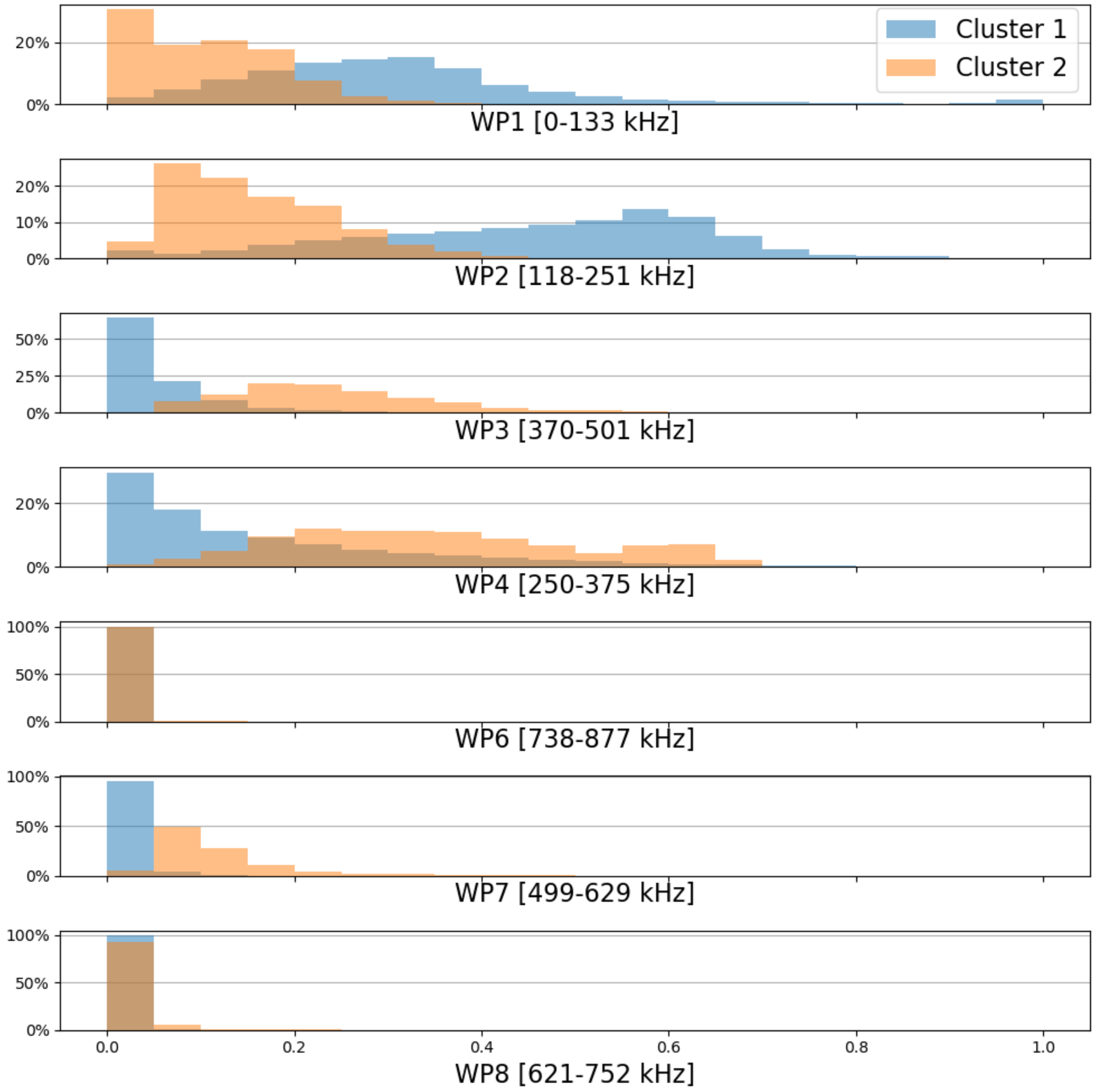


Figure 6.38: Feature histograms of the Incremental K-means on fatigue test CP14: cluster 1 and cluster 2

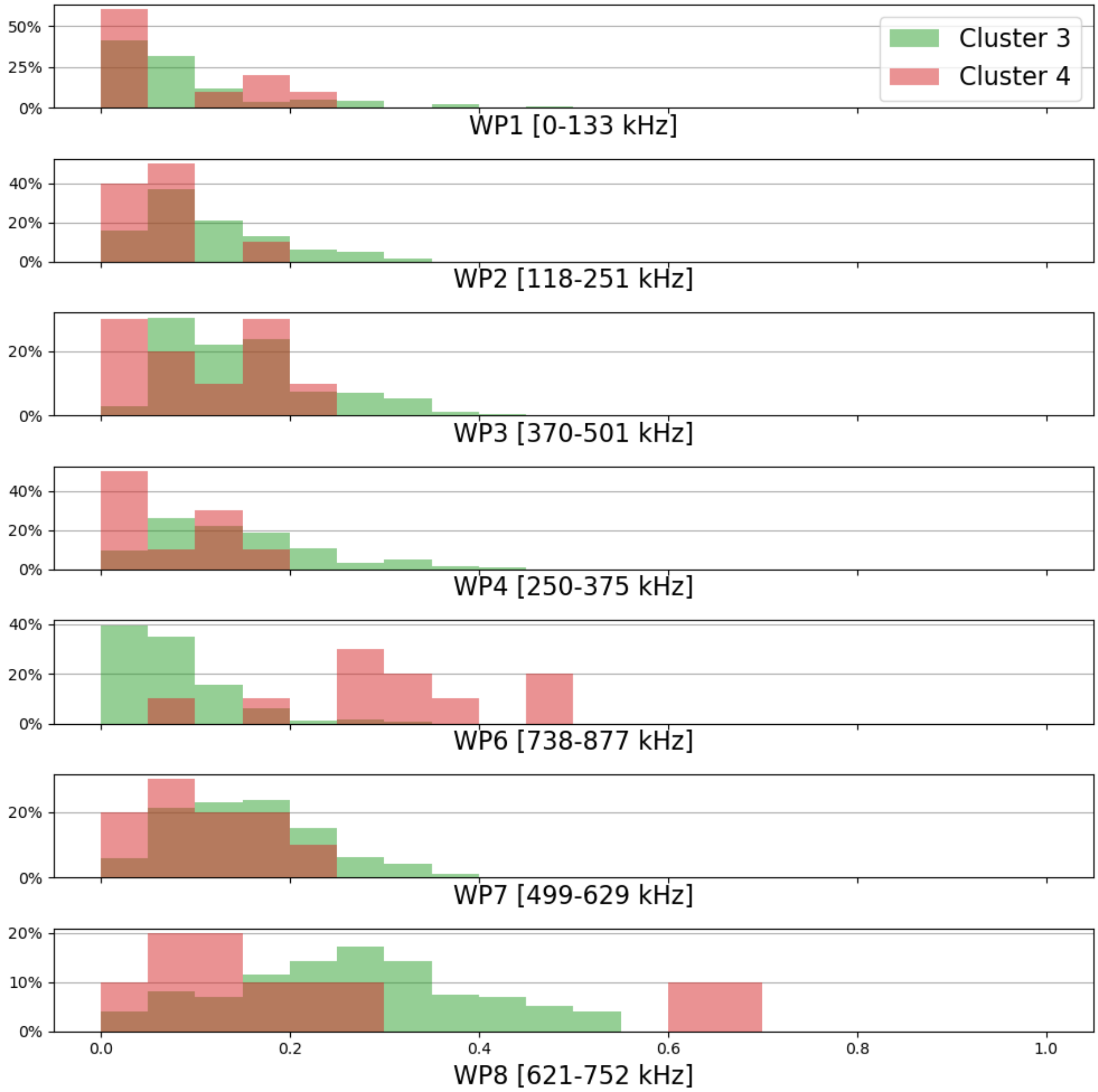


Figure 6.39: Feature histograms of the Incremental K-means on fatigue test CP14: cluster 3 and cluster 4

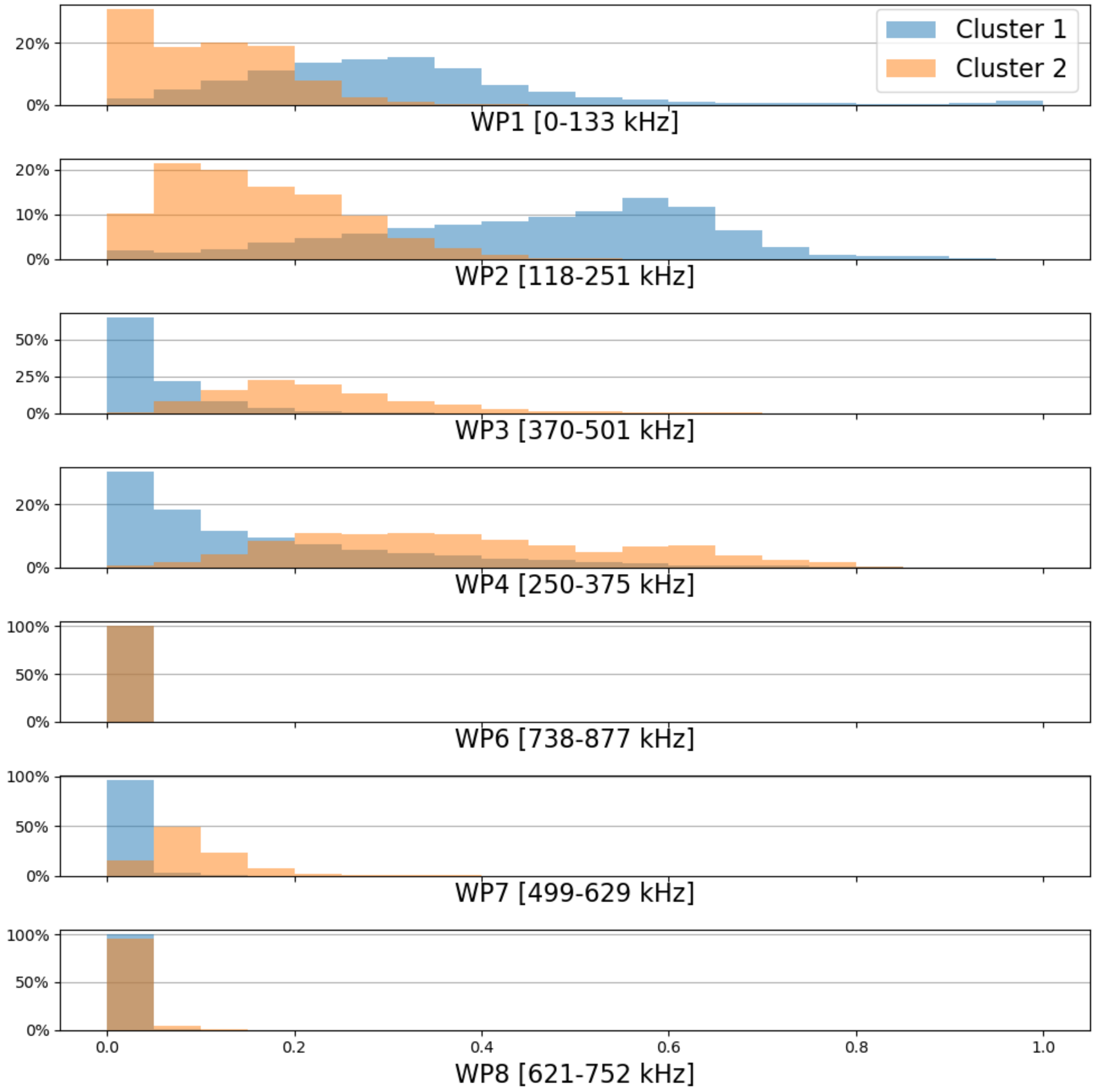


Figure 6.40: Feature histograms of the Adaptive Incremental K-means on fatigue test CP14: cluster 1 and cluster 2

