University of Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study program/Specialization: **Engineering Structures and Materials/Offshore Structures**	Spring semester, 2021 Open
Author: **Yehia Elkhayat**	Yehia Elkhayat Digitally signed by Yehia Elkhayat Date: 2021.06.15 13:01:57 +02'00' (Writer's signature)
Faculty supervisor: ***Hirpa G. Lemu*, UiS** **Sudath C. Siriwardane, UiS** External advisor(s): ***Ole Gabrielsen, DNV***	
Thesis title: **Study of the Characteristics of AET Signals for Structural Health Monitoring**	
Credits (ECTS): **30**	
Keywords: **Acoustic Emission Testing** **Characteristics of Acoustic Emission Signals** **Structural Health Monitoring**	Pages: 65 + Enclosures 71 Stavanger, June 2021

Preface and Acknowledgements

This research project is submitted to fulfil the requirement to complete a Master of Science in Engineering Structures and Materials with Specialization in Offshore Structures from the University of Stavanger in Spring 2021. DNV offered the topic, and it is a continuation of previous theses submitted in 2019 by Dawood and in 2020 by Nguyen. The thesis duration was five months, starting in mid-January 2021 and submitted in mid-June 2021. The topic was therefore chosen as of the exciting tasks defined in the thesis proposal. That included performing a literature study on the acoustic emission signals processing methods, developing a small scale test proposal, performing laboratory tests, and programming tasks to analyze the acoustic emission signals. The tasks the was the most interesting were the programing tasks, where a tool was developed to process the data. Skills in signal processing and analysis, developing test procedures, performing laboratory tests, organizing workflow, setting up achievable goals, presenting achieved work in online meetings and, programing skills were the primary skills gained during the project. This master's thesis study was performed with guidance from Mr Ole Gabrielsen from DNV, Professor Hirpa Gelgele Lemu and Professor Sudath C. Siriwardane from the University of Stavanger. Special thanks to the university professors for being very closely guiding the management of the project. Ole Gabrielsen was always supportive and offered guidance with great enthusiasm on every detail of this thesis. The author is deeply thankful for Ole's effort throughout this project and would emphasize that it was fortunate to work with him. A special thanks to DNV's lab team, Mr Tor Jo Landheim and Mr Pawel Piotrowski, for their support to the project by providing all the needed resources to perform the test and working closely to complete all the required test and project requirement.

Yehia Elkhayat

Stavanger, Norway

15.06.2021

Abstract

This research aims to study the characteristics of acoustic emission signals emitted from a steel sample under loading. The thesis is a continuation of previous theses submitted in 2019 by Dawood and in 2020 by Nguyen. Laboratory tests were planned and performed to collect acoustic emission signals from welded specimens and coated specimens. Some of the collected data were processed. Furthermore, different approaches for correlating signals emitted from an event recorded by several channels were discussed, and tools for implementing them was developed in this research.

The results provided automated tools to correlate signals of an acoustic emission event captured by many sensors. Also compared the number of hits of the welded and unwelded. Further research on correlating signals from the same source is recommended.

Table of Content

List of Figures

List of Tables

List of Abbreviations

AE	Acoustic Emission
AET	Acoustic Emission Testing
GA	Genetic Algorothim
NDE	Non-destructive Evluation
NDT	Non-destructive Tesing
NZC	Number of Zero-crossing
PCA	principal component analysis
PLB	Pencil Lead Break
SHM	Structural Health Montoring
SI	Structural Integrity
SOM	Self-Organizing Map
TEM	Transmission Electron Microscopy

Chapter 1 Introduction

1.1 Background and Motivation

The project was proposed by DNV and is a continuation of the previous work done by Dawood and Nguyen [1], [2].

The number of platforms that passed or are approaching their design lifetime is increasing [3]. Ensuring structural integrity (SI) of the platforms in the oil and gas industry is often done by periodical inspection of the assets to operate safely. Inspections require planning and resources to be executed. The computational power, knowledge in data analysis, and sensors technologies motivate the industry to monitor their assets' online monitoring. Online monitoring of assets is beneficial for obtaining continuous up to date information on the structural members. This is a powerful tool to assess the condition and plan inspections, saving time and cost [3].

The digital twin for structural health monitoring (SHM) field is another approach that online monitoring can contribute to. The digital twin is attracting the attention of the global energy industry [4]. More than 600 offshore structures have exceeded their design lifetime or will exceed it soon. Deciding on the enhancement for those offshore structures requires information about their actual condition, such as the actual capacity of the structural member and the current load carried by the members, which must be fully reachable. The digital twin model is a virtual copy of the actual structure that accurately reflects the structure's current condition. Future behaviour can be expected from the updated knowledge about the structure condition, and necessary actions can be planned [5]. An updated model that reflects the actual asset helps plan the inspection and maintenance of those structures.

Many studies have investigated acoustic emission (AE) signals and how they can lead to information that can reduce maintenance cost when connected to an online monitoring system. For instance, the work presented in [6] established a methodology of online monitoring using AE.

AE is a phenomenon that occurs when a material emits sound waves while experiencing deformation or fracture [7]. AEs are high frequency, transient sound waves. Thus, acoustic emission testing (AET) is non-destructive testing (NDT) that works by attaching sensors to a material to record the AE of a deforming material. The attached sensors detect and record the AE by converting the surface movements caused by the elastic waves to electrical signals. The piezoelectric element of the sensor has high sensitivity, catches the fine surface movements, and converts it to electrical voltage [8]. Dawood and Nguyen [1], [2], under the supervision of Ole Gabrielsen from DNV, have conducted a master thesis on the characteristics of AE signals. The work presented here aims to continue on their findings. Dawood did a laboratory test where AE data were gathered; later on, Nguyen has processed the data from Dawood test using the parameter analysis method.

1.2 Project Objective

The project aims to study the AE signals' characteristics and identify helpful signal processing methods for the characterizing process. This is initiated to develop the online monitoring of the structure, which AE testing is believed to be a valuable tool.

1.3 Scope and Limitation

The project is an experimental based project. A test proposal was prepared and sent to the DNV laboratory in Høvik who conducted the tests. A literature study on AE signal analysis and processing methods to be performed. Selected methods of analysis to be performed on the test data.

The application of online monitoring is not the main focus of this study. This study aim to identify and characterize AE signals for further use in online monitoring applications.

This study is limited to collecting data from a laboratory test applied to a steel sample that undergoes flexural stress by applying a four-point bending test. Other loading setups were not included in this project. Afterwards, processing the data using selected signal analysis methods to develop better characterization AE signals and finally, discussing the result from the data processing.

Due to COVID-19, the student was not present physically at the DNV lab during the test. Therefore, the DNV lab team applied the test. The test was live-streamed to the student, and close communication with the lab engineer team was ensured.

1.4 Research Problem

The comprehensive aim of this research is to identify the AE signals at the yielding point of the material. The challenges occur in:

- Filtering the noise signals that come along with AE signals
- Identifying the source of the signal (ex. rubbing, yielding, crack)
- Correlating signals of the same AE event recorded by two or more channels

The main research problem for this research is to develop an automated algorithm to identify and correlate signals of a particular AE event logged by two channels.

1.5 Project Tasks

For better management of the project, the following tasks were created, and milestones were set up. **Milestones** are indicated by **bold** in the following list

1 Pre-study Report
  1.1 Reading Literature
  1.2 Problem Description and Background
  1.3 Planing Thesis Work
  1.4 Report Writing
  1.5 Submission of Prestudy Report
2 Python Learning and Understanding Bao's Code
3 Literature Study on the Acoustic Emission Signals Post Processing Methods

3.1 Listing the Methods and Selecting Methods to be Used

3.2 Submitting Summary of the Methods to be Used

4 Test Plan

4.1 Specimen Details

4.2 Cutting Procedure

4.3 Testing Procedure

4.4 Submission of Test Plan Report

5 Post Processing Tools

5.1 Coding the Selected Methods

5.2 Testing the Developed Code

5.3 Submission of Processing Methods Tool

6 Performing the Test

7 Performing Data Post Processing

8 Characterizing AE Signals

9 Presenting the Results to the Supervisors

10 Report Writing

10.1 Introduction, Background, Literature, Methodology and Test Plan

10.2 Results, Discussion and Conclusion

10.3 Report Draft Submission

11 Implementing Comments and Enhancing the Final Report

12 Submission of Final Report

1.6 Work Breakdown Structure and Gantt Chart

(Appendix F - Work Breakdown Structure and Gantt Chart) shows the work breakdown structure diagram of the project and the Gantt chart of the project. These were developed to better management for the project tasks.

1.7 Research Approach

This research is an experimental quantitative research that aims to identify the characterize the yielding AE signals. This shall be done by designing and performing a laboratory test; to collect data. Literature study on the available processing tools then implements the selected post-processing tools (ex. Magnitude squared coherence and cross-correlation coefficient [9] to characterize the AE signals finally.

1.8 Thesis Structure

This thesis report includes six chapters. Description of the content of the chapter is as follow

Chapter 1 Introduction

This chapter discusses the motivation and background behind the study and the project objective, scope and limitation.

Chapter 2 Literature Review

This chapter illustrates some of the research papers about processing techniques of AE signals. In addition to experimental cases that used AE signal analysis as part of their work. Also, summarize the previous theses done by previous students.

Chapter 3 Methodology

This chapter explains the methodological approach, the data used in the research, a brief about the used analysis methods.

Chapter 4 Experimentation

This chapter contains a summary of the tests carried out to collect the analysis data.

Chapter 5 Results and Discussion

This chapter presents the results, details about methods used to achieve the processing tools and discusses the details of the results.

Chapter 6 Conclusion and Recommendation for Further Work

This chapter sums up the work done in this thesis and gives recommendations for work that can add to this work.

Chapter 2 Literature Review

2.1 General

AE testing has a high potential to be a very efficient and effective technology used to detect fatigue and fracture behaviour in various materials such as metals, fibreglass, wood, composites, ceramics, concrete, and plastics. Originally introduced as a cheaper alternative to address the previously limited non-destructive evaluation (NDE) technologies, the suitability of this technology in engineering was evaluated in the late 1970s and early 1980s by some researchers from Japan, Europe and the USA [10]. Due to the many advantages of this technique, such as its ability to discriminate between environmental noise and machine vibrations from those of AE signals, it is now used in a wide variety of industries, such as power generation, refineries, structures, pipelines, aircraft. This technique, although passive, allows acoustic energy emitted due to mechanical or physical change to be detected without any energy input. Testing has continued into the 21st century, particularly into the two basic types of AE monitoring strategies. This technique now started being used within geoscience and material science. It can also be used as a complementary technique to others, such as ultrasound. Unfortunately, the AE technique does come with its limitations. It has a broad frequency band, so it requires a significant CPU to run, but the 'sensors are incredibly versatile, which improves the efficiency of the AET. It is understood that analysis is vital in this process, which allows correlations to be drawn. The analysis and data storage can be used for future use, allowing the AET to be employed. However, research and sensor implementation challenges must be remedies before this technique can become an effective tool to digitise a manufacturing process in the future [10]. This chapter will summarize research publication of the methods used to analyze AE signals by mentioning a description and the outcome of the method.

After evaluating the connection between the research topic and the analysis methods discussed in the following sections of the literature, the correlation of the signals was very relevant. It was the main focus to apply it in this research.

2.2 Differentiating and Correlating AE Signals

2.2.1 Study of Micro-Yielding Deformation Before Macro-Yielding in Pure Iron Using Acoustic Emission

AE was used alongside electron backscatter diffraction and transmission electron microscopy (TEM) to study micro-plastic deformation of a Di4 iron ingot with different low angle grain boundary ratios and initial dislocation states before macro-yielding. AE signals generated under tension were studied in detail; to envision the relative activities during micro-plastic deformation. This experiment found that these signals have different waveforms. Before the 1970s, it was believed that deformation before macro-yielding was elastic, so it was impossible to measure the AE signals. However, after the 1970s, more studies took place, and massive AE signals were generated before macro-yielding. Results from previous studies indicate that dislocation multiplication occurs during micro-plastic deformation. Therefore, it is possible to utilize AE techniques to characterize multiplication and motion. Also, when used alongside TEM, the true dislocation of morphology can be determined. More contemporary research, mainly focusing on acoustic emission technology characteristics, helped determine that the relationship between the micro dislocation source observed with TEM and macroscopic AE characteristics was used to describe the micro-plastic deformation mechanism. Experiments took place that showed the samples undergo micro-plastic deformation and begin yielding with a distinct yield platform. The AE signals generated in the rolled annealed sample were collected for analysis. There were two types of signals found: A1 and A2. A1 signals are typically a bursting signal with high amplitude and short duration, whereas A2 signals are considered a mixed signal composed of two types of A1 signals. The results were then shown and discussed. The study found that dislocation multiplication is the only mechanism governing the AE signal generation during micro-plastic deformation. In the end, the evidence showed that the formation of dislocation walls and the higher low angel boundary grain ratio in the rolled annealed sample produce AE signals with higher maximum energy and amplitude [11].

2.2.2 Differentiating Signals Using Cross-Correlation Coefficient and Magnitude Squared Coherence

One of the most significant issues with the practicality of the AE technique is the existence of sources of AE other than crack related, such as rubbing and impacts between different structural components. These other sources often cover the signals emitted from cracks activity, so discriminating the signals to identify the sources is of the utmost importance. One tool that can be applied to identify the sources is comparing similar and different signals. This is done using two methods: the cross-correlation coefficients in the time domain and the magnitude squared coherence in the frequency domain. The cross-correlation coefficients highlight the maximum correlation of signals at a particular time lag by evaluating the correlation of two-time series shifted along with each other. The highest value of this shift is the value used for the maximum correlation. Mathematically the cross-correlation of two signals $f_1(t)$ and $f_2(t)$ can be expressed as

$$r_{xy}(\tau) = \int_{-\infty}^{\infty} f_1(t) \cdot f_2(t+\tau) dt \tag{1}$$

The magnitude squared coherence is calculated by applying the signals' power spectral densities and cross power spectral density. The magnitude squared coherence will be one over the whole frequency range if the signals are entirely identical; however, if the signals differ, the value will be below 1. Magnitude squared coherence can be expressed as

$$C_{xy}(f) = \frac{\left|P_{xy}(f)\right|^2}{P_{xx}(f) \cdot P_{yy}(f)} \tag{2}$$

Where,

$C_{xy}(f)$: $Magnitude\ Squared\ Coherence$

$P_{xy}(f)$: $Cross\ Power\ Spectral\ Density$

$P_{xx}(f)$ and $P_{yy}(f)$: $Power\ Spectral\ Denisty$

After this, a short-time Fourier transform can be performed on signal results in a coefficients matrix by discretization in both frequency and time domains. These results are then plotted on a spectrogram. It is important to remember that the short-time Fourier transform can be used on its own to analyze the data on energy frequencies. Still, the study of simultaneous time-frequency distribution gives essential information for discriminating the signals, so both usually are utilized. The parameters used for the study criteria [12] were proven to be suitable for determining the signals, as unknown sources signals were available as a template for comparison. However, as with all experiments, factors involved could affect the results, so care is needed to overcome them. This study [12] concluded that even more research is required and is being carried out to investigate newer techniques for signal source discrimination [12].

2.2.3 Analysis of Acoustic Emission Parameters from Corrosion of AST Bottom Plate in Field Testing

Improving the accuracy of detection is of great importance. The basis of this study was the identification of corrosion acoustic emission signal of tank bottom. However, in some cases, the corrosion acoustic emission signal of the tank bottom was unknown. The weighted fuzzy clustering recognition method was proposed to avoid this. The aim was to improve the accuracy of the data by explicitly examining the randomness of clustering initialization and using the nearest neighbour method. The focus was on optimizing initial clustering. This data was then used to confirm the cluster number and the centre directly. This method increased the difference of acoustic emission signals, which caused the difference of varied acoustic emission signals to expand. This data redistribution is adjusted with the weighted distance between the gravity and centre to substitute the traditional distance and then allocated data to the set with the minimum weighted distance. This approach is shown to have improved the validity of weighted fuzzy clustering by about 9%. Although the study found that some signals can have differing characteristics, using an acoustic emission detector, the main characteristics of the signal were identified to include rise time, count, energy, duration and average frequency. The study found that pitting acoustic emission signals can be discriminated by two parameters, energy and amplitude, allowing the source to be more easily found. Alongside this, the crack and oxide film acoustic emission signals can also be identified. Therefore, through the use of the analysis showcased within the study, it was concluded that the types of corrosion source could be successfully discriminated by using the method presented in this study [13].

2.2.4 Application of Acoustic Emission Testing in Fault Diagnosis

Gear systems are essential devices used in various kinds of industrial equipment. It affects the whole equipment if it fails, so any faults or failures need to be diagnosed quickly. Noise complicates the diagnosis, 'so conventional vibration testing methods are ineffective. However, AE testing can be used instead: Compared with vibration signals, the frequency spectra of acoustic emission signals are broader. Their high frequencies can effectively inhibit noise interferences and improve diagnosis accuracy. By using AET, any defects inside the material could be detected. The paper [14] discusses using parameter analysis based on basic parameters such as the ring, energy and amplitudes, and waveform analysis methods to analyze the data. Simulations were completed. Through completing these simulations, they verified the feasibility and validity of the morphological opening operation and the multi-scale Top-Hat transform method. Many research studies were presented, and the conclusion is that it is still hard to identify the faults in such equipment by acoustic emission signals. The redundant second-generation wavelet transformation that aims to address one of the acknowledged difficulties with AET of how to point early faults easily in with background noise. This algorithm was split into two processes: decomposition and reconstruction. The experiment showed that the signals were decomposed with redundant second-generation wavelet into three-layer. The results also showed that the redundant second-generation wavelet had de-noised more noise signals than the traditional wavelet, making it a more useful algorithm. 'When applied to gears, the observation was that if there is no abrasion, each gear has an involuted shape and the signal generated in the operation process is the single-frequency harmonic curve whose frequency is the gear mesh frequency; after the gear abrasion, the shape of the gear changes and the signal generated in the operation process is the approximate periodic signal curve whose fundamental frequency is the gear mesh frequency' [14]. When checked internally, the results of analysis and diagnosis matched the on-spot conditions. Therefore, the it was concluded that the validity of this algorithm of the redundant second-generation wavelet transform in the processing and denoising of AE signals was verified [14].

2.2.5 Acoustic Emission Signals Processing Based on Wavelet Analysis

Wavelet analysis is a suitable method and tool for signal processing and analysis. Many scholars share this view. However, these scholars vary depending on how to choose the suitable wavelet basis, as there will be different results depending upon the choice made. It was agreed that wavelet analysis is a promising approach. It is the latest achievement of harmonic analysis and is considered a significant breakthrough of the tools and methods. This analysis method uses the parallel move and telescopic of a function. It is a real-time approach to express the partial characteristics of signals in the time-domain and frequency-domain simultaneously. As well as this, it can describe the corresponding information in the time-domain synchronously; therefore, this is the most suitable method for AET. Each band's centre frequency and bandwidth will change simultaneously in this method and the scale changes. However, the best analysis will be provided with a smaller window in both the time and frequency domains; yet the windows of the time and frequency will restrict each other. Therefore, 'in practical applications, it is necessary to consider the time and frequency domain partial characteristics compromise to the requirements of the practical signal processing. As with all AE experiments, noise signals inevitably play a part. To eliminate noise resulting from friction between the sample ends and pressure plate, the spacer should be used. These spacers are rubber, which has the same thickness, and the core of the axis should coincide with the axis of loading. In the analysis stage of the experiments, it was determined that the method has a high recognition rate for the locating of the AE signal sources. However, it is important to warn that the multi-channel collected data may have large accidental error while the traditional method of calculating AE location was fixed; therefore, the position result is inaccurate and has significant errors [15].

2.2.6 Remote Acoustic Analysis for Tool Condition Monitoring

Acoustic emissions signal technology can also be found in plant machinery. Until recently, organizations chose to run many of their plant machinery until failure and subsequently carry out maintenance as required; however, due to the continuous advancements in manufacturing processes and the increasing complexity of machinery, maintenance teams face new challenges. There are now too many consequences of running machinery in this way, including catastrophic damage to the machinery due to excessive vibration, overheating, breaking of parts etc., not to mention the loss of business due to lack of orders and possible injury to personnel. To maintain machinery, the maintenance would need to be carried out at a time that would reduce the effect on the company. Traditionally, this preventative maintenance method would require many resources and be very costly to the company, and the solution was to replace machine components more frequently. However, now maintenance teams have had to move on to predictive maintenance that is a more efficient and cost-effective maintenance method. To do this, the tool wear must be monitored. This can be done in many ways, such as; vibration analysis, visual inspection (human inspection), power consumption, acoustic emission and sound analysis. Many of these require sensor systems to be placed on or under the system, but this has a high probability of damage. Therefore, it was suggested that 'installing a remote monitoring system can help reduce many of these issues as it allows for protection to the valuable sensor systems. Monitoring and maintenance of the health of machinery and the machine components are also of great importance as all of these factors impact the finished products. It was shown that the use of coatings could extend tool life, as it increases the tools surface hardness decreases friction between the tool and part being machined and allows for better surface temperature distribution by dissipating heat generated during machining, allowing for tool quality to remain for a longer state. There are two main types of tool wear: Flank Wear and Cratering. Flank Wear particularly increases the tool's vibration and has a higher amplitude of sound being emitted. At the same time, machining, whereas Cratering is caused by high temperatures and requires immediate tool replacement. AE is one of the most effective methods for monitoring tool wear. This was done by measuring the vibration levels using accelerometers placed on the system being tested. AE methods are known to be accurate and focus on identifying only significant defects that are actively growing under stress. However, AE is also known to have limitations, such as as the distance from the sound source and the sensor increases, it can result in quick attenuation of the signal, which can result in the reflection of the signals and possibly allow other undesirable noise elements being introduced into the measured data. However, there are many ways to address these issues, as AE sensors are incredibly versatile due to monitoring conditions available for collection and analysis. More detail was looked at for remote acoustic analysis monitoring. It stated that one of the primary advantages of this method is that it can be installed remotely, which is particularly beneficial for micro-machining processed and in very harsh industrial environments. This process is especially desirable due to its ability to be more flexible as it is not designed to fit one purpose. Due to this ability and the option for future faults or failures to be predicted using this technology, remote systems have a great possibility of being integrated effectively into intelligent manufacturing environments. Once the data has been collected, another challenge arises were to keep the data. With this new process, the decentralization of these tasks will allow for the freeing of local server space, opening up this unused space as a production environment. Analyzing the data makes it possible to draw correlations, which can then be employed for predictive outcomes across similar machine, tooling or processing types. This could be influential in many ways, as 'there are numerous applications within industry. Although acoustic analysis can be employed in these applications, there are research and sensor implementation challenges that must be addressed to use acoustic analysis as an effective tool [16].

2.2.7 Third-Order Spectral Characterization of Acoustic Emission Signals

Research shows a limitation in the number of echoes that can be extracted due to their low-level amplitudes. It is known that faults and failure mechanisms cause AE signals. The analysis provides information regarding both the cause of the breaks and the media. The vibratory waves propagate, allowing to predict if and when the fault or failure will reoccur. The research states that an average complex AE event comprises different sub-events expressed as two classes. The first of these classes is a breakage getting more prominent, whereas the second is bursts from different sources, provoked by different mechanisms. Mainly the study of the main burst and the longitudinal reflections at the chord borders; however, there are also 'reflections that take place at the transversal borders. From these, the desirable sources need to be extracted. A bi-spectral analysis which was performed with a twofold purpose. The first purpose is to enhance the characterization of the AE longitudinal events over the measurement background noise, while the second is 'to find our more reflections, which are masked by the transversal reflections with the same order of magnitude, and cannot be detected using power spectrum and wavelet packet analysis. The third-order spectra reveal two or three more echoes, which are supposed to be additional higher-order frequency components, which cannot be discovered using second-order methods. Sometimes, transversal reflections may be hidden. Higher-order spectra were used, and the results can be applied in future work to characterize failure mechanisms of pipes for the oil industry. The experiment is set up by attaching one Sensor to the outer surface of the ring-type sample, which is under mechanical excitation. This resulted in bending in the inside upper face, which is thought to concentrate the elastic waves. At this time, it is possible to distinguish 2 or 3 AE events in each signal. The issues faced as it is difficult to distinguish the main AE event from the secondary events (reflections or echoes), both in the frequency and time domains. This is due to two reasons the noise signals, which is in general symmetrically distributed, and the lack of information regarding the phase of the components. However, it is important to note that all the frequency components are equally important. The study found that as the decomposition level increases, no more reflections are found and concluded that a frequency shift characterizes the existence of the main AE event, which takes the highest value, and the second lower-amplitude echoes. Therefore, it was determined that the better the resolution, the higher the magnitude levels associated with relevant frequency components [17].

2.2.8 Time-Frequency Analysis of Acoustic Emission Signals in CFRP

The most typical representative of advanced composite materials is carbon fibre reinforced plastic composites. The AE signals emitted from these are mostly nonstationary random signals, but some difficulty is faced 'in signal processing and analysis. Using these composites to find helpful information 'from the spectrums of different time-frequency methods. In the analysis of AE signals, which was first completed in the 1940s, the objective of time-frequency analysis was mainly about nonstationary signals and time-varying signals. There are three methods in time-frequency analysis discussed within the paper [18] STFT, CWT and HHT. These can reveal frequency contents and reveal the trend of frequency changes with time changing well. For STFT, the accuracy is brought down as frequency cross terms have been found. CWT is similar to STFT, but the difference is that the window function used in CWT is scaled adjustable.

In comparison, frequency cross-terms were not found in HHT. Therefore, it can be determined that HHT can better reflect the time-frequency characters of signals because of the self-adaptability of the EMD and the independence of the time resolution and frequency resolution after HHT. A compression test was completed, and the signals collected were divided into three main groups matrix cracking, delaminations and fibre breakage. The results were analyzed, and it was concluded that in the low range, frequency changes smoothly.

However, delaminations contain many frequency components from a low-frequency range to a high range, but the energy is mainly distributed in the middle and low-frequency range. It was concluded that the HHT method would play an essential role in the nonstationary signal analysis [18].

2.3 Clustering Approaches for AE Signals

2.3.1 Optimizing Acoustic Emission Data Clustering by a Genetic Algorithm Method

Many challenges occur in the analysis of AE signals of damaging materials. The examination of the damage of structural materials is a vital point for the control of durability and reliability in the structure lifetime. One such obstacle is characterizing the acoustic emission data collected during mechanical tests. The data set from acoustic emissions experiments can be drawn up, which can then be analyzed to cover the contributions of the new algorithm. Complex data often causes issues with clustering; however, the method used in this study [19] grant a better clustering of AE data and allows complex data sets to be clustered. This is particularly prevalent when a cluster is either significantly lower than other clusters in the data, very far from the other clusters in the collected data. These are also termed outliers, or if the clusters vary drastically in size. There are many sources of acoustic emissions, including plastic deformation, crack initiation and cracks propagation, fibres raptures or particles, interfacial decohesion and transformation of phases like; martensitic transformation. There can be many processes involved in the damaging of the materials. However, previous work by (Moevus M) determined the correlation between damage mechanisms and the parameters of AE signals waveform. This process showed that AE signals could represent their sources from the material experiencing, allowing the acoustic signature to be found. Assumptions are made for the studies to take place. Many pattern recognition techniques are already in use, including the Fisher Analysis, the Principal Components Analysis, the k-means and the neural networks. The issues regarding clustering lie in classifying the collected AE signals recorded during mechanical tests into a set of natural clusters with no previous knowledge. The k-means methods were used to find segmentation for a specific user-defined number of clusters. Gutkin, R showed that Self Organizing Map combined with k-means clustering leads to better clustering condition with lower computational attempts. Indeed, the different clusters were identified during the mechanical tests. In another test (by Moevus, M), the AE data revealed the natural structure of data. This meant several different types of signals and clusters were followed along with the expected separation between damage mechanisms. Despite this, the most important mechanism's signals – fibre failures signals – were not discriminated against, the small number of related signals is maybe the reason. Therefore, the clustering algorithm needs to be improved to consider the minority mechanism that may be critical. Some research suggests that Genetic Algorithm (GA)-based clustering techniques give better results than those obtained with the method of k-means.The experimental results were comparing the GA-based clustering method with the k-means algorithm [19]. Amerged real AE data collected under different situations and on different material to have different AE signals with different characteristics and shapes. The study [19] took two sensors that were mounted directly on the test specimen. Then, 18 of the most relevant descriptors were selected, after which a Principal Component Analysis was completed to define new uncorresponded features. The optimal number of clusters, known as k, then needed to be determined. This was formed using the Davies-Bouldin index, of which the best solution corresponds to the lowest value of the index. Then the quality of the clustering by the silhouette value was measured for each point which is a measure of how similar this point is compared to other points of its cluster compared to the points in other clusters. The closer the measurement to +1, the more this shows that those points are very close to neighbouring clusters. The closer the measurement to 0, the more this shows that those points do not belong in one cluster; the closer the measurement to -1, the more this shows 'points are considered in the wrong cluster. These can then be plotted on a distribution plot. The data was split into 6 data sets: basic data set, minority-cluster data set, outliers-cluster data set, dense data set, mechanical test-inspired data set, and real test data set. A comparison between the two algorithms, k-means and strategy genetic, was done to evaluate the applicability of the assessment criteria and assess the capability of the algorithms to categorize the 4 clusters building the data set. It is important to note; both algorithms did not have equivalent results. The study concluded that Cluster 1 was always effective even when it is the beginning of the heating. It was associated with noise. Cluster 2, largely minority, appeared just during the cooling. It was noticed that AE signals were only logged at a lower temperature in the phase of cooling, approving that the algorithm grant for the discrimination of descriptive signals in a data set containing noise. In conclusion, the clustering solution quality can be better by signals with low silhouette values on the side, knowing that the correlation between the obtained classes and the material damage mechanism is specific for each study and shall evaluated for each case based on the knowledge of the studied material [19].

2.3.2 Neural Networks and Signal Subspace Projections as Robust Clustering of Acoustic Emission

One of the biggest challenges within acoustic emissions is differentiating the events from crack growth and other noise origins. A novel algorithm is introduced that consists of two steps. Firstly, the noise was removed from the events using combined tools:

- Covariance analysis
- Principal component analysis
- Differential time delay estimates

Secondly, the data are then processed using a self-organizing map neural network, which provides split neurons for the noise and AE signals. In order to lessen the dimension of the data, the short-time Fourier transform was applied to maintain the time-frequency features of the rest of the events. This was then authenticated with two sets of data.

Structures require increased reliability and safety standards, which depends on the early detection of failures. The advantage of AE is that it can provide continuous real-time monitoring while the structure is in service. However, this technique is flawed, as it struggles to differentiate the events of interest from other noise events. Nevertheless, as real AE events are recorded in an environment of noise due to many factors like vibration, fretting, electromagnetic interference, and so forth, and rejection of those noises is required before relating AE events with crack initiations or progressive failures, it largely becomes an issue of recognizing the pattern and classifying the random processes. One solution to this problem is to apply neural networks that can automatically discover features and patterns in a more significant collection of relatively random observations. This is a two-step process: step one is to separate the noise events from the events of interest by using a combination of covariance analysis, principal component analysis and differential time delay estimates; step two is to process the data left after step one, which groups AE signals and noise signals to different neuron outputs. The short-time Fourier transform was used to enhance the efficacy of the process. It was used for the classification to reduce the dimension of the data. In the end, two sets of reliable data were determined by inspection. In order to find the data in each circuit, there is a flexible filter and setting of gain. The signal that was outputted is divided into a data of signal and a trigger of the signal. This trigger was increased slowly until the system started to record signals frequently before the fatigue strains occur. The level at which this plateaued was then identified as the noise level of the system. The study confirmed that AE produced from growing cracks are burst, which was higher frequency than grip noise and short waves. An accurate, albeit simple, process for differentiating sources of AE and noise is the differential time-delay estimate, which was used in cases where the AE source are spatially separated. As this is a complex process for automation, more schemes are being explored to differentiate the AE and noise in different groups to reach a clean set of AE events. A principal component analysis was used to lower the dimensionality of the data set. A large number of iterateable variables were needed while keeping the various presentations in the data set as much as possible. This is made possible by changing the original data to another new representation, the principal component, ordered in a way where the first few components keep most of the variation present in the original data. This method was used mainly to visualize the original data in the subspace of principal vectors, and the corresponding PCs are being the coordinates. The study concluded that most of the energy is available in the lower frequency components. Still, the higher dimensions characterize the higher frequency details for the signal and the wideband noise recorded in the measurements. If a large crack forms and grows consistently, the generated AE events have very high similarity. By describing the similarities within the cross-correlation between event and other adjacent events. Therefore, it appears that there was a potential of using cross-correlation among succeeding events over a time window as an indicator of the changes in the development and growth stage of a crack. A spectrum of vibrations with low frequencies content can be filtered. This filtration process can be carried out by a definite accurate fine filter which has to be recommended. The study proposed a method that put aside the noncracking events. Then by applying a Kohonen network to cluster related AE, which emitted from cracked parts. By analyzing AE and other accomplished waves, it can concluded that four clusters: the AE cluster around the origin and three non-AE clusters. A significant portion of the non-AE is put aside, but some remain non-separated due to their characteristic close to the AE signals. No definite specifications of the AE. According to that, the use of the Kohonen network seems will be appropriate. Then the AE, which is related to the crack, can be predicted among the other wave spectra. The analytical study shows that the algorithm will efficiently discriminate the AE signals of interest from the other non-AE signals [20].

2.3.3 Acoustic Emission Clustering Using the Kohonen Network

As one of the most hopeful techniques for monitoring mechanical systems, the AE-based technique was used to intuit signals emitted. It is important to note that these must be identified even through the damping of surrounded waves. For this reason, the Kohonen network was a good option, as long as for the implementation and training process of the network algorithm. AE signals are emitted from the sudden launch of internal energy during stress and failure. The alternative approach of spread networks, which influence signal discrimination, which Based on large-scale parallel network machine learning procedures for simply processing. The classifiers found during this process generates severed curly surfaces, defining properties of structures and giving data. This is an automatic method that only uses the input data. The study deals with the in-flight and multi-sensor lab data. Through the use of the AE automatic receivers, the Pinger that is fixed to the rotator connection link is set to on. A small pressure was added; when the energy released, a travelling acoustic signal was emitted from the same area. The study used an experimental setup to prove that the unique stacking of sensors is one of the most significant factors which distinguish AE signal emitted from crack-related and other interwoven signals. the core elements of Fourier transforms of the data can be applied for the network calculations, resulting in the clusters proving that the Korhonen network is an efficient as a reliable source in discriminating acoustic emission signals [21].

2.3.4 Pattern Recognition Approach to Identify Natural Clusters of Acoustic Emission Signals

The approach was based on a comprehensive investigation that was taken for all combinations of signal characteristics extracted from the received acoustic emission signals. It is challenging to decipher the AE signals, which are attenuated and unclamped, emitted from the specimen with no extra information.. A multivariant analysis for pattern recognition. Indeed, one such analysis approach is parametric pattern recognition. These techniques split out noise and non-noise signals, such as signals emitted due to friction of electromagnetic inductions, using non-monitored pattern recognition. This is possible according to their discrimination. The complicated issue that scholars face was the recognition of characteristic clusters of acoustic emitted signals. Based on that, the evaluation and classification results are still limited to study the clustering that forms. The proper method is to make automated screening of feature combinations, so to make deep analysis for Joint evaluation of multi-cluster validation indicators to satisfy cluster identification. This was shown both for the datasets of both experimental and analytical. The comprehensive search of global optimization for combining the cluster signal features and how this was performed. It goes on to explain that such comprehensive research methods are simplified. Still, it needs more computational effort and, consequently, suggests that promising features should be preselected before these methods are engaged. The technique is noted down as:

Step one – take all possible feature clusters with certain minimum set features, which can be later used for clustering;

Step two – apply all pre-processing solution for each feature cluster. It is important to know that modifications have to be done according to the problems dealt with. By combining the k-means algorithm and MacQueen's algorithm, clusters are found and calculated, conclusively devising that collects valid indices for four clusters. This number is therefore optimized numerically and selected for the current feature cluster. Low numerical complexity is base to be indices, as the lower the complexity, the more desirable they are for wide-scale automated discrimination. The study found that the primary data set architecture consists of ten features; five are annoying features. Annoying means a feature with steady distribution, which is independent of all other annoying and non-annoying features. After the experiments, the study found that the results of parameter-based recognition techniques that are mostly influenced by identifying and selecting features describing an object from a data set parameter-based recognition techniques are predominantly influenced by the definition and selection of the features describing an object of a dataset. The results were that out of six datasets, the algorithm found four, according to physical correlation proper classification for three clusters, in conclusion, that the method is significant for a limited number of extreme values and retrieves suitable discrimination. Therefore, the hierarchical cluster algorithms can be used to substitute for the currently used k-means [22].

2.3.5 The Use of Cluster Analysis of the Acoustic Emission Signals for Evaluating Damage Severity in Concrete

AE signals were investigated by focusing on the similarities between these signals and the correlation between AE from concrete structure cracks as it is suffering seismic wave propagation. Indeed, through this comparative method, it becomes an effective method for the recognition of fracture nodes. It was noted that using a cluster of two specific AE parameters was a prevalent method for studying AE signals. Sometimes the type of damage is known in advance of the experiments. The controlled pattern recognition is used as in the K-nearest adjacent way; however, sometimes, no data on attended clusters to be collected, in which case the popular k-means algorithm is usually applied. In these instances, Alternatively, the dimensions of a large data set can get reduced utilizing the principal component analysis (PCA). To discriminate numerically different categories of data, procedures known as neural network procedures can be used. One such procedure is the self-organizing map (SOM), Kohonen's transform, explicitly known to develop feature maps related to the vector analysis in the data set and reset these maps in a topologically significant correlation. Both PCA and SOM were utilized in the study as the basis for analysis with different parameters. The main focus was to derive a methodology for the investigation procedure that could handle many data as well as allow the scholars to evaluate particular corresponds about the data integrity of the tested specimen or structure. The PCA algorithm formula by a linear transformation contains the data's "principal components". It is a process that looks to find a specific accumulative percentage of total variations concluded by successive factors. The SOM model is a number of neural processing components in which the input pattern leads to the self-automated map, and each unit decided its processing. This is known as adaptation. The number of units that adapt decreases with time. Such a process leads to the cumulation of large clusters primarily and accurate input discrimination till the end; in order to compare the data across both methods, the natural values to be replaced by logarithmic values. It was determined that the data should be taken in three clusters strictly related to time-dependent parameters, as the three main sag areas showed in the results directly linked to the cluster areas. It was concluded that the use of these new methodologies was practically applicable, and it gives an evaluation. Still, it requires some modifications of the validation procedure to optimize a clear and significant analysis in the cases of outstanding amounts of data [23].

2.3.6 Real-Time Approach to Acoustic Emission Clustering

AE testing is of paramount importance, as it is like a sensor that detects the critical stage and worn before sudden failure. It is considered an essential part of the industry. However, the biggest hindrance to this technique is that it is undoubtedly and significantly affected by noise, so the processing has two main objectives: firstly, to screen AE signals arising on the background of noise, and secondly, to discriminate the active AE source from other sources. AE testing commonly requires the variable parameters to be set. However, in practice, many AE testing is not optimized strictly. They correlate together, making much of the work superfluous, as it cannot be used. Therefore they develop a significant and noise-resistant clustering task to be more difficult, hence the research [24] proposed scheme, which was explicitly designed for AE techniques considering actual testing conditions as it possesses features. The main thought was to use a developing clustering procedure in order to find natural clusters. Notably, a natural cluster was defined as a set of similar features without any essential substructure as determined by a chosen scale. However, this increases the cost, as the solution based on low-cost assumptions can be made in more runs. During the experiment process, it is considered necessary for a serious assessment of the noise level and to create statistical calculations. To retrieve helpful information to further the research process. The outliers can significantly endanger the clustering process, so it is important to ensure the correct balance of the screening process. Using the Silhouette validation technique, the Davie-Boulding validity index, and the Dunn validity index, the new proposed to ASK algorithm was tested.

