ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

SCHOOL OF ENGINEERING

-Forlì Campus-

SECOND CYCLE MASTER'S DEGREE in
INGEGNERIA AEROSPAZIALE / AEROSPACE ENGINEERING
Class LM-20

GRADUATION THESIS
in Aerospace Structures

Preliminary study for the assessment of discontinuity's size through Machine Learning algorithms

CANDIDATE:

Matteo Sarti

SUPERVISOR:

Professor *Enrico Troiani*

CO-SUPERVISOR:

Professor *Marcias Martinez*

Academic Year 2019/2020

ACKNOWLEDGEMENTS:

First of all, I would like to thank my supervisor, Professor Enrico Troiani, as he put me in contact with Professor Martinez at Clarkson University[1] and, moreover, he always supported me during the choices on how to carry out this study.

To Professor Marcias Martinez goes all my gratitude for helping, guiding and advising me during this journey, and for letting me use his machine throughout the simulation's period. Even if it was not possible to meet physically, the work we did together has been a source of great personal growth for me.

I would like also to thank Dr. Francesco Falcetelli, PhD student at the University of Bologna. Since he was one of the previous three students who went to Clarkson University[1] to carry out his Master Thesis in Aerospace Engineering, I had some very profitable conversations with him and I first got to know the environment I would have eventually found. Moreover, since his current PhD is inherent to the Structural Health Monitoring field, we were able to constructively compare some different choices before putting them into practice within this study. He is a really nice and helpful person, and for this reason he has all my gratitude.

I would now like to thank, from the bottom of my heart, all my family: my parents Stefano and Barbara, my two dear grandmothers Luisa and Mirella, my aunt Patrizia, and the many other members of this great family. They always gave me their support, advising and encouraging me in every choice not only during these university years but since I was born.

Another great thanks goes to all my friends: the Squad, the companions of skiing adventures, family friends, bandmates, fellow tennis players, fellow students, and many others. I will not list you all herein, but you know who you are and that you are all really important to me. The genuine people I met throughout my life gave me something special, a stimulus for personal growth, and it is right that they are here remembered and properly thanked.

Last but not least, I think I have to thank myself too. I know, it may seem a little bit selfish, but I had to make many sacrifices to achieve these results in a faculty that is certainly very demanding, but is even more rewarding when you eventually get to the finish line and accomplish your goals on time.

ABSTRACT:

During the last two decades there has been a huge breakthrough in the Structural Health Monitoring field, especially in the study of Acoustic Emissions (AE), to get qualitative and quantitative damage-related information.

This thesis attempts to focus on the possibility of obtaining an automatic estimate of small discontinuity's length in an aluminium plate, by analysing some impinging signals when they interfere with the defect itself. The novel aspect about this analysis is that it was conducted through "trained" classification and regression algorithms that have been able, up to some extent, to automatically classify and predict the desired responses. This means that Artificial Intelligence, in particular Machine Learning techniques, were employed and played an important role within either the identification and the predictive part of this study.

Due to the SARS-CoV-2 global pandemic, and the consequent closure of the US embassies, it was not possible to obtain the Visa and go to Clarkson University[1] to perform the experimental campaign there. Therefore, in order to collect the raw signals for the subsequent analysis, a comparison between Abaqus CAE^{TM} and OnScale^{TM} software was firstly enforced, and eventually the latter was chosen to perform the whole set of numerical simulations exploiting a pitch-catch configuration.

---

[1] Clarkson University in Potsdam, 13699, NY, USA, is the university in which the co-supervisor of this thesis works and where the author was supposed to go. This thesis was indeed co-supervised by Professor Marcias Martinez, head of the HolSIP Laboratory at that university.

ABBREVIATIONS - ACRONYMS:

AE – **A**coustic **E**mission

AI – **A**rtificial **I**ntelligence

ANN – **A**rtificial **N**eural **N**etwork(s)

CBM – **C**ondition-**B**ased **M**aintenance

CFL – **C**ourant **-** **F**riedrichs **-** **L**ewy

CNN – **C**onvolutional **N**eural **N**etwork(s)

CTOD – **C**rack **T**ip **O**pening **D**isplacement

FE – **F**inite **E**lement

GCON – **G**eneral **CON**nectivity (unstructured mesh)

GLW – **G**uided **L**amb **W**aves

GUI – **G**raphical **U**ser **I**nterface

NDE – **N**on **D**estructive **E**valuations

NDT – **N**on **D**estructive **T**esting

PCA – **P**rincipal **C**omponent **A**nalysis

PLB – **P**encil **L**ead **B**reak

PoD – **P**robability **o**f **D**etection

PWAS – **P**iezoelectric **W**afer **A**ctive **S**ensor

PZT – **P**iezoelectric lead **Z**irconate **T**itanate

RMSE – **R**oot **M**ean **S**quare **E**rror

SHM – **S**tructural **H**ealth **M**onitoring

STFT – **S**hort **T**ime **F**ourier **T**ransform

ToA – **T**ime **o**f **A**rrival

WT – **W**avelet **T**ransform

LIST OF FIGURES:

Figure 1.1: PoD curves for some NDE techniques 4
Figure 2.1: SHM flowchart 8
Figure 2.2: Structure lifetime vs. quality plot, with and without SHM . . . 9
Figure 2.3: Passive (a) vs. Active pitch-catch (b) and Active pulse-echo (c) techniques . 10
Figure 2.4: Amplitude and envelope of a modulated signal 12
Figure 2.5: P, S and Rayleigh waves 13
Figure 2.6: Symmetric and Anti-symmetric modes 16
Figure 2.7: Dispersion curves of a 2 mm Al plate 17
Figure 2.8: Scheme of a PZT transducer 19
Figure 2.9: Example of an AE signal recorded by a PZT sensor 20
Figure 3.1: Cosine bell function 25
Figure 3.2: Some features of a hit 27
Figure 3.3: Three representation techniques of the same signal 30
Figure 3.4: Wavelet window across the time-frequency plane 34
Figure 3.5: Database creation sequence of the Shazam algorithm:
(a) Spectrogram, (b) Constellation map, (c) Hash generation, (d) Hash details . 37
Figure 3.6: Map of hash pairs for a 15 seconds sample 38
Figure 3.7: General working scheme of a pattern recognition algorithm . . . 38
Figure 3.8: Result of a clustering process 40
Figure 3.9: Machine Learning algorithms 42
Figure 3.10: Machine Learning workflow 45
Figure 4.1: Comparison of window functions 47
Figure 4.2: 5 cycle 600 kHz Hanning windowed tone burst. 48
Figure 4.3: Dispersion curves provided by the Vallen Systeme software
for the 1.6mm A7075-T651 plate 53
Figure 4.4: Oscillations at the end of the actuated 600 kHz Hanning window signal . 54
Figure 4.5: SRM subdivided area around the test piece 58
Figure 4.6: Tie Constraint between plate and actuator 58
Figure 4.7: 200 kHz S_0 comparison between Abaqus CAE and OnScale . . . 59
Figure 4.8: Example of a standard mesh with the use of keypoints 61
Figure 4.9: 2D example of the three mesh categories 61

Figure 4.10: Example of a model with keypoints along x and y axis . . . 62
Figure 4.11: A_0 abnormal damping as the sim goes on 63
Figure 4.12: Example of a 3D refined structured mesh 64
Figure 4.13: Hybrid mesh and glued surfaces of the model 65
Figure 4.14: A_0 normal propagation (x, y and z velocity graphs) 66
Figure 4.15: Plate geometries 67
Figure 4.16: Assigned loads 68
Figure 4.17: Required values to model the damping behaviour 70
Figure 5.1: Sensors’ positions on the 3D model 73
Figure 5.2: 2_5 and 10_90 at sensor 1:
time domain (A), frequency domain (B) and spectrogram (C) . . 74-75
Figure 5.3: 2_5 and 10_90 at sensor 2:
time domain (A), frequency domain (B) and spectrogram (C) . . 76-77
Figure 5.4: 2_5 and 10_90 at sensor 3:
time domain (A), frequency domain (B) and spectrogram (C) . . 77-78
Figure 5.5: Elements of an amplitude vector. 80
Figure 5.6: Peaks extracted from an amplitude vector 81
Figure 5.7: Elements extracted from an envelope vector 82
Figure 5.8: Peaks extracted from an envelope vector 82
Figure 5.9: Starting window for a new learning session 84
Figure 5.10: Best Confusion Matrix for sensor 1 87
Figure 5.11: Best Confusion Matrix for sensor 2 88
Figure 5.12: Best Confusion Matrix for sensor 3 88
Figure 5.13: Best Confusion Matrix for the merged signals 89
Figure 5.14: Best predicted responses for sensor 1 92
Figure 5.15: Best predicted responses for sensor 2 92
Figure 5.16: Best predicted responses for sensor 3 93
Figure 5.17: Best predicted responses for the merged signals 93
Figure 6.1: An interesting workflow for future AI studies 96