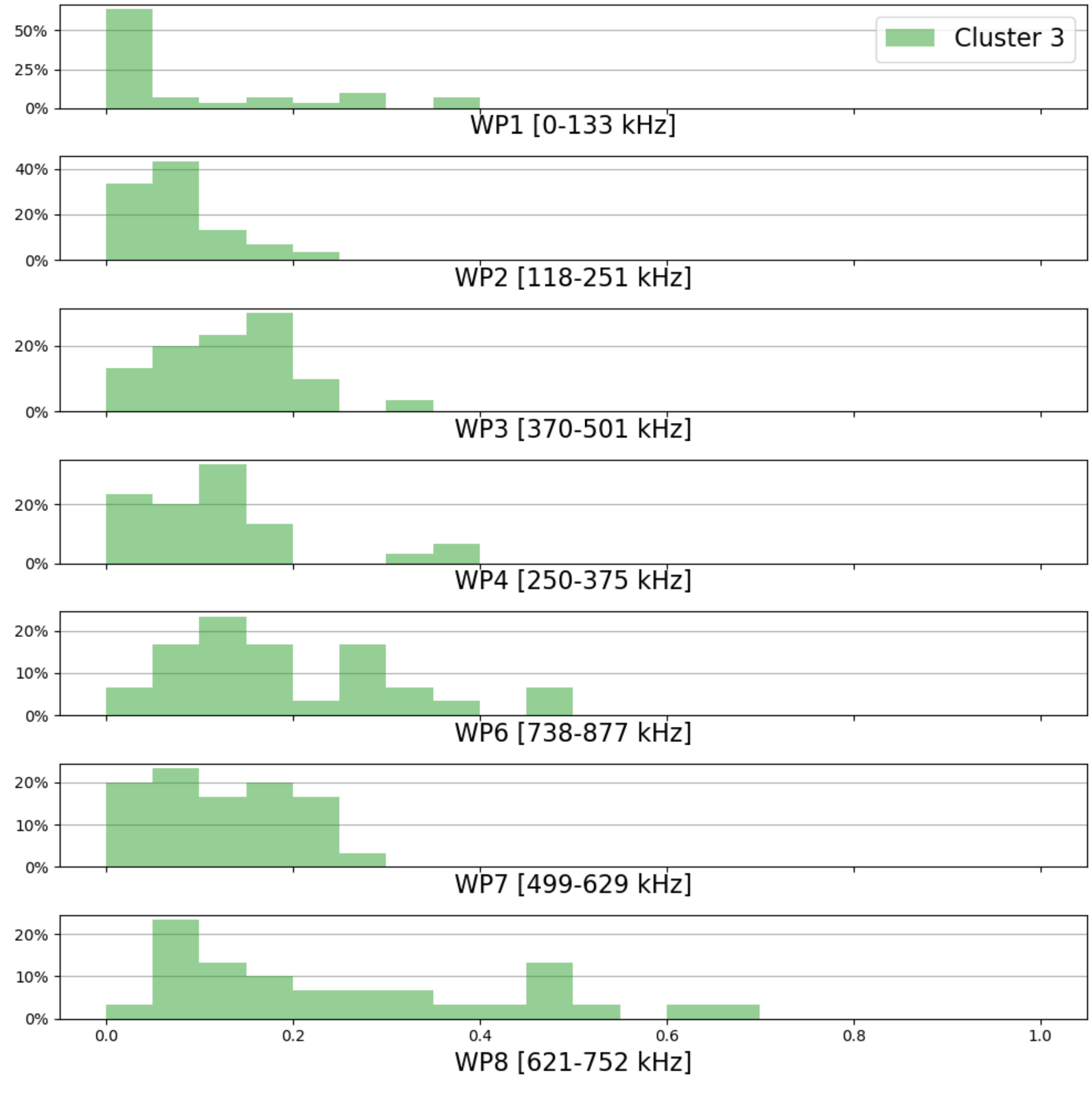
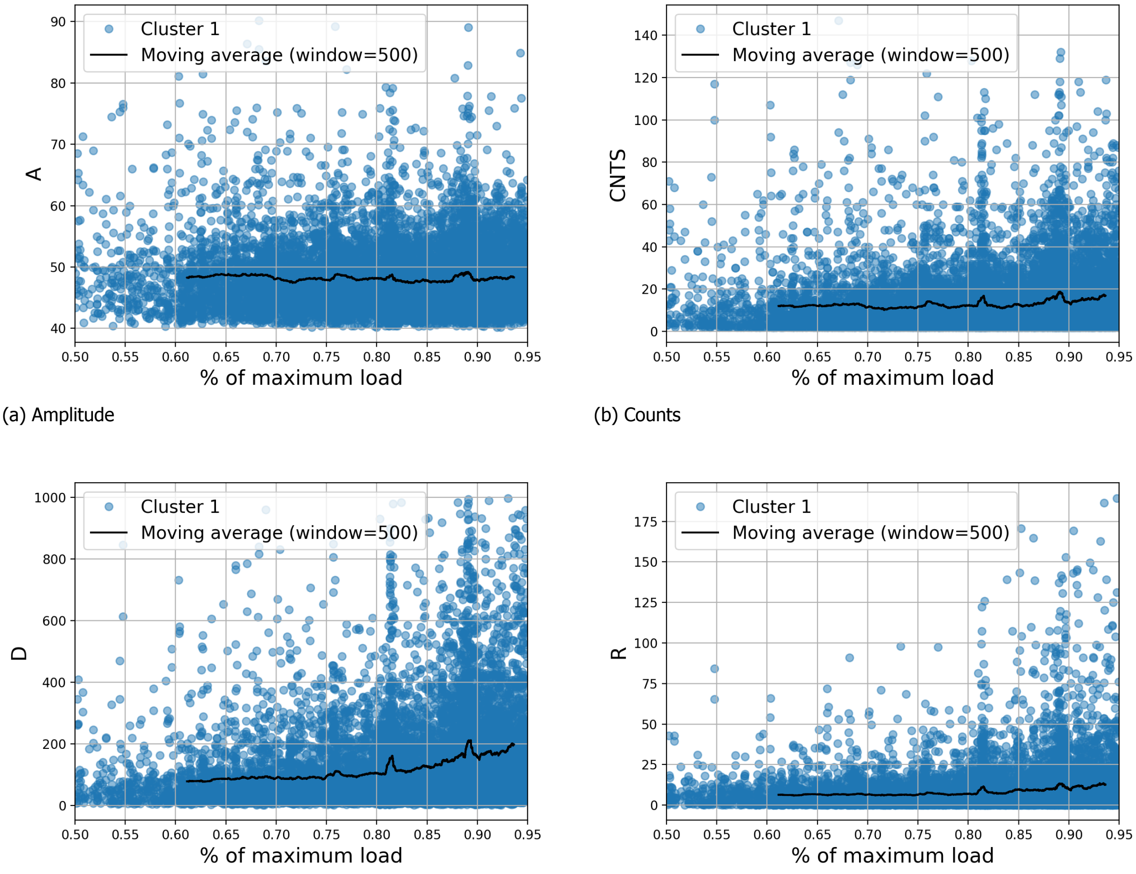


Figure 6.41: Feature histograms of the Adaptive Incremental K-means on fatigue test CP14: cluster 3

6.5. Evolution of matrix cracking signals

In this section the change in matrix cracking signals as a test progresses will be observed. The data used for this analysis will be from the static test CP5. More specifically, the cluster that was most related to the matrix cracking data from the Incremental K-means method. Any conclusions that are drawn from this are under the assumption that the data of this cluster is in reality caused by the matrix cracking damage mechanism. The amplitude, counts, duration and rise time for cluster 1 of the Incremental K-means is shown in fig. 6.42. A moving average with a window of 500 datapoints is also plotted to see the actual progression of the feature. Trials were also performed with a window of 100 and 250, but these lines simply contained too much noise to be able to find any pattern. The amplitude in fig. 6.42(a) remains pretty much constant apart from a few peaks. The number of counts and the rise time show a very slight increase towards the end of the test, visible in fig. 6.42(b) and fig. 6.42(d). The duration, fig. 6.42(c), of the signal shows a more clear increase after 80% of the maximum load. These changes could be due to the fact that damage develops, which causes the medium to change. What is most interesting in these figures is that there are clearly three simultaneous peaks occurring in all plots. These might be related to specific events happening in the test, like large matrix cracks. It could also be that this is from a more rare damage mechanism like fibers breaking that are classified as in this case matrix cracks.

The wavelet packets 1-4 and 6-8 for cluster 1 of the Incremental K-means is shown in fig. 6.43. Laying these next to the previous plots, one can observe the three same peaks in some of the wavelet packet energy percentages. WP1 and WP3 show no observable change during the load in fig. 6.43(a) and fig. 6.43(c), respectively. WP2 in fig. 6.43(b) shows a step after the peak around 80% of the maximum load, which corresponds to the increase in duration from fig. 6.42(c). The other packet that is clearly increasing over time is WP4, fig. 6.43(d). These packets also have a frequency range very close to each other. If some packets show an increase in energy percentages, there has to be a decrease in other packets. This can be seen slighlty in WP6, fig. 6.43(e), and WP8, fig. 6.43(g), but the most significant decrease is seen WP7, fig. 6.43(f). So the most important take away from this is that there is an increase in energy content coming from the frequency range 118-251kHz and 250-375kHz. There could be two reasons for this depending on the clustering performance. Either the clustering process is correct, and the change in medium causes a slight change in frequency content and duration of the signal. Or in a worse scenario the clustering process fails to fully separate the signal and some of the signals coming from another mechanism than matrix cracking are classified as matrix cracking anyway.