The method is true-time and was compared to the algorithms k-means, normal algorithms, and c-means, fuzzy algorithms. During the experiments, it was proven that low energy AE signals are received sporadically during the test process. However, higher AE energy was received at the start and end of the test. The energy increases as crack initiation instant according to internal kinetic energy, and after that, it will decrease as the crack propagation. The results showed that all normal AE signals could be categorized into three categories with mean PSD shapes. Cluster 1 is shown towards the beginning of the test, and it is assumed logically, this cluster initiates with plastic deformation; cluster 2 is shown just after one cycle can be express brittle fracture; cluster 3 is shown as the final cluster, and it is compatible to the fatigue crack as it is created in a zig-zag shape similar to ductile crack propagation. The ASK method agrees with the results found by the k-means method. However, the fuzzy c-means gives different results, showing that ASK and k-means methods both give results better than the fuzzy c-means, which is undoubted [24].

2.3.7 Cluster Analysis of Acoustic Emission for Delaminated Glass Fibre Epoxy

Acoustic emission technology can monitor the damage process in real-time and provide helpful information for understanding the damage evolution of the composites. It was determined that composite specimens started at a 19-32% lower stress level than the stress level at the final fracture. AE can be used to consider the degradation and delamination of materials. The novel AE-based methods are more applicable than conventional methods for characterization of the delamination. It was agreed that complementary monitoring technology could provide abundant results compared with single AE technology. The two complementary technologies are 'combining AE with DIC, which were used together 'to monitor compressive buckling behaviours of the delaminated glass fibre epoxy composites. It was also suggested to base the cluster analysis upon the k-means algorithm and principal component analysis. Four types of composite specimens were prepared in the research, and five specimens of each type of composited laminates were employed. Due to the test conditions, it was not possible to perform the preferred method of two sensors, so only one RS-54A AE sensor was connected to the specimens with a wave-guide rod and preamplifier to monitor the compressive experiments. Similar relationships were found between the specimens in the experiments; however, the results indicate that the delamination defects result in reducing the ultimate strength of the composites. As with all AE experiments, noise signals were found. Here, there were many noise signals, the primary of which are stretching machine and fixtures. After carefully selecting three parameters, the k-means algorithm was used to find that the optimal number of clusters for all types of composite specimens is three. The clusters were found at different frequencies – cluster 1 was the lowest and corresponded to the matrix cracking stage, cluster 3 was the highest compared to the delamination and fibre breakage stage, and cluster 2 was in the middle and 'corresponds to the fibre/matrix debonding stage. It was determined that both damage patterns, matrix cracking and fibre/matrix debonding occur simultaneously. Eventually, the conclusion is that combining the two methods of AE and DIC is helpful to study the deformation and damage evolution of composites during compressive tests. Indeed, it established that this effective technology is a better way to accurately monitor the deformation, damage evolution and failure mechanism of composites [18].

2.4 Study of the AE Signals Characteristics

The project was proposed by DNV and is a continuation of the previous work done by Dawood and Nguyen [1], [2].

In 2019 Dawood [1] applied a three-point bending test and collect and did the analysis using AE win software. The work also included the limitation of AE testing as a structural health monitoring tool.

In 2020 Nguyen [2] used the data from Dawood's test and applied parameter-based analysis on these data. The parameters that were calculated are presented in Table 1 and Figure 1. Based on the calculated parameters from the analysis, the signals were then mapped under three types of signals; description of the three types of signals are presented in Table 2

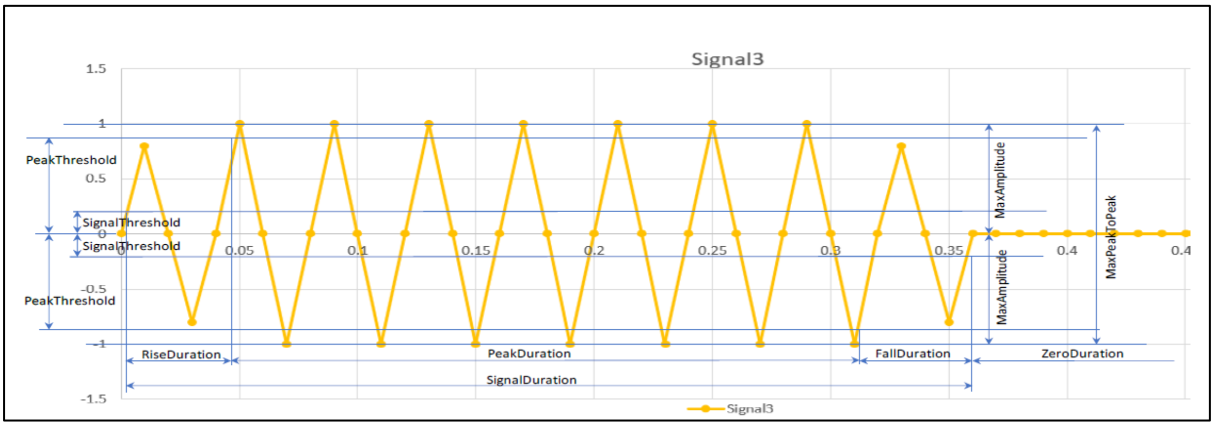


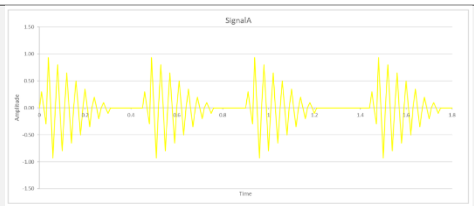
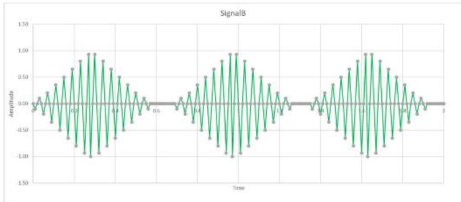
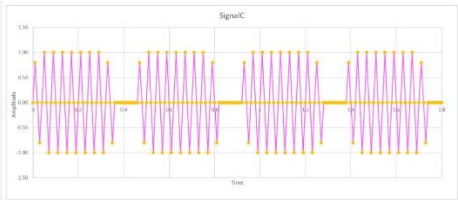
Figure 1 Signal parameters [2]

Table 1 Parameters used in the parametric analysis of signals

Parameter	DESCRIPTION
START	Start time of signal when the signal exceeds the signal threshold (SThres)
END	End time of the signal, i.e. when the signal is below the signal threshold for a duration longer than ZDuration
DURATION	Signal duration (End - Start)
STHRES	Signal threshold; a signal value above this threshold is considered a signal. Input parameter SignalThresholdInPercent
PTHRES	Peak threshold: fraction of maximum signal value. Input parameter PeakThresholdInPercent
ZDURATION	Duration when the signal is below SThres
MEAN	Average signal amplitude
MAX	Largest positive signal amplitude
MIN	Smallest negative signal amplitude
MAXAMP	Absolute max signal amplitude
MAXAMPTIMESTAMP	Timestamp of the absolute max signal amplitude
PEAK2PEAK	Distance between max and min signal amplitude

NZC	Number of zero-crossings between Start and End
RISEDURATION	Duration taken from signal SThres to PThres
FALLDURATION	Duration taken from signal PThres to SThres
PEAKDURATION	Duration above PThres
TYPE	Waveform classification

Table 2 Description of signal types referred in the analysis [2]

SIGNAL TYPE	**DESCRIPTION**	
TYPE A	• Fast rise duration • Very short peak • Long fall duration	
TYPE B	• Equal rise and fall duration • Short peak duration	
TYPE C	• Fast rise and fall duration • Long peak duration	

Nguyen.[2] had the output of the signal processing as a text file, as shown in Figure 2. The process followed to come up with the table in Figure 2 was:

1. Exporting the waveform data from AE Win is a comma-separated value format
2. Merging all the data in one data frame and exporting the data frame into comma-separated value format
3. Removing the pretrigger data, which is 256 μs in every start of waveform data
4. Start looping through the merged data frame and identifying signals. All signals are starting and ending by the amplitude of zero
5. Giving the found signal identification and then calculating the parameters of the signal shown in Table 1
6. Mapping the signals found under the three types shown in Table 2

A script was developed to perform the process, and the output of the process was a text file, as shown in Figure 2.

ID	Start	End	StartRow	EndRow	Duration	SThres	PThres	ZDuration	Mean	Max	Min	MaxAmp	MaxAmpTimestamp	Peak2Peak	NZC	CHMA	RiseDuration	FallDuration	PeakDuration	Type	ID
1	9777981	9778036	1	56	55	20.00	80.00	50	-0.19420	25.63477	-24.71924	25.63477	9777982	50.35401	10	8	1	53	1	A	1
2	9778038	9778098	58	118	60	20.00	80.00	50	-1.41907	22.58301	-23.80371	23.80371	9778043	46.38672	17	14	10	49	1	A	2
3	9778587	9778638	607	658	51	20.00	80.00	50	-0.70011	15.56396	-20.75195	20.75195	9778587	36.31591	13	11	0	46	5	A	3
4	10275101	10275235	771	905	134	20.00	80.00	50	-0.80393	37.53662	-35.70557	37.53662	10275146	73.24219	36	28	45	88	1	Unknown	4
5	10275252	10275491	922	1161	239	20.00	80.00	50	-0.81976	32.04346	-28.68652	32.04346	10275253	60.72998	57	35	1	237	1	A	5
6	10275529	10275612	1199	1282	83	20.00	80.00	50	-1.09569	21.05713	-25.93994	25.93994	10275529	46.99707	17	16	0	49	34	A	6
7	10275681	10275849	1351	1519	168	20.00	80.00	50	-1.00999	27.77100	-28.07617	28.07617	10275773	55.84717	44	33	88	79	1	Unknown	7
8	10904240	10904357	1543	1660	117	20.00	80.00	50	-1.00943	32.04346	-28.68652	32.04346	10904284	60.72998	27	28	44	72	1	Unknown	8
9	10904390	10904490	1693	1793	100	20.00	80.00	50	-0.58899	21.66748	-24.10889	24.10889	10904436	45.77637	27	19	46	53	1	Unknown	9
10	10904562	10904628	1865	1931	66	20.00	80.00	50	-1.09123	14.95361	-22.58301	22.58301	10904578	37.53662	12	10	16	49	1	A	10
11	10904906	10904961	2209	2264	55	20.00	80.00	50	-1.09308	20.44678	-23.80371	23.80371	10904907	44.25049	12	12	1	53	1	A	11
12	10995377	10995431	2311	2365	54	20.00	80.00	50	-1.39024	25.63477	-26.24512	26.24512	10995381	51.87989	12	7	0	49	5	A	12
13	11409425	11409726	3851	4152	301	20.00	80.00	50	-0.79589	45.77637	-38.45215	45.77637	11409470	84.22852	76	32	45	255	1	A	13
14	11409734	11409814	4160	4240	80	20.00	80.00	50	-1.14441	18.00537	-22.88818	22.88818	11409735	40.89355	16	22	1	78	1	A	14
15	11409848	11409903	4274	4329	55	20.00	80.00	50	-0.18865	21.66748	-20.14160	21.66748	11409848	41.80908	11	17	0	49	6	A	15
16	11409904	11409961	4330	4387	57	20.00	80.00	50	-0.94765	18.00537	-23.80371	23.80371	11409904	41.80908	14	12	0	49	8	A	16
17	11409966	11410017	4392	4443	51	20.00	80.00	50	-0.76593	24.10889	-16.17432	24.10889	11409966	40.28321	15	4	0	17	34	A	17
18	11410055	11410153	4481	4579	98	20.00	80.00	50	-0.60412	26.55029	-25.93994	26.55029	11410098	52.49023	20	17	43	54	1	Unknown	18
19	11662689	11662745	4621	4677	56	20.00	80.00	50	-1.40054	25.02441	-32.04346	32.04346	11662690	57.06787	12	5	1	54	1	A	19
20	11663103	11663154	5035	5086	51	20.00	80.00	50	-0.96938	20.14160	-18.92090	20.14160	11663103	39.06250	10	15	0	45	6	A	20
21	11682701	11682752	5392	5443	51	20.00	80.00	50	-1.00528	13.73291	-22.88818	22.88818	11682701	36.62109	9	5	0	49	2	A	21
22	11731011	11731236	6161	6386	225	20.00	80.00	50	-0.85178	39.97803	-63.78174	63.78174	11731016	103.75977	50	7	5	219	1	A	22
23	11731298	11731574	6448	6724	276	20.00	80.00	50	-0.95865	34.48486	-38.45215	38.45215	11731434	72.93701	62	44	136	139	1	Unknown	23
24	11731639	11731710	6789	6860	71	20.00	80.00	50	-1.31956	22.27783	-21.97266	22.27783	11731643	44.25049	14	22	4	66	1	A	24
25	11850345	11850533	6931	7119	188	20.00	80.00	50	-1.17363	52.79541	-44.25049	52.79541	11850350	97.04590	43	16	5	182	1	A	25
26	11850709	11850976	7295	7562	267	20.00	80.00	50	-0.96696	33.26416	-33.56934	33.56934	11850759	66.83350	64	47	55	211	1	A	26
27	11850980	11851113	7566	7699	133	10.56	42.24	50	-0.95912	22.58301	-25.93994	25.93994	11851061	48.52295	32	35	81	51	1	Unknown	27
28	11858739	11858967	7701	7929	228	20.00	80.00	50	-0.91553	70.80078	-107.42188	107.42188	11858746	178.22266	53	7	7	219	2	A	28
29	11859024	11859507	7986	8469	483	21.48	85.94	50	-0.87825	49.43848	-61.95068	61.95068	11859165	111.38916	116	31	141	341	1	A	29
30	11982771	11982822	8471	8522	51	20.00	80.00	50	-1.04119	14.03809	-25.32959	25.32959	11982771	39.36768	11	2	0	49	2	A	30
31	12115979	12116158	9242	9421	179	20.00	80.00	50	-1.17126	41.19873	-43.64014	43.64014	12115980	84.83887	41	15	1	177	1	A	31
32	12116347	12116476	9610	9739	129	20.00	80.00	50	-1.03618	29.29688	-30.21240	30.21240	12116393	59.50928	32	24	50	78	1	Unknown	32
33	12116507	12116606	9770	9869	99	20.00	80.00	50	-1.12514	18.00537	-21.97266	21.97266	12116508	39.97803	26	25	1	97	1	A	33
34	12116614	12116665	9877	9928	51	20.00	80.00	50	-0.77191	18.61572	-20.44678	20.44678	12116614	39.06250	12	14	0	3	48	A	34
35	12116686	12116746	9949	10009	60	8.73	34.91	50	-1.27665	22.58301	-21.97266	22.58301	12116690	44.55567	14	19	4	55	1	A	35
36	12232695	12233463	10011	10779	768	38.02	152.10	50	-0.77009	170.89844	-190.12451	190.12451	12232712	361.02295	217	16	13	750	5	A	36
37	12429082	12429133	10782	10833	51	20.00	80.00	50	-0.88561	14.03809	-24.10889	24.10889	12429082	38.14698	9	3	0	49	2	A	37
38	12829021	12829072	11551	11602	51	20.00	80.00	50	-1.43612	16.17432	-25.02441	25.02441	12829022	41.19873	11	4	1	49	1	A	38

Figure 2 Sample of the output file from of parameters calculation

Chapter 3 Methodology

3.1 Research Objective

The research aimed to study the characteristics of the AE signals by analyzing the signal parameters and relating them to the changes in the material during loading. The context is to identify the AE signals referring to yielding in the material. The data used in the research were from two sources; the primary data set was from an experimental work done in 2021 as a part of this thesis work. The secondary data set was from an experimental work done in 2019 by Dawood [1]. Both experiments had mainly the same approach with minor differences. The summary of the differences between the two experiments is presented in Table 3.

Table 3 Comparison of 2019 and 2021 experiments

<table>
<tr><th>YEAR</th><th>2019 [1]</th><th>2021</th></tr>
<tr><td>NUMBER OF SPECIMENS</td><td>9</td><td>9</td></tr>
<tr><td>DESCRIPTION OF SPECIMENS</td>
<td><table>
<tr><td>Specimen ID</td><td>Dimensions
(widthXthickness X length) mm</td></tr>
<tr><td>A1</td><td>30 X 15 X 500</td></tr>
<tr><td>A2</td><td>15 X 30 X 500</td></tr>
</table></td>
<td><table>
<tr><td>Specimen ID</td><td>Dimensions
(width X thickness X length) mm</td></tr>
<tr><td>PB</td><td>120 X 15 X 500</td></tr>
<tr><td>B1-1</td><td>30 X 15 X 1000</td></tr>
</table></td></tr>
</table>

<table>
<tr>
<td></td>
<td>
<table>
<tr><td>A3</td><td>30X 15 X 500</td></tr>
<tr><td>B1</td><td>30 X 20 X 500</td></tr>
<tr><td>B2</td><td>30 X 20 X 500</td></tr>
<tr><td>B2R</td><td>20 X 30 X 500</td></tr>
<tr><td>C1</td><td>30 X 15 X 500</td></tr>
<tr><td>C2</td><td>30 X 15 X 1000</td></tr>
<tr><td>C2R</td><td>30 X 15 X 1000</td></tr>
</table>

All Specimens were base material (no welds)
Specimens A1 and A2 had preservation coating

</td>
<td>
<table>
<tr><td>B1-2</td><td>30 X 15 X 1000</td></tr>
<tr><td>B1-3</td><td>30 X 15 X 1000</td></tr>
<tr><td>B2-1</td><td>30 X 15 X 3000</td></tr>
<tr><td>B2-2</td><td>30 X 15 X 3000</td></tr>
<tr><td>B3-1</td><td>30 X 15 X 1000</td></tr>
<tr><td>B3-2</td><td>30 X 15 X 1000</td></tr>
<tr><td>B3-3</td><td>30 X 15 X 1000</td></tr>
<tr><td>BD</td><td>30 X 15 X 1000</td></tr>
</table>

Specimens B1, B2 and B3 were welded
Specimens B3 had a coating on the bottom surface

</td>
</tr>
<tr>
<td>NUMBER OF SENSORS</td>
<td>2</td>
<td>

3 sensors for specimens B1 and B3 (2 Type R15a and 1 type R15X)
5 sensors for specimen B2 (2 Type R15a and 3 type R15X)

</td>
</tr>
<tr>
<td>LOADING SETUP</td>
<td>3 point loading test</td>
<td>4 point loading test</td>
</tr>
<tr>
<td>LOGGING TIME</td>
<td>1024 μs</td>
<td>2048 μs</td>
</tr>
</table>

In the 2021 experiment, the four-point test was implemented to avoid applying load directly on the weld, while in 2019, the loading setup was 3-point bending test as the specimens did not have any welds. Also, the number of the sensors was different in the two experiments, as shown in Table 3.

3.2 Methodological Approach

The approach followed in this research was as mentioned in Table 4.

Table 4 Summary of the methodological approach

APPROACH	DESCRIPTION
PRE-STUDY REPORT	A report was containing the plan of the research. It was submitted in the early weeks of the project.
STUDYING THE PREVIOUS THESIS BY BY DAWOOD AND NGUYEN[1], [2]	Covering in details the work done in the thesis on the same topic before. As this year thesis is connected and building upon the previous work done in 2019 and 2020
LITERATURE STUDY ON THE AE SIGNALS POST-PROCESSING METHODS	Performing a literature study to summarize some of the used methods to analyze AE signals
DEVELOP PROPOSAL FOR A SMALL SCALE TEST	To Prepare a test proposal to simulate actual AE signals as data to be used in the thesis.
PERFORMING THE TEST	Performing the test with the help of DNV's lab team
EVALUATE PROCESSING ALGORITHMS	From the literature found, evaluating the processing tools used in the publications
DEVELOPING POST-PROCESSING TOOLS	Coding to prepare an automated analysis tool for processing the data
PROCESSING THE RECORDED AE SIGNALS USING THE DEVELOPED PROCESSING TOOLS	Applying the developed tool to process the data collected from the tests
EVALUATING THE AE SIGNALS	Evaluating, justifying and understanding the findings
REPORTING THE FINDINGS	Reporting the work done and achieved in this thesis

3.3 Data Collection

The primary research data were collected by applying a lab experiment. The experiment was loading a steel specimen to stress the test specimen and ensure that the specimen crossed the yielding point. The AE signals were recorded using AE sensors. The logging was then exported in waveform data points (potential difference vs time) from the logging software into comma-separated value '.csv' format. The logging software was set up to record the 2048 μs waveform of the obtained signals. More details about the experiment are presented in Chapter 4 and (Appendix A - Test Proposal).

Chapter 4 Experimentation

4.1 General

This chapter summarizes the test done for this research project. One of the main research tasks was to develop a small scale test program for various samples and perform the test. The goal was to collect AE signals from different specimens that could represent actual structure. As actual structures will have welded members and coated members, the specimen prepared for the test were welded, and some were coated. This chapter summarizes the preparation done for the specimens, explain the experiment setup and lists the dimensions of the specimen. (Appendix A - Test Proposal) shows the test proposal developed and sent to the lab as a guideline containing a detailed test procedure.

The loading applied was a four-point bending test, as shown in Figure 3. The four-point bending test was done according to ASTM D790-15.

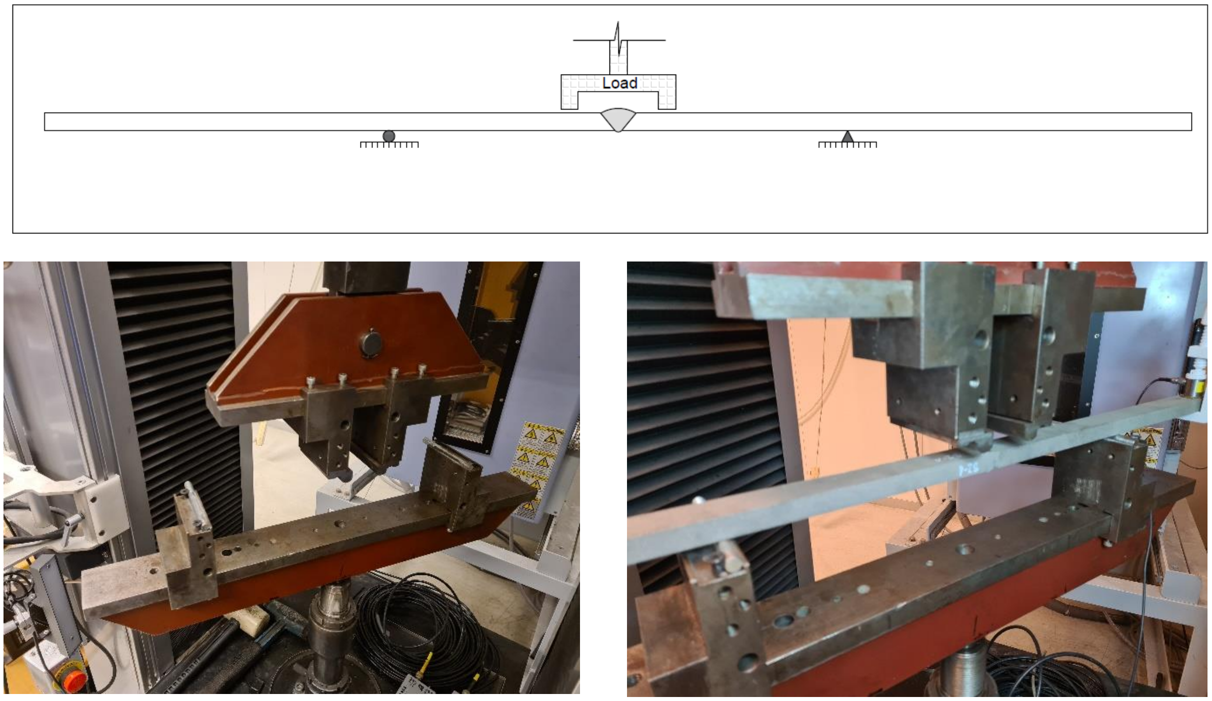


Figure 3 Four-point setup sketch and photos

A pencil lead break test (PLB) was done on the specimen before the loading test to ensure that the channels are recording. The PLB test was applied following the guideline in ASTM E298317:2019. Figure 4 shows the pencil lead break test done in the experiment.

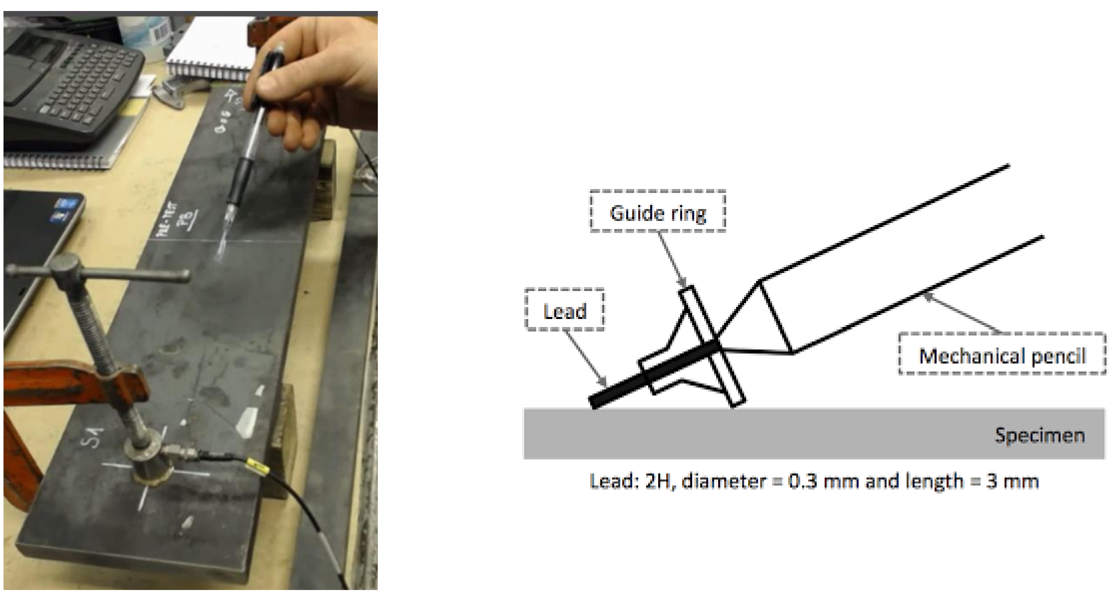


Figure 4 Pencil lead break test

4.2 Preparation

The dimensions of the ordered raw material were 120 mm X 15 mm X 6 m Flat steel in quality S355J2 according to EN 10025-2 and NORSOK M-120 Rev. 5 MDS Y05 of Yield strength 435 MPa and ultimate strength 534 MPa. The raw material was then cut, welded and coated according to the cutting plan mentioned in the test proposal in (Appendix A - Test Proposal). The cutting was the water jet cutting method. The welding was Metal Inert Gas (MIG) and Metal Active Gas (MAG). The preservation coating was removed using sandblasting. The coated specimens had the coat on one surface only, and the coat was subsea coating class 7B / 878. Figure 5 shows the specimen with a coat on one surface.

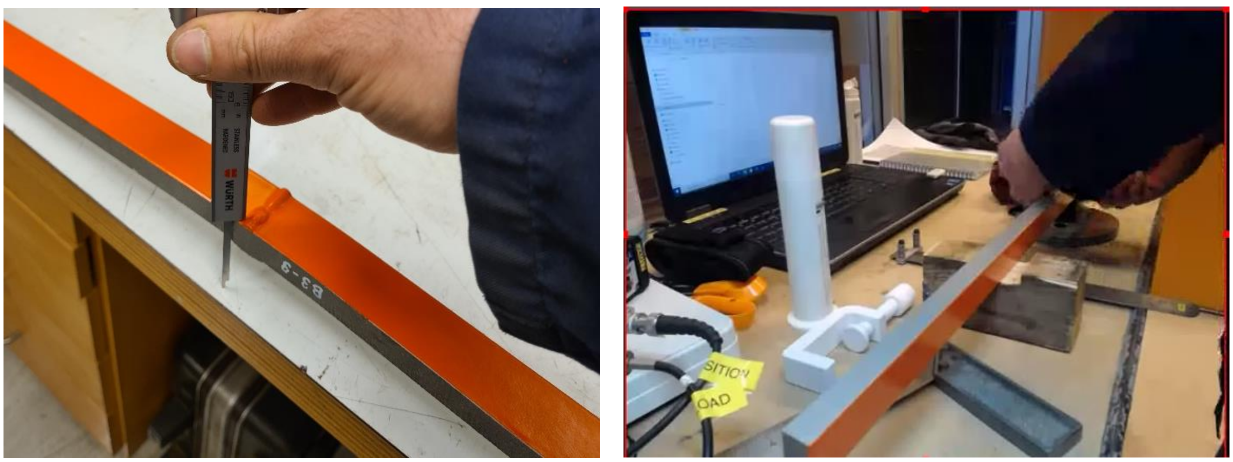


Figure 5 Specimen with a coat

4.3 Specimens

Table 5 lists the dimensions of the specimens, the type of experiment applied on the specimen and reference to the setup sketch. Specimens B1-1 and B1-2 had repetition for the test since the applied load was low and they did not show any deflection. The files for the repeated test were saved under the names B1-1_Rep and B1-2_Rep.

Table 5 Specimen dimensions and experiment type

SPECIMEN ID	DIMENSIONS	TEST DESCRIPTION	TEST SETUP
PB	120 mm X 15 mm X 500 mm	2 times PLB test only	Figure 6
B1-1	30 mm X 15 mm X 1 m	2 times PLB test Four-point test (loading to yield)	Figure 7
B1-2	30 mm X 15 mm X 1 m	2 times PLB test Four-point test (loading to yield)	Figure 7
B1-3	30 mm X 15 mm X 1 m	2 times PLB test Four-point test (loading to yield)	Figure 7
B2-1	30 mm X 15 mm X 2.5 m	2 times PLB test Four-point test (loading to yield)	Figure 8
B2-2	30 mm X 15 mm X 2.5 m	2 times PLB test Four-point test (loading to yield)	Figure 8
B3-1	30 mm X 15 mm X 1 m	2 times PLB test Four-point test (loading to yield)	Figure 9
B3-2	30 mm X 15 mm X 1 m	2 times PLB test Four-point test (loading to yield)	Figure 9
B3-3	30 mm X 15 mm X 1 m	2 times PLB test Four-point test (loading to yield)	Figure 9
BD	30 mm X 15 mm X 1 m	Ball drop 2 times PLB test Four-point test (loading to yield)	Figure 10 Specimen BD test setup

4.4 Experiments Setup

Figure 6 - Figure 10 shows sketches for the setup of the experiments. More details about the setup are mentioned in (Appendix A - Test Proposal).

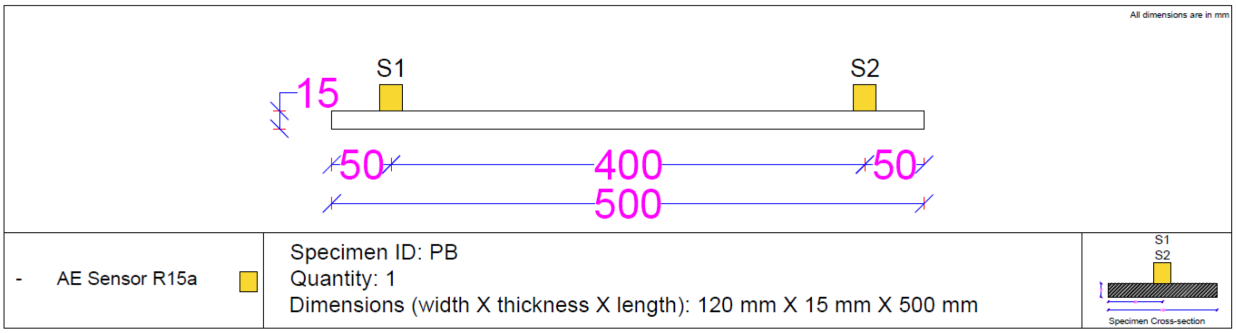


Figure 6 Specimen PB test setup

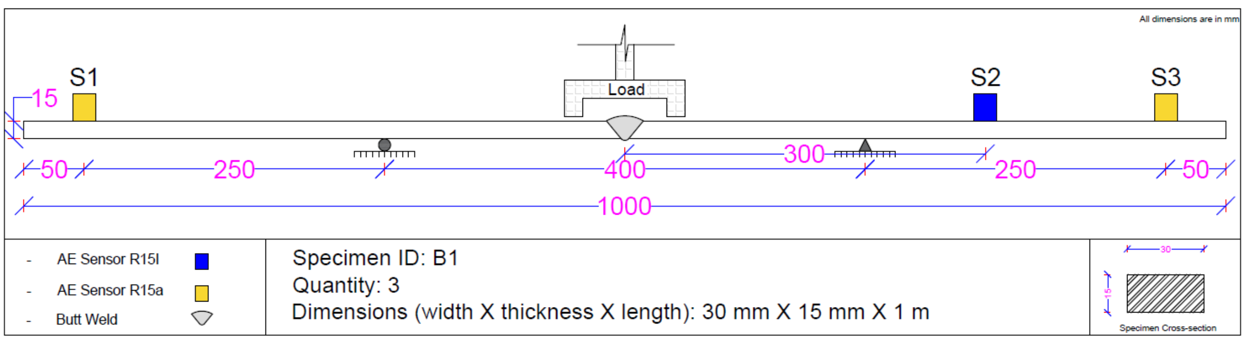


Figure 7 Specimen B1 test setup

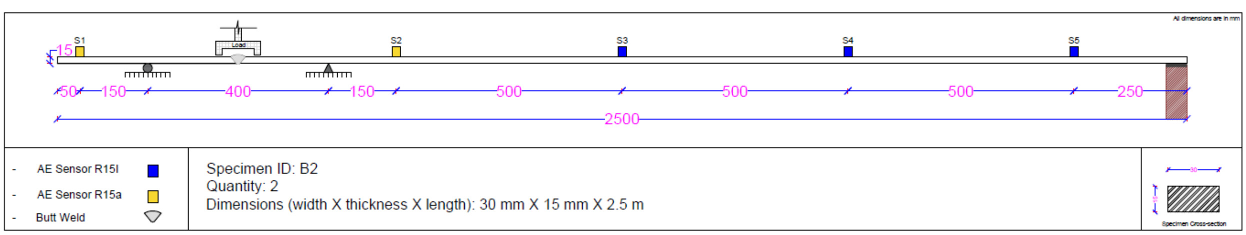


Figure 8 Specimen B2 test setup

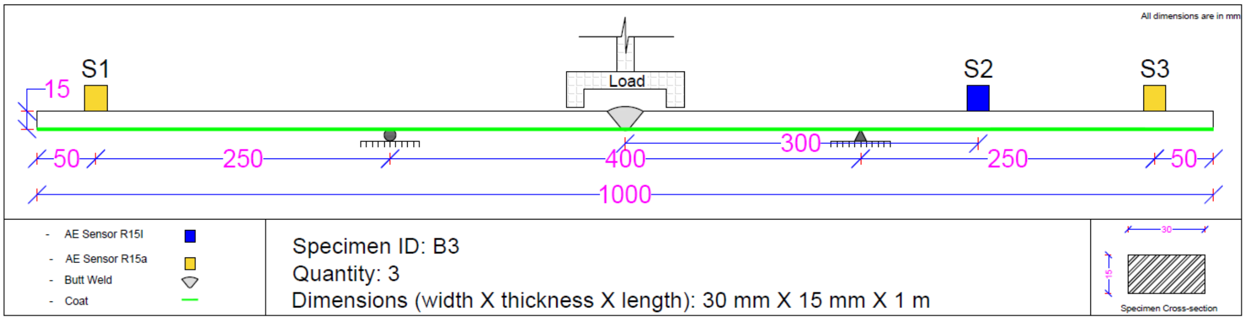


Figure 9 Specimen B3 test setup

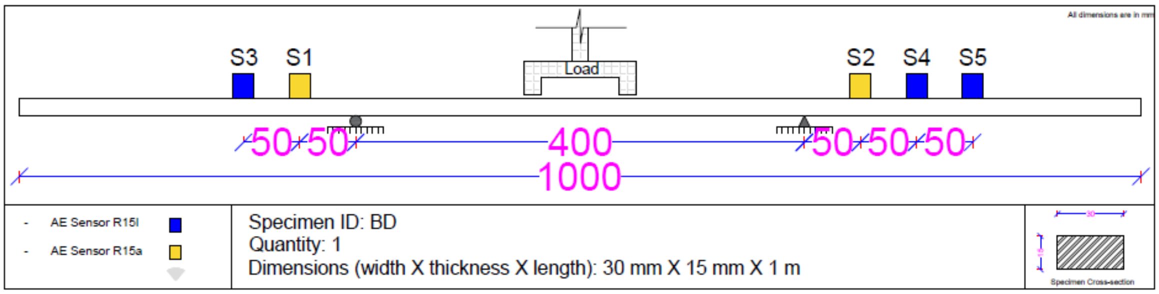


Figure 10 Specimen BD test setup

4.5 Summary

The testing was an essential task for this research topic, and the aim was to record AE signals from the specimen with weld and coats. A test proposal was developed and sent to the lab team, and the author was following up closely on all the details of the test. A summary table was then created to record all the notes during testing. The developed test proposal and the summary table can be found in (Appendix A - Test Proposal)

Chapter 5 Results and Discussion

5.1 General

This chapter describes the analysis methods used, their method of implementation, the aim of using them and finally, the results from using these methods. The analysis carried out in this research focused on developing automated tools to correlate signals of the same AE event but captured by different sensors. The correlation can help identify the AE events' location and therefore filter out AE signals emitted far from the weld. Since the background of the research is to identify the signals at the yielding point and the test done this year was a four-point bending test, the yielding signals are occurring in the middle of the span between the supports. Figure 11 shows the location of the AE events of interest to be used in further research for identifying the yielding signals. The work explained in this chapter is based on the results from a previous master thesis [2], which contains the signal parameters in statistics tables. Two of these tables were taken as the input of the correlation from Nguyen's [2] work, of which one comprises data recorded by one channel. The statistics table done 2020 thesis [2] did not include the energy as a parameter, so it was added in this thesis work. Table 6 shows the analysis title, the method of implementation and the expected outcome, and later in this chapter, a detailed explanation of the method and its evaluation are discussed.

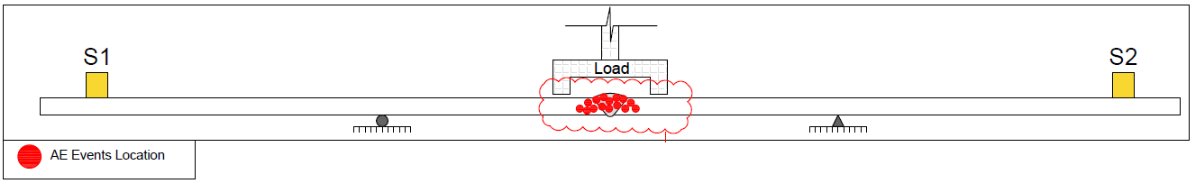


Figure 11 AE events location needed to be considered in the analysis

Table 6 Analysis approach

METHOD	METHOD OF IMPLEMENTATION	GOAL
SIGNAL STRENGTH AND LOAD FOR WELDED SPECIMENS	Developing a Python script to plot the load vs time and signal strength vs time on the same plot.	Correlate AE events with the load curve, compare the AE events of the welded specimens, and compare them with unwelded specimens.
CORRELATION (TIMESTAMP)	Developing a Python script to match the signals of two channels based on the timestamp. Using the start time of the signals of two different channels, the signals were matched based on the closest timestamp.	To identify the location of the AE signals and filter out all the signal that are not near the weld area.
CORRELATION USING DEFINED TIME RANGE	Developing a Python script to match signals from two channels within a defined time range.	To identify the location of the AE signals and filter out all the signal that are not near the weld area.
CORRELATION USING DEFINED TIME RANGE (NUMBER OF ZERO-CROSSING AND ENERGY)	The percentage difference is calculated for two parameters: the number of zero-crossing (NZC) and the energy from the matched signals of the defined time range.	To enhance the approach used to define signal by a time range. Also, to better strictly match signals of the same AE event recorded by two different channels.