LIST OF TABLES:

Table 3.1: Sensors coordinates and distance from the discontinuity . . . 22
Table 3.2: Full factorial design process 23
Table 4.1: PZT 5A Navy II material properties 51
Table 5.1: Correlation index between 2_5 and 10_90 74
Table 5.2: Classification’s results 85-86
Table 5.3: Regression’s results 90-91

TABLE OF CONTENTS:

1. About the topic 1
   1.1. Damage characterization 2
   1.2. Scientific question 4
   1.3. Thesis structure 5

2. Introduction 7
   2.1. Structural Health Monitoring 7
   2.2. Elastic waves 11
      2.2.1. Lamb waves 13
   2.3. Piezoelectricity and few general features about PZT transducers . . 19

3. Overview 21
   3.1. Methodology 21
   3.2. Simulation outlines: the choice of an active method 23
   3.3. Techniques of signal analysis 27
      3.3.1. Hilbert Transform 28
      3.3.2. Time-frequency representation of a signal 29
         3.3.2.1. Short Time Fourier Transform 31
         3.3.2.2. Gabor Transform 32
         3.3.2.3. Wigner-Ville Transform 32
         3.3.2.4. Wavelet Transform 33
   3.4. The Shazam algorithm 36
   3.5. Artificial Intelligence and Machine Learning techniques . . . 39

4. Numerical simulations 46
   4.1. Hanning windowed tone bursts 46
   4.2. Actuation and sensing of a signal through PZT transducers . . . 48
      4.2.1. Characteristics for their selection 49
      4.2.2. Actuation 52

    - 4.2.3. Detection 55
  - 4.3. Abaqus CAE^{TM} vs OnScale^{TM} 56
    - 4.3.1. Mesh construction and evaluations 60
  - 4.4. Model description 66
    - 4.4.1. Geometry, material, loads and boundary conditions . . . 67
    - 4.4.2. Mesh size and time steps 69
    - 4.4.3. Signal attenuation 70

5. Signal analysis and implementation of the algorithms 72
  - 5.1. Signal pre-processing 72
  - 5.2. Examination of the signals and interesting features 73
  - 5.3. Signal processing 79
  - 5.4. Machine Learning algorithms 83
    - 5.4.1. Classification's results 85
    - 5.4.2. Regression's results 89

6. Final considerations 95
  - 6.1. Recap and future works 95
  - 6.2. Conclusions 97

Bibliography – Sitography 99

Appendix A 105

CHAPTER 1

1. ABOUT THE TOPIC

An acoustic emission (AE) signal is triggered by many phenomena occurring to the material such as, for example, crack initiation and growth. When a crack is present, the stress level at the crack tip is extremely high and therefore acoustic emissions are primarily emitted from the tip of a growing crack.

In general, it is possible to state that they originate from stresses and strains and some of the stronger acoustic emissions take place when a material is loaded either near / at / beyond its yield stress, therefore undergoing plastic deformation. Contemporarily, at microscopic level, the atomic planes slip and release energy in the form of elastic transient waves which then propagate through the material.

The more severe is the triggering event, the higher is the amount of energy released, and thus the amplitude of the generated waveform. This amplitude is also proportional to the crack propagation's velocity: indeed, fast growing cracks would produce larger AE signals with respect to slow ones.

Once this signals are detected and converted into their digital counterpart, it is possible to exploit various AE testing techniques: a further analysis of these signals with specialized equipment may then lead to valuable qualitative and quantitative information regarding the crack features.

In the following section, the main issues related to damage characterization (in particular sizing) are stated. This will help defining how Machine Learning algorithms could be helpful in order to identify different outcoming signals associated to an input wave interfering with a discontinuity.

Experiments will also be required in a future development of this preliminary work to validate these results. This is mainly due to the simplifications that are introduced and explained throughout this thesis, of which the most relevant is for sure the presence of a signal interfering with a non-physical already present discontinuity of predetermined geometry, instead of proper modelled acoustic emissions associated with realistic crack propagation phenomena.

1.1. Damage characterization

Most of the existing methods using guided waves for damage characterization only focus on identifying its nature and position, but not its size [1]–[8]. The reason is that most of them exploit imaging reconstruction algorithms that do not provide accurate quantitative measures. Indeed, up to the very recent years, the damage growth has been mostly assessed exploiting either these kind of algorithms or damage indexes properly defined and processed to obtain a graphical reference of the growing crack. Recently, He et al. [9] attempted to understand whether two signal's features (amplitude and phase change) may be exploited as crack length indicators; moreover they also questioned whether finite element (FE) simulations could replace experiments in gathering raw data in order to simplify the research activity and make it cheaper. The aim of this thesis is somehow related to these topics since the goal is to collect raw data by numerical simulations and evaluate if their features can be automatically recognized by Machine Learning algorithms, and related to the discontinuity's size.

Su et al. proved that the active sensing, together with a diagnostic imaging algorithm based on linear and non-linear features of the acoustic waves, is effective in providing a quantitative estimation of the damage, ranging from microscopic flaws to macroscopic cracks [10].

Zhao et al. showed that strong attenuations and scattering of acoustic waves are present when dealing with characterization of complex and large geometries [11]. Instead, in this thesis it was considered a 1.6 mm thick plate with a rectangular geometry available in the HolSIP Laboratory at Clarkson University. This is due to the dispersion curves[1] being already experimentally calculated for this thickness and material, and for possible future validation's experimental activities.

In the same article, the crack growth is evaluated using a circular array of piezoelectric transducers[2], however these techniques require a very detailed calibration and unfortunately need for a large array of transducers. Indeed, the methods utilizing piezoelectric transducers (called "active") usually require a large number of sensible points and, sometimes, a baseline signal to be compared with the signal recorded on the damaged structure in order to have an accurate defect estimation. A novel example of a possible array configuration and signal processing technique to estimate the location and size of the damage is reported in [12], where the plate's structure is divided in many cells.

Despite these drawbacks, scattering can be effectively exploited to characterize the damage's

---

[1] Refer to figure 4.3 [13] to see them.

[2] Refer to section 2.3 and especially to 4.2 for a deeper explanation of these devices.

size: as an example, Eremin et al. considered in [14] to use the scattering resonance frequencies of the acoustic wave to non-destructively estimate composite delamination, since these frequencies proved to be related to the crack's variations of size and depth along its path.

Pavelko stated in [15] that the opening and closing movements associated to the crack motion change the acoustic impedance of the material, hence they affect the propagation of the acoustic waves through it. Moreover, the fracture surface's roughness and its local plastic deformation constitute other critical factors and obstacles for the propagation of elastic waves.

The crack itself also causes a partial reflection, hence a decrement of the wave energy, that could be used to assess the damage's size. Therefore, it is also important to evaluate the position of the sensors on the structure in order to receive a distinguishable signal.

Giurgiutiu and Poddar stated in [16] that nowadays very few passive methods[3] exist to quantitatively identify the crack's size, and this technique was actually the primary choice for this work if it would have been possible to access the laboratories to perform experimental collection of raw data. Since this was not the case due to the pandemic, in this thesis the simulations were performed exploiting an active method[4] in a pitch-catch scenario. The motivations regarding the choice of this particular method are better explained in section 3.2.