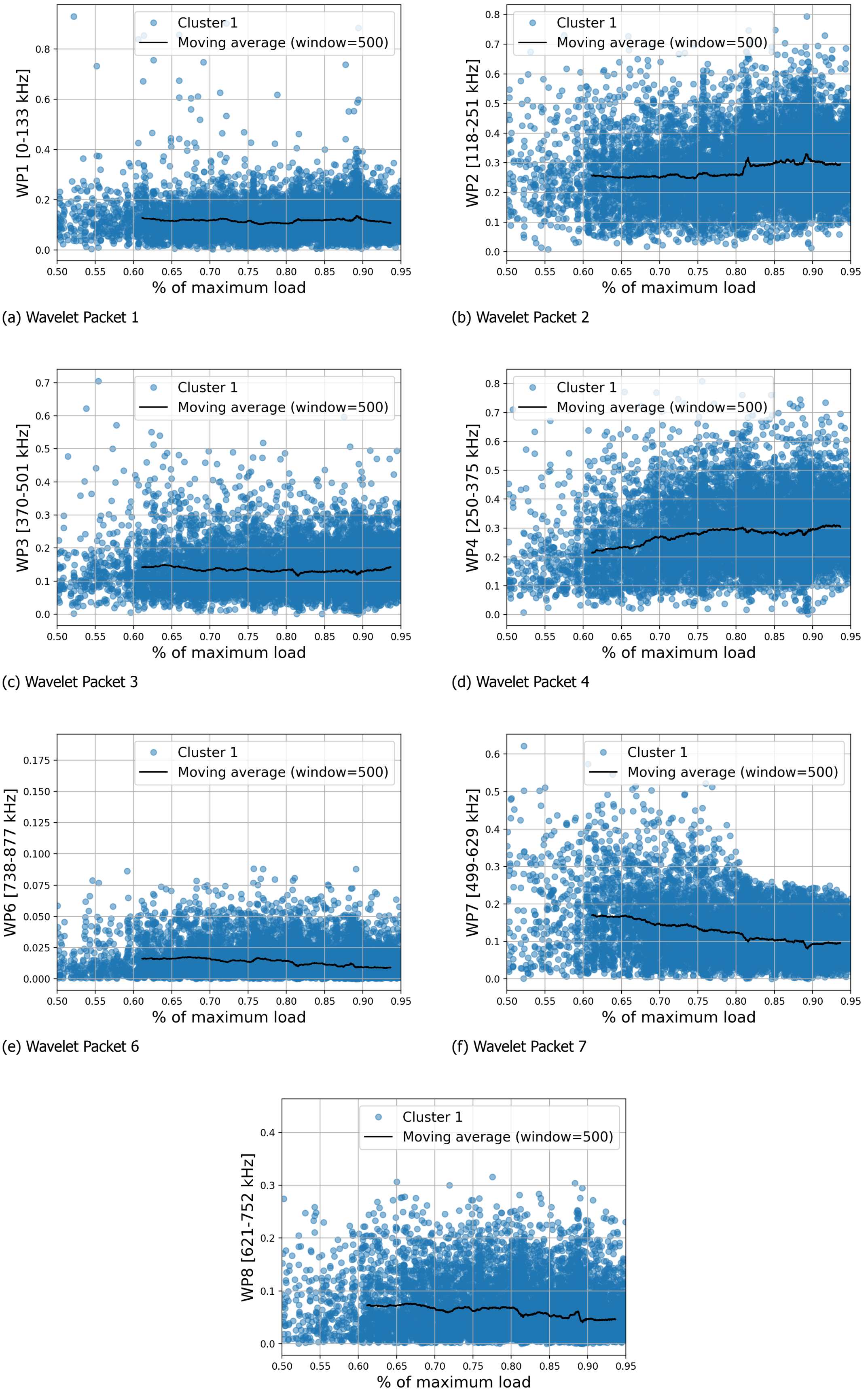
(a) Amplitude

(b) Counts

(c) Duration

(d) Rise time

Figure 6.42: Evolution of matrix cracking signals for basic AE features



(a) Wavelet Packet 1

(b) Wavelet Packet 2

(c) Wavelet Packet 3

(d) Wavelet Packet 4

(e) Wavelet Packet 6

(f) Wavelet Packet 7

(g) Wavelet Packet 8

Figure 6.43: Evolution of matrix cracking signals for wavelet packet relative energy percentages

7 Conclusions and recommendations

This chapter describes the main conclusions that answer the main research questions supported by the results. After the conclusion follow some recommendations that can help in further research regarding the topic of clustering and classifying damage mechanisms based on AE.

7.1. Conclusion

The goal of this project was to create a system that could classify damage mechanisms real-time in composites using AE signals. A set of algorithms has been developed, which together with priorly measured matrix cracking signals was able to separate signals belonging to this damage mechanism and signals coming from other sources like delamination or fiber breakage.

It is known when a carbon fiber reinforced polymer coupon with unidirectional 90°layup is loaded under tension, matrix cracks develop. This in general eventually leads up to failure by a complete transverse crack. A coupon with the same material with a crossply layup also develops damage from delamination and fiber breakage, when loaded in tension. Matrix cracking is often a precursor of delamination due to stress concentrations. Delamination in turn also causes stress concentrations and therefore acts as a precursor for matrix cracks and fiber breakage.

For each AE hit, basic parameters are recorded: amplitude, duration, rise time, counts, and energy. These features are combined to generate more usable features. However, previous studies showed more relevant information for classifying signals can be found in the frequency domain. The waveform of the AE signal is analyzed to determine the peak, average, rise and decay frequency. Next to this, the wavelet transform is applied to the waveform to determine the energy contents for different frequency ranges. It was found that the Daubechies 10 wavelet performs best for matrix cracking signals based on the energy to entropy ratio of the decomposition and the computation time. The analysis of different wavelet types showed that it was able to process a 1000 signals under a second, which is more than sufficient for any real-time AE analysis. Wavelet packet transform up to level 3 is used to decompose the signal into octaves. The energy percentage of each decomposition is determined, which function as features.

Two processing steps are required before the AE signals can be clustered and classified. First, the data has to be standardized. After that, all features have a mean of 0 and a standard deviation of 1, thus each feature contributes to subsequent steps equally.

The second step is to reduce the dimensionality of the data. The data is reduced to two dimensions in order to visualize the results and to ensure that the clustering process works properly. Skipping this step would lead to a lot of small isolated clusters, due to the fact that distance metrics do not work well in higher dimensional space. Principal Component Analysis is used for this purpose.

A total of 30 features of the AE signals were obtained. A correlation matrix was created and three variance metrics are used to create a score for each feature. If two features have a Pearson correlation coefficient higher than 0.8, the feature with the lowest variance score is eliminated. This resulted in a ranking of features ordered by their variance score. A high variance score indicates a large spread of data, which is beneficial for clustering the data.

The signals of the unidirectional specimens and a crossply specimen were analyzed to identify the most significant differences. The most obvious differences were found in the wavelet packet energy percentages. More specifically, packets with peak frequency ranges of 118-251kHz, 250-375kHz, 499-629kHz and 621-752kHz. The reason for this is that matrix cracking signals occur at the lower frequency ranges, and delamination and fiber breakage at the higher frequency ranges. The actual influence of the layup on the AE signals proved to be difficult due to the low amount of signals emerging from the unidirectional tests.

A set of trials was performed to find the optimal set of features for further clustering analysis. The optimal feature set that characterized the signals in an optimal way, was found to be the wavelet packet energy percentages, median frequency, rise time over duration, average frequency and amplitude. The cluster analysis is performed using the K-means method, for a total of 5 quasi static tests. For all specimens the optimal number was found to be 2, where all of the specimens showed that a single cluster corresponded very well with the matrix cracking signals. The clustered data was remapped to basic AE features to see if they are distinguishable, but this was not the case. A second set of trials was performed with an adaptation to make the transition to the real-time clustering process. The adaptation is that the data is standardized with the data from the unidirectional tests. This ensures smooth data processing and prevents a lot of computation time during the clustering process. The optimal feature set for these trials was found to be the wavelet packet energy percentages only. The cluster analysis resulted in one result with three clusters, and the rest with two clusters. Again, there was a clear distinction between the cluster that corresponded with the matrix cracking signals and the cluster(s) that did not. A remapping to basic AE features, showed some differences between clusters, but insufficient for separation. All in all, the wavelet packet features proved to be successful for clustering the AE data and correlating a cluster to matrix cracking, regardless of the standardization process.

A real-time clustering algorithm was developed, derived from the post processing version. The features were however processed beforehand for the ease of repeatability and analysis. The Principal Component Analysis is replaced with Incremental Principal Component Analysis. This method can reduce the dimensionality of data as well, by taking in batches of data. Two real-time clustering methods were used and analyzed: Incremental K-means and Adaptive Incremental K-means. For the quasi static tests, Incremental K-means lead to similar results as K-means. The method proved to be successful in clustering and classifying the data in real-time. It was able to correlate incoming data to the matrix cracking data, with only a few seconds of computation time. The Adaptive Incremental K-means was contrary to the other methods not able to cluster the data appropriately. All data was allocated to a single cluster, due to the merging conditions of this method. It was able to find the cluster that correlated most to the matrix cracking data, but this was not very insightful.

The incremental K-means and Adaptive Incremental K-means were also applied to the data of a fatigue test. For this data the methods performed similarly. Both methods resulted in one dominant cluster that was most correlated to the matrix cracking data. Since the fatigue test was performed at cycles of 70% of the load, it was expected that matrix cracking would be the dominant source of AE. However, both clustering algorithms did not work optimally. There was another cluster that had some overlap with the matrix cracking signals, which was also found by the similarity metrics. The enormous amount of data resulting from this test, as well as the difference in layup and loading condition with the unidirectional test are most likely the cause of this misperformance. Regarding the time-wise performance, both algorithms showed that they were able to keep up with the incoming data, with a slight edge. Again, the wavelet packet features proved to be characteristic for the signals. This was illustrated in an in-depth analysis which showed that each cluster had a distinctive distribution in some of the wavelet packet features. Why the Adaptive Incremental K-means worked for the fatigue case and not for the quasi static case remains an interesting point of discussion. It certainly has to do with the calculation of the parameter d, which is the only parameter that has to be determined for this method. The known matrix cracking signals are used in the calculation of this parameter, where it is assumed that these signals are normally distributed. If this is a false assumption it could be responsible for the poor performance. It is important to realize that the loading condition is different for the fatigue case, causing different AE signals since the damage progression is slower. The way that the clusters develop also strongly influence the merging process. If there remains a sufficient distance between the clusters, they will not be merged which is what happened with the fatigue case.