CORRELATION USING DEFINED TIME RANGE (CROSS-CORRELATION COEFFICIENT)	The cross-correlation coefficient is then calculated for the matched signals to identify the signals of the same AE event.	To enhance the approach of the correlation using the number of zero-crossing and energy. Also, to make robust reference for identifying signals of the same AE event.
CATEGORIZATION OF SIGNALS	Applying the script developed by Nguyen [2] on the tests performed in this thesis and summarizing the number of signals from each category.	To characterize the signal of the welded and coated specimens of the tests performed in this research.
SOURCE LOCATION	Explaining the approach followed in locating the sources of the AE signals.	To have a base for developing a robust approach for locating AE signals source.

5.2 Signal Strength and Stress vs Time for Welded Specimens

Developing a plot to visualise the amplitude of the AE signals about the load applied on the specimen is essential to connect signals to the stress stages. Therefore, an algorithm was developed to create plots similar to the plot shown in Figure 12, where it presents the load in (KN) vs time in (s) in red and the amplitude in (dB) vs time in (s).

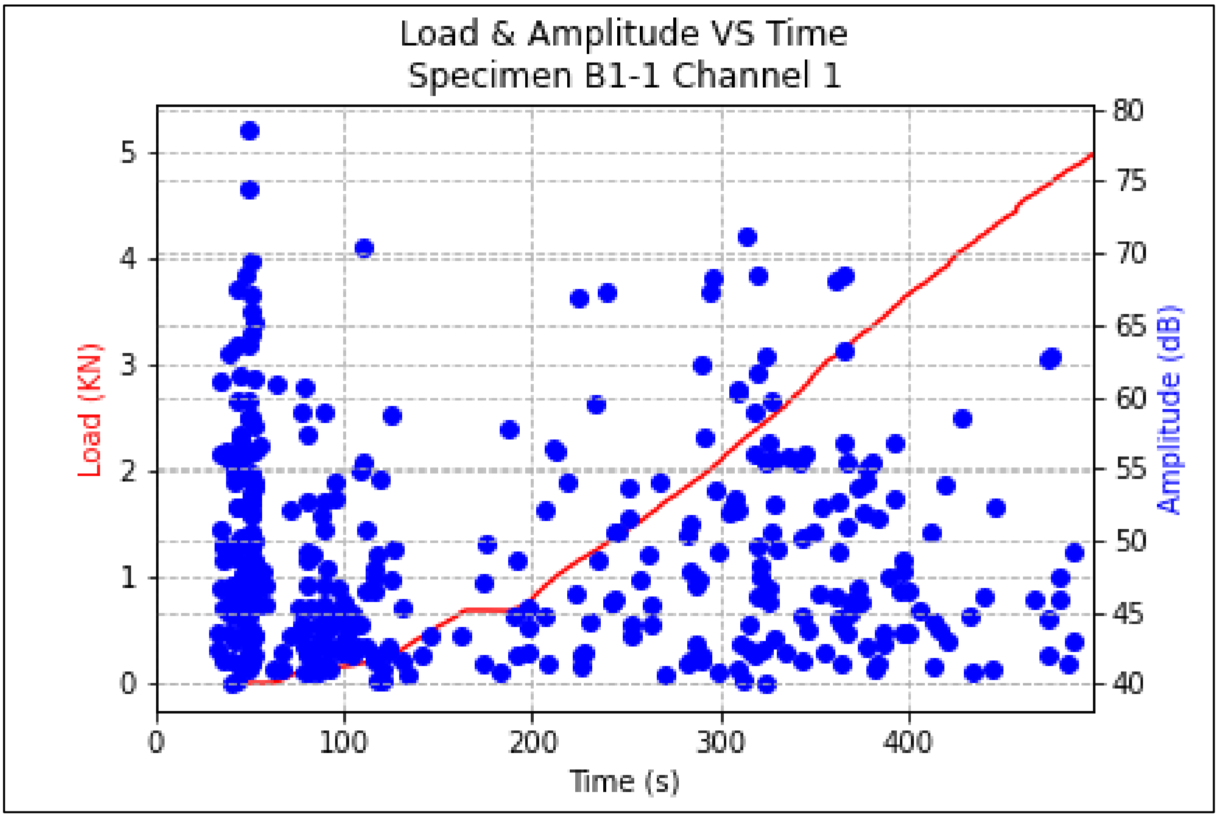
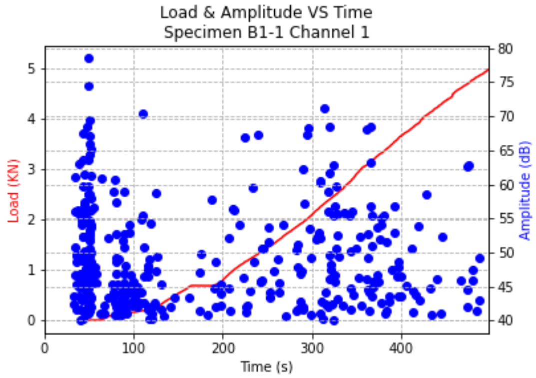


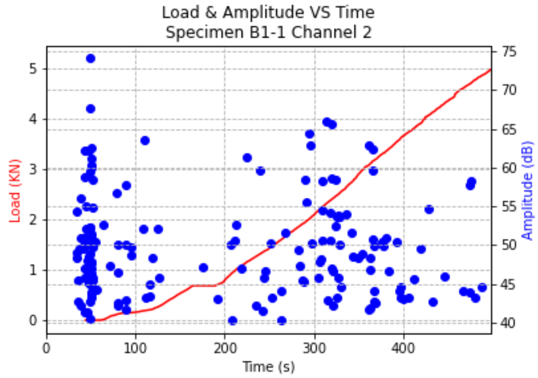
Figure 12 Sample output for the load and amplitude vs time plots

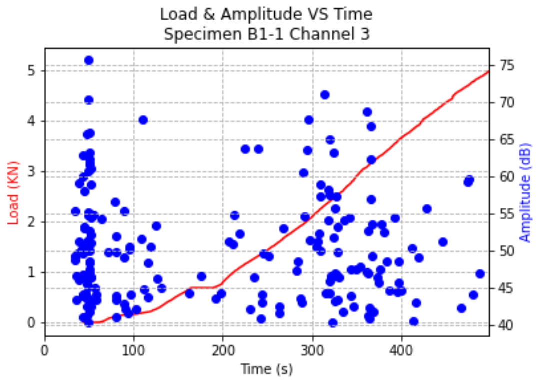
The script developed for these plots is presented in (Appendix E – Developed Code). The plots for all the test specimens can be found in (Appendix B - Signal Strength and Load VS Time Plots). Each specimen has a table that presents the plot for each channel and the specimen sketch, as shown in Table 7. The comparison between welded and unwelded specimens based on this result is presented in section 5.6.1.

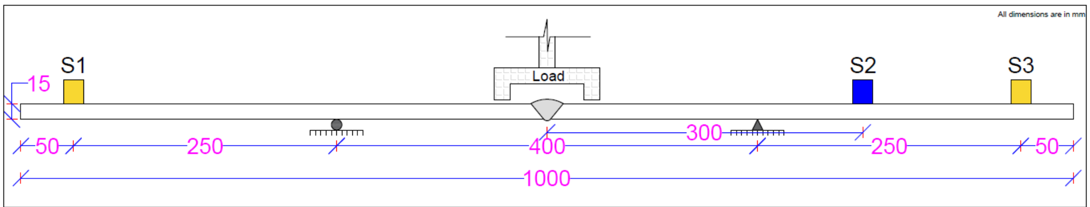
Table 7 Sample of result table in Appendix B

Specimen B1-1



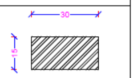






- AE Sensor R15I
- AE Sensor R15a
- Butt Weld

Specimen ID: B1
Quantity: 3
Dimensions (width X thickness X length): 30 mm X 15 mm X 1 m



Specimen Cross-section

Table 8 Signal Strength and load vs amplitude plots summary summarise the plots' trends in (Appendix B - Signal Strength and Load VS Time Plots). Here to note that the test was done with two different types of AE sensors, R15a and R15I. The sketches in the result table indicate the different sensors.

Table 8 Signal Strength and load vs amplitude plots summary

SPECIMEN	PLOTS TREND
B1-1	• The three channels showed similar trends. • Channels two and three had a very similar distribution for AE signals compared to channel one.
B1-2	• Channels one and two had a very similar distribution for AE signals compared to channel three. • Channel three had fewer signals compared to channels one and two.
B1-3	• The three channels had a similar distribution for the signals
B2-1	• Channels one and two had more signals than the other channels. • Channels three, four and five had fewer signals compared to channels one and two.
B2-2	• Channels one and two had very few signals. • Channels three, four and five had many more signals compared to channels one and two. • The signals in these channels one, two, and three were with low amplitude. These channels were far from the mid-span, so usually, the amplitude shall be lower. • During the testing for B2-1, it was noticed that channels three, four and five did not record many signals. Therefore, the threshold for recording was dropped to make sure they can record signals from specimen B2-2.
B3-1	• The three channels had similar distribution for the signals. • This was a coated specimen. The number of AE hits was not increased due to the coat.
B3-2	• The three channels had similar distribution for the signals.

	• This was a coated specimen. The number of AE hits was not increased due to the coat.
B3-3	• For some unknown reason, channel one did not record any signal; however, it was used afterwards and recorded. • This was a coated specimen. The number of AE hits was not increased due to the coat.
BD	• This specimen was an unwelded specimen, and generally, all the channels showed more signals. • Channels one and two showed fewer signals than the other channels. • Channels three, four and five showed many more signals, with many of them on similar amplitude and following in close time range.
B1-1_REP	• The three channels had similar distribution for the signals.
B1-2_REP	• The three channels had similar distribution for the signals.

5.3 Correlation of Signals

5.3.1 Timestamp

The first approach used in correlating signals from two different channels matched the signals using the start timestamp. The aim is to capture the same AE event by both channels and prove that the channels are recording the same signal shape. Since these signals travel very fast, matching AE signals emitted from the same source and recorded by two different channels could be done using timestamp difference. A python script was developed to find the nearest timestamp from the other channels. Meaning that if a signal in channel one has a start timestamp of '123456', the code will loop through the timestamps of the signals in channel one to find the nearest start timestamp to '123456'.

The algorithm of the developed script was

1. Read the statistics table the output from Nguyen's [2] work for the two intended channels.

2. Using the concept of binary search, select a target from any of the two channels and loop to find the minimum time difference.
3. Once the minimum time difference is found, write their data matched with signals from the two different channels in one row.
4. The process is done by selecting the targets from the first channel and repeated selecting the target from the second channel. This ensures that all the signals have a match.
5. Then the duplicated data from step 4 are removed since some data can be matched twice.
6. Then the time difference and the position where the signals were estimated is calculated.
7. Print the output from matching the signals, as given in Table 9.

Table 9 Sample output table from matching the start timestamp of two channels

	ID_C1	StartTime_C1	MaxAmpl_C1	ID_C2	StartTime_C2	MaxAmpl_C2	TimeDifference_0.1µs	AbsoluteTimeDifference_0.1µs	Position_m
0	1.0	3808417.0	4956.05469	1.0	3808419.0	4465.02686	-2.0	2.0	-0.0012
1	1.0	3808417.0	4956.05469	2.0	3995481.0	57.06787	-187064.0	187064.0	-112.2384
2	2.0	3869330.0	709.22852	1.0	3808419.0	4465.02686	60911.0	60911.0	36.5466
3	2.0	3869330.0	709.22852	3.0	3995786.0	34.79004	-126456.0	126456.0	-75.8736
4	3.0	3995486.0	39.36768	2.0	3995481.0	57.06787	5.0	5.0	0.0030
5	4.0	3995813.0	36.62109	3.0	3995786.0	34.79004	27.0	27.0	0.0162
6	5.0	3996022.0	31.43311	2.0	3995481.0	57.06787	541.0	541.0	0.3246
7	5.0	3996022.0	31.43311	4.0	3996077.0	32.34863	-55.0	55.0	-0.0330
8	5.0	3996022.0	31.43311	5.0	3996500.0	20.14160	-478.0	478.0	-0.2868

8. Finally, make plots to show the estimated source location of the signal. Figure 6 and Figure 7 show a sample of the output plot from the developed script. The plot shows the location of two sensors, where the blue refers to C1, and the magenta refers to C2. The plot also shows the maximum amplitude value recorded by both channels, where green refers to the maximum amplitude for C1 and red to the maximum amplitude for C2. The difference between Figure 13and Figure 14 is the location of the AE events. This was done by calculating the location by multiplying the time difference between the two channels by the longitudinal speed of the sound wave in steel 6000 m/s, as shown in Table 9.

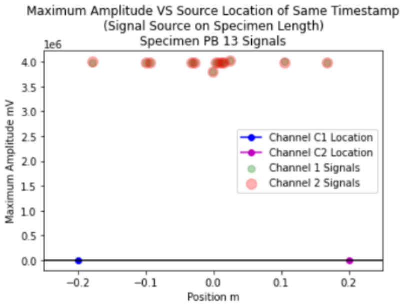


Figure 13 Sample output showing the expected location of AE events

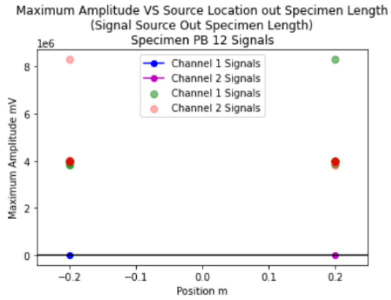


Figure 14 Sample output showing AE events expected to be not from the specimen

The script developed to achieve this part is documented in (Appendix E – Developed Code). However, the approach did not satisfy the planned result. This is because a single signal from a channel can have two or more signal from the other channel matched based on the closest timestamp. In contrast, only one signal from the other channel will be the true match that refers to the same AE event.

5.3.2 Defining a Time Range with Percentage of Number of Zero-Crossing and Energy

Since the previous approach had its disadvantage, another approach was developed that predefines a time range that the signals of the same event travelling to two different sensors will delay. As shown in Figure 15, an AE event occurred at a small distance to the left side of the centre, and the signal will spread in the material longitudinally to be captured by the two sensors since the AE event was not exactly in the centre so that S1 will be recording the signals before S2. Since the longitudinal speed of the sound wave in steel is known to be around 6000 m/s, and the length of the specimens is known, the time range can then be calculated. For the case in Figure 15, the distance between both channels is 900 mm. Then the time range can be between 0 to 150 μs.

$$Time = \frac{Distance}{Velocity}, Time = \frac{0.9\ m}{6000\ \frac{m}{s}} = 0.000150\ s\ = 150\ \mu s$$

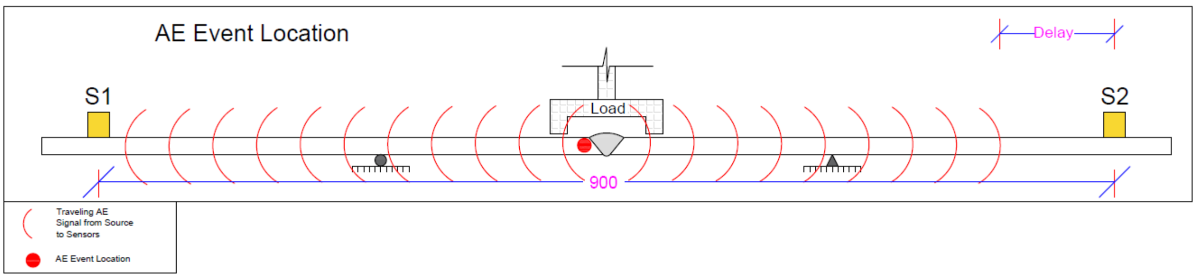


Figure 15 Sketch of AE event travelling to two sensors

The algorithm of the developed script was

1. Read the statistics table of the output from Nguyen's [2] work for the two intended channels.
2. Add a new column holding the channel number to be able to combine in one dataset.
3. Combine the data from the two channels in one dataset.
4. Sort the data based on the start time of the signals.
5. Using the defined time range, the difference in time of every two consecutive rows is calculated. If they are less than the time range and the data in the consecutive rows are from two different channels, then these two rows are moved in the same row.
6. Calculate the time difference the percentage difference for data from both channels for two parameters NZC and the energy.
7. Plot a bubble plot for the ratio of NZC and maximum amplitude vs the time difference as shown in Figure 16 and Figure 17. The bubble size refers to the maximum amplitude value; the big size bubble refers to the higher maximum amplitude. At the same time, the colour indicates the value with reference to the colour bar on the right side of the plot.

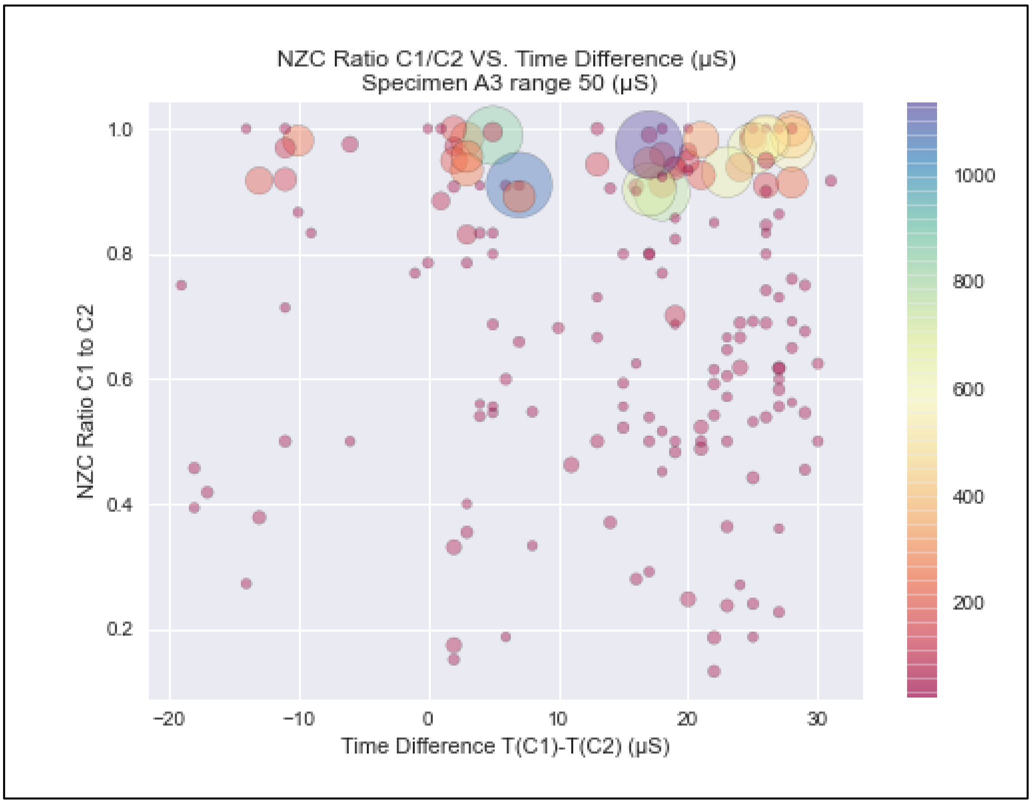


Figure 16 Sample output for the ratio of NZC maximum amplitude vs time difference

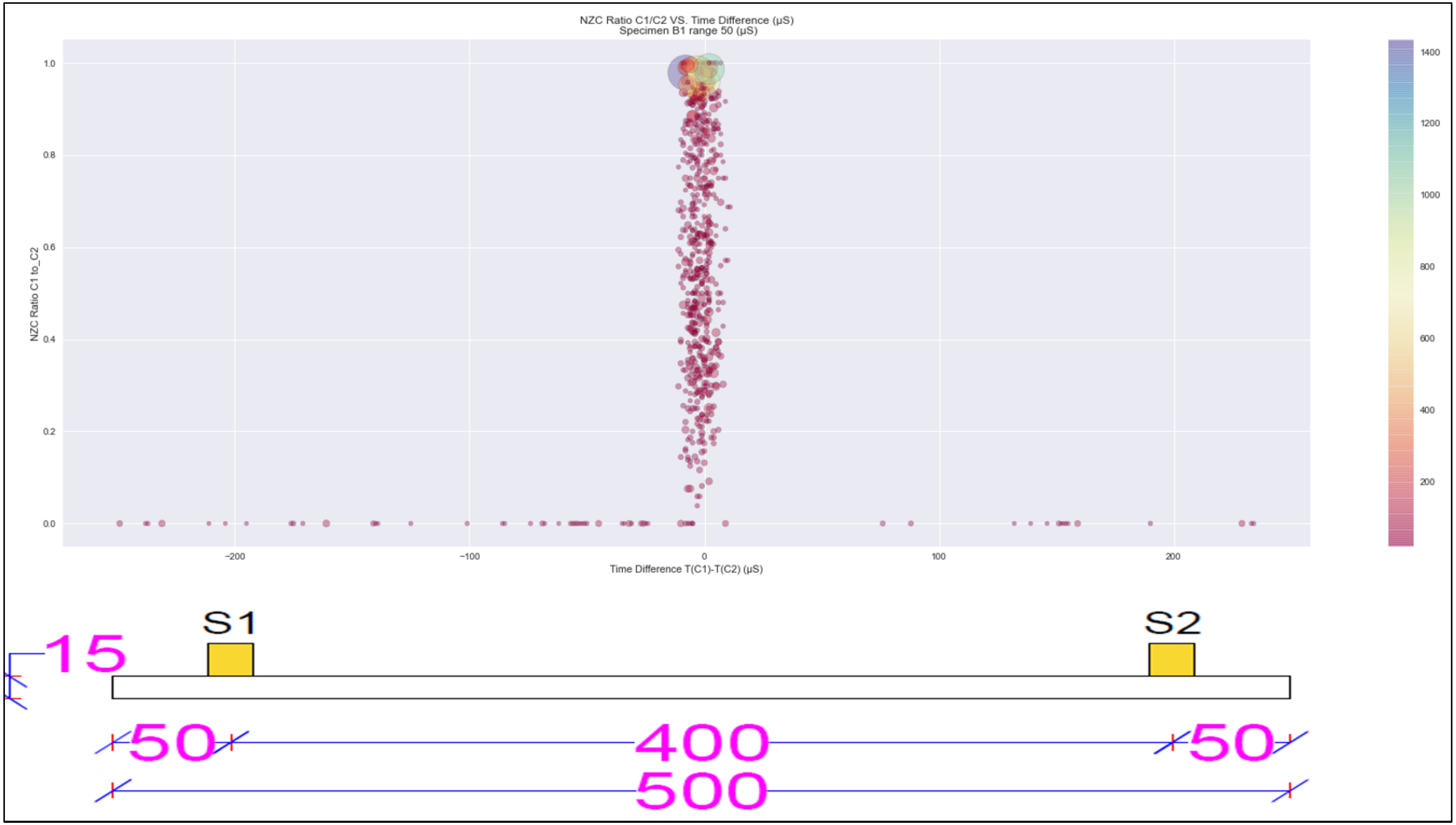


Figure 17 Sample output for the ratio of NZC maximum amplitude vs time difference with the specimen sketch – specimen B1 from 2019 test

This approach will match every signal since the AE events are happening very fast after each other. For example, signal X1 from channel one can be matched with signals Y1, Y2, Y3 from channel two since they are all within the defined time range. To define which Y1, Y2, Y3 is the actual match for X1, they are emitted from the same source. A parametric comparison shall be carried out. Many parameters can be used to match the signals from two channels. The two parameters that this research focused on to compare the signals are the NZC and the energy. The maximum amplitude cannot be a reference since it depends on the channel's sensitivity and how it is attached to the test specimen. Also, the amplitude depends on the distance between the source and the logging sensor. A percentage difference of the NZC and energy to be calculated where a high percentage indicates that signals are more likely to be from the same source that the signal source is located midway between the two sensors. The approach discussed in this section aimed to match signal based on the NZC and signal energy; however, these AE signals can be very similar since they come from a very close time range. In some cases, basing signals from different channels on the percentage difference and the energy cannot be robust enough. So another indication shall be used as an indicator of a strong match for signals from different channels.

5.3.3 Cross-Correlation Coefficient in the Time Domain

Adding another parameter to point at the AE event captured by two channels is needed to complete the matching analysis. A factor that is used in the signals analysis is the cross-correlation coefficient. The cross-correlation coefficient measures how two time-series signals are similar. The coefficient can have a value between -1 and 1, where values closer to 1 refer to a high correlation, values closer to 0 refer to a weak or no correlation and values close to -1 means inverse correlation. The cross-correlation coefficients highlight the maximum correlation of signals at a particular time lag by evaluating the correlation of two-time series shifted along with each other. The highest value of this shift is the value used for the maximum correlation. Mathematically the cross-correlation of two signals $f_1(t)$ and $f_2(t)$ can be expressed as

$$r_{xy}(\tau) = \int_{-\infty}^{\infty} f_1(t) \cdot f_2(t + \tau) dt \tag{1}$$

Then, normalizing the cross-correlation function to have time dependant coefficient.

The use of this correlation aims to point out the actual match of the signals. Consider the case in Table 10 Example for the cross-correlation approach. The case presented is similar to many cases in the output of the approach mentioned in section 5.3.2. Signal ID 1 from channel one had four matched signals from channel two within the defined time range. Based on the percentage of NZC and energy, it can be noticed that the best match will be row two or four. Next is to identify which of these two rows are the true match referring to the same AE event. The cross-correlation coefficient can be calculated for all the rows, and the row that will have a coefficient value closer to 1 has a higher chance to match the signals for the same AE event. The cross-correlation Python script can be found in (Appendix E – Developed Code).

Table 10 Example for the cross-correlation approach

INDEX	SIGNAL ID C1	SIGNAL ID C2	%NZC	%ENERGY	CROSS-CORRELATION COEFFICIENT
1	1	1	50	50	0.3
2	1	2	80	60	0.6
3	1	3	40	60	0.2
4	1	4	90	80	0.95

5.3.4 Visualizing the Signals of the Same AE Event

After determining the AE signals with the highest cross-correlation coefficient, a final step is to plot the two signals in the same plot and observe the same AE event signals. This will lead to more confidence in the result. First, Nguyen [2] developed the tool for plotting the AE signals from one channel. Then the script was updated in this thesis work to include the signal plots from two channels. Figure 18 shows the output of the script. The signal strength vs time in the top plot, Fast Fourier Transform (FFT) in the bottom plot and the signals parameters for both channels on the right.

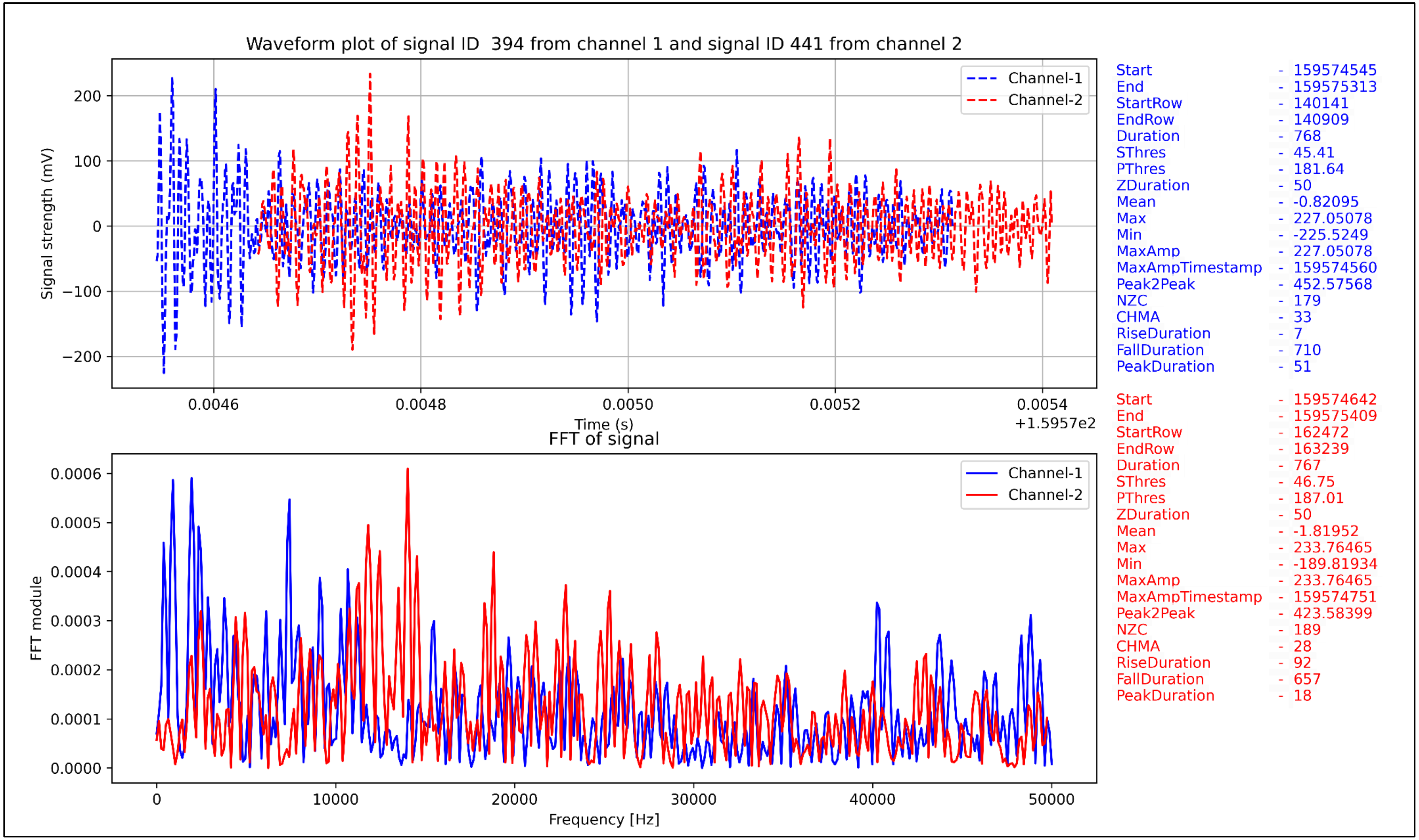


Figure 18 Sample plot to visualize the signals from both channels

5.4 Categorization of Signals

- The categorization of the AE signals for the welded specimens follows the same approach developed by Nguyen [2]. Where the signals parameters are calculated, and the statistics table is the output of the script. More details about the parameters of the signals and the description of the signal types can be found in section 2.4 and Table 2. A sample of the statistics table is shown in Figure 2. The summary of the number of signals is presented in Table 11. Where most of the signals were categorized as type A which has a fast rise duration, very short peak and Long fall duration. No signals were classified as type C in 2021 tests, where on the other hand, few signals were categorized as type C for the unwelded specimens [2]. Channels C3, C4 and C4 for B2-1, B2-2 and BD specimens did not categorise their recorded hits using the developed Python script by Nguyen [2]. It is indicated by '-' in Table 11. Note that these channels are from the model R15-I, and in test B2-2 and BD, the amplitude threshold was dropped to 20 to make sure they are recording. This was decided after noticing that they reported few hits in specimen B2-1.

Table 11 Summary of the signals categories of the specimens from 2021 tests

SPECIMEN	CHANNEL	A	B	C	E	UNCATEGORIZED	TOTAL
B1-1	C1	478			15	97	590
	C2	384	1		10	79	474
	C3	341	5		7	104	457
B1-2	C1	90			3	19	112
	C2	113			3	28	144
	C3	64	1		2	12	79
B1-3	C1	34			1	7	42
	C2	49				14	63
	C3	45			2	6	53
B2-1	C1	13				3	16
	C2	11			2	4	17
	C3	5			1	1	7

	C4	4				1	5
	C5	6				3	9
B2-2	C1	10					10
	C2	6				1	7
	C3	-	-	-	-	-	-
	C4	-	-	-	-	-	-
	C5	-	-	-	-	-	-
B3-1	C1	61			3	11	75
	C2	54			2	15	71
	C3	63	2			13	78
B3-2	C1	26			2	8	36
	C2	42			2	7	51
	C3	27				5	32
B3-3	C1						
	C2	2					
	C3	14				2	16
BD	C1	26	1			2	29
	C2	39			1	5	45
	C3	-	-	-	-	-	-
	C4	-	-	-	-	-	-
	C5	-	-	-	-	-	-
B1-1_REP	C1	156			1	13	170
	C2	114	1		1	18	134
	C3	166				21	187
B1-2_REP	C1	2			1		4
	C2	18			6	1	25
	C3	39				2	41

5.5 Source Location

The source location approach was based on calculating the time difference between the signals of the same AE event and multiplying the time difference by the speed of the sound signal longitudinally in steel (6000 m/s) to locate the AE event on the specimen. Then a plot similar to the one shown in Figure 19 can be created showing the location of the event and the recorded value of the maximum amplitude by each channel. The script used to develop this plot is mentioned in section 5.3.1 and can be found in (Appendix E – Developed Code).

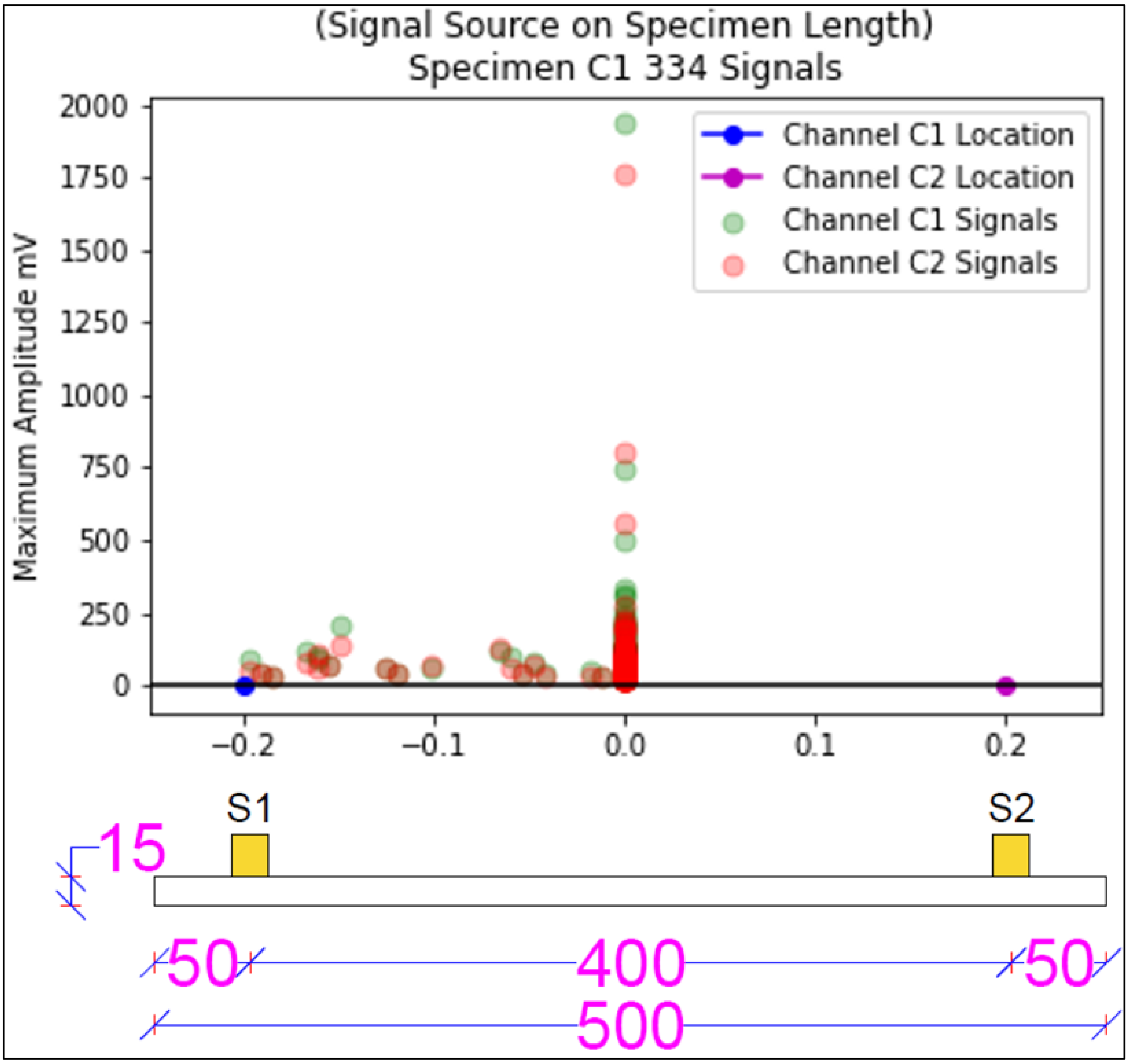


Figure 19 Sample plot showing the AE event location

5.6 Summary

This section viewed the approaches followed to develop the analysis used in this research and discussed the approaches used to develop the automated tool to identify the signals of an AE event recorded by two sensors by applying many correlation approaches. The approaches discussed were improved to achieve a robust correlation algorithm to point the AE event recorded by two or more channels. Also discussed an approach to identifying the source location. In the summary part of this chapter, the following is summarized

- The comparison between the number hits for the welded and unwelded specimens
- The comparison between the amplitude and load vs time for the welded and unwelded specimens
- Evaluation for the correlation algorithm
- The number of sensors

5.6.1 Welded VS Un-Welded Specimens

The tests performed in this research were all welded specimen except for one specimen (BD), which was base material. The test performed in 2019 was all done base material. Here to note that some of the specimens used in 2019 test had preservation coating, which was noticed during testing and caused a high number of AE hits. However, in 2021 test sandblasting was applied to all the test specimens to remove the preservation coat. As a result, the number of hits for the welded specimens were lower than the number of hits for the unwelded specimens. Comparing the plot from Dawood [1] in Figure 20 for the unwelded specimen and the plots in (Appendix B - Signal Strength and Load VS Time Plots). Generally, the AE hit of the unwelded specimen continued when the load curve flattened; on the other hand, many welded specimens had the AE hits before the curve is flat. The reason is that the unwelded specimens had more and stronger molecular bonds and fewer defects on a micro-scale level. In comparison, the welded specimens had more defects and less strong bonds, and fewer bonds. Few bonds and weaker bond will cause minor AE compared to more strong bonds.