The reason why passive methods are rare may lie behind the fact that with active sensing the generated signals can have the amplitude the user prefers, even a very high one, which enables the sensors to detect the acoustic waves even for high scattering phenomena and in a noisy environment. On the other hand, when dealing with an acoustic emission generated directly from the growing crack, the amplitude might be very small, thus making the caption extremely complicated (pre-amplification is necessary).

Researchers like Bhuiyan et al. developed in [17] a novel technique of using piezoelectric transducers to detect low amplitude AE signals triggered by a fatigue crack, exploiting different sensors configurations in a passive way. Another example of a passive technique is the one exploiting local vibrational modes along the crack surfaces that may arise directly from the primarily tip-emitted acoustic waves interfering with the defect itself. Analysing this phenomena could help estimating the crack length as reported in [18].

---

[3] It is a method where there are no actuators involved, therefore the signal is recorded by sensors which are simply "listening" to the surrounding environment.

[4] It is a method where, on the contrary to the previous one, there are both actuators and sensors involved. The signal is defined by the user, sent to the actuator which excites the material, and eventually recorded by the sensors after a travelling period. To have a deeper insight about this distinction refer to section 2.1.

1.2. Scientific question

As already mentioned, acoustic waves propagating in a solid medium have been widely studied, especially throughout the last two decades, and many results have been found in terms of detection and localization of damages. However, these techniques are still under great research activity and are not employed in the industry because they are challenging to exploit when dealing with the whole structure of an aircraft.

The reason why these methods are not certified, and consequently used on large scale, is primarily related to the still high uncertainty and poor replicability of the results. This leads to the lack of a statistical model describing the effectiveness of the measuring technique, the so-called Probability of Detection (PoD) curves. Indeed, these graphs report the ratio between correctly detected defect and the whole defects present (i.e. a probability) on the ordinate, vs a defect parameter (e.g. length) on the abscissa. These curves are mandatory and needs to report values above a predefined threshold in order to be accepted by the certification authorities therefore, until they are properly defined, this kind of structural monitoring will remain a research topic.

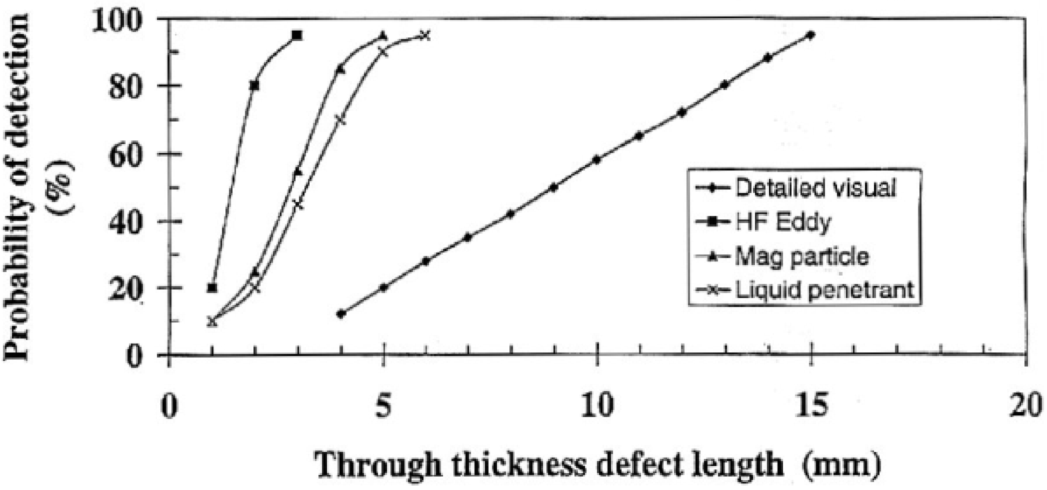


Figure 1.1: PoD curves for some NDE techniques [19]

Examining the chart reported in figure 1.1, it is possible to state that new Structural Health Monitoring (SHM) inspection techniques should be able to detect defects of 2 mm in length with a PoD of around 70-80% in order to be comparable with the other non-destructive evaluation (NDE) methods like penetrant liquids, radiography, ultrasonic testing and, mostly, eddy currents [19].

The latter are currently successfully employed to study cracks at early fracture stage, however the main requirements of a direct access to the structure and a point-by-point measure are great limitations for these technologies. An alternative is therefore represented by SHM, especially through embedded Lamb wave techniques.

As a matter of fact, acoustic signals are unfortunately very susceptible to noise, vibrations and other disturbing phenomena (such as external loads, temperature, humidity, and even the geometry and boundary conditions themselves) that can significantly alter the captured signal. For this reason, thresholds settings, implemented by proper filtering operations during the signal analysis, are extremely important; however, they tend to be set according to a specific case study, thus resulting in a not replicable *hit detection law* [20].

Another important aspect is that the damage, after being detected and localized, needs to be properly quantified and, as it was mentioned in section 1.1, this is still an ongoing matter in the engineering field, where many different techniques are being developed and tested.

This thesis wants indeed to deal with the problem of quantifying a damage exploiting acoustic signals, but it does that through a novel technique firstly used by N. Facciotto in his Master Thesis [21] who was inspired by the well known Shazam^{TM} music recognition software.

The idea constituting the foundations of this preliminary work is that each event is supposed to have its own *fingerprint*, which basically carries some features identifiable within that sole detected signal. These could then be exploited by Machine Learning algorithms able to automatically recognize and build relations between them in order to eventually, and hopefully, understand the related defect's size.

The necessity of an automatic process, which is also quite novel in the damage quantification researches, is due to the increasing amount of data usually collected during these kind of studies that would lead to a complicated and time consuming human-based analysis.

Based on the aforementioned reasons, the scientific questions the author wants to answer are:

- *Are there features carrying quantitative information about the size of a small defect ?*
- *If so, is there at least an algorithm able to effectively identify them ?*

1.3. Thesis structure

This thesis is structured in several chapters and sections that allow the reader to properly understand the ideas behind this study and the developed path. The preliminary literature review has been very useful, especially for better understanding the general scientific field this thesis is developing through, which is the Structural Health Monitoring (SHM), and the fundamental pillars of the wave propagation phenomena.

Chapter 1 provided an outline of the main challenges and questions this study is trying to answer, moreover it focused the attention on the present state of the art regarding the damage characterization techniques.

Chapter 2 presents the literature review regarding Structural Health Monitoring and elastic waves propagation (in particular Lamb waves) since the reader might be interested in knowing the general qualitative and mathematical features of this particular research field. Furthermore, a brief introduction about piezoelectricity and piezoelectric transducers is given.

Chapter 3 gives an overview of the subjects addressed in this study starting from the general methodology, going through the choice of an active pitch-catch configuration for the numerical simulations, main signal analysis' methods, principal features of the Shazam^{TM} algorithm and, eventually, the different Machine Learning techniques.

Chapter 4 comprises all the aspects related to the numerical simulation part of the thesis. In it, the input signal exploited, the selection of the PZT transducers, the excitation and sensing methods, the modelling software and parameters needed in order to get the proper output signal to be subsequently processed are reported. Moreover, a comprehensive comparison between two different commercial software (Abaqus CAE^{TM} and OnScale^{TM}) is presented.

In Chapter 5, the recorded time domain output signals are firstly pre-processed, examined and eventually processed to prepare them for the subsequent step. The signal analysis through the Machine Learning algorithms is also presented, together with the best results.

Chapter 6 deals with the final considerations and conclusions. The future works necessary to develop what has been presented in this preliminary study are also addressed herein.

Eventually, all the articles, books and websites consulted in the development of this thesis are listed in a detailed bibliography.

CHAPTER 2

2. INTRODUCTION

During the last few years, the leading companies of the aerospace commercial industry have been launching some new efficient aircraft models that significantly reduce the fuel consumption. This is achieved not only through the employment of new high by-pass ratio engines but also through lighter and stiffer structures allowing, at the same time, weight reduction and increment of passenger comfort. Two examples regarding this latter aspect could be the lower cabin altitude and larger windows present on the Boeing 787 Dreamliner.