Overall, the real-time clustering algorithms have proven to be successful in clustering and classifying incoming AE signals in an efficient way. With room for optimization the proposed methods seem fit to be applied in real-time applications. Before this can be done however, more testing and validation of these methods is required.

At last, the clustered data from a quasi static test processed with the Incremental K-means was analyzed. The cluster that correlated most to the matrix cracking data was tracked over time to see how the signals evolved. This highlighted specific events during the test and also showed gradual changes towards the end of the test. The main parameters that were influenced were the duration of the signal, and some wavelet packets of specific frequencies.

7.2. Recommendations

The most valuable addition to this study would be a proper validation of the clustering results. Although Digital Image Correlation was performed during the test, a number of practical issues lead to the inability of using this analysis to provide a validation. Also, another use of a camera would be to aim it at the side of the specimen in order to track the matrix cracks. Software like ImageJ could then be used to track and localize the matrix cracks from high resolution pictures. This in combination with proper localization of the AE data can be used to obtain more information about the relation between the matrix cracking damage and signals.

Another way to validate the clustering results would be to stop in case of the quasi static test after 95% of the load and perform a C-scan after this. Similarly, this would also be useful for analysing the damage in the fatigue tests. Performing a C-scan before the tests would also provide useful insights in the difference in response of the different specimens.

Another important addition to the research could be to do more tests with different layups. A specimen with -45/+45 layers when loaded under tension leads to matrix cracking and delamination between the different layups. In combination with the matrix cracking data from the unidirectional 90 specimens can provide more insights in the difference in AE signal caused by different damage mechanisms. Similarly, a specimen with a unidirectional 0 layers leads to fiber breakage, matrix cracking and longitudinal splits. This would all give insight in how the signals from different mechanisms differ and might also show how the layup affects the AE signals.

Lastly, there is still a lot of room for optimization of the algorithms. Apart from switching to a more efficient programming language, the current code could also be optimized further. The evaluation metric that finds the correlation between the clusters and the matrix cracking data, the Davies-Bouldin index, is recalculated for each batch of signals. A better way would be the keep track of the input and change the metric to function more efficiently in a real-time sense. Also, the current bottleneck of the algorithm is that it is a function of the size of the input data. If the AE signals could be fed to the algorithm in real-time without having to save and read the datafiles, that would significantly improve the time-wise performance. Reading large datafiles currently takes more time than the actual analysis of data. This would already be reduced a lot if the waveform files can be loaded in batches as well. The Incremental Principal Component Analysis is currently also updated with batches up to a certain point. This can be made more data-driven and efficient by tracking the changes in the covariance matrix and setting a threshold after which the Principal Components are not updated anymore.

Bibliography

[1] Y. A. Dzenis and J. Qian, *Analysis of microdamage evolution histories in composites,* International Journal of Solids and Structures **38**, 1831 (2001).

[2] H. Wadley and J. Simmons, *Microscopic Origins of Acoustic Emission* (ASNT Handbook on Acoustic Emission, NDT Handbook Vol. 5, Section 3, Ed. P.M. McIntyre, 1987).

[3] F. Kishinouye, *An experiment on the progression of fracture (a preliminary report) [in japanese],* Jishin [Journal of the Seismological Society of Japan] **6**, 25 (1934).

[4] Y. Ech-Choudany, M. Assarar, F. M.-N. D. Scida, and B. Bellach, *Unsupervised clustering for building a learning database of acoustic emission signals to identify damage mechanisms in unidirectional laminates,* Applied Acoustics **123**, 123 (2017).

[5] Z. long Zhou, J. Zhou, L. jun Dong, X. Cai, Y. chao Rui, and C. tao Ke, *Experimental study on the location of an acoustic emission source considering refraction in different media,* Springer Nature **7** (2017).

[6] J.-P. Favre and J.-C. Laizet, *Acoustic emission analysis of the accumulation of cracks in cfrp cross-ply laminates under tensile loading,* Journal of Acoustic Emission **9**, 97–102 (1990).

[7] T.Bohmann, M.Schlamp, and I.Ehrlich, *Acoustic emission of material damages in glass fibre-reinforced plastics,* Composites Part B: Engineering **155**, 444 (2012).

[8] W. Roundi, A. E. Mahi, A. E. Gharad, and J.-L. Rebiere, *Acoustic emission monitoring of damage progression in glass/epoxy composites during static and fatigue tensile tests,* Applied Acoustics **132**, 124 (2018).

[9] E. Pomponi and A. Vinograov, *Real-time acoustic emission classification: New evolutionary approach,* European Conference on Acoustic Emission Testing & 7th International Conference on Acoustic Emission **30** (2012).

[10] T. Proctor, *An improved piezoelectric acoustic emission transducer,* Journal of the Acoustical Society of America **71**, 1163 (1982).

[11] R. K. Miller, E. v. K. Hill, and P. O. Moore, *Acoustic Emission Testing* (American Society for Nondestructive Testing, 2005).

[12] P. J. Groot, P. A.M.Wijnen, and R. B.F.Janssen, *Real-time frequency determination of acoustic emission for different fracture mechanisms in carbon/epoxy composites,* Composites Science and Technology **55**, 405 (1995).

[13] S. Goutianos, *Acoustic emission characteristics of unidirectional glass/epoxy composites under mixed-mode fracture,* SN Applied Sciences **1**, 474 (2019).

[14] R. Talreja and J. Varna, *Modeling Damage, Fatigue and Failure of Composite Materials* (Elsevier Ltd., 2016).

[15] E. K. Gamstedt and S. I. Andersen, *Fatigue Degradation and Failure of Rotating Composite Structures – Materials Characterisation and Underlying Mechanisms* (Risø National Laboratory, Roskilde, Denmark. Forskningscenter Risoe. Risoe-R, No. 1261(EN), 2001).

[16] R. Joffe, *Damage accumulation and sitffness degradation in composite laminates* (Doctoral Thesis. s.l.: Lulea University of Technology, 1999).

[17] S. S. Rao and B. Subramanyam, *Analysis of acoustic emission signals using wavelet transformation technique,* Defence Science Journal **58**, 559 (2008).

[18] L. Debnath and F. A. Shah, *Wavelet Transforms and Their Applications* (Springer Science+Business Media New York, 2015).

[19] A. K. Alexandridis, A. D. Zapranis, and A. D. Zapranis, *Wavelet Neural Networks: With Applications in Financial Engineering, Chaos, and Classification* (John Wiley & Sons, Incorporated, 2014).

[20] D. L. Fugal, *Conceptual Wavelets In Digital Signal Processing* (Space & Signals Technical Publishing, 2009).

[21] The PyWavelets Developers, *Discrete wavelet transform (dwt),* https://pywavelets.readthedocs.io/en/latest/ref/dwt-discrete-wavelet-transform.html (), accessed: 2019-10-15.

[22] The PyWavelets Developers, *Inverse discrete wavelet transform (idwt),* https://pywavelets.readthedocs.io/en/latest/ref/idwt-inverse-discrete-wavelet-transform.html (), accessed: 2019-10-15.

[23] A. Taspinar, *A guide for using the wavelet transform in machine learning,* http://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/, posted: 2018-12-21.

[24] A. Abbate, J. Koay, J. Frankel, S. C. Schroeder, and P. Das, *Signal detection and noise suppression using a wavelet transform signal processor: Application to ultrasonic flaw detection,* IEEE Transactions On Ultrasonics, Ferroelectrics, And Frequency Control **44** (1997).

[25] A. R. Oskouei, M. Ahmadi, and M. Hajikhani, *Wavelet-based acoustic emission characterization of damage mechanism in composite materials under mode i delamination at different interfaces,* eXPRESS Polymer Letters **3**, 804 (2009).

[26] C. Barile, C. Casavola, G. Pappalettera, and P. K. Vimalathithan, *Damage characterization in composite materials using acoustic emission signal-based and parameter-based data,* Composites Part B: Engineering **178**, 107469 (2019).

[27] S. K. Najafi, H. Sharifnia, M. A. Najafabadi, and E. Landis, *Acoustic emission characterization of failure mechanisms in oriented strand board using wavelet-based and unsupervised clustering methods,* Wood Science and Technology **51**, 1433 (2017).

[28] M. Misiti, Y. Misiti, G. Oppenheim, and J. M. Poggi, *Wavelet toolbox for matLab* (The MathWorks Inc, 1997).

[29] A. Marec, J.-H. Thomas, and R. Guerjouma, *Damage characterization of polymer-based composite materials: Multivariable analysis and wavelet transform for clustering acoustic emission data,* Mechanical Systems and Signal Processing **2**, 1441 (2008).

[30] T. Loutas, V. Kostopoulos, C. Ramirez-Jimenez, and M. Pharaoh, *Damage evolution in center-holed glass/polyester composites under quasi-static loading using time/frequency analysis of acoustic emission monitored waveforms,* Composites Science and Technology **66**, 1366 (2006).

[31] R. R. Coifman and M. V. Wickerhauser, *Entropy-based algorithms for best basis selection,* IEEE Transactions On Information Theory **38** (1992).

[32] D. Strömbergsson, P. Marklund, K. Berglund, J. Saari, and A. Thomson, *Mother wavelet selection in the discrete wavelet transform for condition monitoring of wind turbine drivetrain bearings,* Wind Energy **22**, 1581 (2019).

[33] A. Hinneburg and D. A. Keim, *Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering,* Proceedings of the 25th International Conference on Very Large Databases , 506 (1999).

[34] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, *Incremental learning for robust visual tracking,* International Journal of Computer Vision **77** (2008).

[35] H. Saleh, *Machine Learning Fundamentals* (Packt Publishing, 2018).

[36] T. Kanungo, *An efficient k-means clustering algorithm: Analysis and implementation,* IEEE transactions on pattern analysis and machine intelligence **24** (2002).

[37] M. Parsian, *Data Algorithms* (O'Reilly Media, Inc., 2015).