Table 12 Comparision of the number of hits for the welded and unwelded specimens

Welded Specimens		**Unwelded Specimens**	
Specimen ID	Number of Hits	**Specimen ID**	Number of Hits
B1-1 (2021)	• C1: 292 • C2: 173 • C3: 201	**A1 (2019)**	• C1: 3900 • C2: 4000
B1-2 (2021)	• C1: 77 • C2: 80 • C3: 38	**A2 (2019)**	• C1: 3500 • C2: 3550
B1-3 (2021)	• C1: 25 • C2: 27 • C3: 43	**A3 (2019)**	• C1: 410 • C2: 420
B2-1 (2021)	• C1: 16 • C2: 21 • C3: 1 • C4: 2 • C5: 5	**B1 (2019)**	• C1: 105 • C2: 138
B2-2 (2021)	• C1: 4 • C2: 5 • C3: 31 • C4: 31 • C5: 31	**B2 (2019)**	• C1: 295 • C2: 330
B3-1 (2021)	• C1: 28 • C2: 26 • C3: 30	**B2R (2019)**	• C1: 245 • C2: 225
B3-2 (2021)	• C1: 16 • C2: 20 • C3: 23	**C1 (2019)**	• C1: 260 • C2: 330
B3-3 (2021)	• C1: 0 • C2: 15	**BD (2021)**	• C1: 63 • C2: 65

	• C3: 18		• C3: 635 • C4: 635 C5: 635
B1-1_Rep (2021)	• C1: 82 • C2: 60 • C3: 94		•
B1-2_Rep (2021)	• C1: 6 • C2: 23 • C3: 22		

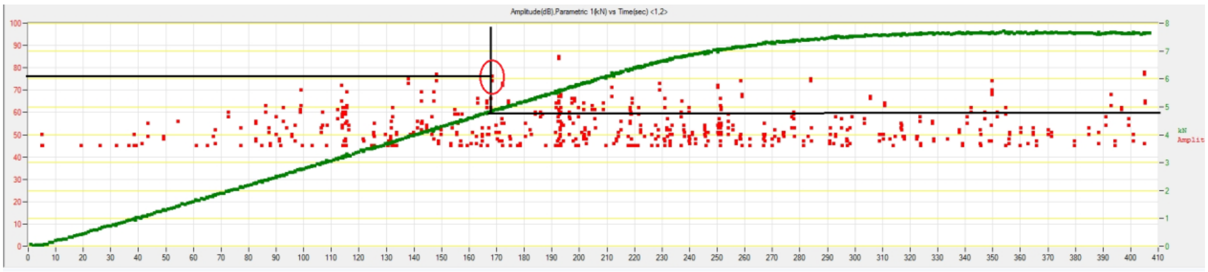


Figure 20 Amplitude and Load vs time unwelded specimen Dawood [1]

Chapter 6 Conclusion and Recommendation for Further Work

The thesis included a literature review on the processing methods to be used to analyse the AE signal. A small scale test was part of the work carried out in this research. A test proposal was developed for recording AE signals from welded specimens, and the test was performed successfully. Some analysis tools were developed using Python to have ready automated tool for processing the data collected. The research context was to characterize the AE signals, identify their source and finally be able to point out the yielding signals. The direction taken in this research was mainly to develop a robust approach to correlate the signals from two different channels and easily identify the signals of the same AE event recorded by different channels. In addition to a comparison between the welded and unwelded specimens. This chapter will include a brief discussion on the processed tools and discuss the recommendation for further work.

6.1 Conclusion

AE can be interpreted as the sound of the material when it experiences stress or deformation. Therefore, sensors attached to the material to listen to the AE and present signals can be a valuable tool to understand the changes happening to the material. AE is a sound wave travelling in all direction. Sensors attached to the material shall record the same signal for the same AE event. The thesis focused mainly on developing tools correlating signals from the same source recorded by different channels. The correlation approach started by finding the signals with the minimum time difference between the start time of the signals, then was developed to define a time range and calculates the percentage difference for the NZC and energy of the matched signals within the defined time range. After concluding that the percentage difference for the NZC and energy is not enough to correlate the signals, the cross-correlation coefficient was added to the matching with a defined time range. Finally, the improved approach to correlate signals with the percentage difference for the NZC and energy and cross-correlation coefficient shall be a robust approach to identify the AE signals of the same AE event. The difference between welded and unwelded specimens in terms of AE hits number. Also, its relation to the load vs time plot was discussed.

The experiment was done using different models of AE sensors; R15I and R15a. The history of some of the sensors is not known. All the sensors were not calibrated before the experiments. A PLB test was done, and all the sensors showed acceptable amplitude, which indicates that we can trust their recording. However, more work is needed to evaluate the accuracy of the sensors, their sensitivities and define their uncertainties.

The approach for the cross-correlation coefficient is not fully functional. The cross-correlation coefficient script is available, but it was not connected to the primary correlation with a defined time range script. Therefore, more development is needed to connect the two scripts. Nevertheless, the correlation approach is helpful to validate the signals' reading by plotting the signals with a high correlation coefficient, and the plotted signals are expected to be the same since they are expected to be emitted from the same source.

In summary, the research aimed to develop a Python script to perform the three main tasks:

1- Correlate signals of the same AE event logged by two channels

The approach followed was improved during this research. It started by matching the signals of the two channels by the closest timestamp of the signal’s start time. Then, another direction of thought was introduced; to define a time range and match the signals of the two channels within this defined time range. Since the single signal from a channel had many matches from the other channel within the defined time range.

The approach needed improvement. The percentage difference of the parameters NCZ and energy was used to indicate the similarity of the signals. However, it did not appear as an easy indicator for the signal similarity. The approach used until the percentage difference of the NZC and energy is adequate; however, it was not enough to indicate the signals of the same AE event. Therefore, the cross-correlation coefficient was introduced as another indicator. This was tried separately but was not connected to the main script of the defined time range correlation. The concept of how it should be implemented is discussed in section 5.3.3.

2- Determine the location of the AE event

Determining the location of the AE event was based on the time difference of the matched signals. Then, multiply it by the longitudinal speed of the sound wave in the steel, where the result is the location of the AE event of the matched signals. The approach is theoretically valid, but to ensure that it is accurate, the matched signals shall be proven to be from the same AE event before using the approach to identify the location

3- Characterize the signals using the parametric analysis

The Python script and the categories of signals from Nguyen [2] were implemented. The collected AE signals from the welded and coated specimens of this research experiment were categorized. However, the categories were not correlated to any phenomena.

6.2 Recommendation for Further Work

The algorithm used to correlate the signals shall be improved to have the cross-correlation coefficient script. Plotting the matched signals with a high coefficient and finally developing a tool to ensure that the signals with a high coefficient of correlation are from the same source without visualising the signals plots. The following recommendation shall be considered for continuing this research topic and for similar future work:

1- Evaluation of Correlation Algorithms

Evaluating the approach discussed in section 5.3.3 by applying t to the tests data from 2019 and 2021. After finding the high coefficient of correlation, the signals to be plotted to prove visually that the high correlation coefficient indicated a high similarity of the AE signals.

2- Correlation and source location using more than two sensors

Developing an approach similar to the one explained in section 5.5 to locate AE events recorded by more than one sensor. Also, research more about AE source location.

3- Magnitude squared coherence

Apply the magnitude squared coherence to the data gathered in the tests. The magnitude square coherence is another method to measure the similarity of two signals in the frequency domain. More about the method can be found in [12].

4- Signal-based approach

This research classified the signals using the parametric approach of signal analysis. The other method is to use the signal based approach. More about the method can be found in '*GROSSE, C. U. & OHTSU, M. 2008. Acoustic emission testing, Springer Science & Business Media'.*

5- Compare the result from the welded and unwelded specimen in details

Considering the sketch in Figure 21. Suppose an AE signal is emitted from the left of the weld. The signal travelling to the left for sensor one will travel in a homogenous material. While the signal travelling to the right sigh to sensors two and three will be cross the weld be non-homogenous and has a different molecular structure. This is a case that is recommended to be investigated. What can happen to the signal crossing the weld and identify the AE in such a case? The other interesting question is how the signals get affected when they get reflected by the specimen boundaries. As shown in the sketch, signals reflected on the end of the specimen has a chance to be captured more than once. Another case is also for the signal hitting the specimens transversely on the short width or height. How these be identified and what happens to the signals, and what parameters shall be the correlation of signals based on these cases. Figure 21 shows a sketch for a case where the AE event is on the side of the weld and the AE signal in red is not affected by the weld, where the signal blue is affected by the weld. After hitting the end of the specimens, the two signals changes. Different colours indicate this.

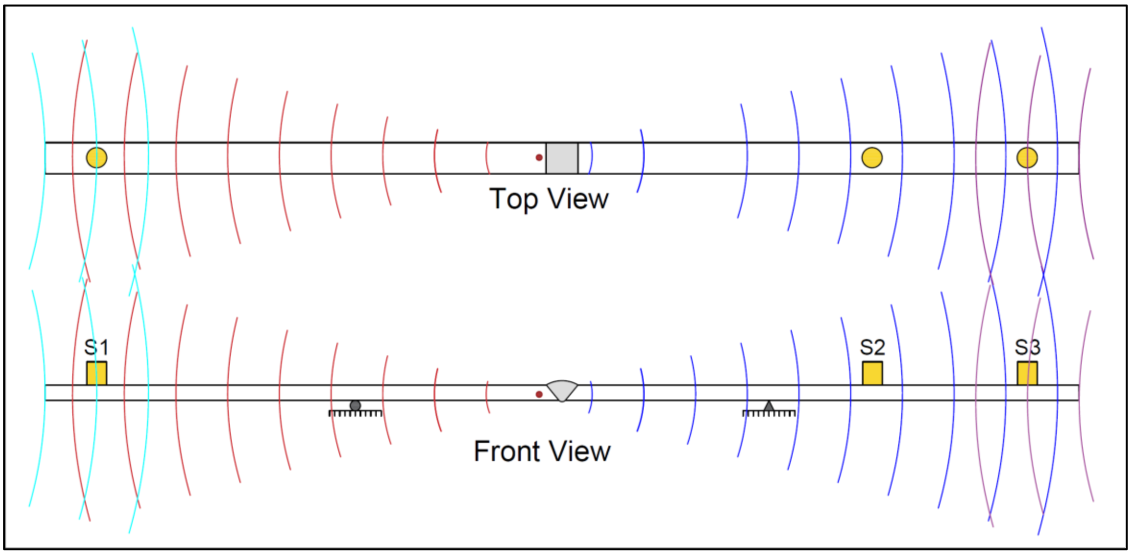


Figure 21 Sketch for further work

6- Bubble plot using other parameters

Plot the bubble plot in Figure 16 for the matched signals using other parameters like peak duration, fall duration or energy.

References

[1] K. Dawood, “Master Thesis - Acoustic Emission Testing for Offshore Structures,” University of Stavanger, 2019.

[2] B. Nguyen, “Master Thesis - Study of the Characteristics of Acoustic Emission Signals,” University of Stvavanger, 2020.

[3] O. Gabrielsen, “Proposal for Master Thesis - Study the Characteristics of Acoustic Emission Signals.,” 2020.

[4] J. Stump, “Offshore industry embraces digital twin technology | Offshore,” *Offshore Magazine*, Nov. 01, 2020. https://www.offshore-mag.com/production/article/14185502/offshore-industry-embraces-digital-twin-technology (accessed Mar. 04, 2021).

[5] S. Hansen, “Digital Twin: Learning from the Offshore Industry - Ramboll Group,” *T&D World*. https://ramboll.com/ingenuity/digital-twin-based-asset-management (accessed Mar. 07, 2021).

[6] D. B. Duthie and F. Gabriels, “Remote Monitoring of Offshore Structures using Acoustic

Emission," *11th Eur. Conf. NDT, ECNNDT 2014*, no. Ecndt, 2014, [Online]. Available: http://www.ndt.net/events/ECNDT2014/app/content/Paper/222_Duthie.pdf.

[7] B. Muravin, "Acoustic emission science and technology," *J. Build. Infrastucture Eng. Isr. Assoc. Eng. Archit.*, no. Figure 1, pp. 1–12, 2009.

[8] V. Systems, "Acoustic Emission Sensors," 2017.

[9] M. Kaphle, A. C. C. Tan, D. P. Thambiratnam, and T. H. T. Chan, "Analysis of acoustic emission data for structural health monitoring applications," no. December, 2010.

[10] S. Gholizadeh, Z. Lemana, and B. T. H. T. Baharudinb, "A review of the application of acoustic emission technique in engineering," *Struct. Eng. Mech.*, vol. 54, no. 6, pp. 1075–1095, 2015, doi: 10.12989/sem.2015.54.6.1075.

[11] Q. Tian *et al.*, "Study of micro-plastic deformation in pure iron before macro-yielding using acoustic emission, electron backscattered diffraction and transmission electron microscopy," *Mater. Sci. Eng. A*, vol. 771, no. July 2019, p. 138645, 2020, doi: 10.1016/j.msea.2019.138645.

[12] M. K. and A. C. C. Tan, *Differentiating Signals from Different Sources of Acoustic Emission for Structural Health Monitoring Purposes*, Volume 158., vol. 13. Springer, 2013.

[13] Y. Yang, C. Hui, Y. Ping, F. Yuan, and J. Ling, "Identification of Acoustic Emission Signal of Tank Bottom Corrosion Based on Weighted Fuzzy Clustering," in *Advances in Acoustic Emission Technology*, 2013, pp. 395–404.

[14] L. Gao, F. Zai, S. Su, H. Wang, P. Chen, and L. Liu, "Study and application of acoustic emission testing in fault diagnosis of low-speed heavy-duty gears," *Sensors*, vol. 11, no. 1, pp. 599–611, 2011, doi: 10.3390/s110100599.

[15] Y. M. Kang, "Acoustic emission signal processing based on wavelet analysis," *ISRM Int. Symp. - 5th Asian Rock Mech. Symp. 2008, ARMS 2008*, no. November, pp. 353–356, 2008.

[16] J. Coady, D. Toal, T. Newe, and G. Dooly, "Remote acoustic analysis for tool condition monitoring," *Procedia Manuf.*, vol. 38, no. 2019, pp. 840–847, 2019, doi:

10.1016/j.promfg.2020.01.165.

[17] J. J. González de la Rosa, R. Piotrkowski, and J. Ruzzante, “Third-order spectral characterization of acoustic emission signals in ring-type samples from steel pipes for the oil industry,” *Mech. Syst. Signal Process.*, vol. 21, no. 4, pp. 1917–1926, 2007, doi: 10.1016/j.ymssp.2006.08.011.

[18] C. Lu, P. Ding, and Z. Chen, “Time-frequency analysis of acoustic emission signals generated by tension damage in CFRP,” *Procedia Eng.*, vol. 23, pp. 210–215, 2011, doi: 10.1016/j.proeng.2011.11.2491.

[19] A. Sibil, N. Godin, M. R’Mili, E. Maillet, and G. Fantozzi, “Optimization of acoustic emission data clustering by a genetic algorithm method,” *J. Nondestruct. Eval.*, vol. 31, no. 2, pp. 169–180, 2012, doi: 10.1007/s10921-012-0132-1.

[20] V. Emamian, M. Kaveh, A. H. Tewfik, Z. Shi, L. J. Jacobs, and J. Jarzynski, “Robust clustering of acoustic emission signals using neural networks and signal subspace projections,” *EURASIP J. Appl. Signal Processing*, vol. 2003, no. 3, pp. 276–286, 2003, doi: 10.1155/S1110865703210027.

[21] W. Emamian, M. Kaveh, and A. H. Tewfik, “Robust clustering of acoustic emission signals using the Kohonen network,” *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 6, pp. 3891–3894, 2000, doi: 10.1109/ICASSP.2000.860253.

[22] M. G. R. Sause, A. Gribov, A. R. Unwin, and S. Horn, “Pattern recognition approach to identify natural clusters of acoustic emission signals,” *Pattern Recognit. Lett.*, vol. 33, no. 1, pp. 17–23, 2012, doi: 10.1016/j.patrec.2011.09.018.

[23] L. Calabrese, G. Campanella, and E. Proverbio, “Use of Cluster Analysis of Acoustic Emission Signals in Evaluating Damage Severity in Concrete Structures,” *J. Acoust. Emiss.*, vol. 28, pp. 129–141, 2010.

[24] E. Pomponi and A. Vinogradov, “A real-time approach to acoustic emission clustering,” *Mech. Syst. Signal Process.*, vol. 40, no. 2, pp. 791–804, 2013, doi: 10.1016/j.ymssp.2013.03.017.

Appendices

Appendix A - Test Proposal

This section is a developed version of the test final proposal used as a reference to the test performed in this thesis. The scope of the thesis included developing a test proposal. The test proposal was developed and sent DNV laboratory at Høvik, Oslo, to perform the test.

Purpose

The purpose of this test was to record real-time (AE) signals from a specimen being loaded. The data then is to be analyzed to serve the aim of this research which is to correlate the AE signals to the material changes underloading.

Questions

Some essential questions need to be asked for this test.

- What other parameters can cause AE in the test environment? How to identify and filter other AE?

All surrounding of the test shall be noticed. As much as possible, the test environment shall remain the same for all the specimens. Filtering the unneeded AE will be done by filtering unwanted frequencies after identifying all the unwanted sources. One example of a needless AE source in the test environment is the emissions from the friction between the support and the specimen.

- How to validate the accuracy of the sensors?

Validation of the accuracy of the sensors shall be ensured by:

  - Calibrating the AE sensors before using
  - Applying Pencil Lead Break (PLB) test and calculating the cross-correlation coefficient of the signals from the different sensors

- How to identify the time where the material is yielding?

Identifying the exact time where the material started yielding would be beneficial information; since it will help analyze the signals that occurred at the same timestamp. This can be done by attaching a strain gauge. Due to the complexity of the test set up if the strain gauge to be connected, the time of yielding will be estimated theoretically.

Acoustic Emission Testing for Steel Specimens

A four-point bending test will be carried out to put the material under stress. The material will emit AE that will be recorded by the AE sensor mounted to the specimen.

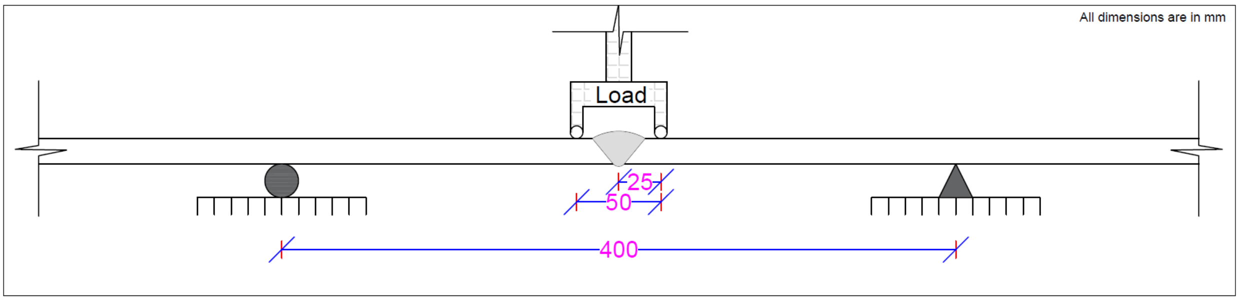


Figure 22 Sketch of the dimensions to supports and loading cell for the four-point test

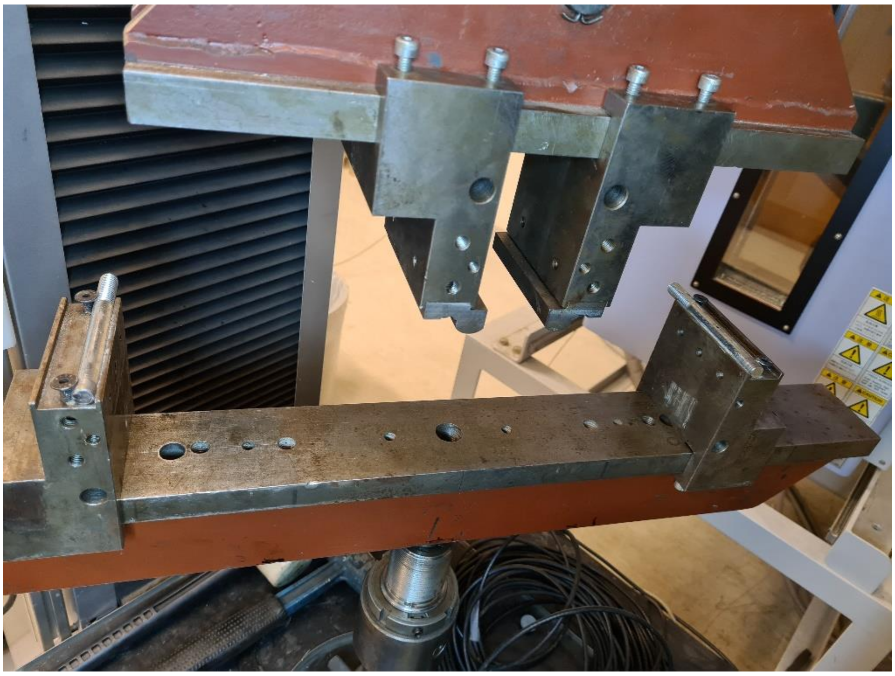


Figure 23 Four-point test supports and load cell

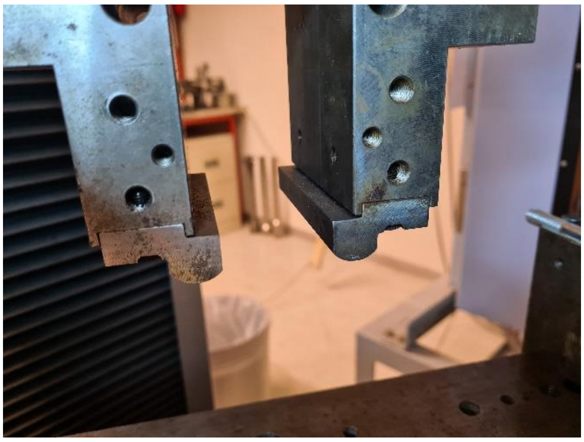


Figure 24 Loading cell

Figure 25 Support

Materials

Table 13 Test material

MATERIAL	DESCRIPTION	QUANTITY	USE
STEEL SPECIMEN	• Dimensions 120 mm X 15 mm X 6 m • Flat steel in quality S355J2 according to EN 10025-2 and NORSOK M-120 Rev. 5 MDS Y05 • Yield strength 355 MPa and ultimate strength 490 MPa	1	Use it as test specimens. Figure 26
COUPLANT	Any	1	To be applied between the surface of the test specimen and the face on AE sensors
MARKER	Light colour	1	To mark the position of the sensors and write the specimen ID
MECHANICAL PENCIL WITH PENCIL LEAD	A lead of 0.3 mm diameter	1 Pencil 1 lead box	To perform PLB test

Figure 26 Test specimens after cutting

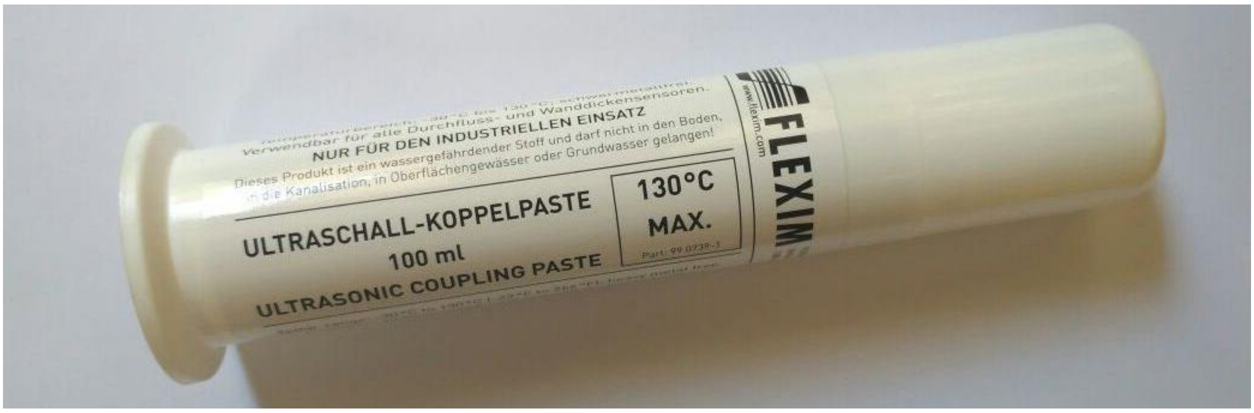


Figure 27 Ultrasonic coupling paste used in the experiment

Apparatus

Table 14 Test apparatus

TYPE	***QUANTITY***
R15A (FIGURE 28 & FIGURE 29) AND R15I SENSORS (FIGURE 30 & FIGURE 31)	2X R15A and 3XR15I
AMPLIFIERS (FIGURE 32)	2 for the R15A sensors
WIRES	Enough to make the test setup
HOLDER (FIGURE 33)	5
DATA ACQUISITION DEVICE (FIGURE 34)	1
COMPUTER WITH AEWIN SOFTWARE (FIGURE 35)	1
CALIPER (FIGURE 36)	1
RULER	1
LOADING MACHINE (SHIMADZU) (FIGURE 37)	1
VIDEO RECORDING CAMERA WITH TRIPOD	1

Figure 28 R15a sensor

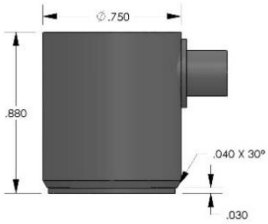


Figure 29 R15a sensor dimensions

Figure 30 R15I sensor

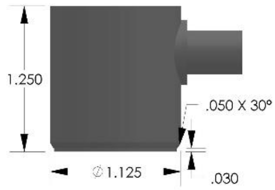


Figure 31 R15I sensor dimensions

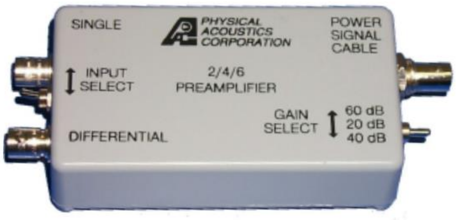


Figure 32 Preamplifier used in the test

Figure 33 3d Printed holder used in the test

Figure 34 Data acoustion device

Figure 35 Computer with AE Win software

Figure 36 Caliper

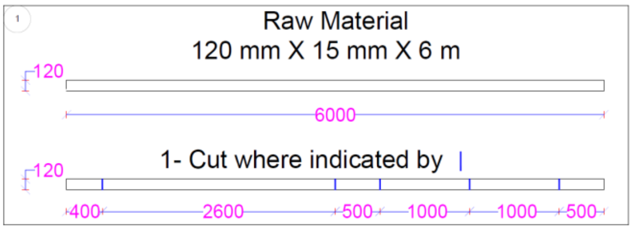
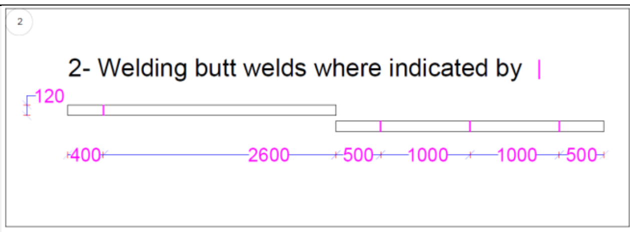
Figure 37 Shimadzu loading machine

Specimens Preparation

The supplied dimensions are 120 mm X 15 mm X 6 m. Preparation of the test samples was done as explained in this section.

Table 15 Summarize the procedure implemented to cut the test specimens. The cutting was to done using the water jet cutting method to ensure accuracy. The specimen was then welded Metal Inert Gas (MIG) and Metal Active Gas (MAG). The welding type was not the focus of this research. The chosen methods of welding were agreed upon between the lab and the welding company based on their experience.

Table 15 Cutting procedure

1- The specimen was cut as shown in Figure 38 **The purpose was to have:** • **(6)X 30 mm X 15 mm X 1 m** • **(2)X 30 mm X 15 mm X 2.5 m** • **(1)X 120 mm X 15 X 500 mm** **After the final step of cutting**	 *Figure 38 Specimen cutting - 1*
2- The pieces were welded together (Butt Weld) as shown in Figure 39	 *Figure 39 Specimen cutting - 2*

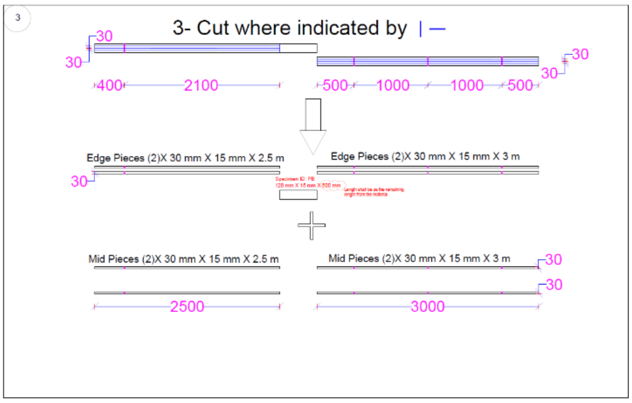
<table>
<tr>
<td>3- 3 longitudinal cuts as shown in Figure 40. Leaving two mid pieces with 30mm +-0.1mm width (if that is a practical tolerance). The remaining material on each edge was discarded.
The result from this step was:
Remaining Edge Pieces to avoid weld end effect (was discarded)
• (2)X 30 mm X 15 mm X 2.5 m
• (2)X 30 mm X 15 mm X 3 m
Mid Pieces
• (2)X 30 mm X 15 mm X 2.5 m
• (2)X 30 mm X 15 mm X 3 m
• (1)X 120 mm X 15 X 500 mm</td>
<td>

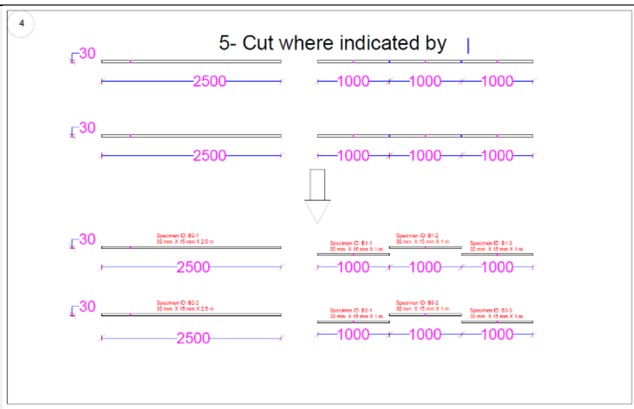
Figure 40 Specimen cutting - 3</td>
</tr>
<tr>
<td>4- Useing the mid pieces with dimensions (30 mm X 15 mm) from step no 3, the pieces were then cut into the test pieces as shown in Figure 41</td>
<td>

Figure 41 Specimen cutting - 4</td>
</tr>
</table>

During testing, an additional specimen was prepared for the ball drop test and four-point loading test. The purpose was to have an unwelded specimen and recorded the AE signals from this specimen. This specimen was cut from specimen B3-1, and the dimension was 30 X 15 X 1000 mm.

Table 16 Summary of test specimens

SPECIMEN ID	***DESCRIPTION***	***QUANTITY***	***DIMENSIONS*** *WIDTH X THICKNESS X LENGTH*	***PRIORITY***	***NOTES***
PB	Unwelded	1	120 mm X 15 mm X 500 mm	High	Only 2 X PLB test
B1 (1-3)	Butt welded at midspan	3	30 mm X 15 mm X 1 m	High	2 X PLB test
B2 (1-2)	But welded at 400 mm from the side. As shown in Figure 44	2	30 mm X 15 mm X 2.5 m	High	1 X loading until yielding test
B3 (1-3)	But welded at midspan and coated on the bottom surface	3	30 mm X 15 mm X 1 m	Low	
BD	Unwelded and uncoated	1	30 mm X 15 mm X 1 m		Ball drop 2 X PLB test 1 X loading until yielding test

Ball Drop Test

This test was applied only on one specimen that was unwelded and uncoated. It was cut specimen B3-1, and the dimension was 30 X 15 X 1000 mm. The ID was 'BD' referring to the ball drop test. After the ball drop test, the four-point loading test was applied to this specimen as well.

The Ball Drop test was applied using a 9.5 mm diameter steel ball on the specimen from 15 cm height. This process was repeated three times.

Testing Order and Setup

The testing was carried on a total of 10 specimens. Table 17 summarize the type of test applied on each specimen and the reference test setup figure.

Table 17 Test description for each specimen and reference test setup

TEST ORDER	***SPECIMEN ID***	***TEST DESCRIPTION***	***TEST SETUP***
1	PB	2 times PLB test only	Figure 42
2	B1-1	2 times PLB test Four-point test (loading to yield)	Figure 43
3	B1-2	2 times PLB test Four-point test (loading to yield)	Figure 43
4	B1-3	2 times PLB test Four-point test (loading to yield)	Figure 43
5	B2-1	2 times PLB test Four-point test (loading to yield)	Figure 44
6	B2-2	2 times PLB test Four-point test (loading to yield)	Figure 44
7	B3-1	Figure 45	Figure 45
8	B3-2	2 times PLB test Four-point test (loading to yield)	Figure 45
9	B3-3	2 times PLB test Four-point test (loading to yield)	Figure 45
10	BD	Ball drop 2 times PLB test Four-point test (loading to yield)	

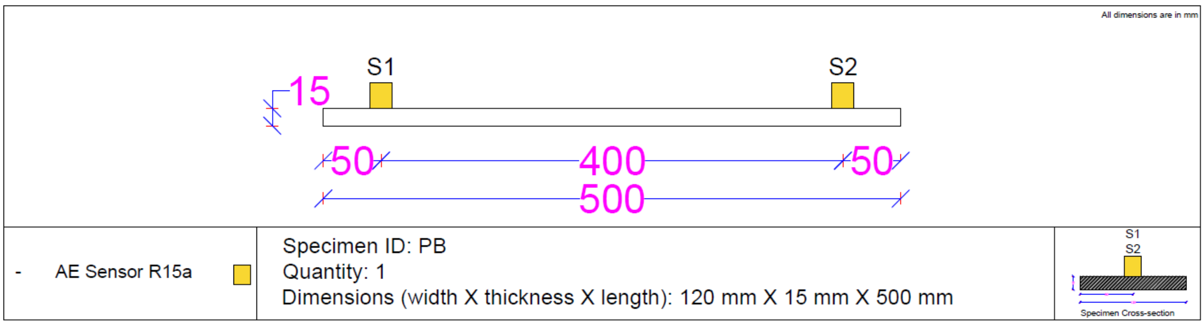


Figure 42 Specimen PB test setup

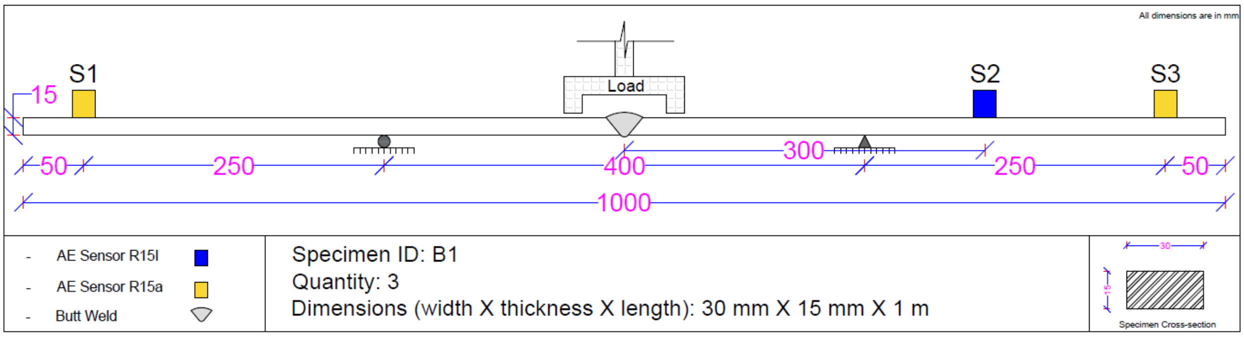


Figure 43 Specimen B1 test setup

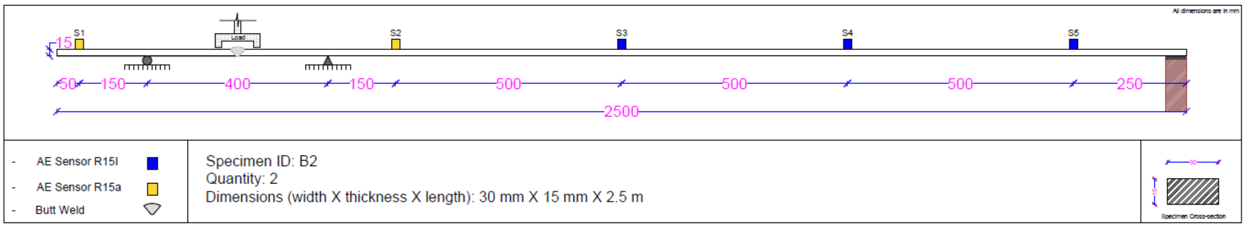


Figure 44 Specimen B2 test setup

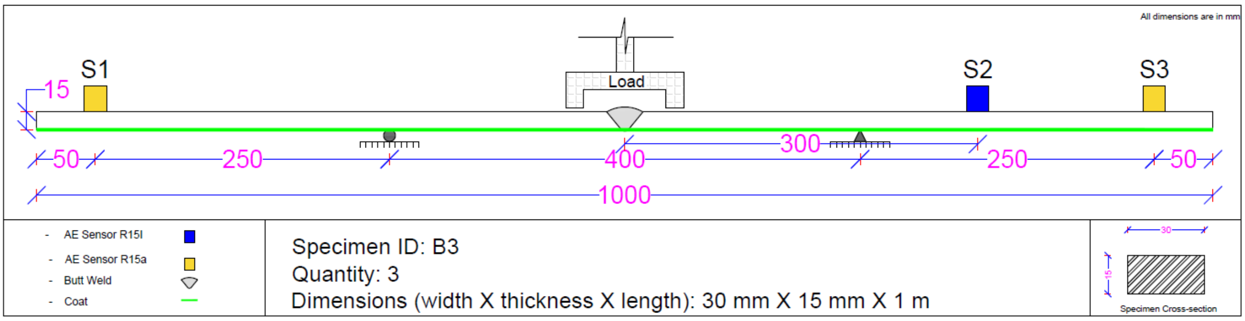


Figure 45 Specimen B3 test setup

Pre-Test Preparation

Before starting the testing, some preparation needed to be done for the specimens. The following form was sent to the lab to be followed as a guideline for the pre-test preparation. It was filled in on the test days, and the filled-in forms were sent back after the test was done.

AET Pre-Test Preparation
Specimen ID: Location: Date: Start Time:

Finish time:		
Procedure	Comment	Check Mark
1- Marking the specimens ID, location of the sensors, location of supports and sensors ID • Specimen's ID is to be written on the front surfaces. • Use the ruler and the marker to mark the locations of the sensors and supports. The dimensions shall be centre to centre ○ For the welded specimens, all dimensions shall be with reference to the centre of the weld root. ○ For the unwelded specimen, mark the centre of the specimen on all the surfaces. All dimensions shall be with reference to the centre of the specimen ○ The sensor's location is to be marked on the top and front surfaces. • The sensor's ID is to be written on the front surfaces.		
2- Mark the PLB test location • Considering the orientation where the weld root is the bottom and weld face is the top • Mark the PLB test location on the top surface ○ For the welded specimens, the mark shall be in the centre of the weld face. ○ The unwelded specimens shall be in the centre between the supports marking and the centre of the specimen		
3- Mark the cables with the ID of the sensor.		