The demand of lighter structures goes hand in hand with the necessity of keeping a very high safety level, and it is here that Structural Health Monitoring (SHM) may play an important role.

2.1. Structural Health Monitoring

A precise definition of SHM is given in [22] where it is stated: "*The process of implementing damage identification strategies for aerospace, civil and mechanical engineering infrastructure is referred to as Structural Health Monitoring*".

The techniques gathered inside this field are sometimes quite different, but they all aim at real time monitoring the health state of a structure continuously throughout its entire working period. This is in contrast with the current non-destructive methods, where external (i.e. non-embedded) devices are employed.

Some drawbacks linked to the employment of NDT&E techniques are [19]:

- The necessity of a regular scheduled-based maintenance;
- Massive disassembling operations in order to provide access even to the smallest components that have to be inspected;
- Either the increased safety factors during the design phase or the need of reinforcements to ensure the safety of the operations in between successive inspections, where the damage may grow unpredictably fast or defects may go undetected.

  This weight increment restricts the performances, reduces the payload and raises the fuel burnt, consequently leading to an augmentation of the operational costs and of the environmental pollution.

The SHM steps can be divided as following [23]:

- *Sensing*, defined as the collection of the most adequate data from the monitored structure. For example, mechanical (e.g. surface displacement, stress, strain) or acoustic (e.g. impedance) properties can be recorded exploiting different types of commercially available sensors. An important aspect to take into account regards also the location and the number of sensors in order to control the most critical points of the structure, which have to be identified a priori by the designer.
- *Diagnosis*, defined as the extraction of damage related features from the acquired data exploiting properly developed algorithms to understand, if a damage exists, where, what and how severe it is.
- *Prognosis*, defined as the assessment of the residual operational life of the structure through different methodologies, including the rapidly growing Artificial Intelligence (AI).

  The more the prognosis is accurate the more it is possible to perform the so-called *condition-based maintenance* (CBM) rather than the scheduled-based maintenance (linked to the widely used damage tolerance design approach) decided by the manufacturer and mandatory after a certain amount of flight hours [24].

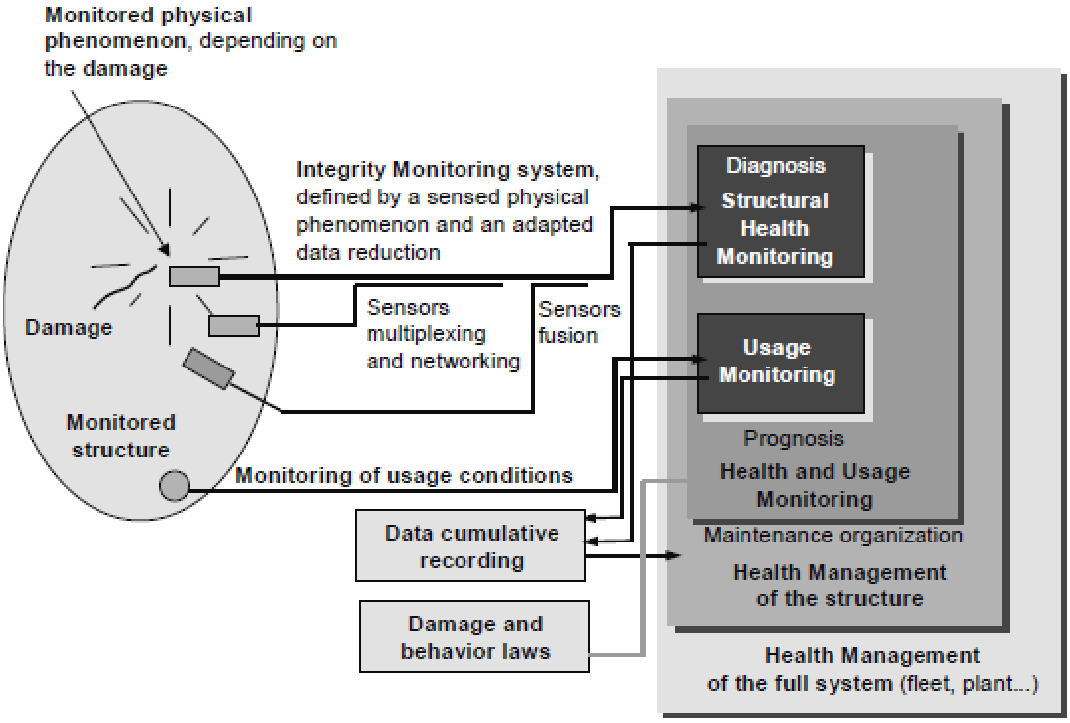


Figure 2.1: SHM flowchart [23]

Referring to figure 2.1, the flowchart at the base of SHM is here better explained. First of all, the sensors receive a signal travelling on the monitored structure and send it to the structural health monitoring subsystem for its processing and consequent diagnosis. Meanwhile, the usage monitoring subsystem is performing its evaluation about the environmental conditions and the structure's mechanical behaviour. Then, linking these information together through the merging of the two subsystems, the prognosis is provided. To perform this final task, the damage's features must be known to the overall health management system.

What is described here is the so-called *intelligent structure*, that is actually the final result at which all the researchers in this field aim to come putting together all the discoveries made throughout the years.

Thanks to CBM, which is an integration to the current damage tolerance approach, there will be a time reduction of the maintenance layover since the airplane would be taken away from operations only when needed and not on mandatory scheduled dates. As a consequence also the maintenance cost would drop while the safety of the aircraft would be ensured by systems constantly checking the health state of the structure throughout its life span.

The real time checking of a structure through embedded transducers could lead to a better exploitation of the materials at disposal, together with an increment of the structural safety: indeed, if the SHM system detects a damage prior to the scheduled inspection, the airplane can be properly fixed prior to its scheduled date. This cutting-edge technology may also replace some of the rigid inspection protocols enforced to reduce the probability of human error.

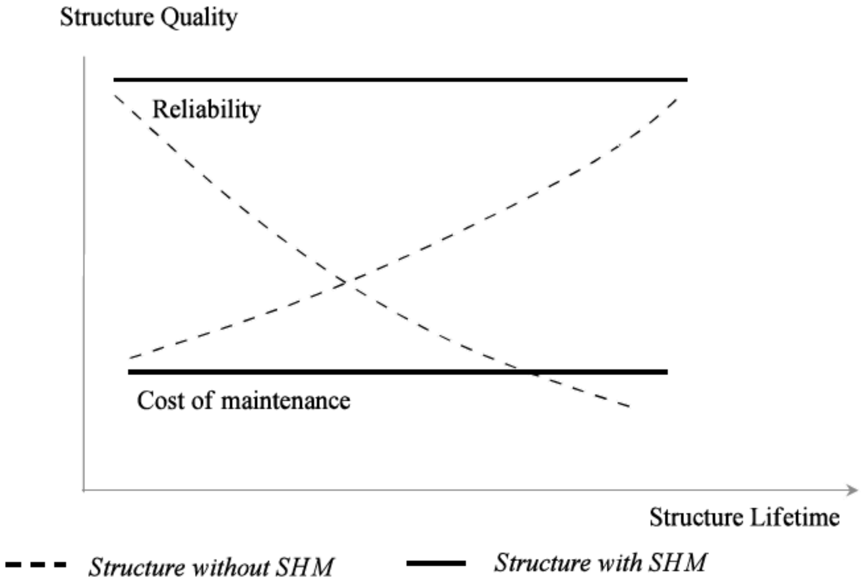


Figure 2.2: Structure lifetime vs. quality plot, with and without SHM [23]

There exist a lot of SHM methods, but it is possible to group them in two broad classes highlighted in figure 2.3 [23]:

- *Passive* techniques are defined as the methods exploiting only sensors to “listen” to any change in the structure. These signals are recorded and analysed through specific algorithms in order to evaluate the structural health state.
- *Active* techniques are instead defined as the methods exploiting actuators sending a signal onto the structure (excitation phase), and properly placed sensors recording the dynamic response, that is later processed. Baseline data representing the damage-free scenario are sometimes needed in order to compare the damaged and the pristine system’s dynamic responses.