[38] E. Pomponi and A. Vinograov, *A real-time approach to acoustic emission clustering,* Mechanical Systems and Signal Processing **40**, 791 (2013).

[39] J. MacQueen, *Some methods for classification and analysis of multivariate observations,* Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability , 281 (1967).

[40] T. Ott, A. Kern, W.-H. Steeb, and R. Stoop, *Sequential clustering: tracking down themost natural clusters,* Journal of Statistical Mechanics: Theory and Experiment **11**, P11014 (2005).

[41] C. Manning, P. Raghavan, and H. Schutze, *Introduction to information retreival* (Cambridge University Press, New York, 2008).

[42] D. L. Davies and D. W. Bouldin, *A cluster separation measure,* IEEE Transactions on pattern analysis and machine intelligence **1** (1979).

[43] P. J. Rousseeuw, *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,* Journal of Computational and Applied Mathematics **20**, 53 (1987).

[44] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis* (Wiley Interscience, New York, 1990).

[45] J. Dunn, *Well separated clusters and optimal fuzzy-partitions,* Journal of Cybernetics **4**, 224 (1974).

[46] Hexcel Corporation, *HexPly 6376 datasheet,* https://www.hexcel.com/user_area/content_media/raw/HexPly_6376_eu_DataSheet.pdf, (2016), online; accessed 30 December 2019.

[47] A. Hosoi and H. Kawada, *Fatigue life prediction for transverse crack initiation of cfrp cross-ply and quasi-isotropic laminates,* Materials **11**, 1182 (2018).

[48] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way* (Academic Press, 2008).

[49] M. Bonamente, *Random Variables and Their Distributions. In: Statistics and Analysis of Scientific Data. Graduate Texts in Physics.* (Springer, New York, NY, 2017).

[50] A. G. Asuero, A. Sayago, and A. G. González, *The correlation coefficient: An overview,* Critical Reviews in Analytical Chemistry **36** (2007).

[51] J. SCohen, *Statistical Power Analysis for the Behavioral Sciences* (Lawrence Eribaum Associates, 1988).

A

Data Histograms

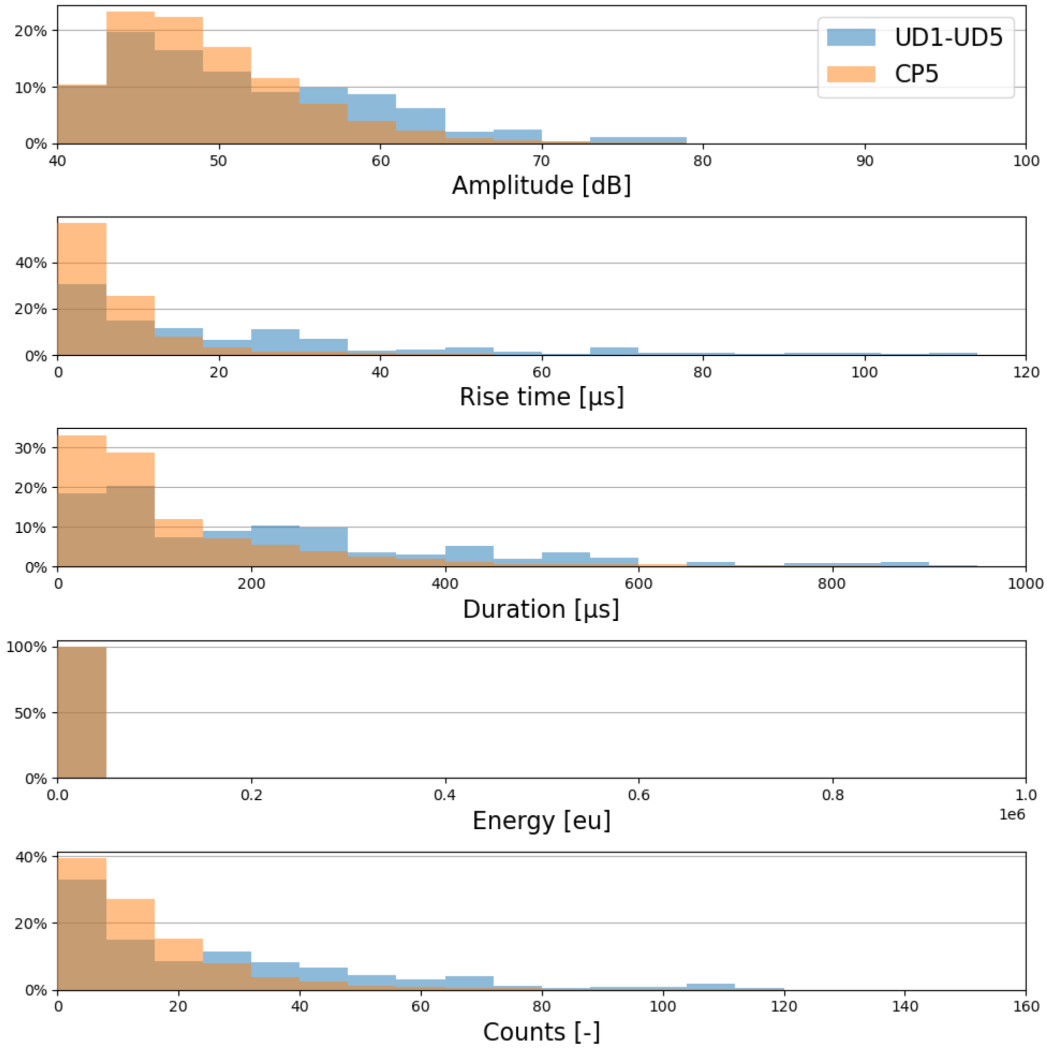


Figure A.1: Histogram of basic AE features for UD1-UD5 and CP5 data

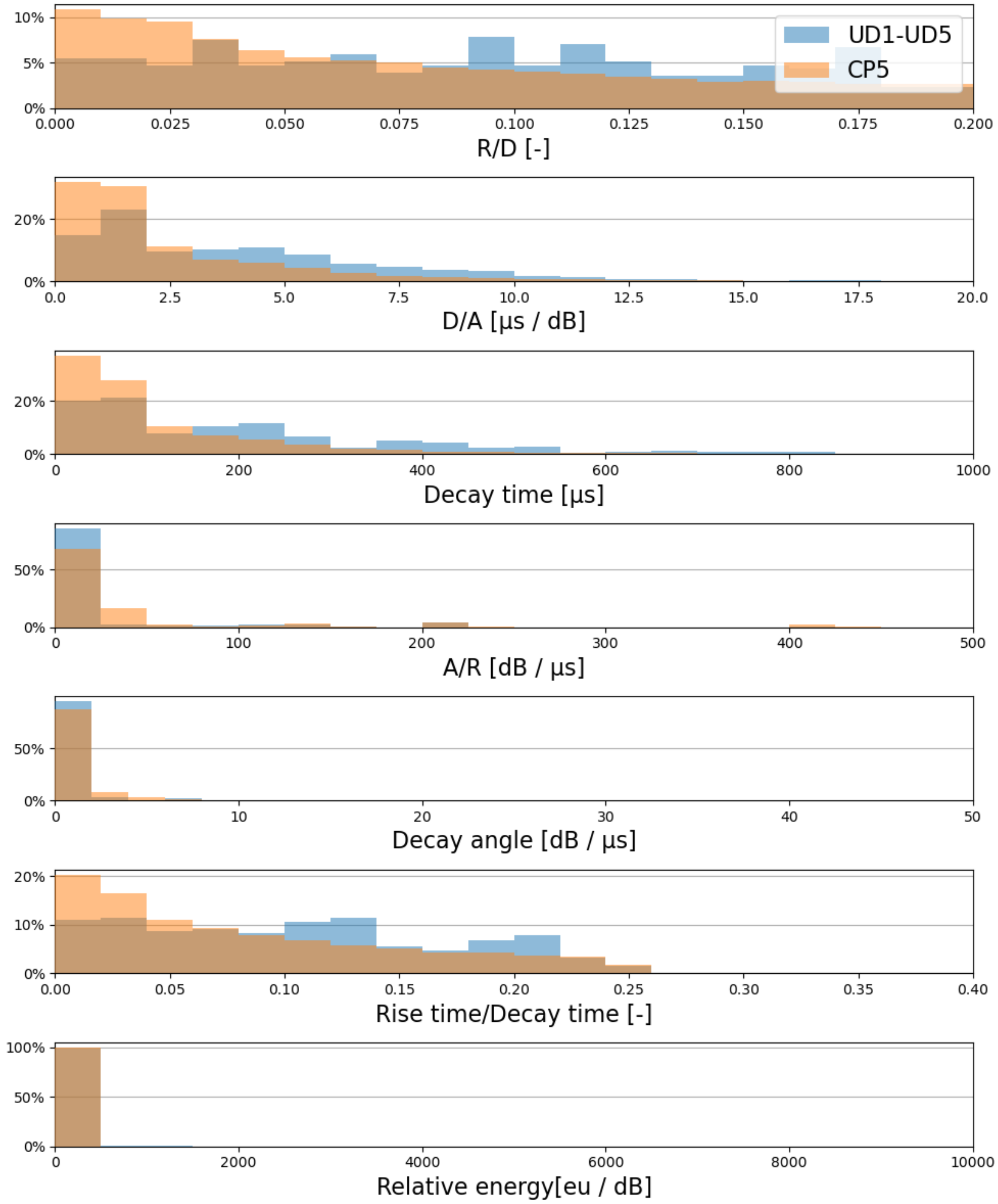


Figure A.2: Histogram of features calculated with basic AE features, for UD1-UD5 and CP5 data