• Specimen PB will have 2 sensors (S1 and S2) • Specimens B1 and B3 will have 3 sensors (S1, S2 and S3) • Specimens B2 will have 5 sensors (S1, S2, S3, S4 and S5)		
4- Connect the preamplifier to the sensor		
5- Connect the amplifier to the data acquisition system		
6- Connect the data acquisition system to the computer		
7- Mounting the AE sensors • Apply a small amount of couplant on the centre of the sensitive face of the AE sensor • Carefully press the AE sensor on the surface of the test specimen where the sensor's marks exist • Ensure that the couplant is spread evenly from the centre to a bit outside the sensor surface • Ensure that the couplant is applied in the thinnest practical layer • Carefully attach the magnetic holder to the material and the sensors and tighten the screw to hold the sensor to the material ("Acoustic Emission Sensors Specification")		
8- Place the recording camera viewing the testing machine		
9- Setup the loading machine • The speed of the crosshead 0.017 mm/s	Yield strength from material certificate ………435……………. MPa Calculated load to be applied	

• The load is to be calculated based on the values of the material properties from the material datasheet. See more details in the 'Applied Load Calculation' section below	………………6800……………….. N to cross yielding	
10- Test the available sensors. In case some of the sensors are not good to be used, sensors setup shall be changed		
11- Create folders with every specimen ID total of 9 folders. Folders name shall have no space ex. ('B1-2') • Create 1 folder for general-photos In the 9 folders named after the specimen ID. Create 4 subfolders for: • AE Win files. Folder name 'AE_win_files' • SHIMADZU files. Folder name 'shimatzu_files' • Video recording and photos of the test. The folder name 'video_recording' • Filled test procedure form. The folder name 'test_procedure_form'		
12- Take picture of: • The area where the test will be carried out • All the specimens together with the clear specimen ID on every specimen • Each specimen alone with the specimen ID • All test material. Refer to table 1 • All apparatus. Refer to table 2		

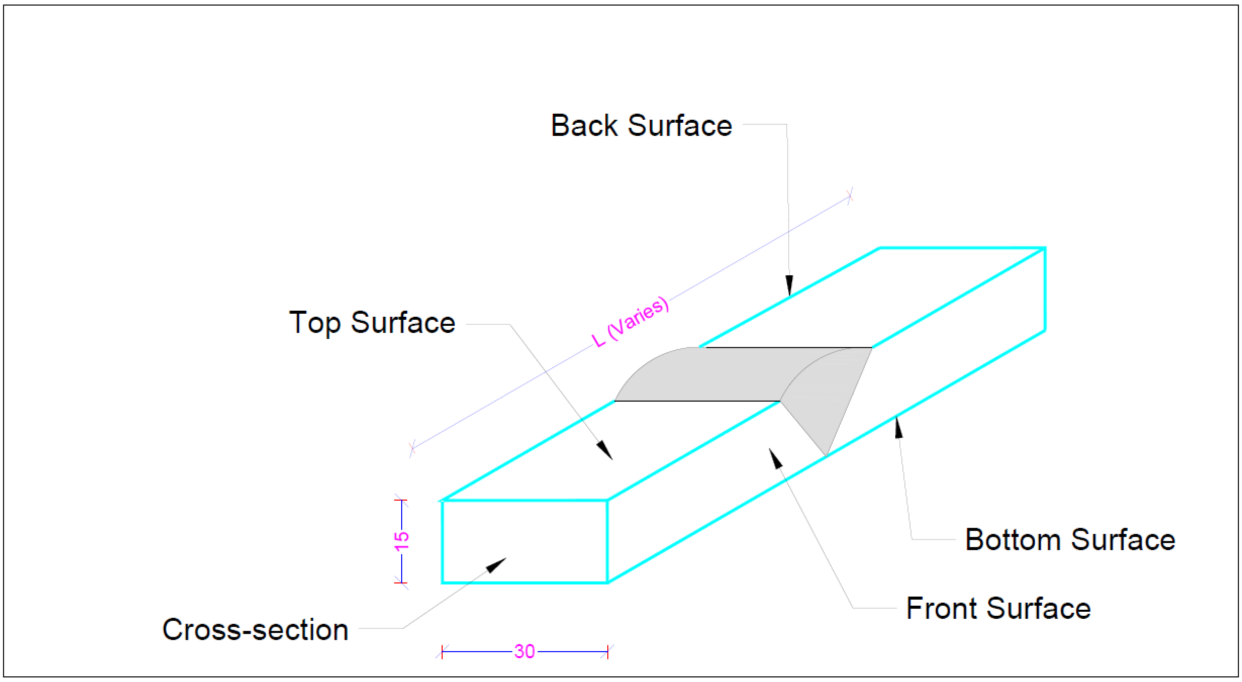


Figure 46 Welded specimen orientation and surfaces definition

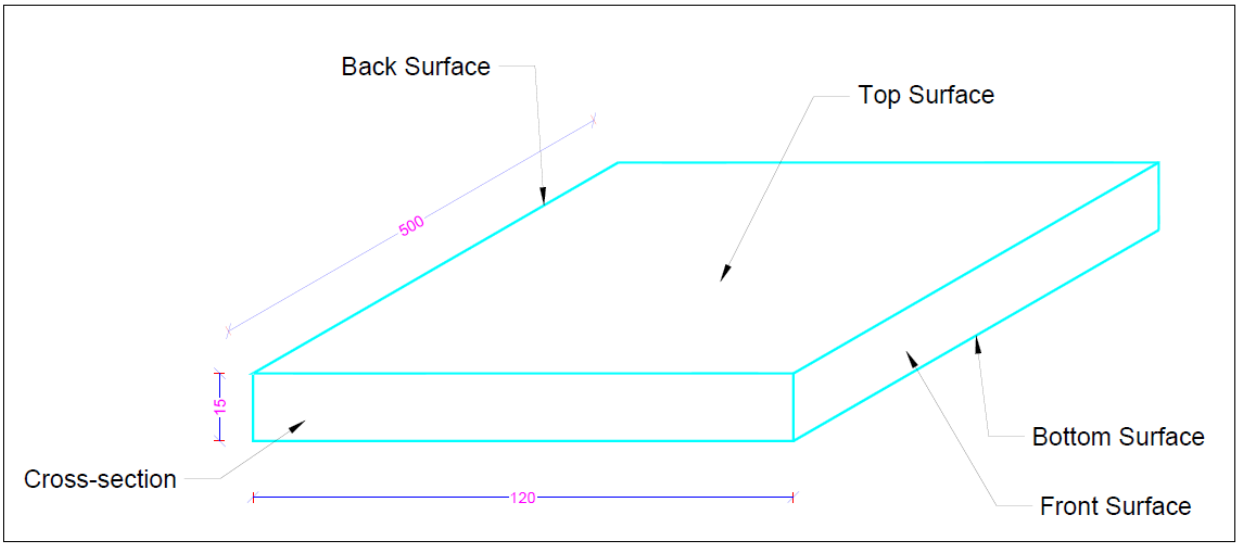


Figure 47 Unwelded specimen orientation and surfaces definition

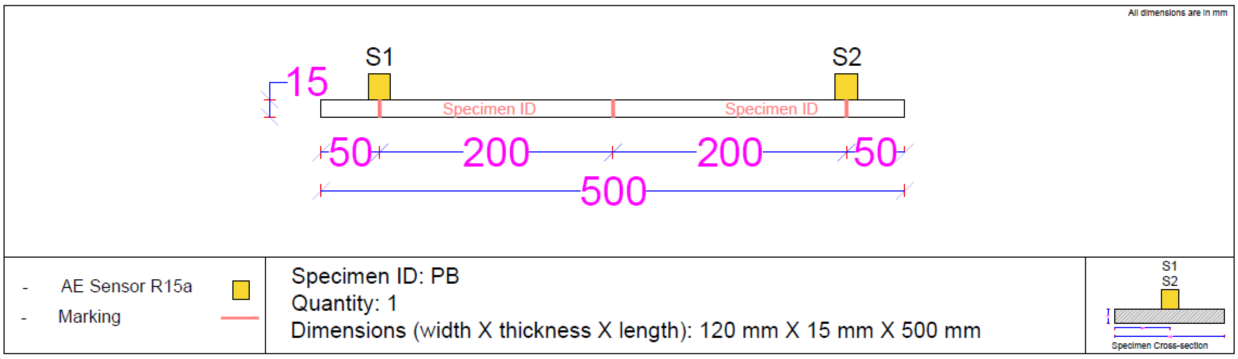


Figure 48 Specimen PB marking and sensors location from centre

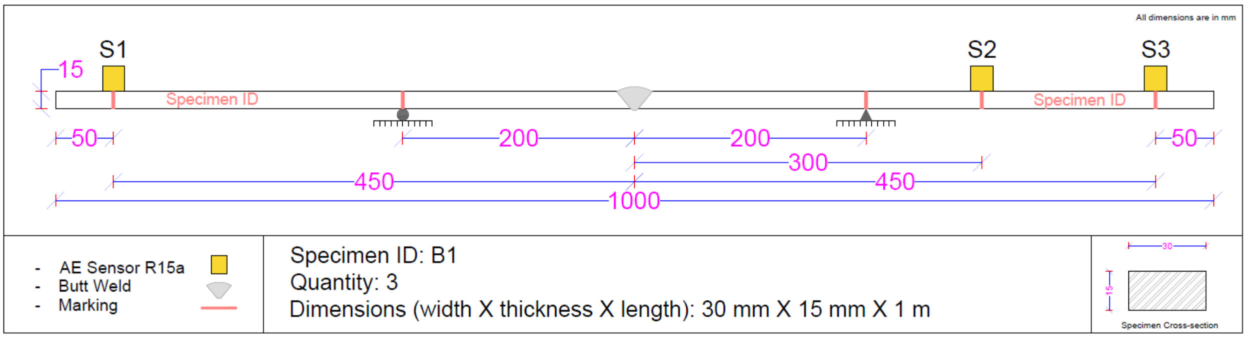


Figure 49 Specimen B1 marking and sensors location from the centre of the weld

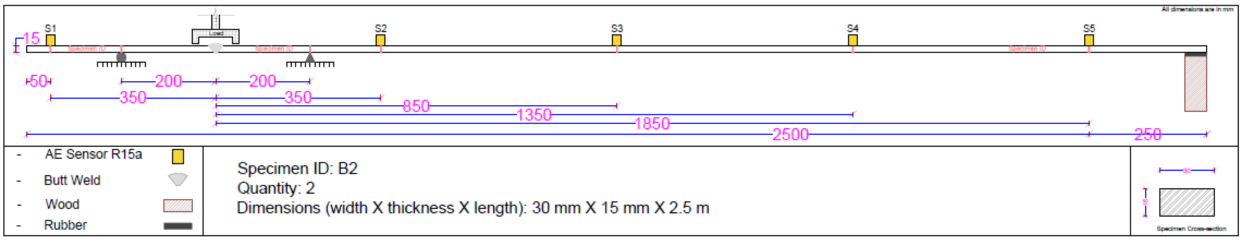


Figure 50 Specimen B2 Marking and sensors location from the centre of the weld

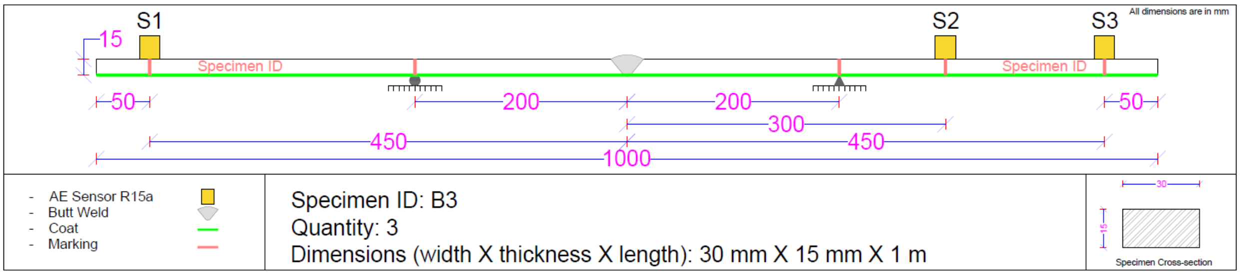


Figure 51 Specimen B3 Marking and sensors location from the centre of the weld

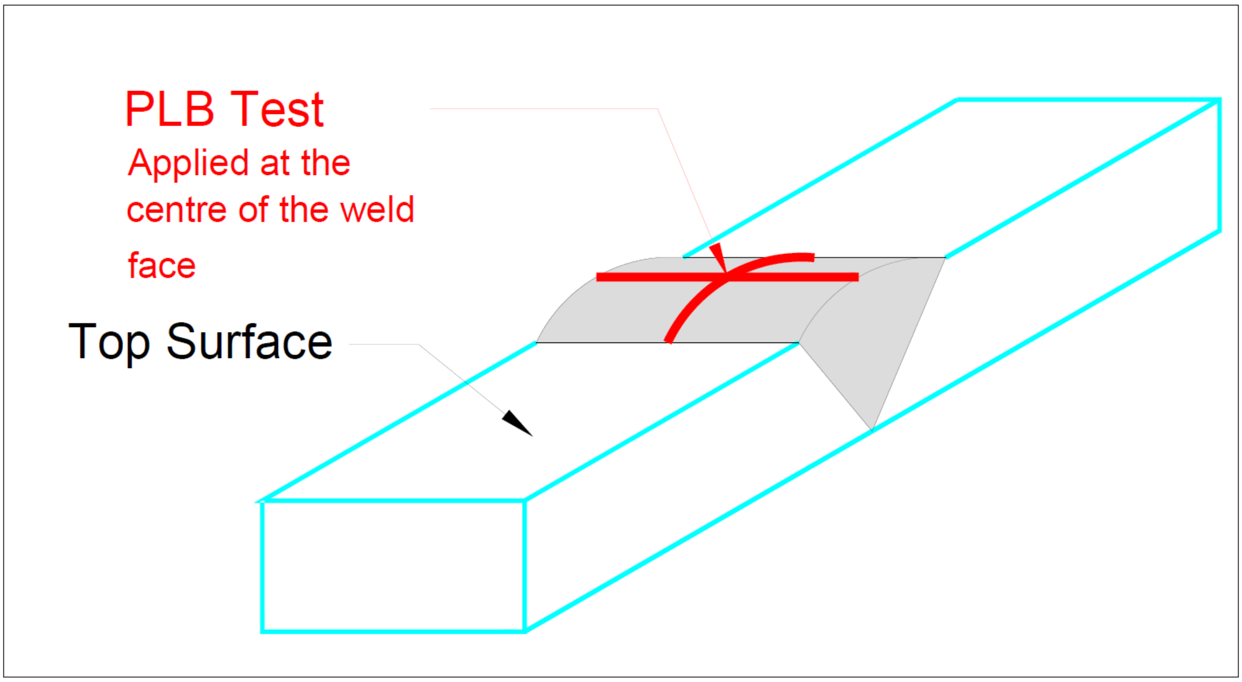


Figure 52 Pencil Break Lead test location on the welded specimen

Applied Load Calculation

The following calculation was done to calculate the minimum load to be applied to the specimen.

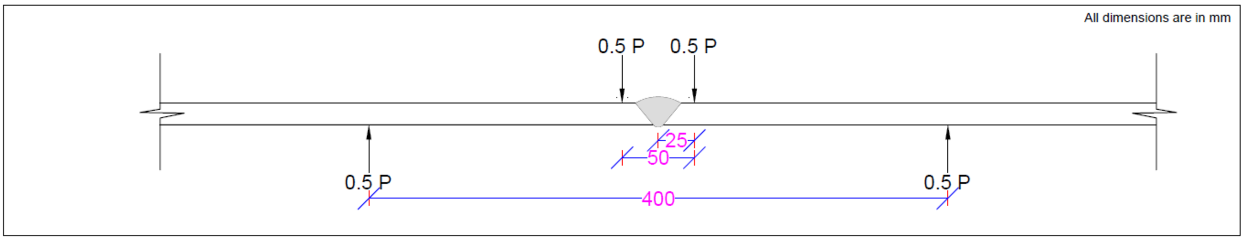


Figure 53 Sketch for load/reaction of the test specimen

$$\Sigma M = 0\,; \qquad M_{mid\ span} = 0.5P \times 200 - 0.5P \times 25 = \frac{175}{2}P \quad (eq.1)$$

$$\sigma_{applied} > \sigma_y$$

$$\sigma_{applied} = \frac{My}{I} \quad (eq.2)$$

Substituting eq. 1 in eq two and resolving for P

$$\frac{\sigma_{applied} I}{\frac{175}{2} \times \frac{15}{2}} = P$$

Substituting σ_y from the material certificate as $\sigma_{applied}$

$$\frac{430 \times \frac{1}{12} \times 30 \times 15^3}{\frac{175}{2} \times \frac{15}{2}} = P = 5529\ N$$

Increasing P by 20% to ensure that yielding is reached. The minimum load applied was not below $P = 6635\ N$

Test Procedure

The following test procedure was sent to the lab to be followed as a guideline for the tests. It was filled in on the test days, and the filled-in forms were sent back after the test was done.

AET Thesis Test Procedure		
Specimen ID: Location: Date: Start Time: Finish time:		
Procedure	Comments	Check Mark
1- Check that the PC has AE Win software running, and the logging time is set to 2048 microseconds. Setup the software for the test requirement based on the guidelines from 'AE Win User Manual.'		
2- Measure the actual dimensions of the specimens. Write down the measured dimensions (Width X Thickness X Length). • Width and thickness to be measure by calliper in mm	(W= ……….. mm X Th= ………… mm X L=……….. m)	

<table>
<tr><td>Length to be measured by the ruler in m</td><td></td><td></td></tr>
<tr><td>3- Mounting the AE sensors
Apply a small amount of couplant on the centre of the sensitive face of the AE sensor
Carefully press the AE sensor on the surface of the test specimen where the sensor's marks exist
Ensure that the couplant is spread evenly from the centre to a bit outside the sensor surface
Ensure that the couplant is applied in the thinnest practical layer
Carefully attach the magnetic holder to the material and the sensors and tighten the screw to hold the sensor to the material ("Acoustic Emission Sensors Specification")</td><td></td><td></td></tr>
<tr><td>4- Measure the deflection of the specimen before loading; By putting it on a flat surface. Using the calliper, measure the maximum deflection on the sides of the weld. Write down the value</td><td><table><tr><td></td><td>L (mm)</td><td>R (mm)</td></tr><tr><td>Before Loading</td><td></td><td></td></tr><tr><td>After Loading</td><td></td><td></td></tr></table></td><td></td></tr>
<tr><td>5- If deflection was found. Take a picture showing the direction of the deflection
Save the picture in the specified folder with the name 'deflection_before_test'</td><td></td><td></td></tr>
<tr><td>6- Performing Pencil Lead Break test. Test to be performed 2 times
Considering that the specimen is rested on the supports, where the weld root is the bottom and the</td><td></td><td></td></tr>
</table>

weld face is the top. The PLB shall be applied on the top surface • Apply the test on the marked point ○ In the centre of the weld face for the welded specimen ○ In the centre between the supports for the unwelded specimen • Ensure that the specimen is stable and no motion will occur during the PLB test • Ensure the consistency of the repetitive PLB test		
a- Length between 2 and 3 mm of the lead shall be prepared to be broken. Always break the same length in the repetition of the test		
b- Point the lead to the marked breaking point with an angle of 30° between the lead and the top surface		
c- Break the lead by touching it to the marked point on the top surface. Ensure that only the lead touches the specimen		
7- Ensure that the AE software has received signals from every channel the value shall be close to 100 dB		
8- Save the data from AE Win in format, and the file shall be named 'PLB_SpecimenID_Test number.'		

• Test number: Shall have the value 1 or 2 since the 2 repetitions of the PLB will be done • SpecimenID: for ex. (B2-1) The file name shall have no space. Use' _' instead of space		
9- Repeat step 6 consistently and ensure the lead has the same length and is broken in the orientation on the same point		
10- Repeat step 7 and 8		
11- Rest the specimen on the roller support. The following is to be considered: • The support shall be centred on the support's marks on the specimen surface • The specimen to be rested in the orientation where the weld root is the bottom, and the weld face is top • The specimen ID written on the front surface shall be clear to the recording camera		
12- Prepare for the loading test. **Specimen PB** will have **no** loading test • Lower the crosshead toward the specimen. Ensure that the loading piston does not touch the specimen		
13- Take a picture of the test setup		
14- Start video recording with the camera viewing the test setup		
15- Start recording from the AE Win		

16- Start the loading from the loading machine. • Write the start time and finish time of the test • Write the load applied	Start time……………………………… Finish time…………………………… Load applied…………………………..N	
17- Stop recording from AE Win		
18- Save the data from AE Win in the native format, and the file shall be named 'Loading_SpecimenID' • SpecimenID: for ex. (B2-1) • The file name shall have no space. Use' _' instead of space		
19- Lift the crosshead, remove the sensors from the specimen		
20- Stop video recording and save the recorded videos in the specified folder		
21- Measure the permanent deflection of the specimen after testing; by putting it on a flat surface. The specimen shall be concaved up. Using the calliper, measure the maximum deflection in the centre of the weld face. Record down the value in step 4		
22- Take a picture showing the deflection • Save the picture in the specified folder with the name 'deflection_after_test'		

Summary

Summary of the information gathered and events that occurred during testing.

Experiment Type	S1	Hits _S1	S2	Hits _S2	S3	Hits _S3	S4	Hits _S4	S5	Hits _S5	Load Applied (KN)	Deflection (mm)	Notes	Test Setup
PLB on table	R1 5a	6	R1 5a	7										
PLB on table	R1 5a	20	R1 5a	22									2 times PLB were performed in the same recording	
PLB on table	R1 5a	5	R1 5I	4	R1 5a	4							2 times PLB was performed in the same recording The location setup was not modified on AE win software for this specimen	
Four-point Bending Flexural Test using SHIMADZU	R1 5a	292	R1 5I	173	R1 5a	201					5.2	0.63	The recording started while the loading machine had some preliminary adjustments. Some of the signals at the start are expected to be from the machine adjuments The location setup was not modified on AE win software for this specimen	
PLB on table 1	R1 5a	2	R1 5I	3	R1 5a	1								

PLB on table 2	R1 5a	5	R1 5l	3	R1 5a	7					
PLB on machine	R1 5a	9	R1 5l	6	R1 5a	7					
Four-point Bending Flexural Test using SHIMADZU	R1 5a	82	R1 5l	60	R1 5a	94			7.5	2.86	This test was a repetition of the loading test since minor deflection was found in the 1st test
PLB on table 1	R1 5a	3	R1 5l	3	R1 5a	5					Specimen B1-2 had bubbles on the welded area. Theses bubles were removed using flat file tool
PLB on table 2	R1 5a	4	R1 5l	4	R1 5a	3					
Four-point Bending Flexural Test using SHIMADZU	R1 5a	77	R1 5l	80	R1 5a	38			5.5	2.36	
PLB on table 1	R1 5a	2	R1 5l	7	R1 5a	7					
PLB on table 2	R1 5a	1	R1 5l	3	R1 5a	4					
PLB on machine	R1 5a	1	R1 5l	1	R1 5a	2					
Four-point Bending Flexural Test using SHIMADZU	R1 5a	6	R1 5l	23	R1 5a	22			7.8	6.48	This test was a repetition of the loading test since minor deflection was found in the 1st test
PLB on table 1	R1 5a	5	R1 5l	4	R1 5a	4					Specimen B3-1 had a mistake in placing the sensors and was noticed after finishing the loading test Sensor S2 was located at 350 mm from the centre instead

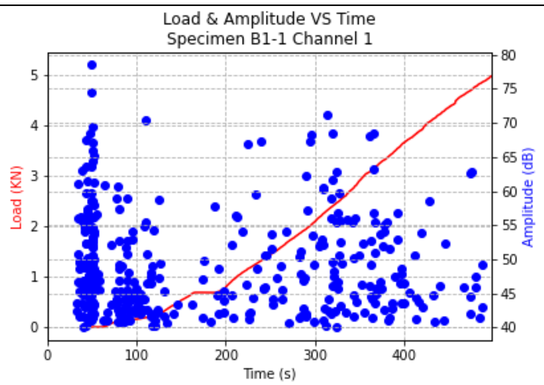
													of 300 mm. This resulted in inaccurate result on the location graph from AE win
PLB on table 2	R15a	5	R15I	3	R15a	3							
Four-point Bending Flexural Test using SHIMADZU	R15a	25	R15I	27	R15a	43					6.8	4.87	
PLB on table 1	R15a	4	R15a	6	R15I	1	R15I	1	R15I	1			
PLB on table 2	R15a	2	R15a	3	R15I	2	R15I	1	R15I	1			
PLB on machine	R15a	2	R15a	1	R15I	1	R15I	1	R15I	1			
Four-point Bending Flexural Test using SHIMADZU	R15a	16	R15a	21	R15I	1	R15I	2	R15I	5	8.5	27.8	
PLB on table 1	R15a	2	R15a	2	R15I	10	R15I	9	R15I	10			It was noticed that the R15I sensors didn't capture many signals on Specimen B2-1. The reason is that they were far from the source location. So the preamplifier for S3, S4 & S5 were reduced to 30 dB.

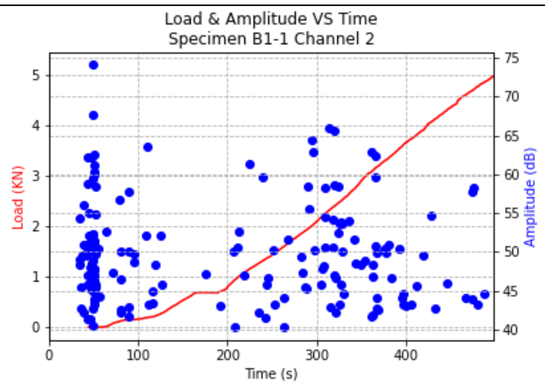
PLB on table 2	R15a	2	R15a	2	R15l	6	R15l	6	R15l	6			
PLB on machine	R15a	2	R15a	1	R15l	1	R15l	1	R15l	1			
Four-point Bending Flexural Test using SHIMADZU	R15a	4	R15a	5	R15l	31	R15l	31	R15l	31	8.2	20.81	
PLB on table 1	R15a	3	R15l	1	R15a	2							
PLB on table 2	R15a	1	R15l	1	R15a	1							
Four-point Bending Flexural Test using SHIMADZU	R15a	28	R15l	26	R15a	30					7.49	1.95	
PLB on table 1	R15a	2	R15l	1	R15a	2							
PLB on table 2	R15a	6	R15l	4	R15a	4							
Four-point Bending Flexural Test using SHIMADZU	R15a	16	R15l	20	R15a	23					9	52.24	
PLB on table 1	R15a	4	R15l	2	R15a	5							
PLB on table 2	R15a	2	R15l	2	R15a	3							
PLB on machine	R15a	4	R15l	3	R15a	5							
Four-point Bending Flexural Test using SHIMADZU	R15a	0	R15l	15	R15a	18					8.5	34.82	SHIMADZU started losing force and at the same time a hit with high amplitude of almost 100 dB was recorded on all the sensors

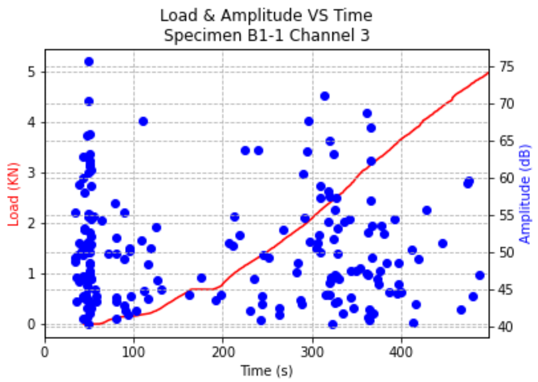
Dropping 9.5 mm steel ball from 15 cm height	R1 5a	2	R1 5a	3	R1 5I	5	R 15 I	5	R 15 I	5		
Dropping 9.5 mm steel ball from 15 cm height	R1 5a	4	R1 5a	5	R1 5I	4	R 15 I	4	R 15 I	4		
Dropping 9.5 mm steel ball from 15 cm height	R1 5a	15	R1 5a	13	R1 5I	19	R 15 I	19	R 15 I	19		
PLB on table 1	R1 5a	4	R1 5a	3	R1 5I	8	R 15 I	8	R 15 I	8		
PLB on table 2	R1 5a	3	R1 5a	3	R1 5I	9	R 15 I	9	R 15 I	9		
PLB on machine	R1 5a	7	R1 5a	9	R1 5I	9	R 15 I	9	R 15 I	9		
Four-point Bending Flexural Test using SHIMADZU	R1 5a	63	R1 5a	65	R1 5I	635	R 15 I	635	R 15 I	635	8.4	60.71

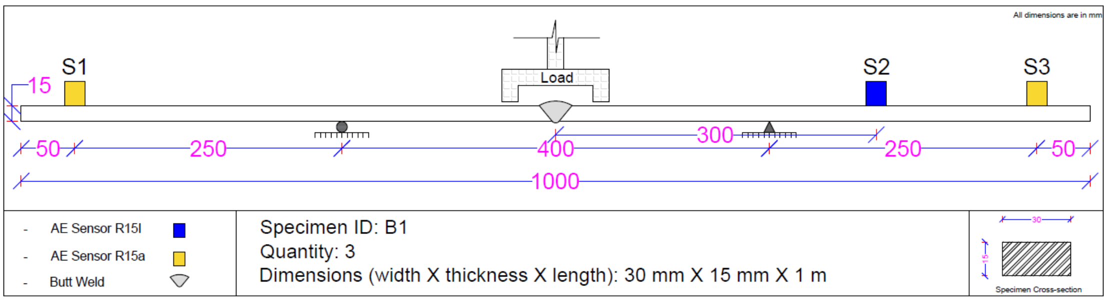
Appendix B - Signal Strength and Load VS Time Plots

Specimen B1-1

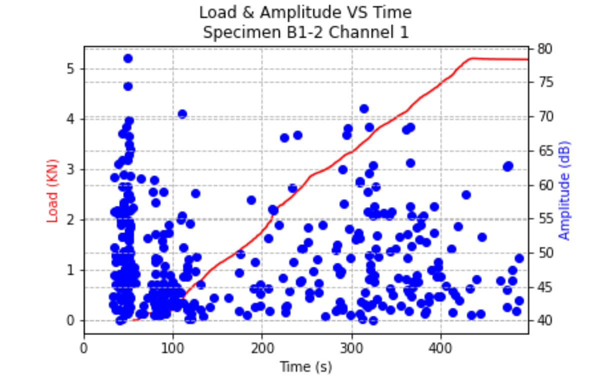


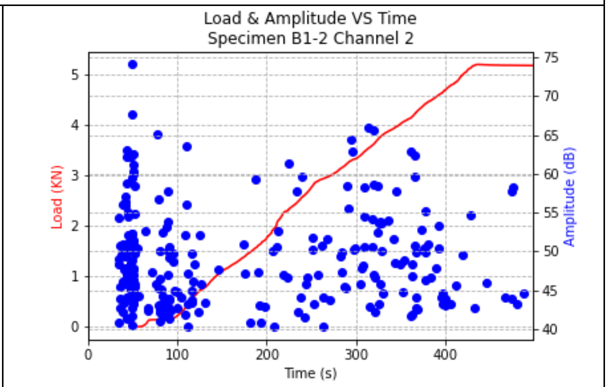


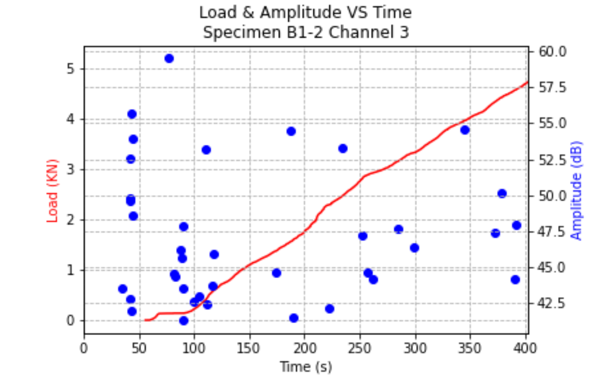


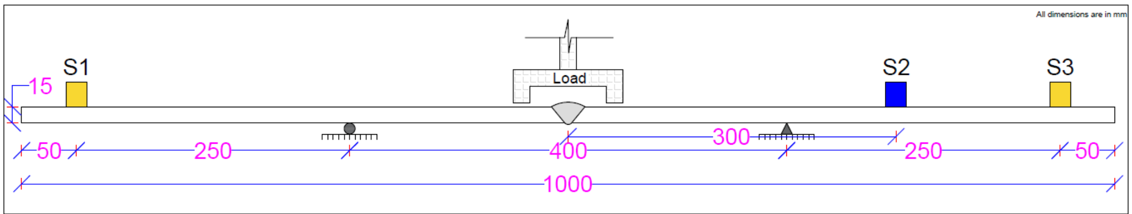


Specimen B1-2



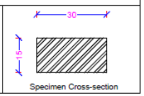




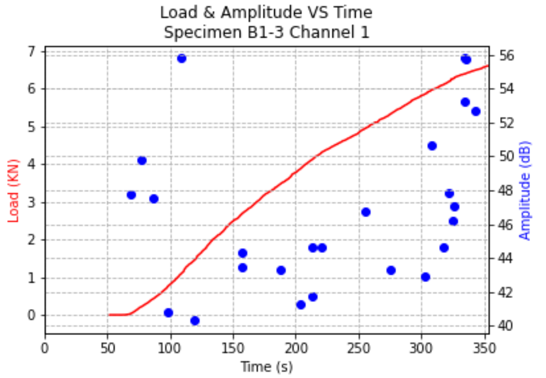


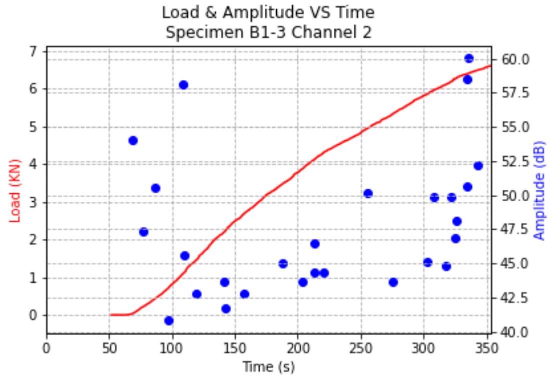
- AE Sensor R15I
- AE Sensor R15a
- Butt Weld

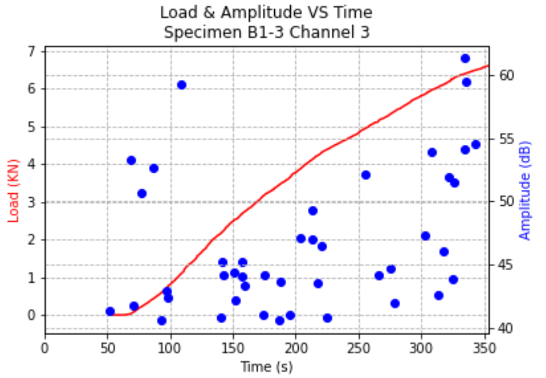
Specimen ID: B1
Quantity: 3
Dimensions (width X thickness X length): 30 mm X 15 mm X 1 m

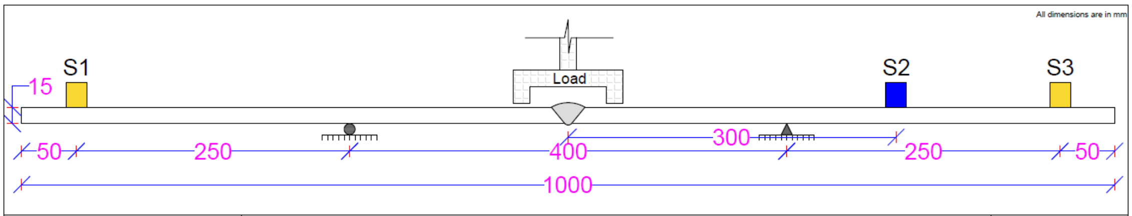


Specimen B1-3



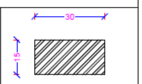






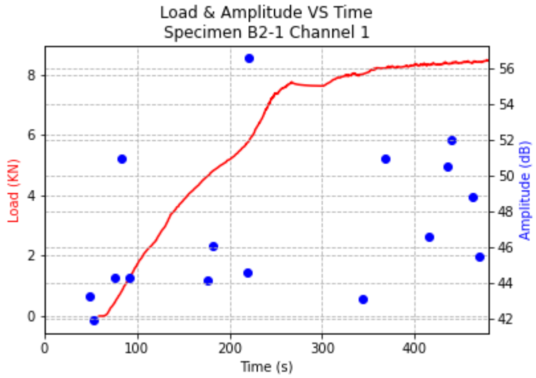
- AE Sensor R15I
- AE Sensor R15a
- Butt Weld

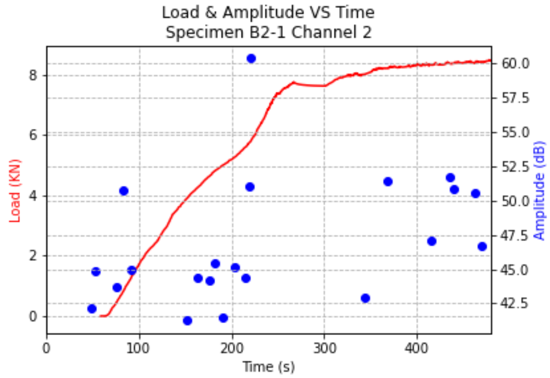
Specimen ID: B1
Quantity: 3
Dimensions (width X thickness X length): 30 mm X 15 mm X 1 m