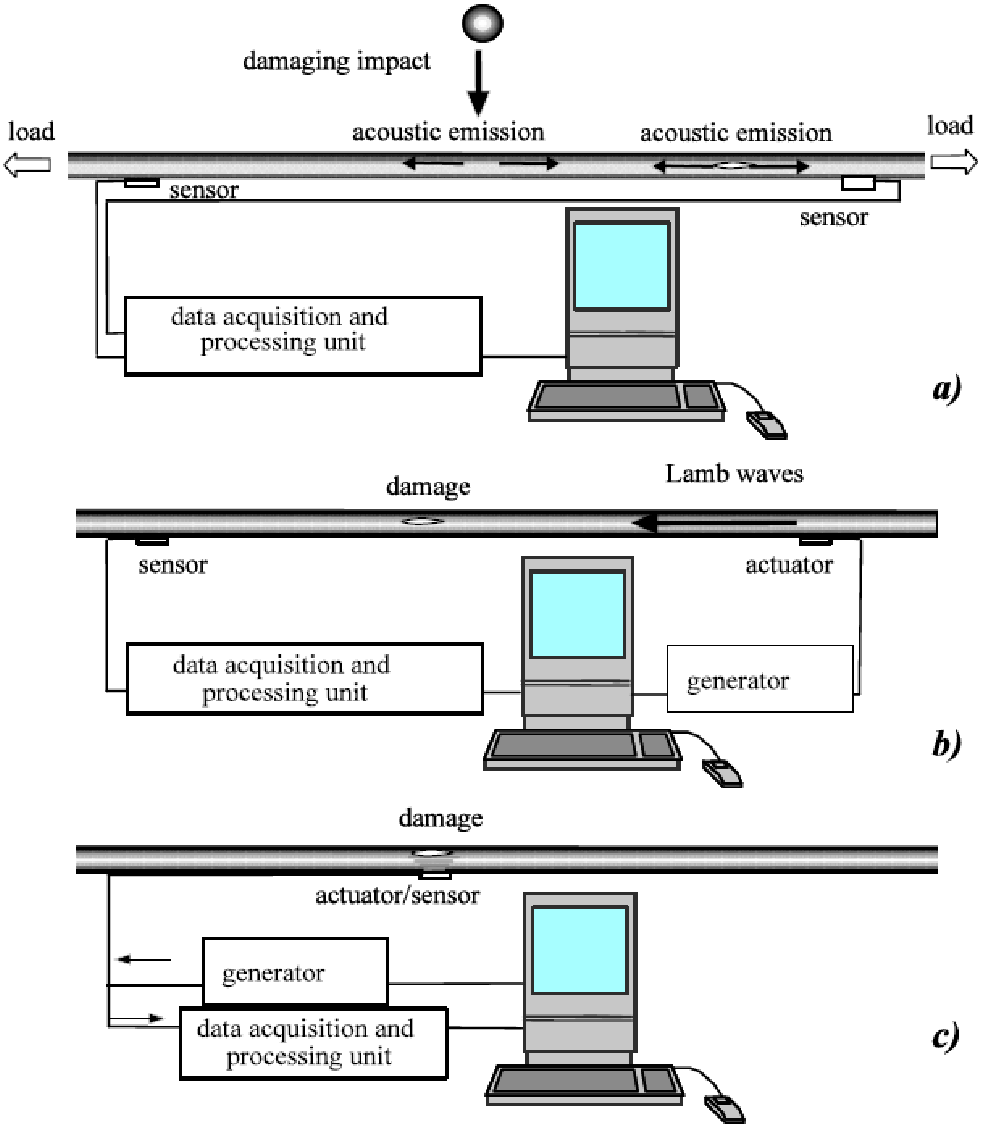


Figure 2.3: Passive (a) vs. Active pitch-catch (b) and Active pulse-echo (c) techniques [23]

Modern SHM systems consist of a network of transducers incorporating active and passive techniques, i.e. Guided Lamb Waves (GLW) and Acoustic Emission (AE).

AE is one of the most recent and promising technique and is properly a passive method for the monitoring of structures because, as already mentioned at the beginning of Chapter 1, the emission source is located at the damage site and its nature can be very different: crack opening, plastic deformation, friction, impacts, delamination, matrix cracking and fiber rupture are all AE sources generating peculiar broadband signals that have to be properly monitored. Fiber optics, piezoelectric and strain sensors are some examples of sensors exploited in the passive techniques.

The expression “acoustic emission” is, however, also used to indicate active methods in which an input signal defined by the user is triggered into the material leading to the propagation of elastic waves. If the material is bounded by two surfaces (e.g. in a plate) it is more correct to name the emission as “guided waves”. GLW are commonly studied as an active technique for damage detection. Pencil Lead Break (PLB), Ultrasonic probes, Piezoelectric and Laser actuators are some examples of triggering techniques.

2.2. Elastic waves

The acoustic waves propagating in a solid material are related to the particles vibrational motion around their equilibrium position caused by elastic displacement and restoration internal forces [58].

According to the direction of the particle’s speed vector, these waves can be grouped in two broad classes moving with different phase velocities:

- *Longitudinal* waves are characterized by the same direction between the particle’s speed vector and the wave’s phase speed vector. They are also called “pressure waves” (P).
- *Transverse* (Shear) waves are characterized by a particle’s speed vector and a wave’s phase speed vector which are orthogonal. These can be encountered only in solid materials and are usually generated by extracting energy from the longitudinal waves.

To better understand the previous definitions it is important to state the difference between *group* and *phase* velocity of a wave [59].

The wave’s phase velocity C_p is the one representing the rate at which the phase of a specific wave changes and it can be though as the propagation velocity of a crest but it does not necessarily coincide with the wave's proper propagation velocity. The phase of a particular wave is expressed by a value ranging from 0 to 2π and it is taken at a certain time during the wave evolution. The phase velocity is defined by relations (2.1):

$$C_p = \frac{\lambda}{T} = \lambda * f = \frac{\omega}{k} \quad \text{[m/sec]} \tag{2.1}$$

where λ is the wavelength [m], T is the wave period [sec], f is the wave frequency [Hz=1/sec], $\omega=2\pi f=2\pi/T$ is the angular frequency [rad/sec], and $k=2\pi/\lambda$ is the wavenumber [rad/m].

Waves usually encountered in the practical applications considered in this thesis are multifrequency, therefore a second velocity must be defined to have a complete understanding of the topic. This is the group velocity C_g, it represents the motion of the envelope, hence the energy propagation through the material, and it is defined by equation (2.2):

$$C_g = \frac{\partial\omega}{\partial k} \tag{2.2}$$

This velocity is exploited to assess the Time of Arrival (ToA) of a wave packet.
The envelope of a signal, highlighted in figure 2.4, is the curve tracing the signal amplitude peaks (it is the shape of a modulated wave). Therefore, the amplitude is an instantaneous value while the envelope is the curve formed by the peaks succession.

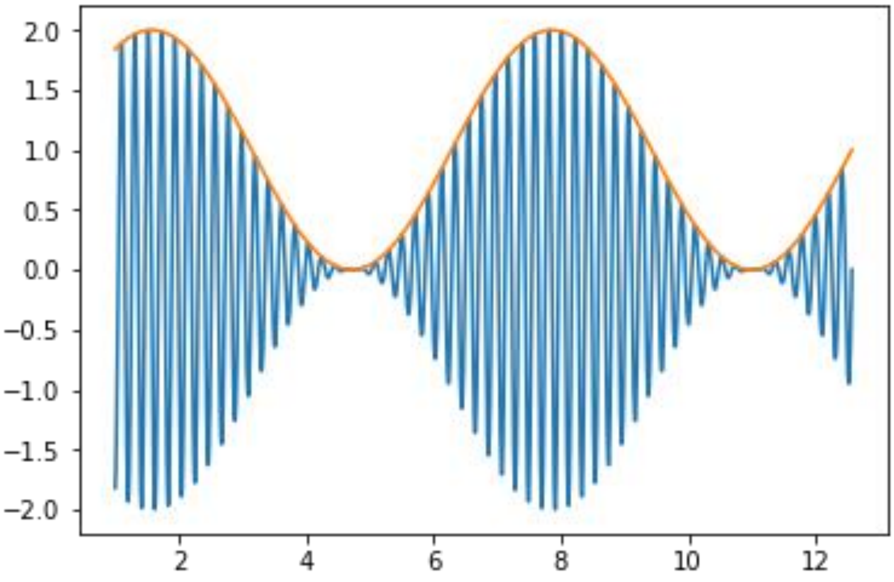


Figure 2.4: Amplitude and envelope of a modulated signal

2.2.1. Lamb waves

When a solid material is unbounded, the acoustic waves propagating through it are referred to as *bulk* waves (since they propagate inside the bulk of the material) and they possess some universal properties. However, these waves can be encountered only for very thick components (where this dimension is much higher, theoretically infinite, with respect to the wavelength) and this is not the case for aerospace structures which are usually made of thin parts assembled together.