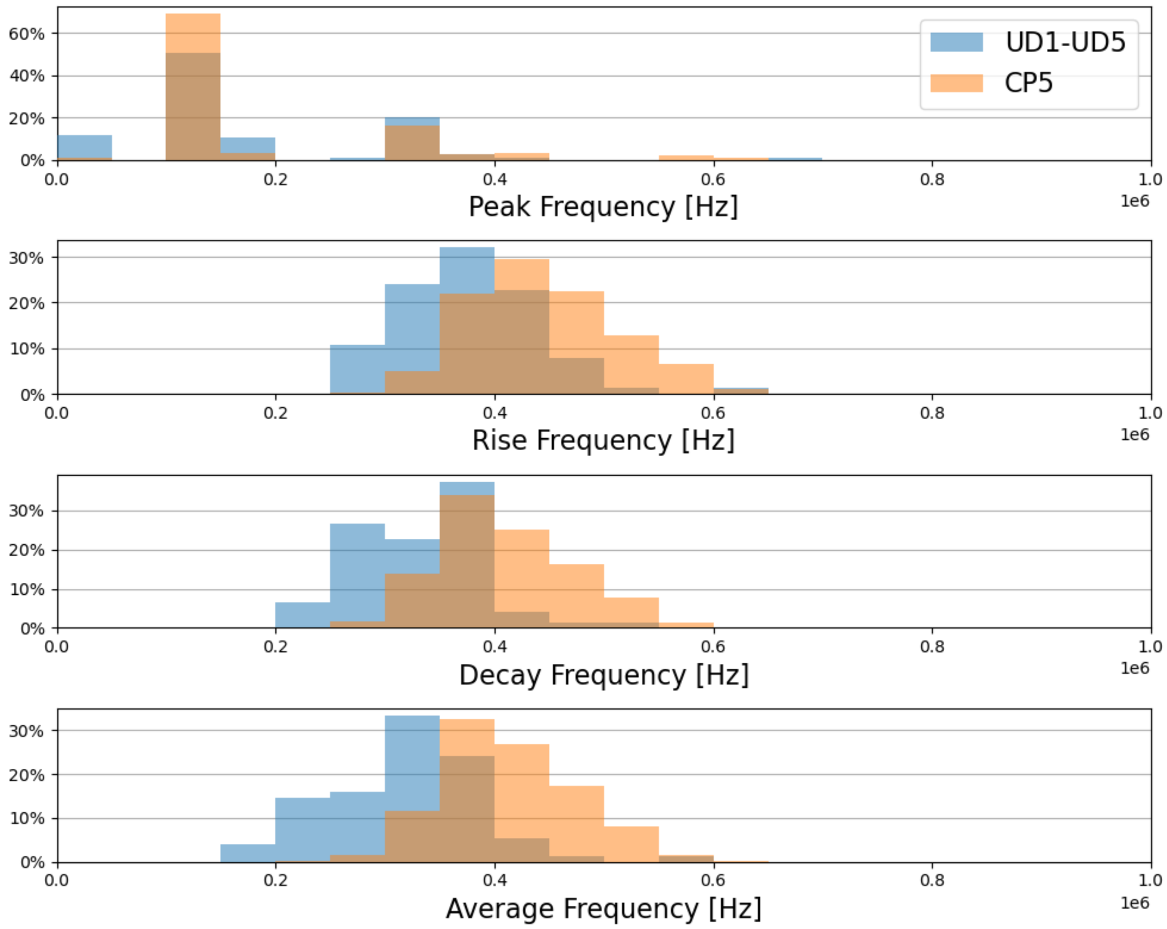


Figure A.3: Histogram of frequency features for UD1-UD5 and CP5 data

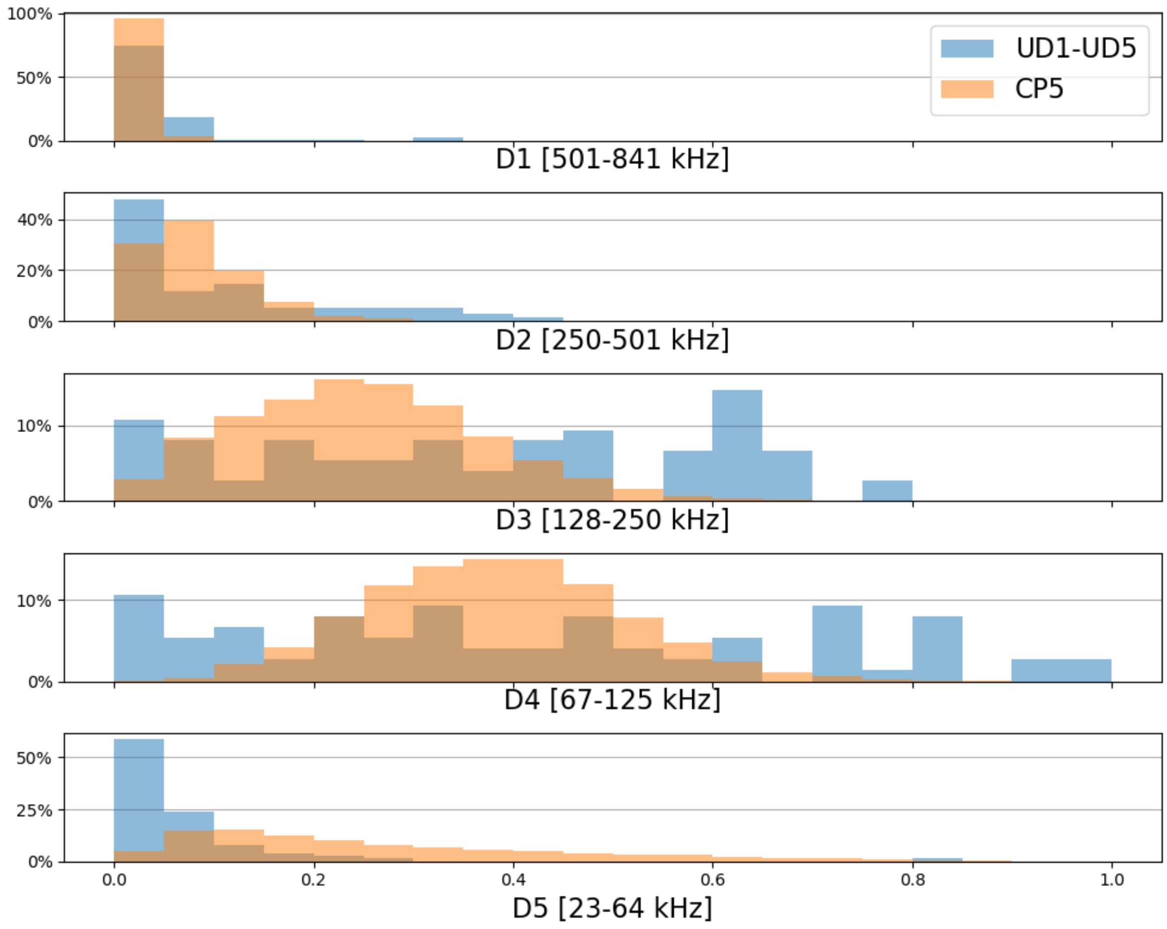


Figure A.4: Histogram of wavelet detail level features for UD1-UD5 and CP5 data

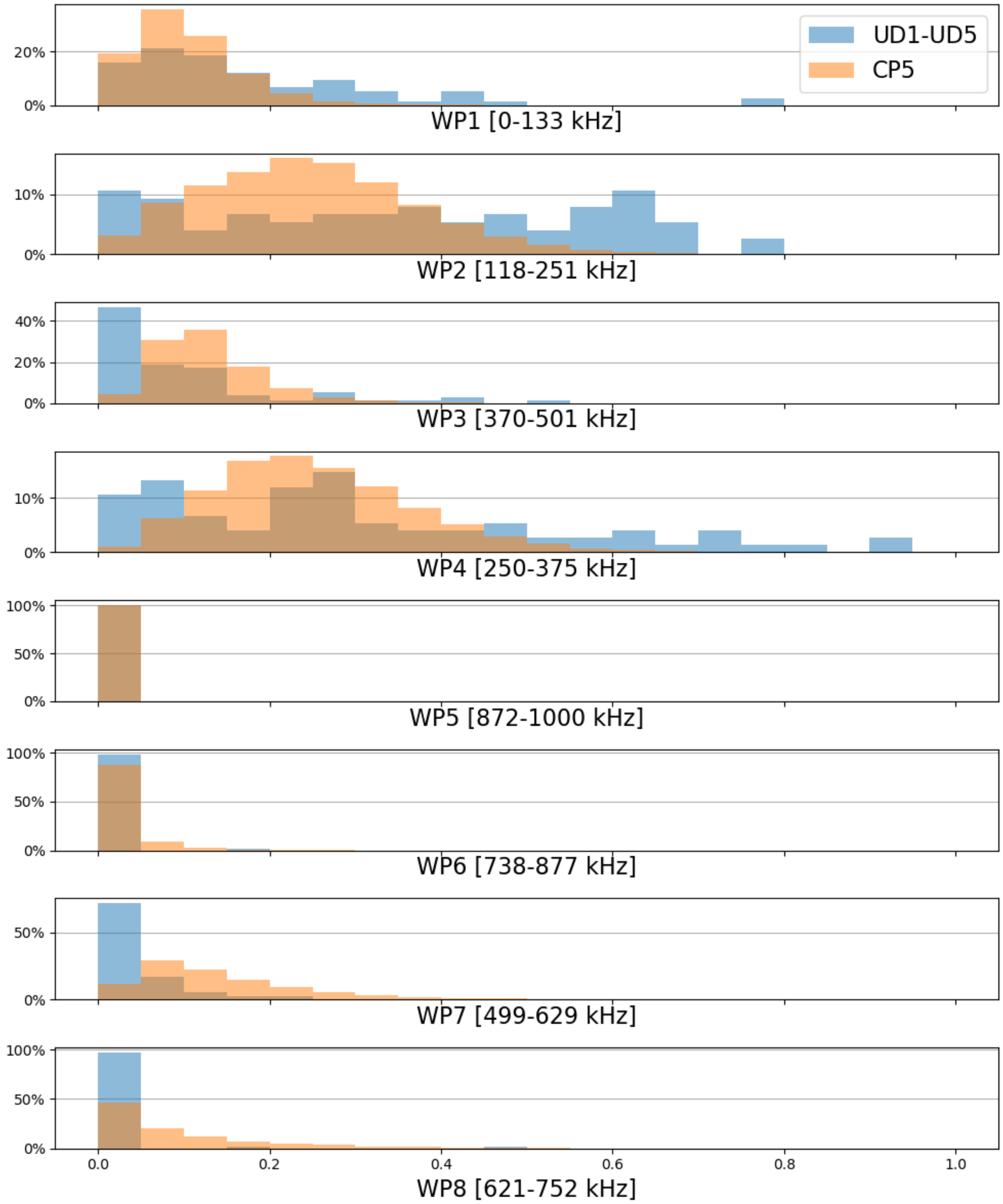


Figure A.5: Histogram of wavelet packet features for UD1-UD5 and CP5 data

B

Clustering Results