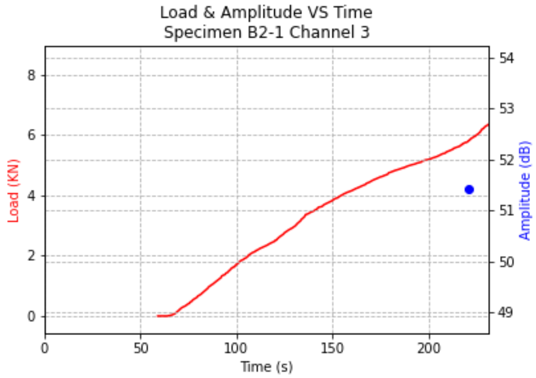


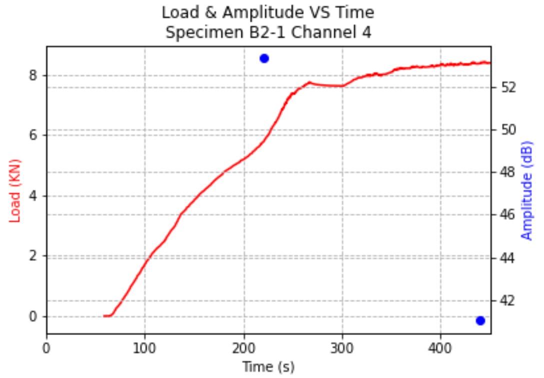
Specimen Cross-section

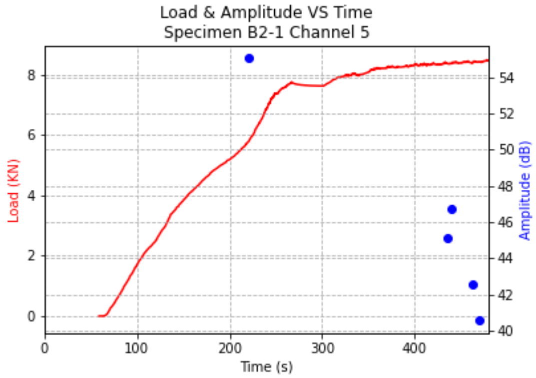
Specimen B2-1

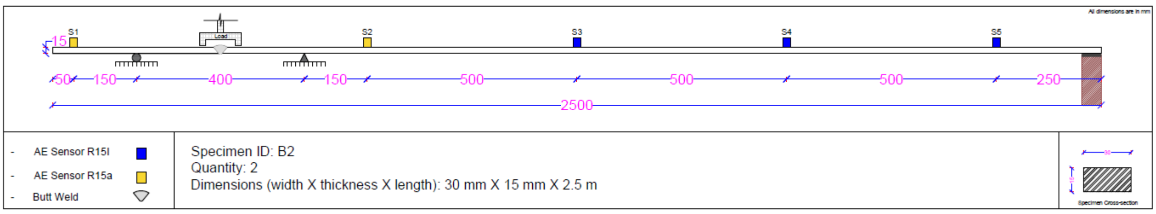




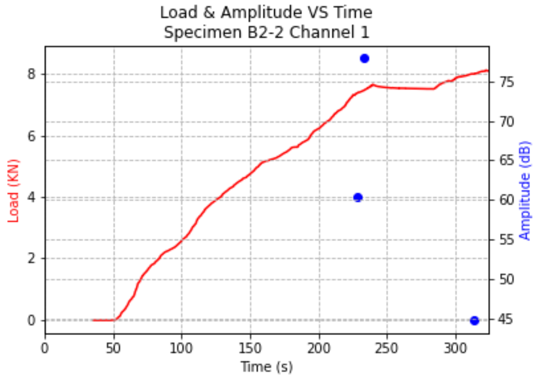


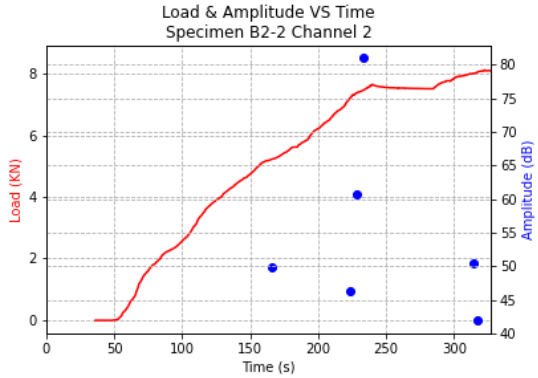


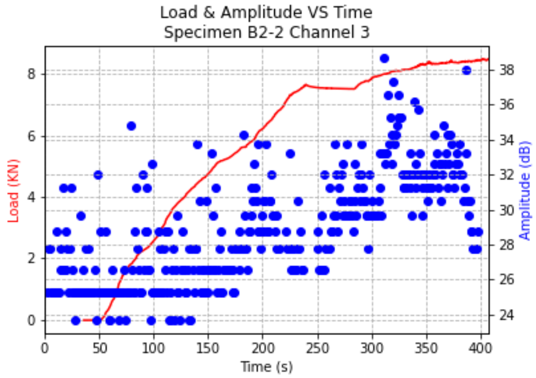


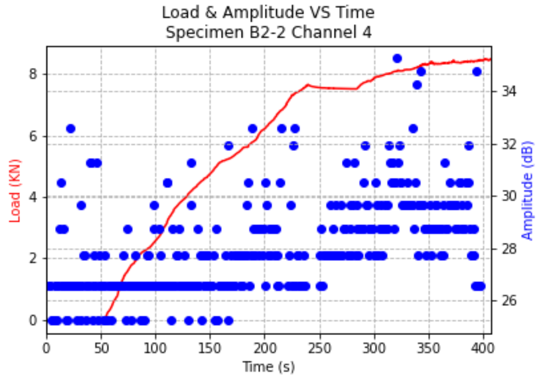


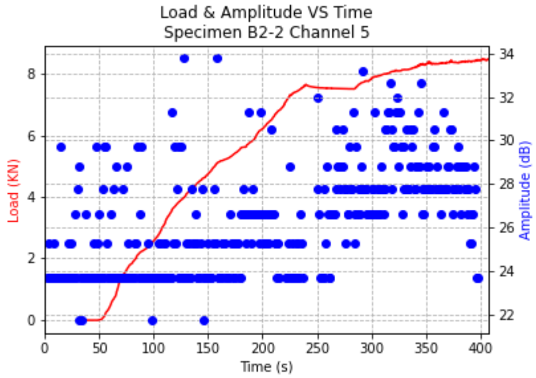
Specimen B2-2

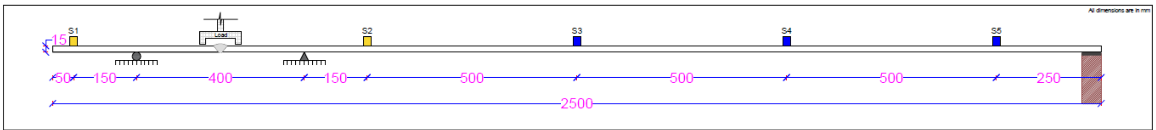








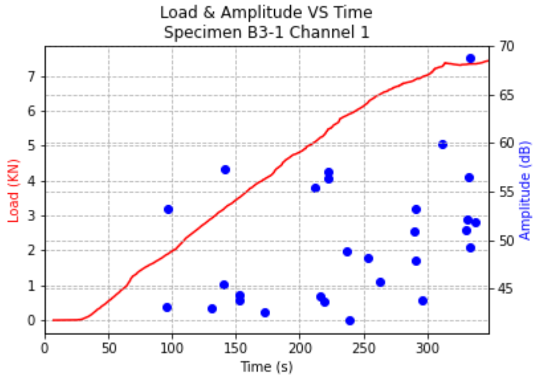


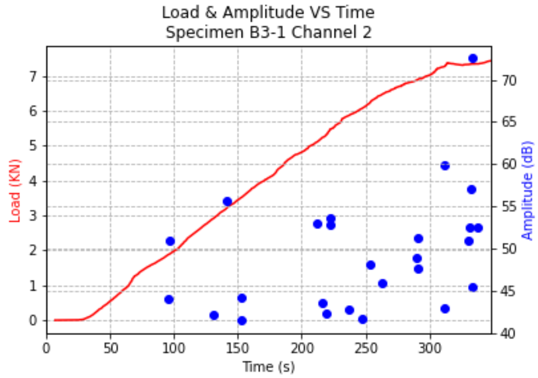


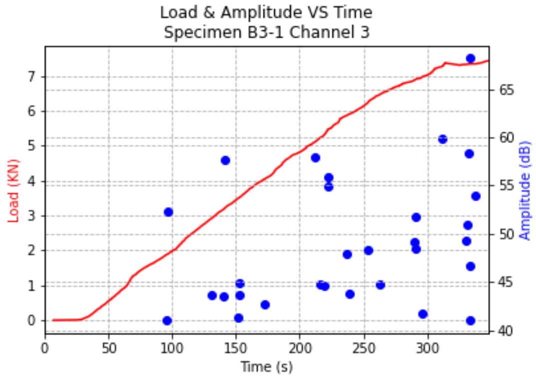
- AE Sensor R15I
- AE Sensor R15a
- Butt Weld

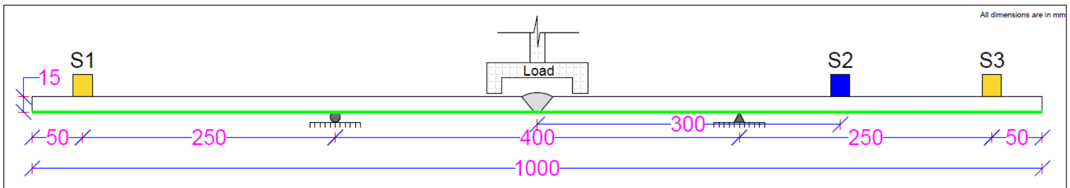
Specimen ID: B2
Quantity: 2
Dimensions (width X thickness X length): 30 mm X 15 mm X 2.5 m

Specimen B3-1



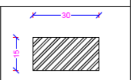






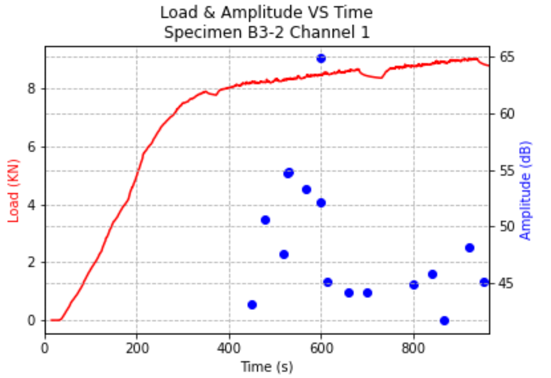
- AE Sensor R15I
- AE Sensor R15a
- Butt Weld
- Coat

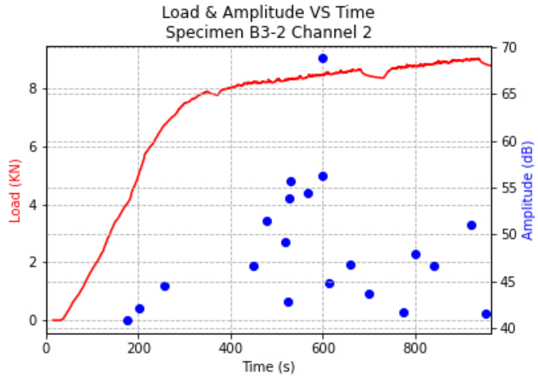
Specimen ID: B3
Quantity: 3
Dimensions (width X thickness X length): 30 mm X 15 mm X 1 m

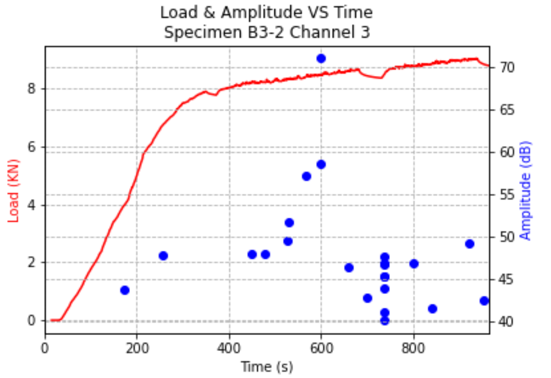


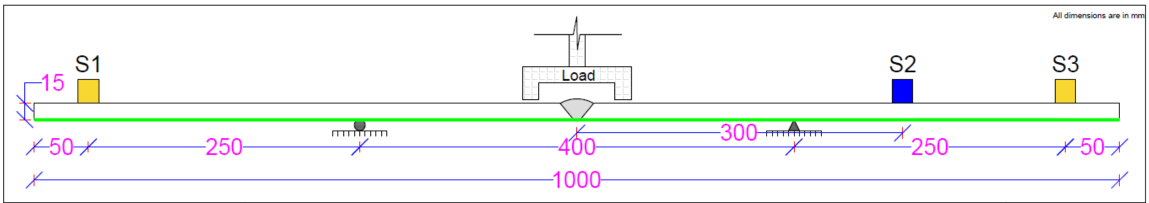
Specimen Cross-section

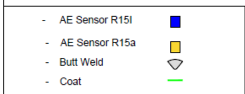
Specimen B3-2



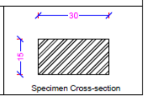




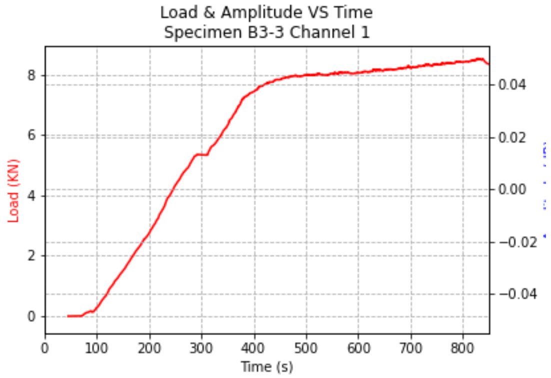


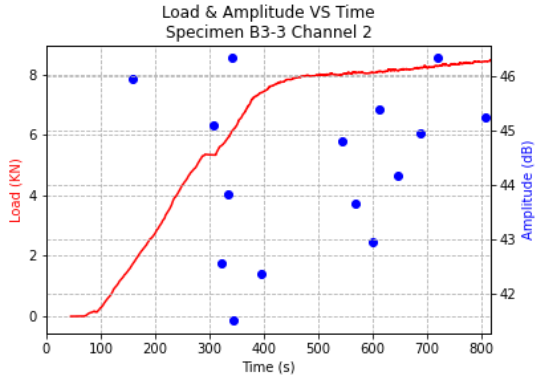


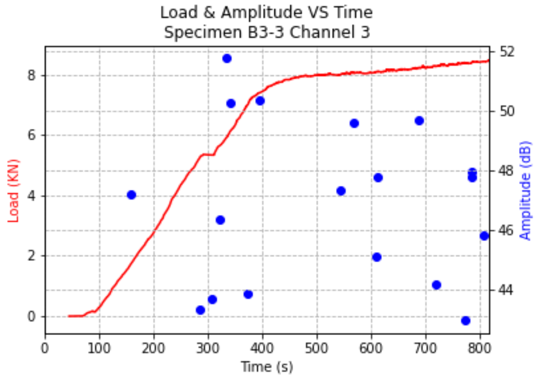
Specimen ID: B3
Quantity: 3
Dimensions (width X thickness X length): 30 mm X 15 mm X 1 m

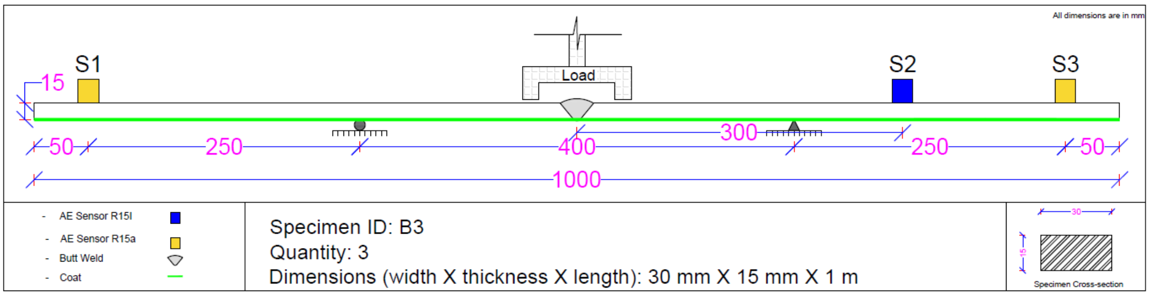


Specimen B3-3



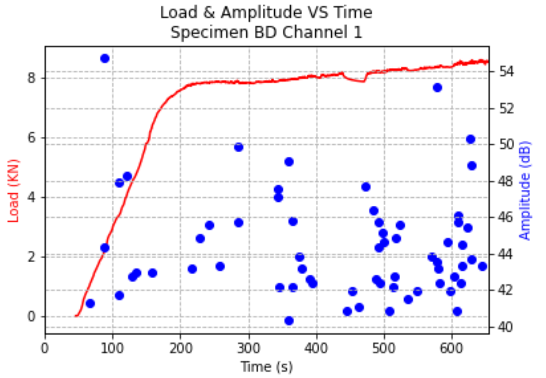


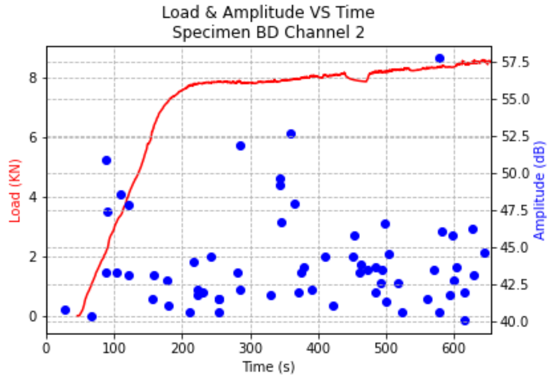


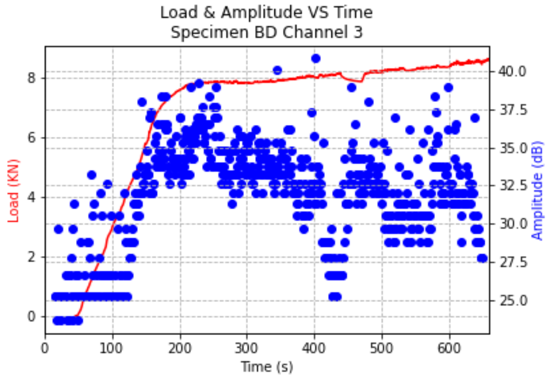


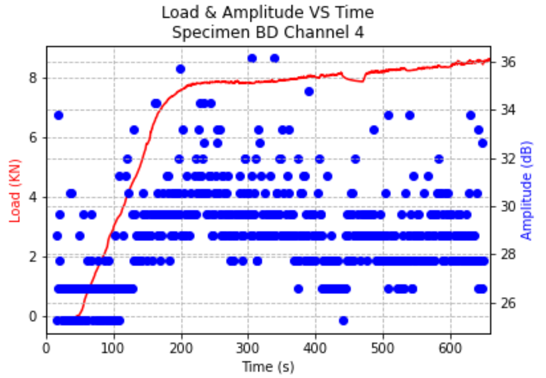
Specimen ID: B3
Quantity: 3
Dimensions (width X thickness X length): 30 mm X 15 mm X 1 m

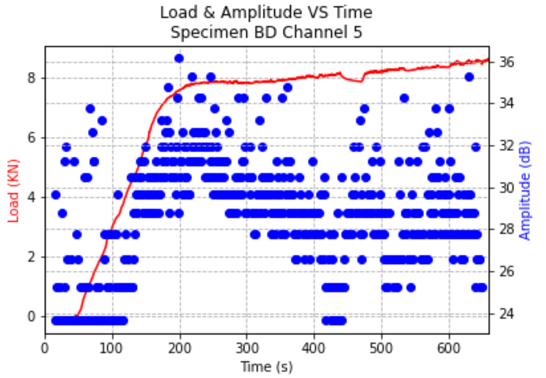
Specimen BD

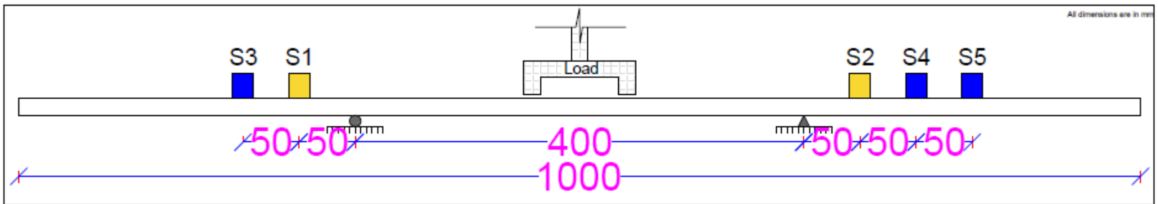










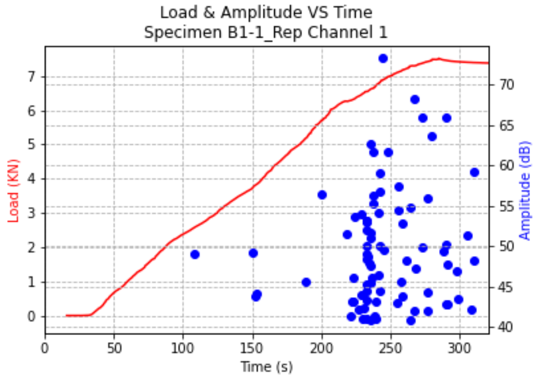


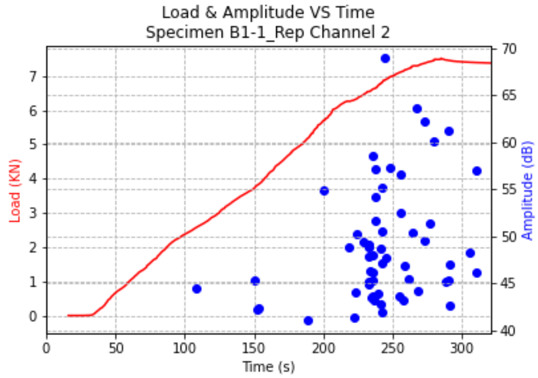
- AE Sensor R15I
- AE Sensor R15a

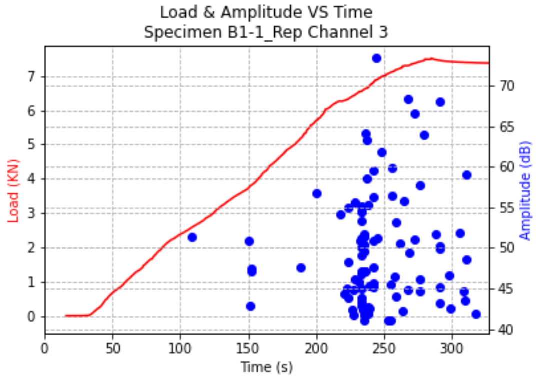
Specimen ID: BD
Quantity: 1
Dimensions (width X thickness X length): 30 mm X 15 mm X 1 m

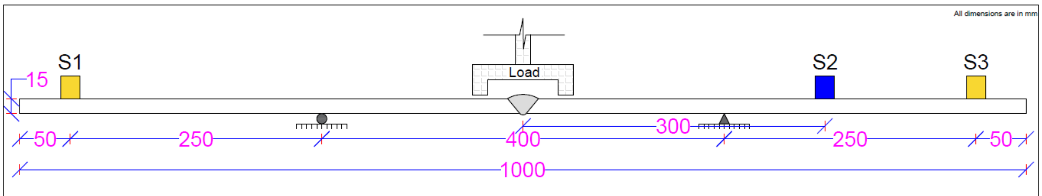
Specimen Cross-section

Specimen B1-1_Rep



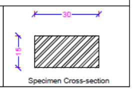




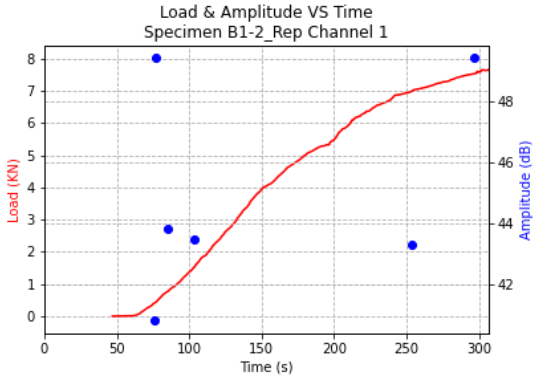


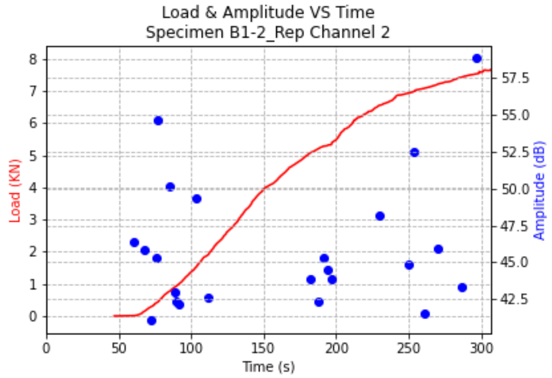
- AE Sensor R15I
- AE Sensor R15a
- Butt Weld

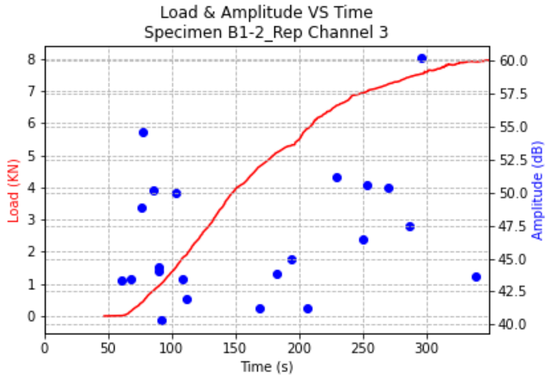
Specimen ID: B1
Quantity: 3
Dimensions (width X thickness X length): 30 mm X 15 mm X 1 m

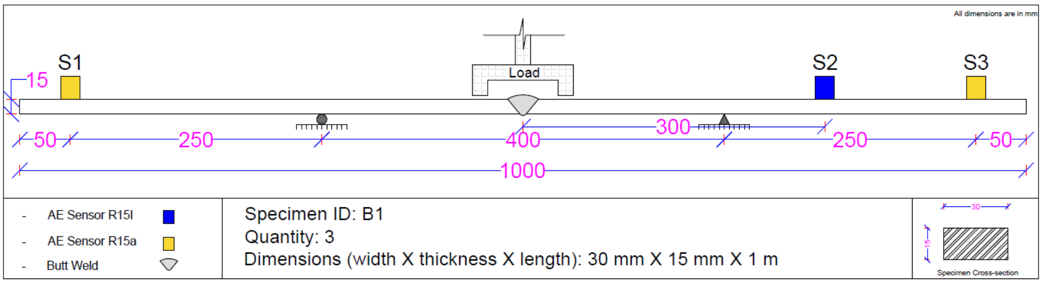


Specimen B1-2_Rep









Specimen ID: B1
Quantity: 3
Dimensions (width X thickness X length): 30 mm X 15 mm X 1 m

Appendix C – Specimens Documents

CO #	Item #	Del #	Heat	Lot	Your art #	Qty	Description
	100	RP8351 06	42589			1	FLATTSTÅL S355J2 120 x 15mm x 6 m

NORSK STÅL

This document is electronically reproduced and is identical to the original.

Duferco Danish Steel

Duferco GROUP

DK-3300 Frederiksværk - Telefon +45 47767600

BESCHEINIGUNG / CERTIFICATE

Stabstahl/Bars

Seite / Page: 1/1 Nr./No.: 124150

Type: EN 10204 / 3.1

Ihrer Auftrag/Your order P03012030

Unser Auftrag/Our order 73479

Datum/Date: 14-01-2021

Lieferstelle/Delivery address:

Norsk Stål AS, avd Horten
Nedre Vei 8
NO-3187 - Horten
NORWAY

NORSK STÅL AS LAGER
Postboks 123
NO-1378 - Nesbru
NORWAY

Lieferbedingung / Specification:
EN 10025-2:2019 S355J2+AR

Toleranz Tolerance: EN10058

	Pos.	Product Type	Abmessungen/Dimensions			Stk/Pcs	Gewicht/Weight	Schmelze/Heat
1	2	Flad	6000	50.0	5.0	1	2025	42311
2	2	Flad	6000	50.0	5.0	1	1998	42312
3	2	Flad	6000	50.0	5.0	1	2032	42320
4	10	Flad	6000	70.0	15.0	1	2047	40610
5	8	Flad	6000	40.0	15.0	1	2035	42315
100 6	12	Flad	6000	120.0	15.0	1	2065	42589
7								
8								
9								
10								

Total weight: 12202

Lieferzustand/Condition of delivery
Walzzustand / As rolled

Kennzeichnung/Marking CE

Sachverständigen/Quality inspector:

Schmelznr/Heatno.
10.000-29.000 Konverter-verfahren/Oxygen converter
30.000-99.999 Elektro-Ofen/El-arc-furnace
Im Pfanne raffiniert/Ladle refined

	C	Mn	Si	P	S	Cr	Cu	Ni	Mo	Sn	Al	Nb	Ti	V	B	N	Ceq = Carbon-Equivalent (IIW - formula)
1	14	135	21	15	10	8	27	12	3		4	0	12	37		90	42
2	14	135	21	15	7	9	27	12	3		5	1	13	37		100	42
3	14	132	21	23	8	12	27	10	3		0	1	15	43		70	42
4	13	135	22	13	13	7	26	13	3		1	1	15	38		80	41
5	13	131	20	12	12	8	24	10	3		0	1	15	34		50	40
6	10	134	19	17	23	7	31	12	2		2	1	1	39		120	38
7																	
8																	
9																	
10																	
	%x100		%x1000			%x100				%x1000					%x10.000		%x100

	Zugversuch/Tensile test			Kerbschlagbiegeversuch/Impact test/ISO - V					Hardness
	ReH	Rm	A5	1	2	3	Mittel/Average	Temp	
1	411	549	30	47	41	35	41	-20	
2	388	564	31	41	36	45	41	-20	
3	403	559	27	47	44	49	47	-20	
4	379	512	30	148	150	150	149	-20	
5	367	504	35	145	157	133	145	-20	
6	435	534	29	48	64	53	55	-20	
7									
8									
9									
10									
	MPa		%	Joule				°C	HB

DDS requires, that all billet suppliers make statement in writing to the effect, that the billets delivered have not been exposed to and / or contaminated with radiation.

Bei Material mit einer Dicke unter 10 mm, werden die Kerbschlagproben zu einer Breite von entweder 7.5 mm oder 5.0 mm bearbeitet.

For material with thickness less than 10 mm, the impact testpieces are machined to a width of either 7.5 mm or 5.0 mm.

Wir bestätigen, dass die Lieferung den Forderungen der obengenannten Lieferbedingungen und des Auftrages entspricht.
We hereby certify, that the material has been produced and tested in compliance with the mentioned specification and with the requirement of the order.

Duferco Danish Steel A/S

Mauro Bucciolini
Quality Department

CE
08
0045 CPR-0820
EN 10025-1

See www.duferco.dk for Declaration Of Performance: DDS 009

INSPECTION CERTIFICATE (3.1) - Chemical analysis
TEST REPORT (2.2) - Mechanical properties

Cert no: 25005174
Date: 08-01-2019

PURCHASE ORDER : 10005 / 27.11.18

PRODUCT NAME: NST Carbomig 2N
TYPE OF PRODUCT: Solid MAG Wire
STANDARD CLASSIFICATION:

EN ISO 14341-A	G46 2 M21 3Si1
	G42 2 C1 3Si1
AWS/SFA. 5.18 / SFA5.18	ER70S-6

DIAMETER / WEIGHT: 0.80 mm Kg.5 D-200 (Kg.1.000) / 0.80 mm P.L.W. D-300 (Kg.5.400)
1.00 mm P.L.W. D-300 (Kg.4.320) / 1.00 mm DRUM 250 KG. (Kg.2.000)
1.20 mm DRUM 250 KG. (Kg.3.000)

HEAT N°: 2056484

CHEMICAL COMPOSITION

acc to EN 10204 - 3.1

C	Mn	S	P	Si	Cu	Al	Mo	Ni	Cr	Ti	N	
0.073	1.452	0.013	0.010	0.855	0.003	0.002	0.002	0.011	0.020	0.001	0.0055	

MECHANICAL PROPERTIES OF ALL WELD METAL

(Shielding gas 80% Argon + 20% CO_2, and 100% CO_2)
Typical data / acc to EN 10204 - 2.2

PROTECTIVE GAS	YIELD STRENGHT $R_{p0.2}$	TENSILE STRENGH R_m	ELOGANTION A_5	Charpy Impact V-Notch Temperature (°C)	Absorbed Energy (J)
M21	470 Mpa	560 Mpa	26%	-30°	> 47
CO_2	440 Mpa	530 Mpa	26%	-20°	> 47

POR/BY/DE
DPT. QUALITY

CE
0035 15
0035 – CPR – C906
DoP NSTAS - 49
EN 13479

COMMENT: We hereby confirm that the material herein described has been manufactured, sampled, tested and inspected in accordance with refered standars. Product supplied under a QA Programme fulfilling the EN ISO 9001 standard. This certificate is produced electronically and is valid without signature.

Norsk Sveiseteknikk AS
Headquarter
P.O.Box 109, 3301 Hokksund, Norway.

Telephone: + 47 99 27 80 00
Telefax: + 47 94 74 02 27

E-mail: nst@nst.no
www.nst.no

Perfect Welding by NST
This certificate has been inspected and approved
Visit us at www.nst.no for info and certificates

NST Carbomig 2N

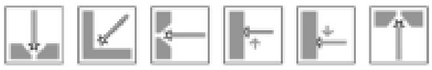
AWS: A5-18/SFA5.18: ER70S-6
EN ISO 14341-A: G 46 2 M21 3Si1
EN ISO 14341-A: G 42 2 C1 3Si1

Homogen tråd for sveising av vanlige konstruksjonsstål.

Generell beskrivelse:

NST Carbomig2N, er en kobberbelagt homogen (SG2) tråd for Mig/Mag sveising av ulegert stål med CO_2 eller Mixgass Ar/CO_2 som dekkgass.
Tråden har stor parameterboks meget god sveisbarhet som gir ett meget godt resultat.

Sveisestillinger:



Strømart:

DC+

Gasstype/mengde:

Ar/CO_2 eller ren CO_2

12 –20 l/min

Typical chemical composition of welding wire:

C	Si	Mn								
0,08	0,90	1,50								

Mekaniske verdier i rent sveisemetall:

Brudd og flytegrense			Slagseighet	
Flytegrense Mpa	Bruddgrense Mpa	Forlengelse %	Charpy V (J) -30 ºC	
≥420	500-640	≥22	≥47	

Forpakningsdata:

0,8mm x 15Kg + fat Ø51cm
1,0mm x 15Kg + fat Ø51cm
1,2mm x 15Kg + fat Ø51cm

Godkjenninger:

TÜV, CE,
- andre vurderes etter behov.

Referanse/dato:

NST Carbomig2N,
Norsk, 29.04.2016.

Perfect Welding www.nst.no

Appendix E – Developed Code

Signal Strength and Load VS Time

Plot Amplitude and Load VS Time

File	Plot_Amp_Load_Time.ipynb
Project	AET signal characteristics
Client	N/A
Purpose	Plot the load data in KN from Shimadzu with the signal amplitude in dB from AE wiin
Prerequisite	Works with Python 3.4 and newer.
Created by	2021-06-10 Elkhayat
Expanded and modified by	
Date/version	2021-06-10/ 1.0: First release

In [3]:
```
from pathlib import Path
from typing import Collection

import pandas as pd  # Import pandas for data manipulation
import argparse  # Import argparse for parsing command line parameters
import numpy as np
import matplotlib.pyplot as plt  # Import matplotlib module for plotting
import matplotlib.patches as mpatches  # Needed for putting label on figure
import pickle  # Used for serializing object to disk, we use this to store the figure to retrieve later
# this allows us to interact with the model without calculating values again.
import sys
import os  # Import os for doing file operations, read the files from specified folder
import re  # Import re for regular expression to select correct files and parse channels and Time
# of test and number of hit
from datetime import datetime  # Used to interact and calculate dates and time for output files

from natsort import natsorted, ns  # sort file names naturally
import math
from math import log
from enum import Enum
from scipy import signal
import matplotlib.pyplot as plt
```

In [4]:
```
os.chdir(r'C:/Users/Yehia/Desktop/1-UiS/4-Thesis DNV GL/19-Python_Final/2021/Plot_Amp_Load_Vs_Time')
print(os.getcwd())
```

```
C:\Users\Yehia\Desktop\1-UiS\4-Thesis DNV GL\19-Python_Final\2021\Plot_Amp_Load_Vs_Time
```

Read the waveform files from AE Win

In [583…
```
#Input
SpecimenID='B1-2_Rep' #to be changed for every run based on the specimen of intereset
InpFolder='./AEwin/Loading_B1-2_Rep' #to be changed for every run based on the specimen of intereset
ChannelId=3 #to be changed for every run based on the channel of intereset
OutFolder='OutPut'
#Getting the AE win files
files = [f for f in os.listdir(InpFolder) if re.match(r'.*_' + str(ChannelId) + '_\\d+_\\w+', f)]
print('list of files',files)
plot_list = []
sorted_files = natsorted(files, alg=ns.REAL)
```

```
list of files ['Loading_B1-2_3_10_92024107.csv', 'Loading_B1-2_3_11_96097708.csv', 'Loading_B1-2_3_12_153044940.csv', 'Loading_B1-2_3_13_166232305.csv', 'Loading_B1-2_3_14_178398686.csv', 'Loading_B1-2_3_15_191033170.csv', 'Loading_B1-2_3_16_213759413.csv', 'Loading_B1-2_3_17_234092065.csv', 'Loading_B1-2_3_18_237455028.csv', 'Loading_B1-2_3_19_253866397.csv', 'Loading_B1-2_3_1_44743929.cs
```

```
v', 'Loading_B1-2_3_20_271011036.csv', 'Loading_B1-2_3_21_280551264.csv', 'Loading_B1-2_3_22_323246287.csv', 'Loading_B1-2_3_2_51968463.csv', 'Loading_B1-2_3_3_60223980.csv', 'Loading_B1-2_3_4_60721057.csv', 'Loading_B1-2_3_5_69177930.csv', 'Loading_B1-2_3_6_73201252.csv', 'Loading_B1-2_3_7_73562063.csv', 'Loading_B1-2_3_8_75505178.csv', 'Loading_B1-2_3_9_87057646.csv']
```

In [584…

```
PlotDF = pd.DataFrame(columns=["Time", "Amp_dB"])# Creating dataframe to hold the time and the calculated amplitude from every file
```

Calculating Signal Amplitude in dB

dB = 20 log (V_{max}/1μ-volt) – (Preamplifier Gain in dB).