When an elastic material is bounded, the problem becomes more complex since new elastic surface waves arise simply due to the particular geometry of the component: it is the case of *Rayleigh* (if the material is bounded by just one surface) and *Lamb* waves (when the bounding surfaces are two, like in a plate) [60].

Rayleigh waves usually penetrate to a depth of one wavelength and their motion combines both longitudinal and transverse movements creating an elliptic particle path, as visible in the following figure:

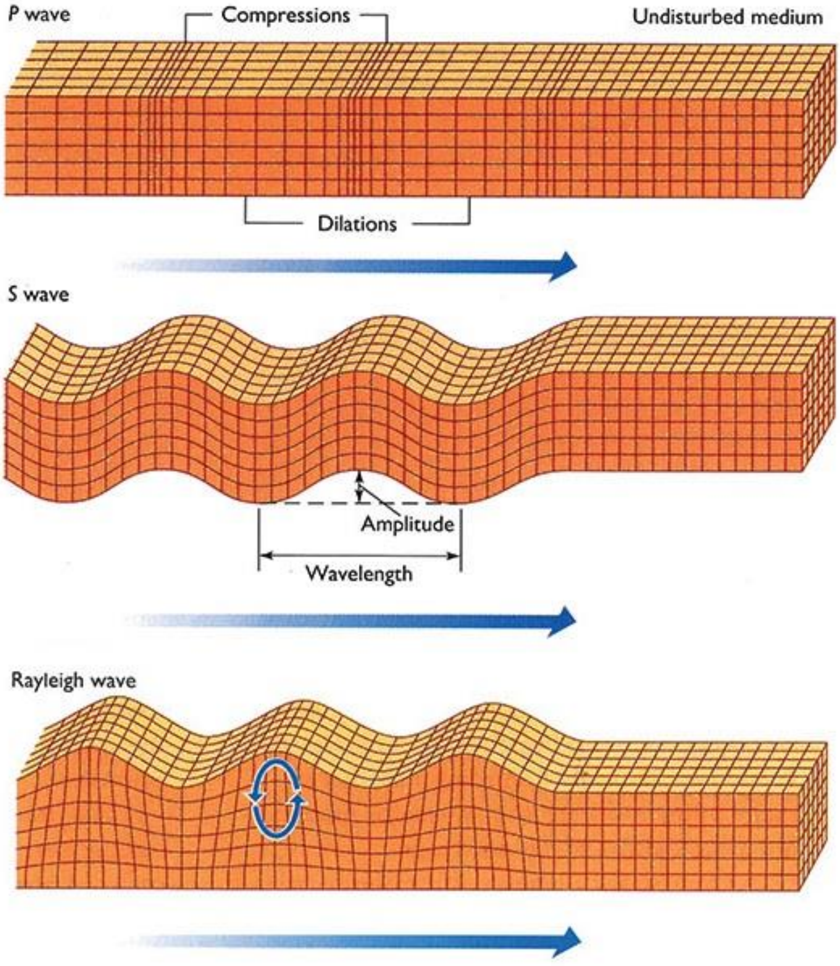


Figure 2.5: P, S and Rayleigh waves

As for Rayleigh waves, the Lamb waves particles' motion is once again elliptical. The latter are the most common *guided* waves, meaning that the wave motion is guided by the two surfaces bounding the plate, and their name derives from the mathematician Horace Lamb who discovered and characterized them in an homogeneous, isotropic and infinite plate[1] during the 1917 [25].

The derivation and explanation of their entire set of equations is out of this thesis' scope, but interested readers can find useful information in reference [26].

What is important to mention is that there exists two *families* of wave modes (symmetrical and anti-symmetrical, about the plate's mid-thickness plane) in infinite plates whereas, in unbounded medium, just the longitudinal and the transverse modes are present.

These latter surface waves are useful in NDE applications since they are very sensitive to many kind of defects and follow the surface's geometry during their motion. For this reason, they can be used to inspect areas that other types of wave may not reach. Moreover, these waves possess a high susceptibility to any kind of interferences and this can be positive (e.g. in case of damage detection) or negative (e.g. in case of near boundaries and presence of noise) according to the purpose they are exploited for, which is usually the SHM.

Lamb waves have also the peculiar feature to travel long distances even in materials having high attenuation (like the ones having lots of inhomogeneities, e.g. CFRP) due to their high propagation's velocity, this leads to the possibility of fast checking a large component without the need of a large amount of equipment [27].

However, this latter property brings to the problem regarding the examination of small components, where the reflections from the edges may become a huge source of signal distortion. The time interval for an acoustic wave to reflect around the test piece (thus leading to another hit detection by the sensor) ranges a lot depending on the damping properties of the material itself. This means that the temporal windows considered for the signal analysis are of crucial importance to get reasonable results and have to be carefully evaluated.

---

[1] In defining the problem, H. Lamb assumed a 2D scenario regarding the particles' motion: they were restricted to the normal to the plane (z) direction and to the wave propagation (x) direction. Motion just along the y-direction was assigned to another set of modes, called shear-horizontal (SH).

This means that GLW and SH modes are complementary and, moreover, they are the only ones possessing the property of travelling with straight and infinite wave fronts along a plate.

As reported in section 2.1, researchers are trying to develop techniques based on the physical principles of Lamb waves in order to go from the qualitative indication of the occurrence of a damage to the final prediction of the structural residual life, passing through a quantitative assessment of the location and severity of the damage itself.

The merits associated to the inspection of a component through Guided Lamb Waves (GLW) can be resumed in [28]:

- The ability to inspect large structures during service (avoiding disassembling and retaining even the coating and the insulation) provided that the detection area is sufficiently small;
- The possibility of checking the entire cross section for a fairly long length;
- The lack of complicated, expensive and movable devices during the inspection;
- The good sensitivity to many different defects;
- The low energy consumption and great cost-effectiveness.

As mentioned, two main Lamb wave modes exist and each of these modes may be composed by multiple harmonics mainly due to the excitation frequency and plate thickness.

Their *dispersive* properties[2] can be obtained by finding the solutions to the propagating Lamb waves equations. By applying the proper boundary conditions fulfilling the plate geometry requirements (i.e. zero stresses on the upper and lower surfaces), the following dispersion relations are found for, respectively, the longitudinal symmetric “extensional” and the transverse anti-symmetric “flexural” modes (depicted in figure 2.6).

$$\frac{\tan(qh)}{\tan(ph)} = -\frac{4k^2qp}{(k^2-q^2)^2} \tag{2.3}$$

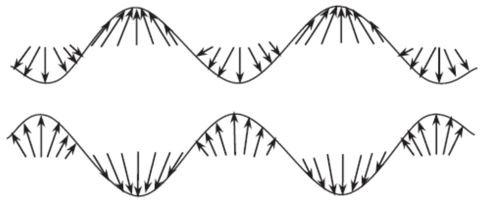
$$\frac{\tan(qh)}{\tan(ph)} = -\frac{(k^2-q^2)^2}{4k^2qp} \tag{2.4}$$

$$\text{where: } p^2 = \frac{\omega^2}{{C_L}^2} - k^2 \quad \text{and} \quad q^2 = \frac{\omega^2}{{C_T}^2} - k^2 \tag{2.5}$$

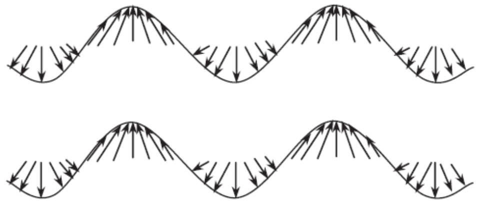
h is half of the plate thickness, k is the wavenumber, ω is the angular frequency, C_L and C_T are respectively the longitudinal and transverse wave’s velocities.