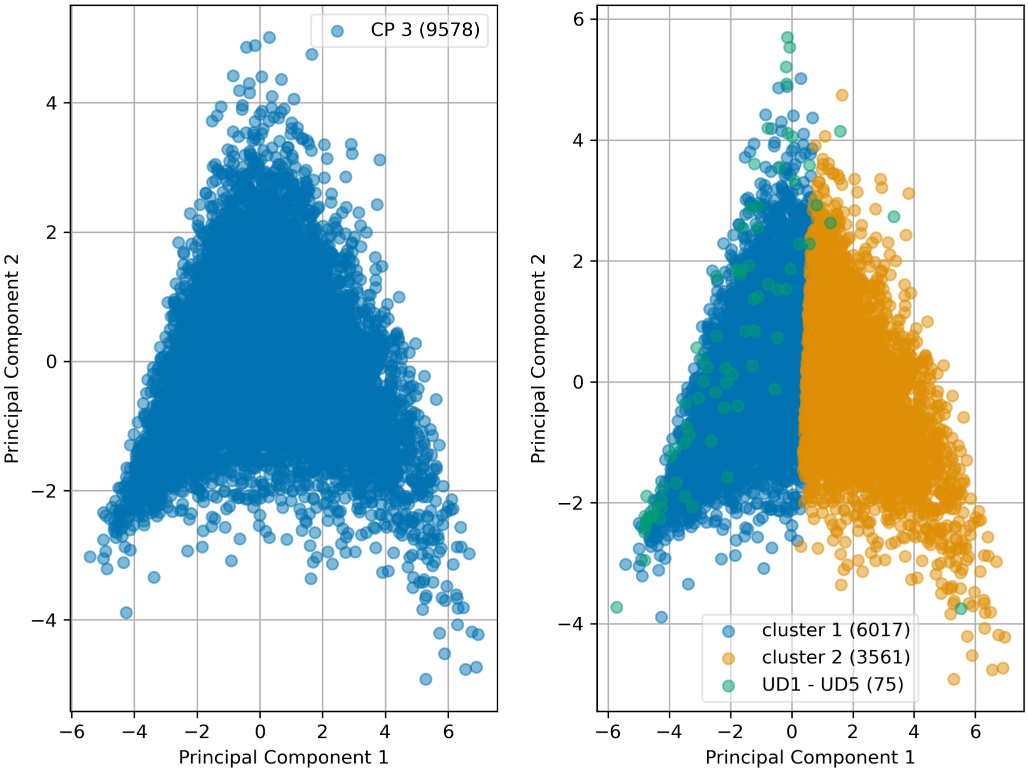


Figure B.1: K-means result on CP3 data

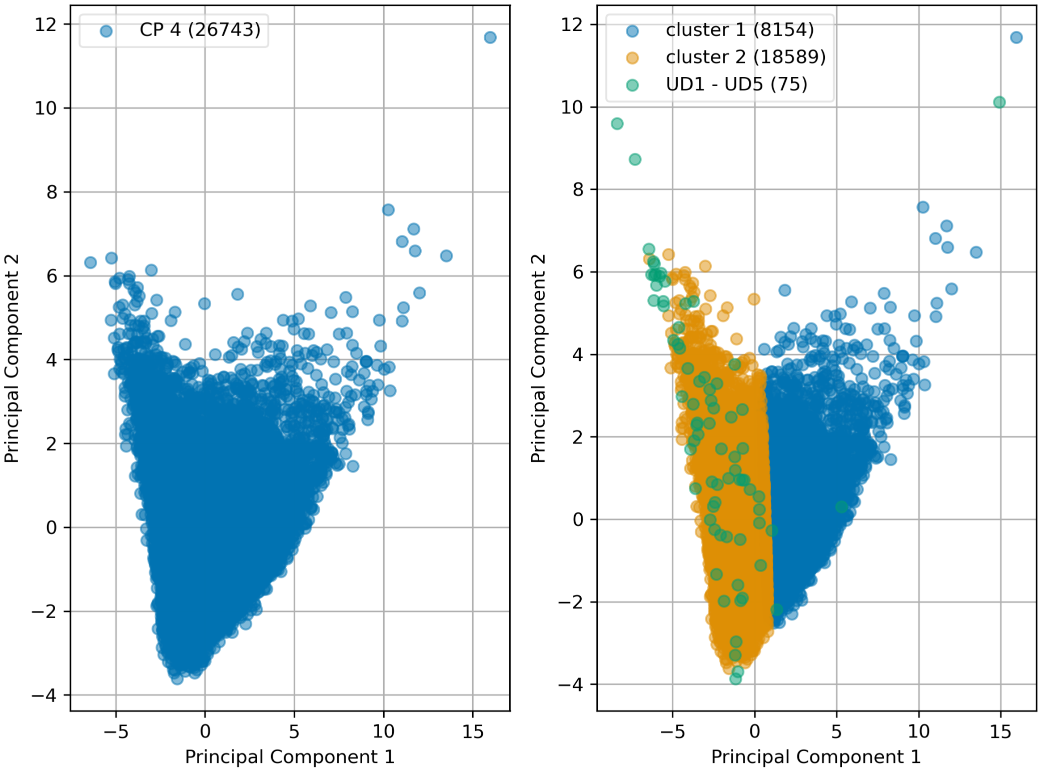


Figure B.2: K-means result on CP4 data

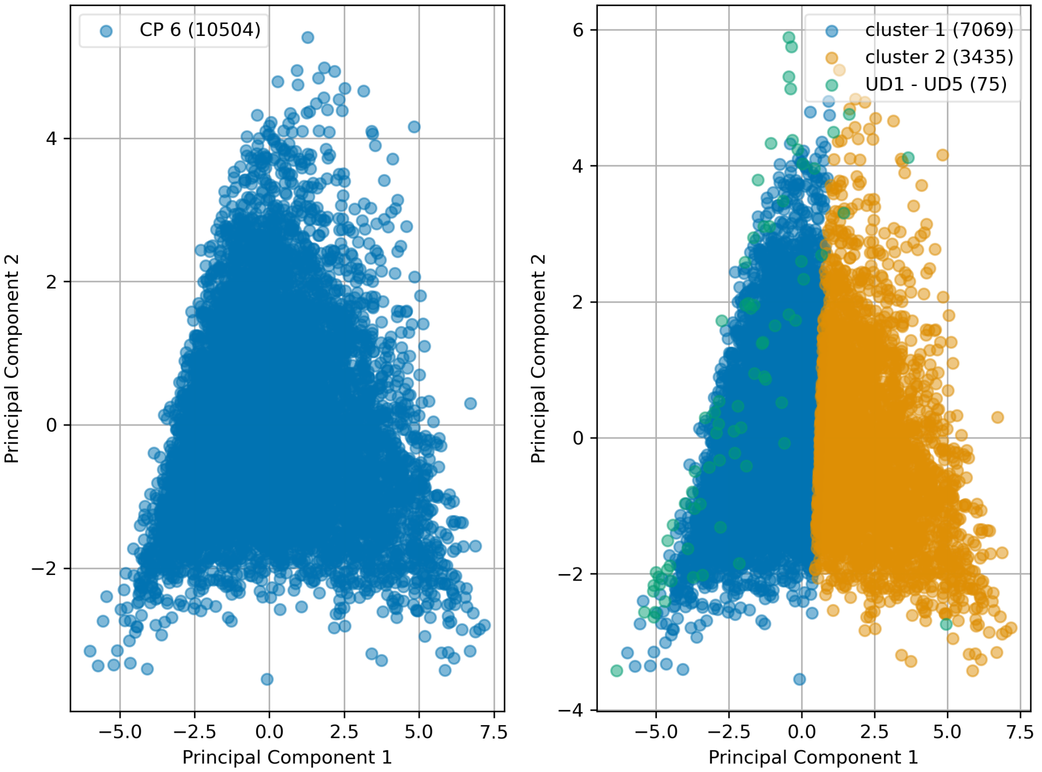


Figure B.3: K-means result on CP6 data

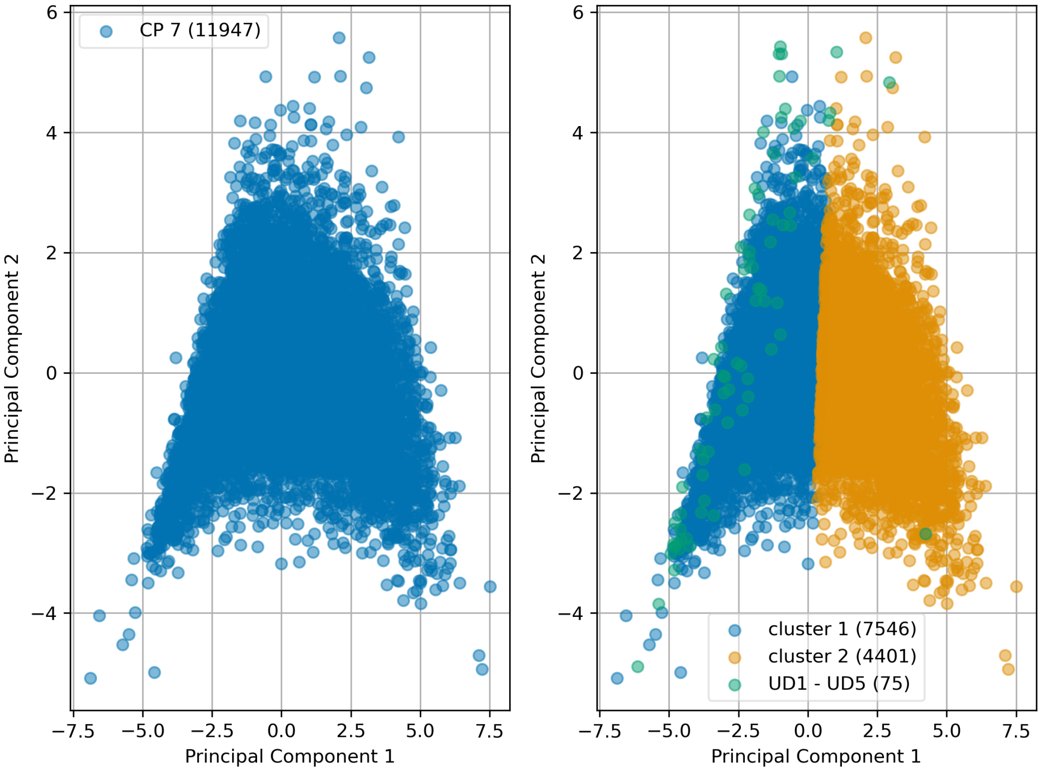


Figure B.4: K-means result on CP7 data

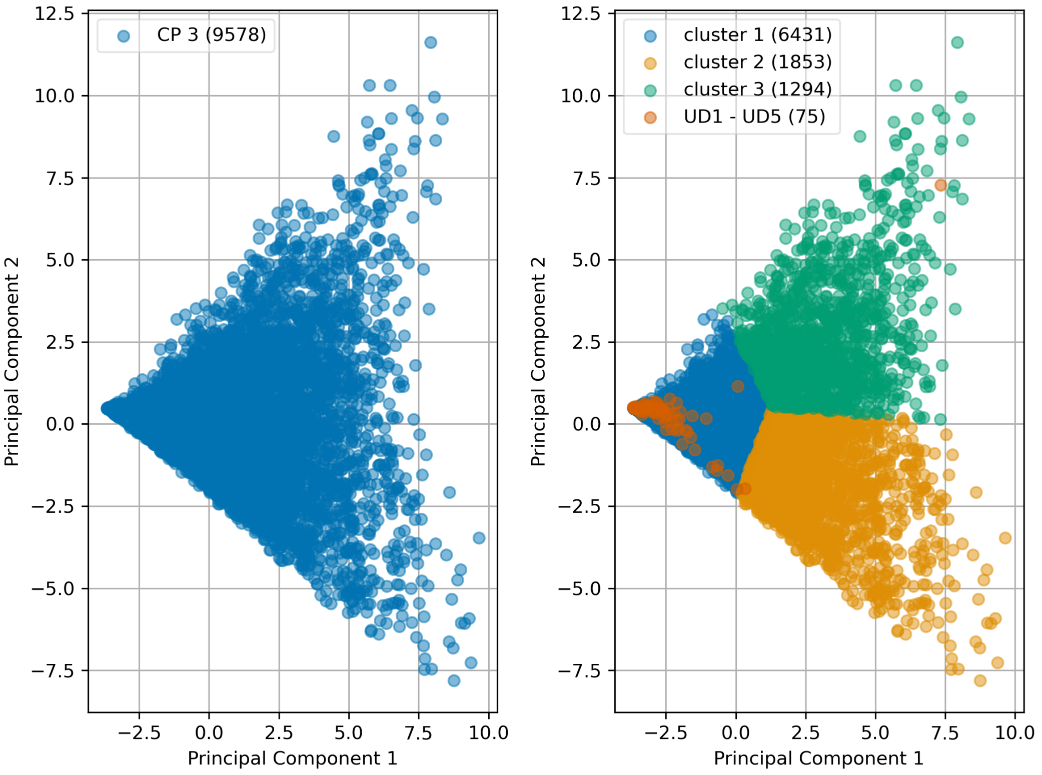