In [585…

```
#reading file and removing the ineeded rows and printing it to csv again
for fileName in sorted_files:
    frame = pd.read_csv(InpFolder+'/'+fileName)
    frame.drop(frame.index[2:9], inplace=True)
    frame.drop(frame.index[0], inplace=True)
    frame.to_csv('.\Test'+'/'+SpecimenID+'/'+fileName)
```

In [586…

```
#reading the printed files again
InpFolder='./Test'+'/'+SpecimenID
files1 = [f for f in os.listdir(InpFolder) if re.match(r'.*_' + str(ChannelId) + '_\\d+_\\w+', f)]
files1 = natsorted(files1, alg=ns.REAL)
```

In [587…

```
#looping to put the time and caluclate the aplitude in dB from every file
x=1/1000000
for fileName in files1:
    frame1 = pd.read_csv('.\Test'+'/'+'/'+SpecimenID+'/'+fileName,sep=',')
    frame1.drop(frame1.columns[1:3], axis='columns',inplace=True)
    TimeOfTest=frame1.iloc[:2,:]
    dfSplit=TimeOfTest['Unnamed: 0'].str.split(":",expand=True,)
    Time=float(dfSplit.iloc[0,3])+float(dfSplit.iloc[1,1])
    frame2=frame1.iloc[2:,:].reset_index(drop=True)
    frame3=pd.to_numeric(frame2['Unnamed: 0'])
    Vmax=abs(max(frame3, key=abs))
    dB=20*math.log10(Vmax/x)-40
    print(float(dfSplit.iloc[0,3]),float(dfSplit.iloc[1,1]),Time,dB)
    PlotDF=PlotDF.append({'Time':Time,
                'Amp_dB':dB},ignore_index=True)
```

```
16.0 44.7439295 60.7439295 43.3158275307091
16.0 51.9684635 67.9684635 43.49492099544335
16.0 60.2239805 76.22398050000001 48.871830368978934
16.0 60.7210575 76.7210575 54.65046610041077
16.0 69.1779305 85.1779305 50.19711785092028
16.0 73.2012525 89.2012525 44.0110678314112
16.0 73.5620635 89.5620635 44.33887534429351
16.0 75.5051785 91.5051785 40.32058251923442
16.0 87.0576465 103.0576465 50.03166774469082
16.0 92.0241075 108.0241075 43.49492099544335
16.0 96.0977085 112.0977085 41.946680496086316
16.0 153.0449405 169.0449405 41.28667346720371
16.0 166.2323055 182.2323055 43.84240211957885
16.0 178.3986865 194.3986865 44.959563472327275
16.0 191.0331705 207.0331705 41.28667346720371
16.0 213.7594135 229.7594135 51.20194101448213
16.0 234.0920655 250.0920655 46.46798356671347
16.0 237.4550285 253.4550285 50.5974604358277
16.0 253.8663975 269.8663975 50.35947525757314
16.0 271.0110365 287.0110365 47.42081759467639
16.0 280.5512645 296.5512645 60.19189772032071
16.0 323.2462875 339.2462875 43.670401921187064
```

In [588…

```
PlotDF
```

Out[588…

	Time	Amp_dB
0	60.743930	43.315828
1	67.968463	43.494921
2	76.223981	48.871830
3	76.721058	54.650466
4	85.177931	50.197118
5	89.201252	44.011068
6	89.562063	44.338875
7	91.505178	40.320583
8	103.057647	50.031668
9	108.024107	43.494921
10	112.097708	41.946680
11	169.044940	41.286673
12	182.232305	43.842402
13	194.398686	44.959563
14	207.033171	41.286673
15	229.759413	51.201941
16	250.092065	46.467984
17	253.455028	50.597460
18	269.866398	50.359475
19	287.011036	47.420818
20	296.551265	60.191898
21	339.246287	43.670402

Read the file from Shimadzu

In [589…

```
#reading the file shimadzu after modyfying it manually
frame_shimadzu = pd.read_csv('.\Shimadzu'+'/'+SpecimenID+'.csv')
```

In [590…

```
frame_shimadzu
```

Out[590…

	Time	TimeModified	Load	def
0	0.000000	47.000000	0.000415	0.000075
1	0.100000	47.100000	0.000201	0.000660
2	0.200000	47.200000	-0.000003	-0.001154
3	0.300000	47.300000	0.000033	-0.001622
4	0.400000	47.400000	0.000164	-0.000902
...	...	...	...	...
4046	404.600019	451.600019	-0.131801	-8.648950

	Time	TimeModified	Load	def
4047	404.700019	451.700019	-0.131916	-8.649650
4048	404.800019	451.800019	-0.131862	-8.650052
4049	404.900019	451.900019	-0.131933	-8.650607
4050	405.000019	452.000019	-0.131821	-8.650688

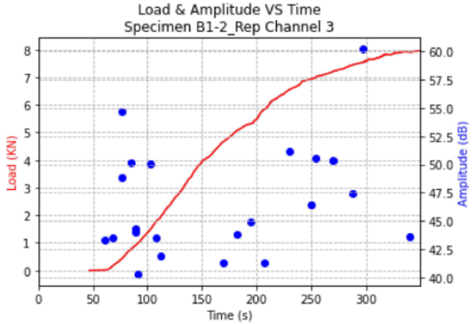
4051 rows × 4 columns

In [591…

```
%matplotlib inline
```

In [592…

```
#Plotting and saving the plot
fig, ax1=plt.subplots()
ax1.plot(frame_shimadzu['TimeModified'],frame_shimadzu['Load'],c='r')
plt.xlabel('Time (s)')
plt.ylabel('Load (KN)',c='r')

plt.grid(linestyle='--')
ax2=ax1.twinx()
ax2.scatter(PlotDF['Time'],PlotDF['Amp_dB'],c='b')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude (dB)',c='b')
plt.xlim(0,max(PlotDF['Time'])+10)

plt.grid(linestyle='--')
plt.title('Load & Amplitude VS Time\n'+'Specimen '+SpecimenID+' Channel '+ str(ChannelId))
fig1=plt.gcf()
plt.show()
plt.draw()

fig1.savefig(OutFolder+'/'+SpecimenID+'Channel '+str(ChannelId)+' Load and Amplitude VS Time'+'.png')
```



<Figure size 432x288 with 0 Axes>

Correlation with Timestamp (Signal Start Time)

File	Corr_Timestamp.ipynb
Project	AET signal characteristics
Client	N/A
Purpose	Match signals from two different channels based on the closest start time
Prerequisite	Works with Python 3.4 and newer.
Created by	2021-06-03 Elkhayat
Expanded and modified by	
Date/version	2021-06-03 / 1.0: First release

In [1]:
```
from pathlib import Path
from typing import Collection

import pandas as pd  # Import pandas for data manipulation
import argparse  # Import argparse for parsing command line parameters
import numpy
import matplotlib.pyplot as plt  # Import matplotlib module for plotting
import matplotlib.patches as mpatches  # Needed for putting label on figure
import pickle  # Used for serializing object to disk, we use this to store the figure to retrieve later
# this allows us to interact with the model without calculating values again.
import sys
import os  # Import os for doing file operations, read the files from specified folder
import re  # Import re for regular expression to select correct files and parse channels and Time
# of test and number of hit
from datetime import datetime  # Used to interact and calculate dates and time for output files

from natsort import natsorted, ns  # sort file names naturally

from enum import Enum
```

In [3]:
```
os.chdir('/Users/Yehia/Desktop/1-UiS/4-Thesis DNV GL/19-Python_Final/2021/Corr_Timestamp/PB')#Setting up the directory
```

In [4]:
```
stats_C1 = pd.read_csv('stats_C1.txt',sep=',')
stats_C2 = pd.read_csv('stats_C2.txt',sep=',')
```

Comparison of Signals Occurring at Close Timestamp by Two Sensors This function finds the closest timestamp from the two sensors and put the corresponding data in the same row. The plot the signal source location vs maximum amplitude. Using binary search every timestamp in sensor 1 is used to look up the closest timestamp in sensor 2. The function works by finding the minimum difference in time. The same procedure is done by timestamp in sensor 2 to avoid any data loss. The duplicated rows are then removed and the data frame is sorted.The source location of the signal is calculated based on the speed at the sound traveling longitudinally in steel 0.006 μm/s and multiplying it to the difference in the timestamp of sensor 1 and sensor2 (Time at sensor 1 – Time at sensor 2). The negative value for the time difference reflects that the signal occurred in a location closer to sensor 1 and positive values reflect that the signal occurred in a location closer to sensor 2.

In [53]:
```
#Pass to the function the Output folder from Bao's ProcessSignal.py adter runing it for sensor1 and 2 (C1 and C2) and specimen ID
def find_closest_timestamp(OutputFolder,specimen_id,ChannelID1,ChannelID2):
#Creating empty dataframe to put the data of the two snesors in one dataframe
    TD=pd.DataFrame(columns=['ID_C'+ChannelID1,'StartTime_C'+ChannelID1,'MaxAmpl_C'+ChannelID1,'ID_C'+ChannelID2,'StartTime_C'+ChannelID2,'MaxAmpl_C'+ChannelID2,'TimeDifference_0.1µs','AbsoluteTi
#Reading the maximum amplitude timestamp and storing it in dataframe
    MaxAmpTimestamp_file_1=stats_C1.MaxAmpTimestamp
    MaxAmpTimestamp_file_2=stats_C2.MaxAmpTimestamp
    time_data_c1=stats_C1
    time_data_c2=stats_C2

#Using binary search to match the closest timestamp of the two sensors
```

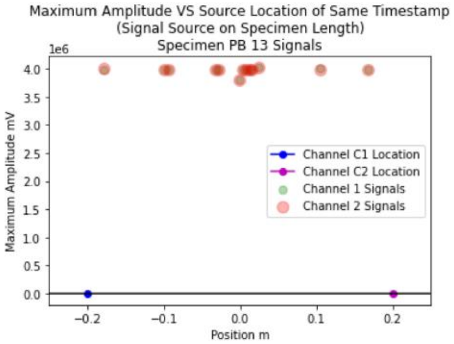
```
#1st step: the timestamp of sensor 1 is used to look up for the closest timestamp in sensor 2
    for i in range(len(time_data_c1)):
        min_diff=sys.maxsize
        low=0
        high=len(time_data_c2)-1
        closest_num=None
        if len(time_data_c1)==0 or len(time_data_c2)==0:
            return None
        target=time_data_c1["Start"][i]
        while low<=high:
            mid=(low+high)//2
            if mid+1<len(time_data_c2):
                min_diff_right=abs(time_data_c2["Start"][mid+1]-target)
            if mid>0:
                min_diff_left=abs(time_data_c2["Start"][mid-1]-target)
            if min_diff_left<min_diff:
                min_diff=min_diff_left
                closest_num=time_data_c2["Start"][mid-1]
                midindex=mid-1
            if min_diff_right<min_diff:
                min_diff=min_diff_right
                closest_num=time_data_c2["Start"][mid+1]
                midindex=mid-1
            if time_data_c2["Start"][mid]<target:
                low=mid+1
            elif time_data_c2["Start"][mid]>target:
                high=mid-1
            else:
                  closest_num=time_data_c2["Start"][mid]
                  midindex=mid
                  break
        diff=time_data_c1["Start"][i]-time_data_c2["Start"][midindex]
        pos=diff*0.0006 #diff is in 0.1 µs and th speed of the wave is about 6000 m/s
#Appending the matched data from bothe sensors after calculating the time difference and source location.
        TD=TD.append({'ID_C'+ChannelID1:time_data_c1["ID"][i],
                      'StartTime_C'+ChannelID1:time_data_c1["Start"][i],
                      'MaxAmpl_C'+ChannelID1:time_data_c1["MaxAmp"][i],
                      'ID_C'+ChannelID2:time_data_c2["ID"][midindex],
                      'StartTime_C'+ChannelID2:time_data_c2["Start"][midindex],
                      'MaxAmpl_C'+ChannelID2:time_data_c2["MaxAmp"][midindex],
                      'TimeDifference_0.1µs':diff,
                      'AbsoluteTimeDifference_0.1µs':abs(diff),
                      'Position_m':pos},ignore_index=True)
#         print(i,'1')#indication for running the code
#2nd step: the timestamp of sensor 2 is used to look up for the closest timestamp in sensor 1
    for i in range(len(time_data_c2)):
        min_diff=sys.maxsize
        low=0
        high=len(time_data_c1)-1
        closest_num=None
        if len(time_data_c1)==0 or len(time_data_c2)==0:
            return None
        target=time_data_c2["Start"][i]
        while low<=high:
            mid=(low+high)//2
            if mid+1<len(time_data_c1):
                min_diff_right=abs(time_data_c1["Start"][mid+1]-target)
            if mid>0:
                min_diff_left=abs(time_data_c1["Start"][mid-1]-target)
            if min_diff_left<min_diff:
                min_diff=min_diff_left
                closest_num=time_data_c1["Start"][mid-1]
                midindex=mid-1
            if min_diff_right<min_diff:
                min_diff=min_diff_right
```

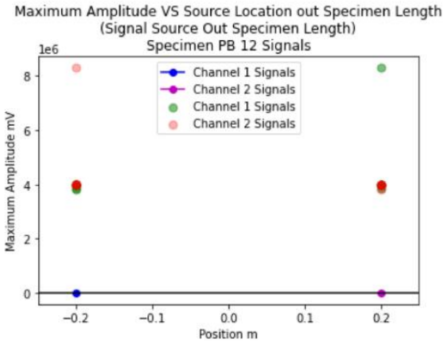
```
                closest_num=time_data_c1["Start"][mid+1]
                midindex=mid-1
            if time_data_c1["Start"][mid]<target:
                low=mid+1
            elif time_data_c1["Start"][mid]>target:
                high=mid-1
            else:
                  closest_num=time_data_c1["Start"][mid]
                  midindex=mid
                  break
        diff=time_data_c1["Start"][midindex]-time_data_c2["Start"][i]
        pos=diff*0.0006 #diff is in 0.1 µs and th speed of the wave is about 6000 m/s
#Appending the matched data from bothe sensors after calculating the time difference and source location.
        TD=TD.append({'ID_C'+ChannelID1:time_data_c1["ID"][midindex],
                      'StartTime_C'+ChannelID1:time_data_c1["Start"][midindex],
                      'MaxAmpl_C'+ChannelID1:time_data_c1["MaxAmp"][midindex],
                      'ID_C'+ChannelID2:time_data_c2["ID"][i],
                      'StartTime_C'+ChannelID2:time_data_c2["Start"][i],
                      'MaxAmpl_C'+ChannelID2:time_data_c2["MaxAmp"][i],
                      'TimeDifference_0.1µs':diff,
                      'AbsoluteTimeDifference_0.1µs':abs(diff),
                      'Position_m':pos},ignore_index=True)
#         print(i,'2')#indication for running the code
#Sorting data and removing duplicated data due to step 1 and 2 for matching the time stamp
    TD.sort_values(by=['StartTime_C'+ChannelID1, 'StartTime_C'+ChannelID2])
    TD.drop_duplicates(subset=['StartTime_C'+ChannelID1, 'StartTime_C'+ChannelID2], keep='last',inplace=True)
    TD.sort_values(by=['StartTime_C'+ChannelID1, 'StartTime_C'+ChannelID2],inplace=True,ignore_index=True)
    # TD.to_csv(OutputFolder+r'\sorted.csv', sep=";")##To audit that the matching process was done correctly
#Print the dataframe in a file in the same outputfolder
    TD.to_csv(OutputFolder+r'\StartTime_MaxAmp_TimeDiff_Position.csv', sep=";")
#Dividing th data into two daftaframes based on the position of the signal
    TDMaxAmpl_in_specimen = TD[(TD['Position_m'] >= -0.2) & (TD['Position_m']<= 0.2)]
    TDMaxAmpl_out_specimen = TD[(TD['Position_m'] < -0.2) | (TD['Position_m']> 0.2)]
    TDMaxAmpl_out_specimen['Position_m'].values[TDMaxAmpl_out_specimen['Position_m'] > 0.2] = 0.2
    TDMaxAmpl_out_specimen['Position_m'].values[TDMaxAmpl_out_specimen['Position_m'] < -0.2] = -0.2
  #plotting 2 plots
      #Signals from the specimen legth
    plt.scatter(TDMaxAmpl_in_specimen['Position_m'],TDMaxAmpl_in_specimen['StartTime_C'+ChannelID1],color='g',marker='o', alpha=0.3,s=50, label='Channel '+ChannelID1+' Signals')
    plt.scatter(TDMaxAmpl_in_specimen['Position_m'],TDMaxAmpl_in_specimen['StartTime_C'+ChannelID2],color='r',marker='o', alpha=0.3,s=100, label='Channel '+ChannelID2+' Signals')
    plt.title('Maximum Amplitude VS Source Location of Same Timestamp'+'\n(Signal Source on Specimen Length)'+'\nSpecimen '+specimen_id+' '+str(len(TDMaxAmpl_in_specimen))+' Signals')
    plt.xlabel('Position m')
    plt.ylabel('Maximum Amplitude mV')
    plt.xlim([-0.25, 0.25])
    plt.plot([-0.2], [0],marker='o', color='b',alpha=1, label='Channel C1 Location')
    plt.plot([0.2], [0],marker='o',color='m', alpha=1, label='Channel C2 Location')
    plt.axhline(y = 0.0, color = 'k', linestyle = '-')
    plt.legend()
    fig1 = plt.gcf()
    plt.show()
    plt.draw()
    fig1.savefig(OutputFolder+r'\Maximum Amplitude VS Source Location of Same Timestamp '+'(Signal Source on Specimen Length) '+specimen_id+'.png')
      #Signals out from the specimen legth
    plt.scatter(TDMaxAmpl_out_specimen['Position_m'],TDMaxAmpl_out_specimen['StartTime_C'+ChannelID1],color='g',marker='o', alpha=0.5,s=50, label='Channel '+ChannelID1+' Signals')
    plt.scatter(TDMaxAmpl_out_specimen['Position_m'],TDMaxAmpl_out_specimen['StartTime_C'+ChannelID2],color='r',marker='o', alpha=0.3,s=50, label='Channel '+ChannelID2+' Signals')
    plt.title('Maximum Amplitude VS Source Location out Specimen Length'+'\n(Signal Source Out Specimen Length)'+'\nSpecimen '+specimen_id+' '+str(len(TDMaxAmpl_out_specimen))+' Signals')
    plt.xlabel('Position m')
    plt.ylabel('Maximum Amplitude mV')
    plt.xlim([-0.25, 0.25])
    plt.plot([-0.2], [0],marker='o', color='b',alpha=1, label='Channel '+ChannelID1+' Signals')
    plt.plot([0.2], [0],marker='o',color='m', alpha=1, label='Channel '+ChannelID2+' Signals')
    plt.axhline(y = 0.0, color = 'k', linestyle = '-')
    plt.legend()
    fig2 = plt.gcf()
    plt.show()
    plt.draw()
```

```
    fig2.savefig(OutputFolder+r'\Maximum Amplitude VS Source Location out Specimen Length '+'(Signal Source Out Specimen Length) '+specimen_id+'.png')
    return TD
```

In [54]:

```
TD=find_closest_timestamp('.\OutPut','PB','1','2')
TD
```





Out[54]:

	ID_C1	StartTime_C1	MaxAmpl_C1	ID_C2	StartTime_C2	MaxAmpl_C2	TimeDifference_0.1μs	AbsoluteTimeDifference_0.1μs	Position_m
0	1.0	3808417.0	4956.05469	1.0	3808419.0	4465.02686	-2.0	2.0	-0.0012
1	1.0	3808417.0	4956.05469	2.0	3995481.0	57.06787	-187064.0	187064.0	-112.2384
2	2.0	3869330.0	709.22852	1.0	3808419.0	4465.02686	60911.0	60911.0	36.5466
3	2.0	3869330.0	709.22852	3.0	3995786.0	34.79004	-126456.0	126456.0	-75.8736
4	3.0	3995486.0	39.36768	2.0	3995481.0	57.06787	5.0	5.0	0.0030
5	4.0	3995813.0	36.62109	3.0	3995786.0	34.79004	27.0	27.0	0.0162
6	5.0	3996022.0	31.43311	2.0	3995481.0	57.06787	541.0	541.0	0.3246
7	5.0	3996022.0	31.43311	4.0	3996077.0	32.34863	-55.0	55.0	-0.0330
8	5.0	3996022.0	31.43311	5.0	3996500.0	20.14160	-478.0	478.0	-0.2868
9	6.0	3996343.0	30.51758	5.0	3996500.0	20.14160	-157.0	157.0	-0.0942

	ID_C1	StartTime_C1	MaxAmpl_C1	ID_C2	StartTime_C2	MaxAmpl_C2	TimeDifference_0.1μs	AbsoluteTimeDifference_0.1μs	Position_m
10	6.0	3996343.0	30.51758	6.0	3996688.0	25.32959	-345.0	345.0	-0.2070
11	7.0	3996597.0	24.10889	4.0	3996077.0	32.34863	520.0	520.0	0.3120
12	8.0	3996649.0	22.88818	4.0	3996077.0	32.34863	572.0	572.0	0.3432
13	9.0	3996780.0	20.44678	5.0	3996500.0	20.14160	280.0	280.0	0.1680
14	9.0	3996780.0	20.44678	8.0	3996946.0	20.14160	-166.0	166.0	-0.0996
15	10.0	3996855.0	25.32959	5.0	3996500.0	20.14160	355.0	355.0	0.2130
16	10.0	3996855.0	25.32959	7.0	3996833.0	25.32959	22.0	22.0	0.0132
17	10.0	3996855.0	25.32959	10.0	3997153.0	29.60205	-298.0	298.0	-0.1788
18	11.0	3996960.0	27.16064	8.0	3996946.0	20.14160	14.0	14.0	0.0084
19	11.0	3996960.0	27.16064	9.0	3997006.0	21.36230	-46.0	46.0	-0.0276
20	11.0	3996960.0	27.16064	11.0	4022563.0	22.58301	-25603.0	25603.0	-15.3618
21	12.0	3997122.0	30.82275	8.0	3996946.0	20.14160	176.0	176.0	0.1056
22	12.0	3997122.0	30.82275	12.0	8302217.0	8587.34131	-4305095.0	4305095.0	-2583.0570
23	13.0	4022603.0	20.44678	11.0	4022563.0	22.58301	40.0	40.0	0.0240
24	14.0	8302214.0	8573.60840	10.0	3997153.0	29.60205	4305061.0	4305061.0	2583.0366

<Figure size 432x288 with 0 Axes>

In [44]:
```
TD
```

Out[44]:

	ID_C1	StartTime_C1	MaxAmpl_C1	ID_C2	StartTime_C2	MaxAmpl_C2	TimeDifference_0.1μs	AbsoluteTimeDifference_0.1μs	Position_m
0	1.0	3808417.0	4956.05469	1.0	3808419.0	4465.02686	-2.0	2.0	-0.0012
1	1.0	3808417.0	4956.05469	2.0	3995481.0	57.06787	-187064.0	187064.0	-112.2384
2	2.0	3869330.0	709.22852	1.0	3808419.0	4465.02686	60911.0	60911.0	36.5466
3	2.0	3869330.0	709.22852	3.0	3995786.0	34.79004	-126456.0	126456.0	-75.8736
4	3.0	3995486.0	39.36768	2.0	3995481.0	57.06787	5.0	5.0	0.0030
5	4.0	3995813.0	36.62109	3.0	3995786.0	34.79004	27.0	27.0	0.0162
6	5.0	3996022.0	31.43311	2.0	3995481.0	57.06787	541.0	541.0	0.3246
7	5.0	3996022.0	31.43311	4.0	3996077.0	32.34863	-55.0	55.0	-0.0330
8	5.0	3996022.0	31.43311	5.0	3996500.0	20.14160	-478.0	478.0	-0.2868
9	6.0	3996343.0	30.51758	5.0	3996500.0	20.14160	-157.0	157.0	-0.0942
10	6.0	3996343.0	30.51758	6.0	3996688.0	25.32959	-345.0	345.0	-0.2070
11	7.0	3996597.0	24.10889	4.0	3996077.0	32.34863	520.0	520.0	0.3120
12	8.0	3996649.0	22.88818	4.0	3996077.0	32.34863	572.0	572.0	0.3432
13	9.0	3996780.0	20.44678	5.0	3996500.0	20.14160	280.0	280.0	0.1680
14	9.0	3996780.0	20.44678	8.0	3996946.0	20.14160	-166.0	166.0	-0.0996
15	10.0	3996855.0	25.32959	5.0	3996500.0	20.14160	355.0	355.0	0.2130
16	10.0	3996855.0	25.32959	7.0	3996833.0	25.32959	22.0	22.0	0.0132

	ID_C1	StartTime_C1	MaxAmpl_C1	ID_C2	StartTime_C2	MaxAmpl_C2	TimeDifference_0.1µs	AbsoluteTimeDifference_0.1µs	Position_m
17	10.0	3996855.0	25.32959	10.0	3997153.0	29.60205	-298.0	298.0	-0.1788
18	11.0	3996960.0	27.16064	8.0	3996946.0	20.14160	14.0	14.0	0.0084
19	11.0	3996960.0	27.16064	9.0	3997006.0	21.36230	-46.0	46.0	-0.0276
20	11.0	3996960.0	27.16064	11.0	4022563.0	22.58301	-25603.0	25603.0	-15.3618
21	12.0	3997122.0	30.82275	8.0	3996946.0	20.14160	176.0	176.0	0.1056
22	12.0	3997122.0	30.82275	12.0	8302217.0	8587.34131	-4305095.0	4305095.0	-2583.0570
23	13.0	4022603.0	20.44678	11.0	4022563.0	22.58301	40.0	40.0	0.0240
24	14.0	8302214.0	8573.60840	10.0	3997153.0	29.60205	4305061.0	4305061.0	2583.0366

```
In [ ]:
```

Correlation with Defined Time Range (NZC, Maximum Amplitude and Energy)

Correlation NZC and Energy with Defined Time Range

In [2]:
```
from pathlib import Path
from typing import Collection

import pandas as pd  # Import pandas for data manipulation
import argparse  # Import argparse for parsing command line parameters
import numpy as np
import matplotlib.pyplot as plt  # Import matplotlib module for plotting
import matplotlib.patches as mpatches  # Needed for putting label on figure
import pickle  # Used for serializing object to disk, we use this to store the figure to retrieve later
# this allows us to interact with the model without calculating values again.
import sys
import os  # Import os for doing file operations, read the files from specified folder
import re  # Import re for regular expression to select correct files and parse channels and Time
# of test and number of hit
from datetime import datetime  # Used to interact and calculate dates and time for output files

from natsort import natsorted, ns  # sort file names naturally

from enum import Enum

os.chdir(r'C:\Users\Yehia\Desktop\1-UiS\4-Thesis DNV GL\Check of Signas in B2\Plot-Signals\5. B1-2 Rep\Stats')
print(os.getcwd())
```

```
C:\Users\Yehia\Desktop\1-UiS\4-Thesis DNV GL\Check of Signas in B2\Plot-Signals\5. B1-2 Rep\Stats
```

Function compare_parameters_c1_c2

The purpose of this function is to identify the same signal logged by the two channels 1&2.

Before running the function some manual preparation for the input files shall be done.

1. In every specimen folder a sub-folder shall be created with the name 'text_files'
2. An edited copies of the text files (1 file for each channel) from the function 'ProcessSignal.py' shall be saved in the created sub-folder
3. The text file shall be edited using 'notepad++' or similar software
```
1. Keep the table header and remove the file header.
2. Replace the spaces with ','
```
   See the example below.

1. Save the file with the same name in '.txt' format

Example:

ID,Start,End,StartRow,EndRow,Duration,SThres,PThres,ZDuration,Mean,Max,Min,MaxAmp,MaxAmpTimestamp,Peak2Peak,NZC,CHMA,RiseDuration,FallDuration,PeakDuration,Type,ID
1,1268319,1268474,1,156,155,20.00,80.00,50,-1.61448,18.61572,-26.55029,26.55029,1268319,45.16601,32,29,0,153,2,A,1

The algorithm works as follow:

1. Reading the .txt files manually modified in different dataframes based on the channel ID
2. Add a column to the dataframe to hold the channel ID
3. Combing the different channels dataframes
4. Creating new dataframe to hold data from the two channels in one row
5. Moving the data from the combined dataframe in step 3 to the new dataframe in step 4
6. Print the created data frame in step 5 to csv file with the name 'MergedData_NoMargin'

Column1	Start Time C1 C2	ID C1	Start C1	End C1	MaxAmp C1	NZC C1	ID C2	Start C2	End C2	MaxAmp C2	NZC C2
0	1268227	1	1268227	1268346	44.86084	24	0	0	0	0	0
1	1268319	0	0	0	0	0	1	1268319	1268474	26.55029	32
2	1268475	0	0	0	0	0	2	1268475	1268536	21.3623	12
3	1268646	2	1268646	1268697	21.05713	9	0	0	0	0	0
4	3194687	3	3194687	3194806	35.09521	30	0	0	0	0	0
5	3197165	4	3197165	3197216	26.55029	8	0	0	0	0	0
6	3219793	5	3219793	3220561	124.2065	178	0	0	0	0	0
7	3219855	0	0	0	0	0	3	3219855	3220623	83.00781	174
8	3237524	6	3237524	3237575	20.44678	9	0	0	0	0	0

1. Sorting the data by column 'Start Time C1 C2' which is the global time column that holds the the time stamp from the combined dataframe in step 3.
2. Checking the absolute time difference between every two consecutive rows. If found to be be less than of equal range in µS and the two rows are for different channels the the data are matched to the same row and one row is changed to 0.
3. Dropping all the rows that are zeros. resulting from the matching in step 8
4. Print the created data frame in step 9 to csv file with the name MergedData_'N'Margin.csv 'N' is the range

Column1	Start Time C1 C2	ID C1	Start C1	End C1	MaxAmp C1	NZC C1	ID C2	Start C2	End C2	MaxAmp C2	NZC C2
20	3379333	13	3379333	3379384	22.27783	12	0	0	0	0	0
21	3379391	14	3379391	3379536	30.82275	32	9	3379420	3379521	0	22
23	3379532	15	3379545	3379670	27.16064	18	10	3379532	3379583	25.63477	15

1. Keeping only the events logged by the two sensors and removing the signals that was not found within range in µS margin.
2. Creating 6 new columns to calculate
   1. Abs. Time Difference: absolute difference between the timestamp of the two sensors
   2. Time Difference T(C1)-T(C2): difference between the timestamp of the two sensors T(C1)-T(C2)
   3. Position (m): Time Difference T(C1)-T(C2)\times 0.006. Speed of Sound Longitudinal= 6000 m/s
   4. Max. Ampl.: Hold the maximum value of the two columns 'MaxAmp C1' and 'MaxAmp C2'
   5. MaxAmpl. Difference: absolute difference between the MaxAmpl of the two sensors
   6. NZC Ratio C1 to C2: (NZC C1)/ (NZC C2)

Column1	Start Time C1 C2	ID C1	Start C1	End C1	MaxAmp C1	NZC C1	ID C2	Start C2	End C2	MaxAmp C2	NZC C2	Abs. Time Difference	Time Difference T(C1)-T(C2)	Position (m)	Max. Ampl.	MaxAmpl. Difference	NZC Ratio C1 to C2
23	3379532	15	3379545	3379670	27.16064	18	10	3379532	3379583	25.63477	15	5	5	0.03	27.16064	1.52587	1.2
67	39510094	37	39510137	39510188	24.41406	12	32	39510094	39510195	25.63477	22	5	5	0.03	25.63477	1.22071	0.545454545
148	76117361	75	76117410	76117527	28.9917	28	75	76117361	76117451	28.07617	22	0	0	0	28.9917	0.91553	1.272727273
165	87518609	83	87518612	87518663	25.02441	10	84	87518609	87518660	25.93994	13	1	-1	-0.006	25.93994	0.91553	0.769230769
210	97327597	107	97327641	97328409	74.15771	211	105	97327597	97328365	85.44922	205	2	2	0.012	85.44922	11.29151	1.029268293
214	99244730	109	99244736	99244791	21.97266	15	107	99244730	99245033	30.82275	99	2	2	0.012	30.82275	8.85009	0.151515152

1. Print the created data frame in steps 11 & 12 to csv file with the name 'MergedData_'N'Margin_EventsBy2Sensors.csv' 'N' is the range
2. Return the final dataframe presented in 'MergedData_'N'Margin_EventsBy2Sensors.csv' to create plots 'N' is the range

In [43]:

```
os.chdir(r'C:\Users\Yehia\Desktop\1-UiS\4-Thesis DNV GL\13-Python working files\ReadData\Bao 2020 output\B2\text_files')
print(os.getcwd())
```

C:\Users\Yehia\Desktop\1-UiS\4-Thesis DNV GL\13-Python working files\ReadData\Bao 2020 output\B2\text_files

In [44]:
```
stats_C1 = pd.read_csv('stats_C1.txt',sep=',')
stats_C2 = pd.read_csv('stats_C2.txt',sep=',')
```

In [45]:
```
def compare_parameters_c1_c2(OutputFolder,specimen_id,range_e,ChannelID1,ChannelID2):

#Reading the output files from Bao's code ProcessSignal.py.
#     file_C1=[f for f in os.listdir(OutputFolder+'/'+specimen_id+'/'+'text_files'+'/') if re.match(r'.*_' + 'C1' + r'.*_'+r'.*.txt', f)]
#     file_C2=[f for f in os.listdir(OutputFolder+'/'+specimen_id+'/'+'text_files'+'/') if re.match(r'.*_' + 'C2' + r'.*_'+r'.*.txt', f)]

    df_C1= stats_C1
    df_C2= stats_C2
# Marking the channel ID by adding new column to hold the channel ID
    df_C1["Channel ID"]=ChannelID1
    df_C2["Channel ID"]=ChannelID2
    STRChannelID1=str(ChannelID1)
    STRChannelID2=str(ChannelID2)
# combining the two dataframes
    StartTime_All=pd.concat([df_C1,df_C2], ignore_index=True)
# creating new dataframe to hold the data from to sensors in one row
    df_merge_margin=pd.DataFrame(columns=['StartTime_C'+STRChannelID1+'_C'+STRChannelID2,'ID_C'+STRChannelID1,'Start_C'+STRChannelID1,'End_C'+STRChannelID1,'MaxAmp_C'+STRChannelID1,'NZC_C'+STRCha
    df_merge_margin_range=pd.DataFrame(columns=['StartTime_C'+STRChannelID1+'_C'+STRChannelID2,'ID_C'+STRChannelID1,'Start_C'+STRChannelID1,'End_C'+STRChannelID1,'MaxAmp_C'+STRChannelID1,'NZC_C'+
    test=pd.DataFrame(columns=['ID'])
# Looping to map the data under the specified sensor using the channel id column
    for i in range(len(StartTime_All)):

        if StartTime_All['Channel ID'][i]==1:
            df_merge_margin=df_merge_margin.append({'StartTime_C'+STRChannelID1+'_C'+STRChannelID2:StartTime_All["Start"][i],
                                                    'ID_C'+STRChannelID1:StartTime_All["ID"][i],
                                                    'Start_C'+STRChannelID1:StartTime_All["Start"][i],
                                                    'End_C'+STRChannelID1:StartTime_All["End"][i],
                                                    'MaxAmp_C'+STRChannelID1:StartTime_All["MaxAmp"][i],
                                                    'NZC_C'+STRChannelID1:StartTime_All["NZC"][i],
                                                    'ID_C'+STRChannelID2:0,
                                                    'Start_C'+STRChannelID2:0,
                                                    'End_C'+STRChannelID2:0,
                                                    'MaxAmp_C'+STRChannelID2:0,
                                                    'NZC_C'+STRChannelID2:0},ignore_index=True)
        elif StartTime_All['Channel ID'][i]==2:
            df_merge_margin=df_merge_margin.append({'StartTime_C'+STRChannelID1+'_C'+STRChannelID2:StartTime_All["Start"][i],
                                                    'ID_C'+STRChannelID1:0,
                                                    'Start_C'+STRChannelID1:0,
                                                    'End_C'+STRChannelID1:0,
                                                    'MaxAmp_C'+STRChannelID1:0,
                                                    'NZC_C'+STRChannelID1:0,
                                                    'ID_C'+STRChannelID2:StartTime_All["ID"][i],
                                                    'Start_C'+STRChannelID2:StartTime_All["Start"][i],
                                                    'End_C'+STRChannelID2:StartTime_All["End"][i],
                                                    'MaxAmp_C'+STRChannelID2:StartTime_All["MaxAmp"][i],
                                                    'NZC_C'+STRChannelID2:StartTime_All["NZC"][i]},ignore_index=True)
#Sorting values
    df_merge_margin.sort_values(by=['StartTime_C'+STRChannelID1+'_C'+STRChannelID2],inplace=True,ignore_index=True)
#Writing the dataframe to csv file
    df_merge_margin.to_csv('MergedData_NoMargin.csv', sep=";")
#Storing the length of the dataframe in variable
    no_margin=len(df_merge_margin)

#Looping to find data within range_e µS and in different channel. If found moving them to one row and putting the row that the data moved from to 0.
    for i in range(len(df_merge_margin)):
        if i==len(df_merge_margin)-1:
            break
        j=i+1

        if (abs(df_merge_margin['StartTime_C'+STRChannelID1+'_C'+STRChannelID2][j]-df_merge_margin['StartTime_C'+STRChannelID1+'_C'+STRChannelID2][i]))>(range_e*10):
```

```
            if (df_merge_margin['ID_C'+STRChannelID1][i]!=0) and (np.all(df_merge_margin['ID_C'+STRChannelID1][i]!=df_merge_margin_range['ID_C'+STRChannelID1])):
                df_merge_margin_range=df_merge_margin_range.append({'StartTime_C'+STRChannelID1+'_C'+STRChannelID2:df_merge_margin["StartTime_C1_C2"][i],
                                                            'ID_C'+STRChannelID1:df_merge_margin["ID_C1"][i],
                                                            'Start_C'+STRChannelID1:df_merge_margin["Start_C1"][i],
                                                            'End_C'+STRChannelID1:df_merge_margin["End_C1"][i],
                                                            'MaxAmp_C'+STRChannelID1:df_merge_margin["MaxAmp_C1"][i],
                                                            'NZC_C'+STRChannelID1:df_merge_margin["NZC_C1"][i],

                                                            'ID_C'+STRChannelID2:df_merge_margin["ID_C2"][i],
                                                            'Start_C'+STRChannelID2:df_merge_margin["Start_C2"][i],
                                                            'End_C'+STRChannelID2:df_merge_margin["End_C2"][i],
                                                            'MaxAmp_C'+STRChannelID2:df_merge_margin["MaxAmp_C2"][i],
                                                            'NZC_C'+STRChannelID2:df_merge_margin["NZC_C2"][i]
                                                             },ignore_index=True)
            elif (df_merge_margin['ID_C'+STRChannelID2][i]!=0) and (np.all(df_merge_margin['ID_C'+STRChannelID2][i]!=df_merge_margin_range['ID_C'+STRChannelID2])):
                df_merge_margin_range=df_merge_margin_range.append({'StartTime_C'+STRChannelID1+'_C'+STRChannelID2:df_merge_margin["StartTime_C1_C2"][i],
                                                            'ID_C'+STRChannelID1:df_merge_margin["ID_C1"][i],
                                                            'Start_C'+STRChannelID1:df_merge_margin["Start_C1"][i],
                                                            'End_C'+STRChannelID1:df_merge_margin["End_C1"][i],
                                                            'MaxAmp_C'+STRChannelID1:df_merge_margin["MaxAmp_C1"][i],
                                                            'NZC_C'+STRChannelID1:df_merge_margin["NZC_C1"][i],

                                                            'ID_C'+STRChannelID2:df_merge_margin["ID_C2"][i],
                                                            'Start_C'+STRChannelID2:df_merge_margin["Start_C2"][i],
                                                            'End_C'+STRChannelID2:df_merge_margin["End_C2"][i],
                                                            'MaxAmp_C'+STRChannelID2:df_merge_margin["MaxAmp_C2"][i],
                                                            'NZC_C'+STRChannelID2:df_merge_margin["NZC_C2"][i]
                                                             },ignore_index=True)
#             else:
#                 df_merge_margin_range=df_merge_margin_range.append({'StartTime_C'+STRChannelID1+'_C'+STRChannelID2:df_merge_margin['StartTime_C'+STRChannelID1+'_C'+STRChannelID2][i],
#                                                             'ID_C'+STRChannelID1:df_merge_margin["ID_C1"][i],
#                                                             'Start_C'+STRChannelID1:df_merge_margin["Start_C1"][i],
#                                                             'End_C'+STRChannelID1:df_merge_margin["End_C1"][i],
#                                                             'MaxAmp_C'+STRChannelID1:df_merge_margin["MaxAmp_C1"][i],
#                                                             'NZC_C'+STRChannelID1:df_merge_margin["NZC_C1"][i],

#                                                             'ID_C'+STRChannelID2:df_merge_margin["ID_C2"][i],
#                                                             'Start_C'+STRChannelID2:df_merge_margin["Start_C2"][i],
#                                                             'End_C'+STRChannelID2:df_merge_margin["End_C2"][i],
#                                                             'MaxAmp_C'+STRChannelID2:df_merge_margin["MaxAmp_C2"][i],
#                                                             'NZC_C'+STRChannelID2:df_merge_margin["NZC_C2"][i]
#                                                              },ignore_index=True)


        elif (abs(df_merge_margin['StartTime_C'+STRChannelID1+'_C'+STRChannelID2][j]-df_merge_margin['StartTime_C'+STRChannelID1+'_C'+STRChannelID2][i]))<=(range_e*10) :
            while abs(df_merge_margin['StartTime_C'+STRChannelID1+'_C'+STRChannelID2][j]-df_merge_margin['StartTime_C'+STRChannelID1+'_C'+STRChannelID2][i])<=range_e*10 or j==len(df_merge_margin)
                if abs(df_merge_margin['StartTime_C'+STRChannelID1+'_C'+STRChannelID2][j]-df_merge_margin['StartTime_C'+STRChannelID1+'_C'+STRChannelID2][i])<=range_e*10 and df_merge_margin['ID_C

                    df_merge_margin_range=df_merge_margin_range.append({'StartTime_C'+STRChannelID1+'_C'+STRChannelID2:0,
                                                            'ID_C'+STRChannelID1:df_merge_margin["ID_C1"][i],
                                                            'Start_C'+STRChannelID1:df_merge_margin["Start_C1"][i],
                                                            'End_C'+STRChannelID1:df_merge_margin["End_C1"][i],
                                                            'MaxAmp_C'+STRChannelID1:df_merge_margin["MaxAmp_C1"][i],
                                                            'NZC_C'+STRChannelID1:df_merge_margin["NZC_C1"][i],

                                                            'ID_C'+STRChannelID2:df_merge_margin["ID_C2"][j],
                                                            'Start_C'+STRChannelID2:df_merge_margin["Start_C2"][j],
                                                            'End_C'+STRChannelID2:df_merge_margin["End_C2"][j],
                                                            'MaxAmp_C'+STRChannelID2:df_merge_margin["MaxAmp_C2"][j],
                                                            'NZC_C'+STRChannelID2:df_merge_margin["NZC_C2"][j]
                                                            },ignore_index=True)

                elif abs(df_merge_margin['StartTime_C'+STRChannelID1+'_C'+STRChannelID2][j]-df_merge_margin['StartTime_C'+STRChannelID1+'_C'+STRChannelID2][i])<=range_e*10 and df_merge_margin['ID
                    df_merge_margin_range=df_merge_margin_range.append({'StartTime_C'+STRChannelID1+'_C'+STRChannelID2:0,
                                                            'ID_C'+STRChannelID1:df_merge_margin["ID_C1"][j],
```