For isotropic materials the longitudinal and transverse wave propagation's velocities C_L and C_T can be expressed directly through the Young modulus E, the Shear modulus G (or the Poisson ratio μ) and the density ρ [61]. This means that each material has its own dispersion relations and curves. The reader can imagine how the problem's complexity rises when anisotropic or orthotropic laminated materials are involved in the analysis.

Many examples of industrial tools for the experimental evaluation and plotting of the dispersion curves are available: the Vallen Systeme^{TM} software exploited in [13] is one of them.



symmetric Lamb mode



anti-symmetric Lamb mode

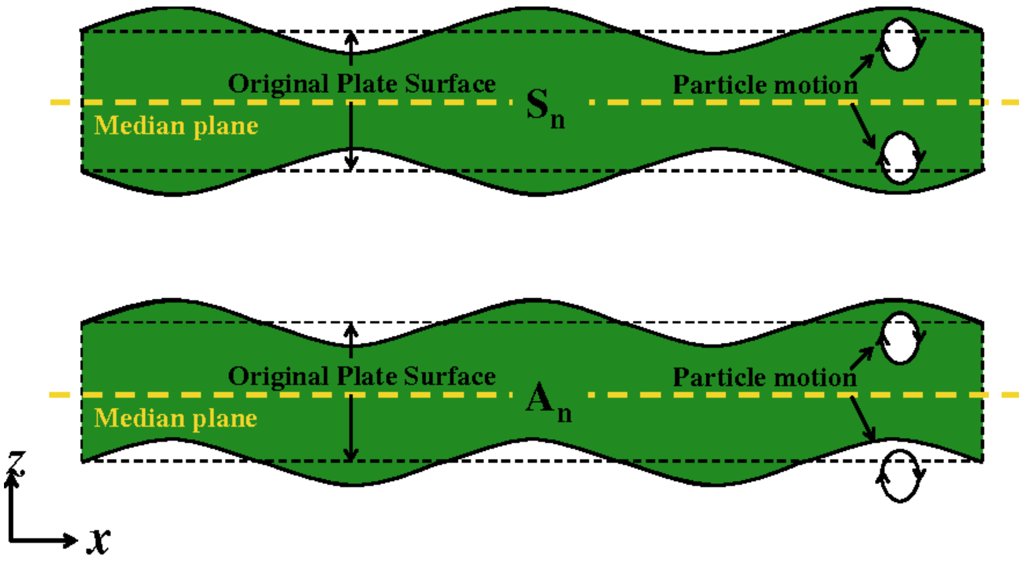


Figure 2.6: Symmetric and Anti-symmetric modes

---

[2] By dispersive property is intended the relation between the wave propagation's velocity and the frequency. A non-dispersive wave has one property independent from the other: this implies that each mode maintains its shape during the motion and the phase velocity coincides with the group velocity.

The Rayleigh-Lamb equations (2.3) and (2.4) relate the propagation velocity with the frequency, hence the Lamb waves are dispersive: this means that, while the modes spread in space and time during their evolution, also the frequency band of the pulse changes. In particular, the driving factor for both the velocities is not simply the frequency but, more properly, the product between frequency and thickness (ωh).

These equations can be solved numerically in an iterative way (e.g. through a bisection or a Newton-Raphson algorithm) to find different solutions, hence different dispersive modes within the same family. For sake of clarity, two plots of the dispersion relations for a 2 mm thick aluminium plate, considering both the phase (a) and the group (b) velocities, are depicted in figure 2.7 [29].

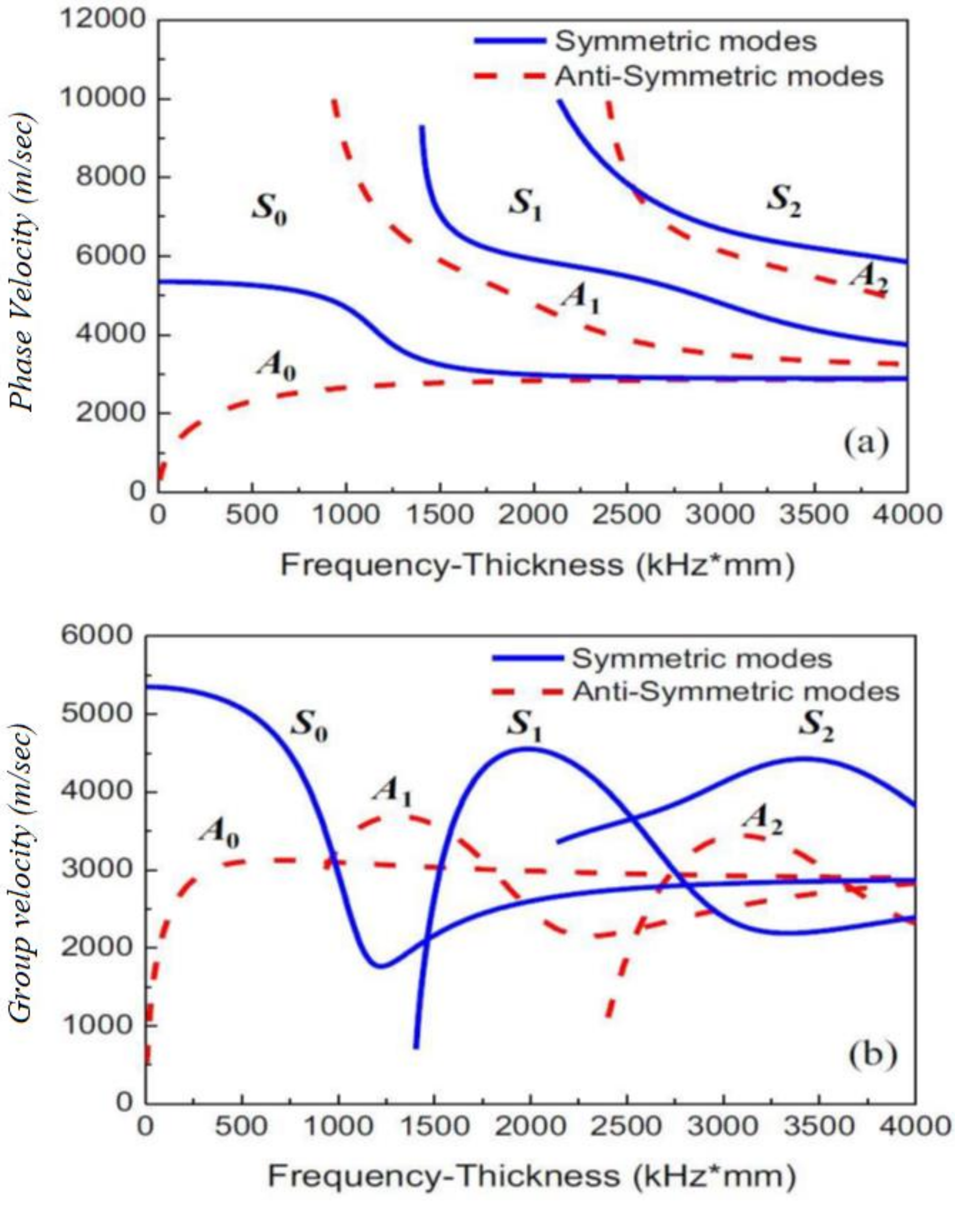


Figure 2.7: Dispersion curves of a 2 mm Al plate

Regarding the dispersive property, it is possible to infer that, below a certain frequency-thickness product, the phase and group velocities of S_0 are almost constant, hence this mode may be considered as very low dispersive within this range.