Figure B.5: K-means result on CP3, UD standardized data

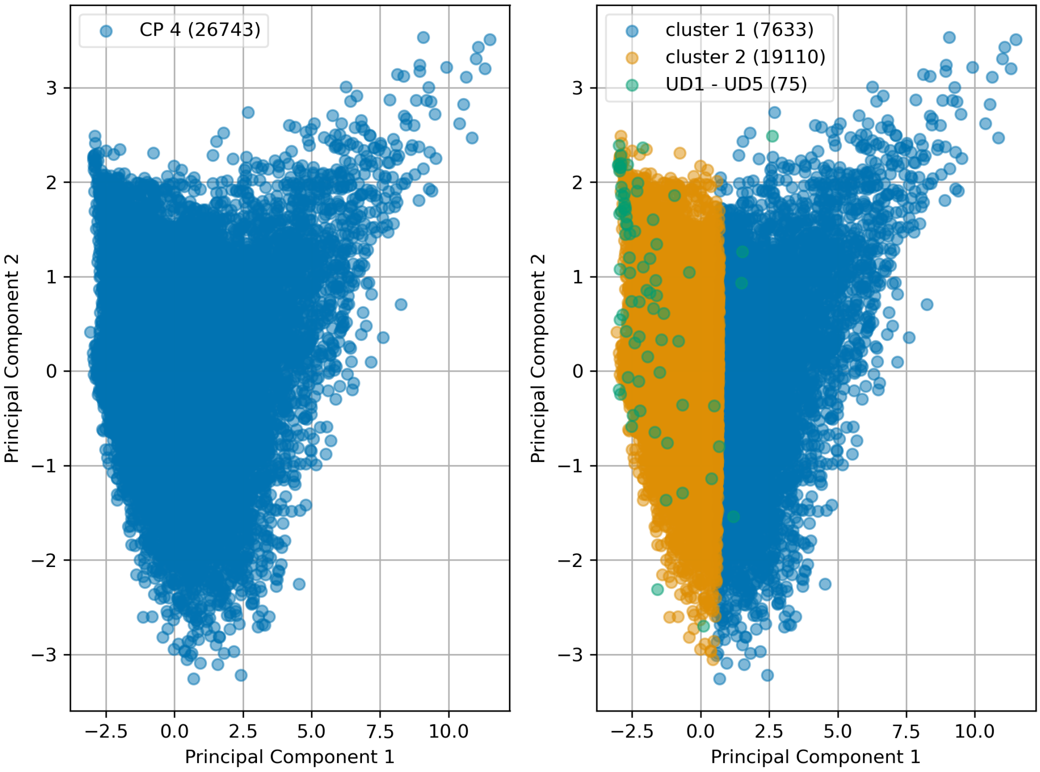


Figure B.6: K-means result on CP4, UD standardized data

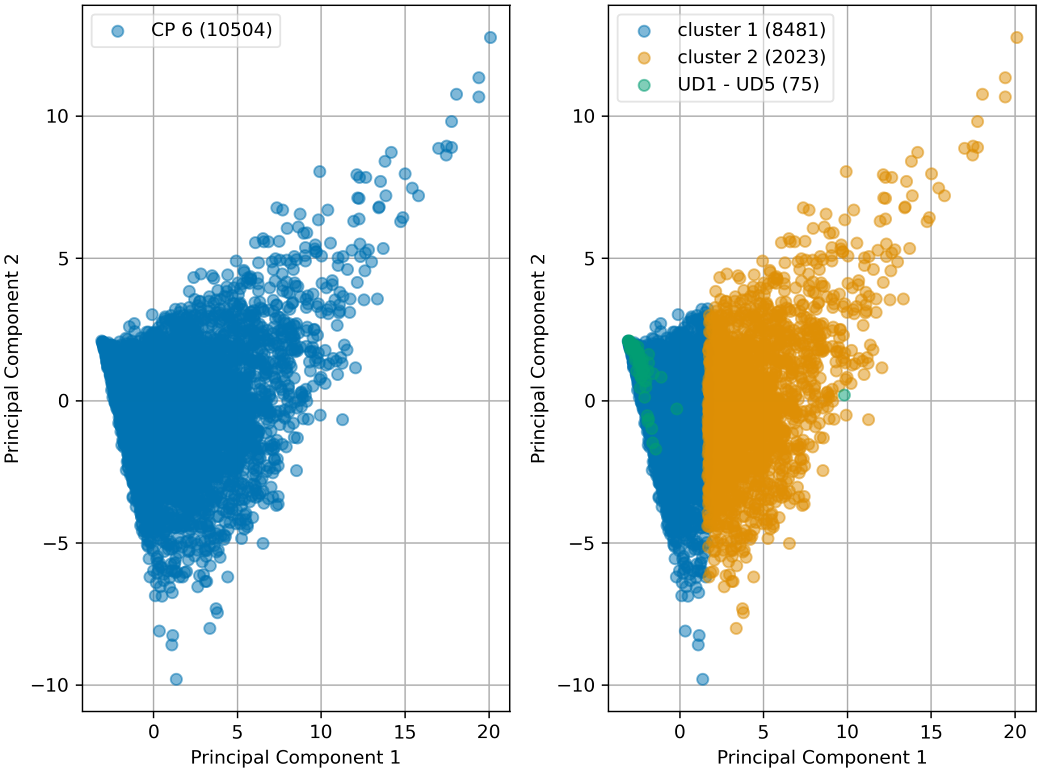


Figure B.7: K-means result on CP6, UD standardized data

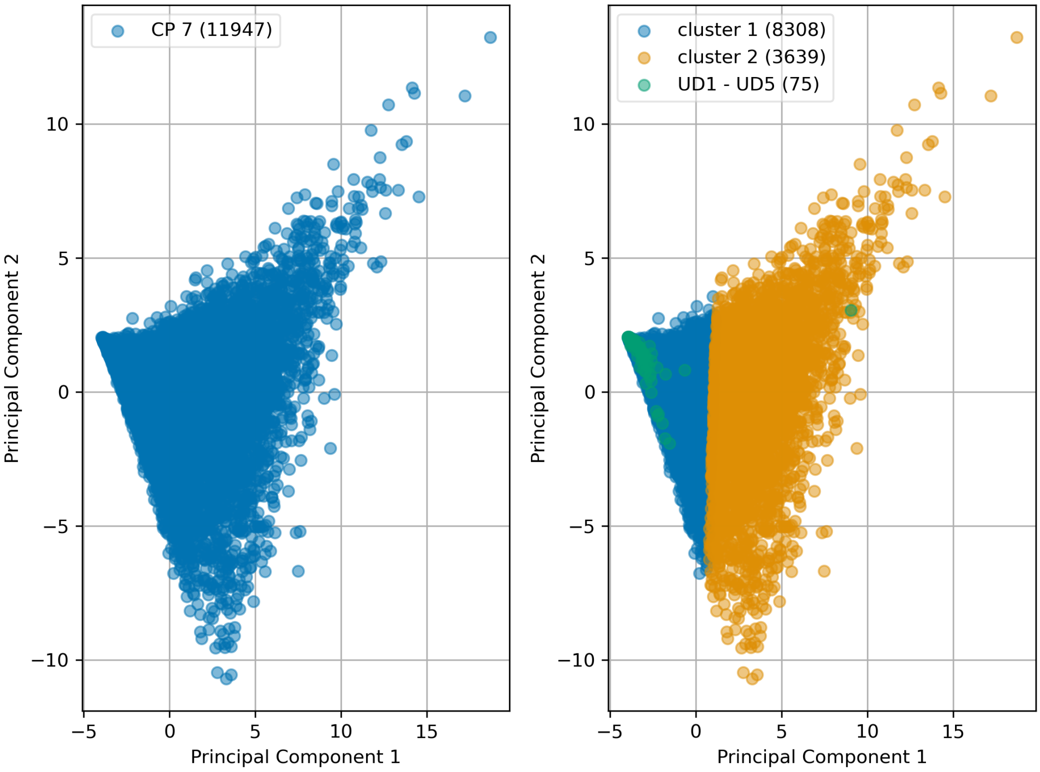


Figure B.8: K-means result on CP7, UD standardized data