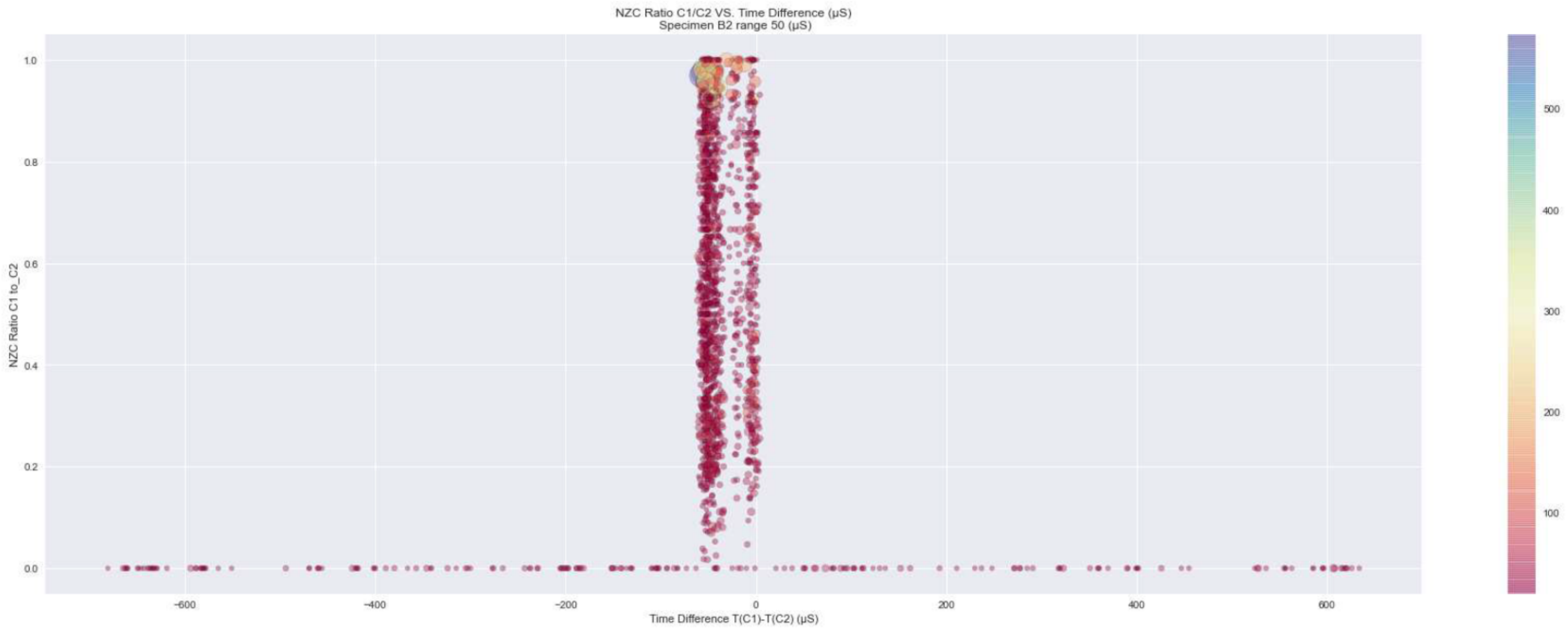
```
                                                                        'Start_C'+STRChannelID1:df_merge_margin["Start_C1"][j],
                                                                        'End_C'+STRChannelID1:df_merge_margin["End_C1"][j],
                                                                        'MaxAmp_C'+STRChannelID1:df_merge_margin["MaxAmp_C1"][j],
                                                                        'NZC_C'+STRChannelID1:df_merge_margin["NZC_C1"][j],

                                                                        'ID_C'+STRChannelID2:df_merge_margin["ID_C2"][i],
                                                                        'Start_C'+STRChannelID2:df_merge_margin["Start_C2"][i],
                                                                        'End_C'+STRChannelID2:df_merge_margin["End_C2"][i],
                                                                        'MaxAmp_C'+STRChannelID2:df_merge_margin["MaxAmp_C2"][i],
                                                                        'NZC_C'+STRChannelID2:df_merge_margin["NZC_C2"][i]
                                                                         },ignore_index=True)
                elif ((df_merge_margin['ID_C'+STRChannelID1][j]!=0 and df_merge_margin['ID_C'+STRChannelID1][i]!=0) and (np.all(df_merge_margin['ID_C'+STRChannelID1][i]!=df_merge_margin_range['ID
                    df_merge_margin_range=df_merge_margin_range.append({'StartTime_C'+STRChannelID1+'_C'+STRChannelID2:df_merge_margin["StartTime_C1_C2"][i],
                                                                        'ID_C'+STRChannelID1:df_merge_margin["ID_C1"][i],
                                                                        'Start_C'+STRChannelID1:df_merge_margin["Start_C1"][i],
                                                                        'End_C'+STRChannelID1:df_merge_margin["End_C1"][i],
                                                                        'MaxAmp_C'+STRChannelID1:df_merge_margin["MaxAmp_C1"][i],
                                                                        'NZC_C'+STRChannelID1:df_merge_margin["NZC_C1"][i],

                                                                        'ID_C'+STRChannelID2:df_merge_margin["ID_C2"][i],
                                                                        'Start_C'+STRChannelID2:df_merge_margin["Start_C2"][i],
                                                                        'End_C'+STRChannelID2:df_merge_margin["End_C2"][i],
                                                                        'MaxAmp_C'+STRChannelID2:df_merge_margin["MaxAmp_C2"][i],
                                                                        'NZC_C'+STRChannelID2:df_merge_margin["NZC_C2"][i]
                                                                         },ignore_index=True)

                elif ((df_merge_margin['ID_C'+STRChannelID2][j]!=0 and df_merge_margin['ID_C'+STRChannelID2][i]!=0) and (np.all(df_merge_margin['ID_C'+STRChannelID2][i]!=df_merge_margin_range['ID

                    df_merge_margin_range=df_merge_margin_range.append({'StartTime_C'+STRChannelID1+'_C'+STRChannelID2:df_merge_margin["StartTime_C1_C2"][i],
                                                                        'ID_C'+STRChannelID1:df_merge_margin["ID_C1"][i],
                                                                        'Start_C'+STRChannelID1:df_merge_margin["Start_C1"][i],
                                                                        'End_C'+STRChannelID1:df_merge_margin["End_C1"][i],
                                                                        'MaxAmp_C'+STRChannelID1:df_merge_margin["MaxAmp_C1"][i],
                                                                        'NZC_C'+STRChannelID1:df_merge_margin["NZC_C1"][i],

                                                                        'ID_C'+STRChannelID2:df_merge_margin["ID_C2"][i],
                                                                        'Start_C'+STRChannelID2:df_merge_margin["Start_C2"][i],
                                                                        'End_C'+STRChannelID2:df_merge_margin["End_C2"][i],
                                                                        'MaxAmp_C'+STRChannelID2:df_merge_margin["MaxAmp_C2"][i],
                                                                        'NZC_C'+STRChannelID2:df_merge_margin["NZC_C2"][i]
                                                                         },ignore_index=True)
                if j==len(df_merge_margin)-1:
                    break
                else:
                    j+=1


    df_merge_margin_range.to_csv('MergedData_'+str(range_e)+'µS.csv', sep=";")
    m_margin=len(df_merge_margin_range)
    df_merge_margin.drop(df_merge_margin[(df_merge_margin['ID_C'+STRChannelID1]==0) & (df_merge_margin['ID_C'+STRChannelID2]==0)].index, inplace = True)
#Writing the dataframe to csv file
    df_merge_margin.to_csv(OutputFolder+'/'+specimen_id+r'/MergedData_'+str(range_e)+'Margin.csv', sep=";")
#Storing the length of the dataframe in variable
    m_margin=len(df_merge_margin)
#Kepping the data that are within range_e µS only and removing the othe data
    df_merge_margin.drop(df_merge_margin[(df_merge_margin['ID_C'+STRChannelID1]==0) | (df_merge_margin['ID_C'+STRChannelID2]==0)].index, inplace = True)
    df_merge_margin.reset_index(drop=True, inplace=True)
#Calaculating new columns
    df_merge_margin_range["Abs. Time Difference"]= abs(df_merge_margin_range["ID_C1"]-df_merge_margin_range["ID_C2"])
    df_merge_margin_range["Time Difference T(C1)-T(C2)"]= df_merge_margin_range["ID_C1"]-df_merge_margin_range["ID_C2"]
    df_merge_margin_range["Position (m)"]=df_merge_margin_range["Time Difference T(C1)-T(C2)"]*0.006
    df_merge_margin_range["Max. Ampl. "]= df_merge_margin_range[['MaxAmp_C'+STRChannelID1,'MaxAmp_C'+STRChannelID2]].max(axis=1)
    df_merge_margin_range["MaxAmpl. Difference"]= abs(df_merge_margin_range["MaxAmp_C1"]-df_merge_margin_range["MaxAmp_C2"])
    NZC_Ratio_C1_to_C2=[]
    for i in range (len(df_merge_margin_range)):
```

```
        NZC_Ratio_C1_to_C2.append(min(df_merge_margin_range["NZC_C1"][i],df_merge_margin_range["NZC_C2"][i])/max(df_merge_margin_range["NZC_C1"][i],df_merge_margin_range["NZC_C2"][i]))
    df_merge_margin_range.reset_index(drop=True, inplace=True)
    df_merge_margin_range["NZC_Ratio_C1_to_C2"]=NZC_Ratio_C1_to_C2
#Writing the dataframe to_Csv file
    df_merge_margin_range.to_csv('MergedData_'+str(range_e)+'µS_Add_Columns.csv', sep=";")
#Storing the length of the dataframe in variable
    m_margin_events=len(df_merge_margin)
#Printing the length of every dataframe
    print(  'Channel 1 signals: ' , len(df_C1) ,'\n',
            'Channel 2 signals: ' , len(df_C2) ,'\n',
            'Length of dataframe holding sorted signals from both channels - No Margin: ' , no_margin ,'\n',
            'Length of dataframe holding sorted signals from both channels after applying '+str(range_e)+' µS margin: ' ,m_margin ,'\n',
            'Length of dataframe holding sorted signals from both channels keeping only the events with '+str(range_e)+' µS margin: ' , m_margin_events)

    plt.style.use('seaborn')
    plt.figure(figsize=(30,10))
    plt.scatter(df_merge_margin_range['Time Difference T(C1)-T(C2)'],df_merge_margin_range['NZC_Ratio_C1_to_C2'],c=df_merge_margin_range['Max. Ampl. '],s=(df_merge_margin_range['Max. Ampl. ']),cm
#     plt.figure(figsize=(15,30))
    plt.title('NZC Ratio C1/C2 VS. Time Difference (µS)\n Specimen '+specimen_id+' range '+str(range_e)+' (µS)')
    plt.colorbar()
    plt.xlim()
    plt.xlabel('Time Difference T(C1)-T(C2) (µS)')
    plt.ylabel('NZC Ratio C1 to_C'+STRChannelID2)
    plt.show
    plt.savefig(OutputFolder+'/'+specimen_id+'/'+r'/NZC Ratio VS Time Difference µS Specimen'+specimen_id+' range '+str(range_e)+' (µS).png')
    return df_merge_margin
```

In [46]:

```
ge_margin=compare_parameters_c1_c2('./2021 output','B2',50,1,2)
```

```
<ipython-input-45-f83f58b237ce>:191: RuntimeWarning: invalid value encountered in double_scalars
  NZC_Ratio_C1_to_C2.append(min(df_merge_margin_range["NZC_C1"][i],df_merge_margin_range["NZC_C2"][i])/max(df_merge_margin_range["NZC_C1"][i],df_merge_margin_range["NZC_C2"][i]))
Channel 1 signals:  652
 Channel 2 signals:  703
 Length of dataframe holding sorted signals from both channels - No Margin:  1355
 Length of dataframe holding sorted signals from both channels after applying 50 µS margin:  1355
 Length of dataframe holding sorted signals from both channels keeping only the events with 50 µS margin:  0
```


NZC Ratio C1/C2 VS. Time Difference (µS)
Specimen B2 range 50 (µS)
NZC Ratio C1 to_C2
Time Difference T(C1)-T(C2) (µS)
1.0
0.8
0.6
0.4
0.2
0.0
−600
−400
−200
0
200
400
600
500
400
300
200
100

Cross-Correlation Coefficient

In [1]:

```
from pathlib import Path
from typing import Collection

import pandas as pd  # Import pandas for data manipulation
import argparse  # Import argparse for parsing command line parameters
import numpy as np
import matplotlib.pyplot as plt  # Import matplotlib module for plotting
import matplotlib.patches as mpatches  # Needed for putting label on figure
import pickle  # Used for serializing object to disk, we use this to store the figure to retrieve later
# this allows us to interact with the model without calculating values again.
import sys
import os  # Import os for doing file operations, read the files from specified folder
import re  # Import re for regular expression to select correct files and parse channels and Time
# of test and number of hit
from datetime import datetime  # Used to interact and calculate dates and time for output files

from natsort import natsorted, ns  # sort file names naturally

from enum import Enum
from scipy import signal
import matplotlib.pyplot as plt

os.chdir(r'C:/Users/Yehia/Desktop/1-UiS/4-Thesis DNV GL/19-Python_Final/2021/Cross-Correlation')
print(os.getcwd())
```

```
C:\Users\Yehia\Desktop\1-UiS\4-Thesis DNV GL\19-Python_Final\2021\Cross-Correlation
```

In [2]:

```
stats_C1 = pd.read_csv('stats_C1.txt',sep=',')
stats_C2 = pd.read_csv('stats_C2.txt',sep=',')

data_C1 = pd.read_csv('20210604110759_PB_C1_S20_P80'+'.csv',  sep=";")
data_C2 = pd.read_csv('20210603104404_PB_C2_S20_P80'+'.csv',  sep=";")
data_C1.set_index('Time')
data_C2.set_index('Time')
```

Out[2]:

Time	Signal	MaxAmplitude
1057648.0	0.000000	14.95361
1057649.0	-0.014954	14.95361
1057650.0	-0.005798	14.95361
1057651.0	0.003357	14.95361
1057652.0	0.006714	14.95361
...	...	...
8382098.0	0.000000	10.37598
8382099.0	0.000000	10.37598
8382100.0	0.000000	10.37598
8382101.0	0.000000	10.37598
8382121.0	0.000000	10.37598

12551 rows × 2 columns

In [3]:

```
for i in range (len(stats_C1)):
    for j in range (len(stats_C2)):
```

```
    ID_C1=stats_C1['ID'][i]
    ID_C2=stats_C2['ID'][j]

    StartTime_C1 = stats_C1.Start[i]
    EndTime_C1 = stats_C1.End[i]

    StartTime_C2 = stats_C2.Start[j]
    EndTime_C2 = stats_C2.End[j]

    data_C1[StartTime_C1:EndTime_C1]
    data_C2[StartTime_C2:EndTime_C2]

    corr = signal.correlate(data_C1.Signal, data_C2.Signal, mode='same')
    lags = signal.correlation_lags(len( data_C1[StartTime_C1:EndTime_C1]), len(data_C2[StartTime_C2:EndTime_C2]))

    fig, (ax_orig, ax_noise, ax_corr) = plt.subplots(3, 1, figsize=(4.8, 4.8))
    ax_orig.plot(data_C1.Signal[StartTime_C1:EndTime_C1])
    ax_orig.set_title('C1')
    ax_orig.set_xlabel('Sample Number')
    ax_noise.plot(data_C2.Signal[StartTime_C2:EndTime_C2])
    ax_noise.set_title('Signal with noise')
    ax_noise.set_xlabel('Sample Number')
    ax_corr.plot(lags, corr)
    ax_corr.set_title('Cross-correlated signal')
    ax_corr.set_xlabel('Lag')
    ax_orig.margins(0, 0.1)
    ax_noise.margins(0, 0.1)
    ax_corr.margins(0, 0.1)
    fig.tight_layout()
    plt.show()
```

Change the CSV Files From AE Win

File	ChangeCSV.ipynb
Project	AET signal characteristics
Client	N/A
Purpose	Prepare the CSV files from AEwin to be ready for use by ProcessSignal Script
Prerequisite	Works with Python 3.4 and newer.
Created by	2021-06-03 Elkhayat
Expanded and modified by	
Date/version	2021-06-03 / 1.0: First release

In [ ]:

In [1]:

```
from pathlib import Path
from typing import Collection
import sys
import pandas as pd  # Import pandas for data manipulation
import argparse  # Import argparse for parsing command line parameters
import numpy
import matplotlib.pyplot as plt  # Import matplotlib module for plotting
import matplotlib.patches as mpatches  # Needed for putting label on figure
import pickle  # Used for serializing object to disk, we use this to store the figure to retrieve later
# this allows us to interact with the model without calculating values again.
import os  # Import os for doing file operations, read the files from specified folder
import re  # Import re for regular expression to select correct files and parse channels and Time
# of test and number of hit
from datetime import datetime  # Used to interact and calculate dates and time for output files

from natsort import natsorted, ns  # sort file names naturally

from enum import Enum
```

In [2]:

```
os.chdir('/Users/Yehia/Desktop/1-UiS/4-Thesis DNV GL/19-Python_Final/2021/ProcessSignal/Input/1. PB/21042021')#Setting up the directory
```

The raw file from AEwin is in this form

SOURCE FILE NAME: C:\Users\USER\Documents\Yehia\2. PB\PB.DTA	
DATE: Wednesday	April 21 2021
TIME: 13:01:26	
SAMPLE INTERVAL (Seconds): 0.0000010000	
SIGNAL UNITS: volts	
TIME UNITS: Seconds	
DATA TYPE: WAVEFORM	
NUMBER OF DATA POINTS PER WAVEFORM: 2048	
CHANNEL NUMBER: 1	
HIT NUMBER: 1	
TIME OF TEST: 3.8084150	

SOURCE FILE NAME: C:\Users\USER\Documents\Yehia\2. PB\PB.DTA

-0.00244

-0.00183

-0.00122

-0.00061

-0.00153

The 1st two 12 rows shall be removed and the month & year columns shall also be removed. This is done in two steps.

In [3]:
```
files = [f for f in os.listdir() if f.endswith( '.csv' )]
```

In [4]:
```
files
```

Out[4]:
```
['PB_1_1_3808415.csv',
 'PB_1_2_3867524.csv',
 'PB_1_3_3869330.csv',
 'PB_1_4_3995472.csv',
 'PB_1_5_4022553.csv',
 'PB_1_6_8302214.csv',
 'PB_2_1_1057649.csv',
 'PB_2_2_3808418.csv',
 'PB_2_3_3900043.csv',
 'PB_2_4_3995481.csv',
 'PB_2_5_4022525.csv',
 'PB_2_6_8302217.csv',
 'PB_2_7_8380311.csv']
```

Step1

Read the raw files and remove the 12 top rows and print in csv. The printed file has three csv.

In [5]:
```
for fileName in files:
    frame = pd.read_csv(fileName)
    frame.drop(frame.index[0:10], inplace=True)
    frame.to_csv('.\Waveform\PB'+'/'+fileName,header=None)#change the output folder
```

In [6]:
```
files1 = [f for f in os.listdir('.\Waveform\PB') if f.endswith( '.csv' )]
```

In [7]:
```
files1
```

Out[7]:
```
['PB_1_1_3808415.csv',
 'PB_1_2_3867524.csv',
 'PB_1_3_3869330.csv',
 'PB_1_4_3995472.csv',
 'PB_1_5_4022553.csv',
 'PB_1_6_8302214.csv',
 'PB_2_1_1057649.csv',
 'PB_2_2_3808418.csv',
 'PB_2_3_3900043.csv',
 'PB_2_4_3995481.csv',
 'PB_2_5_4022525.csv',
 'PB_2_6_8302217.csv',
 'PB_2_7_8380311.csv']
```

Step 2

Read the output CSV files from step 1 and remove the 2 columns with NAN values and print to another CSV file to overwrite the old one. Considering no index to printed in the CSV file.

In [8]:
```
for fileName1 in files1:
    frame1 = pd.read_csv('.\Waveform\PB'+'/'+fileName1,sep=',')
    frame1.drop(frame1.columns[1:3], axis='columns',inplace=True)
    frame1.to_csv('.\Waveform\PB'+'/'+fileName1,header=None,index=False)#here change the output folder
```

Final Output in the CSV Files Shall be

-0.00244
-0.00183
-0.00122
-0.00061
-0.00153
-0.00092
-0.00061

Plot Signals from Two Channel

File	Plot_Signal.ipynb
Project	AET signal characteristics
Client	N/A
Purpose	Process detected waveforms and export plots of the waveforms
Prerequisite	Works with Python 3.4 and newer.
Created by	2020-07-06 GABO
Expanded and modified by	2021-06-13 ElKhayat
Date/version	2020-07-06 / 1.0: First release
	2020-06-13 / 1.1 Modified to plot two signals:

[2]:
```
from IPython.core.display import display, HTML

import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

import os
from datetime import datetime

%matplotlib inline
import mpld3
mpld3.enable_notebook()
```

[3]:
```
# os.chdir('/Users/Yehia/Desktop/1-UiS/4-Thesis DNV GL/Check of Signas in B2/Plot-Signals/5. B1-2 Rep/Stats')
os.chdir('/Users/Yehia/Desktop/1-UiS/4-Thesis DNV GL/13-Python working files/ReadData/Bao 2020 output/B2/signal_plot')
os.getcwdb()
```

[3]:
```
b'C:\\Users\\Yehia\\Desktop\\1-UiS\\4-Thesis DNV GL\\13-Python working files\\ReadData\\Bao 2020 output\\B2\\signal_plot'
```

[4]:
```
# filters and functions definition
# https://medium.com/analytics-vidhya/how-to-filter-noise-with-a-low-pass-filter-python-885223e5e9b7

def butter_highpass(cutoff, fs, order=5):
    nyq = 0.5 * fs
    normal_cutoff = cutoff / nyq
    b, a = signal.butter(order, normal_cutoff, btype='high', analog=False)
    return b, a

def butter_highpass_filter(data, cutoff, fs, order=5):
    b, a = butter_highpass(cutoff, fs, order=order)
    y = signal.filtfilt(b, a, data)
    return y

def butter_lowpass(cutoff, fs, order=5):
    nyq = 0.5 * fs
    normal_cutoff = cutoff / nyq
    b, a = signal.butter(order, normal_cutoff, btype='low', analog=False)
    return b, a

def butter_lowpass_filter(data, cutoff, fs, order=5):
    b, a = butter_lowpass(cutoff, fs, order=order)
    y = signal.lfilter(b, a, data)
    return y

def compute_fft_ModPh(data):
    N = len(data)
    data_freq = np.fft.fft(data)

    if (np.remainder(N,2) == 0):
        # N is even
        # 2020-05-27 GABO: Replaced use of / operator with // in slices to ensure that result is integer
```

```
        mod = abs(data_freq[0:N//2+1]);
        ph  = np.angle(data_freq[0:N//2+1]);

        mod[0] = mod[0]/N;
        mod[1:N//2] = mod[1:N//2]/(N//2);
        mod[N//2] = mod[N//2]/N;
    else:
        # N is odd
        mod = abs(data_freq[0:(N+1)//2]);
        ph = np.angle(data_freq[0:(N+1)//2]);

        mod[0] = mod[0]/N;
        mod[1:(N+1)//2] = mod[1:(N+1)//2]/(N/2);
    return mod, ph
```

[5]:
```
OutputFolder='Bao 2020 output'
specimen_id='B2'
```

[6]:
```
data_C1 = pd.read_csv('signal_data_corrected_C1'+'.csv',  sep=";", index_col ="Time (s)")
data_C2 = pd.read_csv('signal_data_corrected_C2'+'.csv',  sep=";", index_col ="Time (s)")
```

[7]:
```
print(data_C2)
```

```
              Signal
Time (s)
5.859997    0.000000
5.859998    0.003357
5.859999    0.024109
5.860000    0.025024
5.860001    0.012512
...              ...
240.735911 -0.003662
240.735912 -0.005493
240.735913 -0.000916
240.735914  0.002136
240.735915  0.000000

[251790 rows x 1 columns]
```

[8]:
```
stats_C1 = pd.read_excel('stats_C1.xlsx')
stats_C2 = pd.read_excel('stats_C2.xlsx')
```

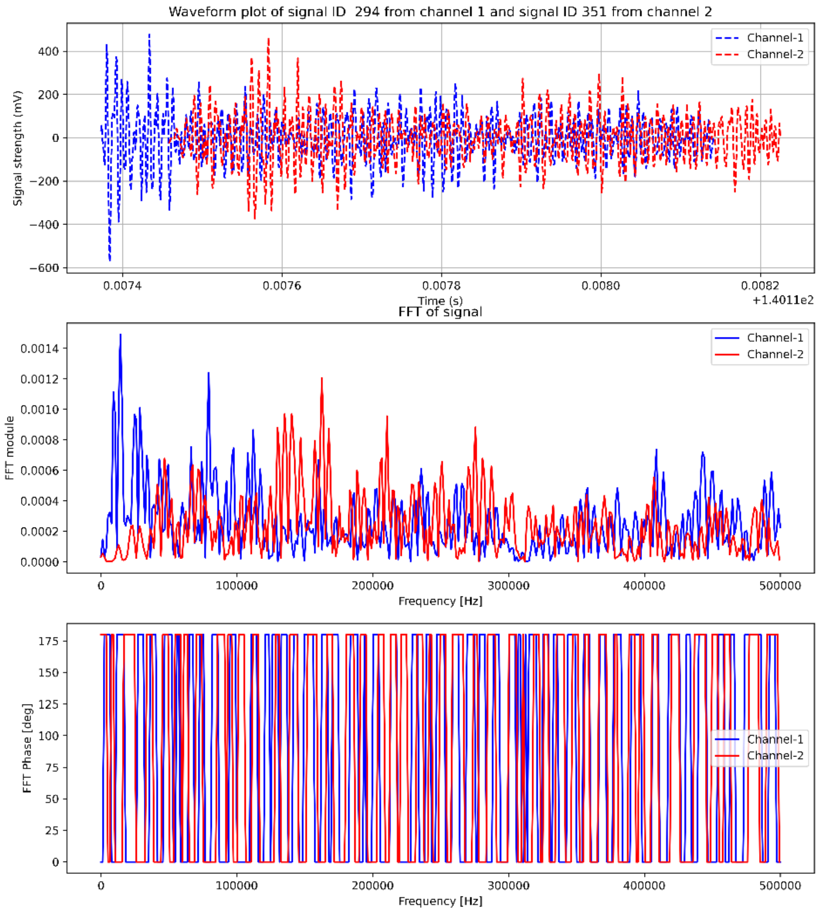
[9]:
```
merged_table=pd.read_csv('MergedData_50µS_Add_Columns'+'.csv',  sep=";")
```

[10]:
```
# Set threshold for plotting wave form (do not plot waveforms with MaxAmp less than this number)
PlotThreshold = 200
counter=0
for i in range(0,len(merged_table)-1,1):
    if merged_table['MaxAmp C1'][i]>PlotThreshold and merged_table['MaxAmp C2'][i]>PlotThreshold:
        ID_C1=merged_table['ID C1'][i]
        ID_C2=merged_table['ID C2'][i]


        i_C1=stats_C1.index[stats_C1['ID'] ==ID_C1]
        i_C2=stats_C2.index[stats_C2['ID'] ==ID_C2]

        StartTime_C1 = float(stats_C1.Start[i_C1]/1e6)
        EndTime_C1 = float(stats_C1.End[i_C1]/1e6)

        StartTime_C2 = float(stats_C2.Start[i_C2]/1e6)
        EndTime_C2 = float(stats_C2.End[i_C2]/1e6)

        fig = plt.figure(figsize=(20,14))
```

```
        ax1 = plt.subplot(311)


        # Plot wave form - converted to mV
        plt.plot(data_C1[StartTime_C1:EndTime_C1]*1e3,'--',color='b',label='Channel-1')
        plt.plot(data_C2[StartTime_C2:EndTime_C2]*1e3,'--',color='r',label='Channel-2')
        plt.xlabel('Time (s)')
        plt.ylabel('Signal strength (mV)')
        plt.legend()

        plt.title("".join(['Waveform plot of signal ID  ',str(stats_C1['ID'][i_C1].values[0]),' from channel 1 and signal ID ',str(stats_C2['ID'][i_C2].values[0]),' from channel 2']))

        # Shrink current axis by 40%
        box1 = ax1.get_position()
        ax1.set_position([box1.x0, box1.y0, box1.width * 0.6, box1.height])

        # Put a legend to the right of the current axis
        for iCol in range(1,20,1):
            plt.text(1.02,1.0-iCol*0.05,stats_C1.columns[iCol],
                     transform=ax1.transAxes,color='blue', bbox={'facecolor': 'white', 'edgecolor':'none', 'alpha': 0.8, 'pad': 3})
            plt.text(1.18,1.0-iCol*0.05," - ",
                     transform=ax1.transAxes,color='blue', bbox={'facecolor': 'white', 'edgecolor':'none', 'alpha': 0.8, 'pad': 3})
            plt.text(1.20,1.0-iCol*0.05,str(stats_C1[stats_C1.columns[iCol]][i_C1].values[0]),
                     transform=ax1.transAxes,color='blue', bbox={'facecolor': 'white', 'edgecolor':'none', 'alpha': 0.8, 'pad': 3})

            plt.text(1.02,0.0-iCol*0.05,stats_C2.columns[iCol],
                     transform=ax1.transAxes,color='red', bbox={'facecolor': 'white', 'edgecolor':'none', 'alpha': 0.8, 'pad': 3})
            plt.text(1.18,0.0-iCol*0.05," - ",
                     transform=ax1.transAxes,color='red', bbox={'facecolor': 'white', 'edgecolor':'none', 'alpha': 0.8, 'pad': 3})
            plt.text(1.20,00.0-iCol*0.05,str(stats_C2[stats_C2.columns[iCol]][i_C2].values[0]),
                     transform=ax1.transAxes,color='red', bbox={'facecolor': 'white', 'edgecolor':'none', 'alpha': 0.8, 'pad': 3})

        # Turn on grids
        ax1.grid(True, which='both',ls="-")

        # Store selected signal data in vector
        fx1_C1 = data_C1[StartTime_C1:EndTime_C1]
        fx1_C2 = data_C2[StartTime_C2:EndTime_C2]
        # compute FFT as module and angle
        mod_C1, ph_C1 = compute_fft_ModPh(fx1_C1)
        mod_C2, ph_C2 = compute_fft_ModPh(fx1_C2)
        if len(mod_C2) != len(ph_C2):
                print(mod_C2)
                print(mod_C2)
        # Number of elements in vector:
        N_C1 = len(fx1_C1)
        N_C2 = len(fx1_C2)
        # Time range of data:
        TimeRange_C1 = EndTime_C1-StartTime_C1
        TimeRange_C2 = EndTime_C2-StartTime_C2
        # Average time between samples:
        TypicalDeltaT_C1 = TimeRange_C1 / N_C1
        TypicalDeltaT_C2 = TimeRange_C1 / N_C2

        fsamp_C1 = 1/TypicalDeltaT_C1
        fsamp_C2 = 1/TypicalDeltaT_C2
        tsamp_C1 = 1/fsamp_C1
        tsamp_C2 = 1/fsamp_C2
#         df_C1 = 1/(N_C1*tsamp_C1)*0.1
#         df_C2 = 1/(N_C1*tsamp_C2)*0.1

        df_C1 = 1/(N_C1*tsamp_C1)
        df_C2 = 1/(N_C1*tsamp_C2)
#         To
        freq_C1 = np.arange(0,(N_C1/2)*df_C1,df_C1)
        freq_C2 = np.arange(0,(N_C2/2)*df_C2,df_C2)
        if len(mod_C1) > len(freq_C1):
```

```
        freq_C2 = np.arange(0,(N_C2/2)*df_C2,df_C2)
        if len(mod_C1) > len(freq_C1):
            freq_C1 = np.arange(0,(N_C1/2+1)*df_C1,df_C1)
        if len(mod_C2) > len(freq_C2):
            freq_C2 = np.arange(0,(N_C2/2+1)*df_C2,df_C2)

        ax2 = plt.subplot(3,1,2)
        plt.plot(freq_C1, mod_C1,color='b',label='Channel-1')
        plt.plot(freq_C2, mod_C2,color='r',label='Channel-2')
        plt.xlabel('Frequency [Hz]')
        plt.ylabel('FFT module')
        plt.legend()
        plt.title('FFT of signal')
        ax3 = plt.subplot(3,1,3)
        plt.plot(freq_C1, ph_C1*180/np.pi,color='b',label='Channel-1')
        plt.plot(freq_C2, ph_C2*180/np.pi,color='r',label='Channel-2')
        plt.xlabel('Frequency [Hz]')
        plt.ylabel('FFT Phase [deg]')
        plt.legend()

        box2 = ax2.get_position()
        ax2.set_position([box2.x0, box2.y0, box2.width * 0.6, box2.height])
        box3 = ax3.get_position()
        ax3.set_position([box3.x0, box3.y0, box3.width * 0.6, box3.height])

        # datetime object containing current date and time
        now = datetime.now()
        dt_string = now.strftime("%Y-%m-%d %H%M")

        PlotFileName = "".join([dt_string,"-Waveform-ID-",str(stats_C1.ID[i_C1].values[0]),"-Channel-1-and-ID-",str(stats_C2.ID[i_C2].values[0]),"-Channel-2"])
        print('Saving plot to: ', "".join([PlotFileName,'.png']))

        # Save plot file to current working directory
        plt.savefig(os.path.join(os.getcwd(),PlotFileName), dpi=600)
        plt.close('all')
        counter+=1
    else:
        print('Signal ',merged_table['ID C1'][i]," channel or Signal ID ",merged_table['ID C2'][i]," has a peak value less than the plot threshold - ",PlotThreshold)
print(counter,' - plots found above the plot threshold - ',PlotThreshold)
```

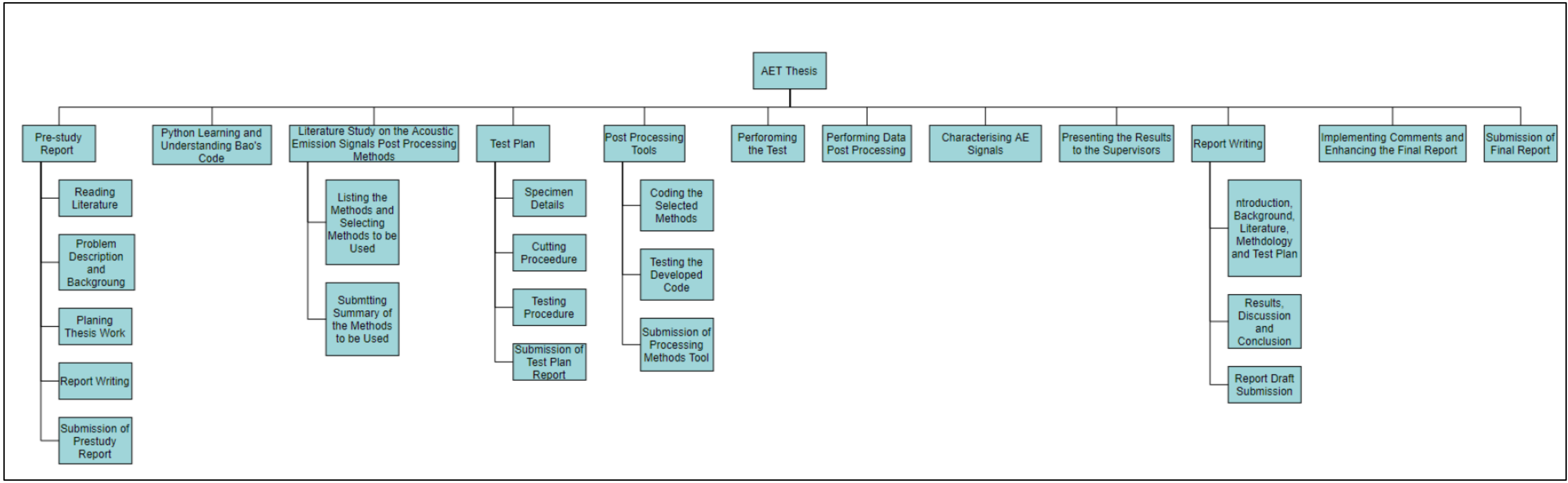
```
Signal  0.0  channel or Signal ID  1.0  has a peak value less than the plot threshold -  200
```



Start	-	140117373
End	-	140118141
StartRow	-	109341
EndRow	-	110109
Duration	-	768
SThres	-	114.62
PThres	-	458.5
ZDuration	-	50
Mean	-	-1.10507
Max	-	476.98975
Min	-	-573.12012
MaxAmp	-	573.12012
MaxAmpTimestamp	-	140117384
Peak2Peak	-	1050.10987
NZC	-	195
CHMA	-	14
RiseDuration	-	11
FallDuration	-	706
PeakDuration	-	51

Start	-	140117458
End	-	140118225
StartRow	-	127822
EndRow	-	128589
Duration	-	767
SThres	-	92.47
PThres	-	369.87
ZDuration	-	50
Mean	-	-1.99817
Max	-	462.34131
Min	-	-372.61963
MaxAmp	-	462.34131
MaxAmpTimestamp	-	140117583
Peak2Peak	-	834.96094
NZC	-	189
CHMA	-	34
RiseDuration	-	104
FallDuration	-	641
PeakDuration	-	22

Appendix F - Work Breakdown Structure and Gantt Chart



AET Thesis Gantt Chart

Period Highlight: | Milestone | Plan Duration | Actual Start | % Complete | % Complete (beyond plan) | Actual (beyond plan)

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
1 Pre-study Report	3	5	3	5	100%
1.1 Reading Literature	3	5	3	5	100%
1.2 Problem Description and Background	3	5	3	5	100%
1.3 Planing Thesis Work	3	5	3	5	100%
1.4 Report Writing	3	5	3	5	100%
1.5 Submission of Prestudy Report	3	5	3	5	100%
2 Python Learning and Understanding Bao's Code	3	21	3	21	100%
3 Literature Study on the Acoustic Emission Signals Post Processing Methods	3	9	3	15	100%
3.1 Listing the Methods and Selecting Methods to be Used	9	2	15	2	100%
3.2 Submitting Summary of the Methods to be Used	10	1	22	1	100%
4 Test Plan	3	5	3	6	100%
4.1 Specimen Details	3	5	3	6	100%
4.2 Cutting Procedure	3	5	3	6	100%
4.3 Testing Procedure	3	5	3	6	100%
4.4 Submission of Test Plan Report	3	5	3	6	100%
5 Post Processing Tools	9	6	9	15	100%
5.1 Coding the Selected Methods	9	6	9	15	100%
5.2 Testing the Developed Code	13	2	13	11	100%
5.3 Submission of Processing Methods Tool	14	1	23	1	100%
6 Performing the Test	15	1	16	2	100%
7 Performing Data Post Processing	16	6	20	4	100%
8 Characterising AE Signals	16	6	22	2	100%
9 Presenting the Results to the Supervisors	21	1	23	1	100%
10 Report Writing	9	10	9	15	100%
10.1 Introduction, Background, Literature, Methodology and Test Plan	9	6	9	10	100%
10.2 Results, Discussion and Conclusion	16	5	21	3	100%
10.3 Report Draft Submission	21	1	21	1	100%
11 Implementing Comments and Enhancing the Final Report	22	2	22	2	100%
12 Submission of Final Report	23	1	23	1	100%