A closer look to figure 2.7 also reveals that just one symmetric and one anti-symmetric mode exist underneath a specific threshold of the frequency-thickness product. They are named as *fundamental* modes and are denoted respectively by S_0 (for the symmetric mode) and A_0 (for the anti-symmetric mode). Higher order harmonics are labelled with an increasing subscript number and exists only above the aforementioned threshold.

From these graphs it is also evident that usually multiple modes are contemporary superposing and propagating through the material. A peculiarity is that they don't interfere with each other and, therefore, the resulting motion is generated by superposition of the different modes.

To excite a single mode, either the symmetric or the anti-symmetric, it is necessary to respectively input a disturbance along the longitudinal or along the out-of-plane direction of the plate; this is due to the main extensional or transversal nature of the modes. For this reason, the S_0 mode is more visible along the longitudinal displacement, while A_0 is more visible in the transversal displacement.

Another positive features of the fundamental modes is that they both possess different group velocities, therefore, if the sensors' location is far enough from the actuation point, there is enough time for the two modes to separate and be clearly visible and analysable. This characteristic is very helpful in the so-called *"mode selection"* process, in which specific modes are separated from the general signal in order to perform a targeted analysis. This is done since one mode may carry some particular information about the damage due to the unique interactions occurred.

These reasons show why many of the researches try to focus on the lower range of the frequency-thickness parameter: the generated modes are here easier to process, therefore it is possible to push towards more pioneering results otherwise not reachable due to the complex analysis of the chaotic and multiple wave packets contemporary moving through the material.

A key aspect in using GLW for non-destructive evaluations (NDE) purposes is indeed the triggering of a particular subset of modes that propagate well and give fairly clear echoes, instead of generating the whole set of waves with the ultimate result of just complicating the overall situation.

Besides dispersion, of primarily importance is also reflection and attenuation. The latter is a phenomena, better addressed in section 4.4.3, comprising geometrical spreading, material damping and scattering.

2.3. Piezoelectricity and few general features about PZT transducers

The proper definition of piezoelectricity (whose effect was discovered in 1880 by the Curie brothers) encloses the capacity, some solid materials possess, of generating electric charge when a mechanical stresses is enforced on them. The word, derived from the Greek language, means indeed *“electric charge coming from pressure”*. The piezoelectric effect is reversible: this means that due to an applied electric field the material undergoes mechanical strain. However, these effects are possible only if the material possesses a net polarization along a specific direction [62].

Some materials exhibiting this behaviour are crystals (e.g. quartz), organic matter (e.g. collagen in human bones), and some ceramics (e.g. lead zirconate titanate). The ceramic materials have a significantly higher piezoelectric constant with respect to the natural crystals and they can be produced by cheap sintering processes. However, there is a drawback: their piezoelectric effect degrades in time and this loss is also related to the temperature they are exposed to.

Some new crystal materials are coming to the market, one of them is the Lead Magnesium Niobate-Lead Titanate (PMN-PT): it offers an improved sensitivity with respect to the Lead Zirconate Titanate but it possesses a lower maximum operating temperature and a higher manufacturing cost [63].

The most famous piezoelectric material exploited for the construction of acoustic transducer’s elements is the Piezoelectric Lead Zirconate Titanate (PZT). It is a ceramic compound, it appears as an insoluble white solid and it has very good piezoelectric, ferroelectric and pyroelectric properties extremely useful for the construction of electrical transducers[3].

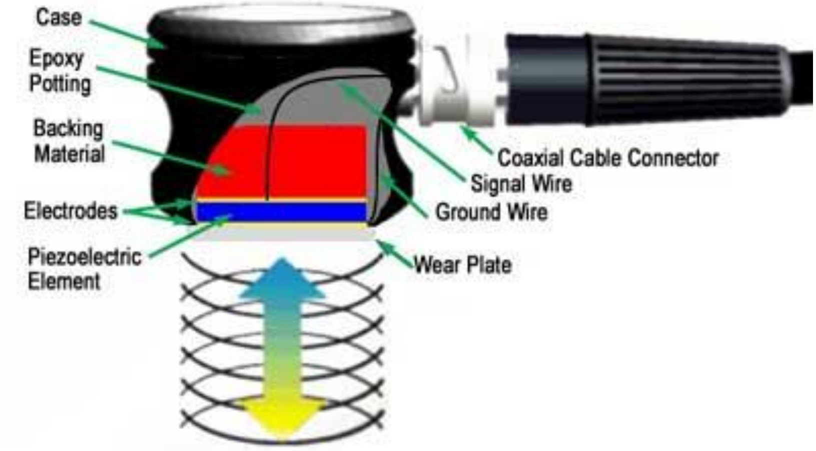


Figure 2.8: Scheme of a PZT transducer

---

[3] The word “transducer” means that this device is capable of transforming a strain into an electrical pulse and vice-versa according to its function as a sensor or as an actuator.

The piezoelectric effect is exploited in several applications, from the quality assurance to the process control measurements. The one of interest for this thesis is the dynamic generation and detection of acoustic waves using small, light and cheap devices [24].

PZT transducers may be surface mounted or embedded in the airframe, thus constituting the sensing “neural network” the structure needs for developing a proper SHM system.

The PZT wafer transducer operates differently from the conventional ultrasonic probes employed nowadays for the airframe’s NDE. The differences are in the excitation mechanism of the Lamb waves, the sensor-structure coupling, the nature of the device itself and, more simply, the size and cost. Despite these differences, the PWAS proved to constitute a good alternative to the ultrasonic probes due to the favourable characteristics mentioned inside reference [30].

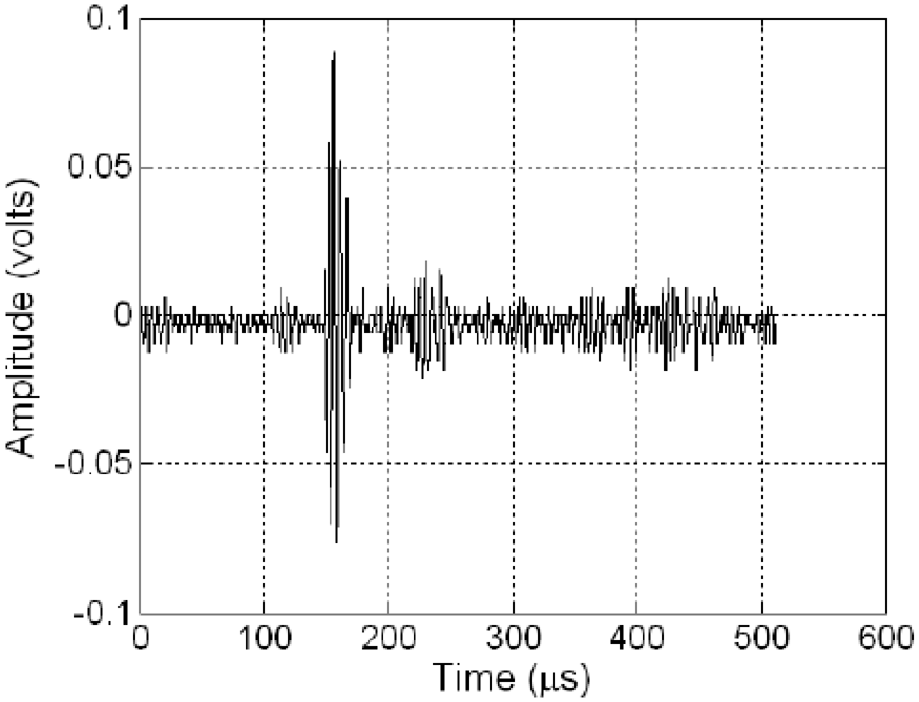


Figure 2.9: Example of an AE signal recorded by a PZT sensor [33]

CHAPTER 3

3. OVERVIEW

In this chapter, an overview of the thesis' main topics is presented. All the general choices and aspects that will be encountered throughout this study are listed and addressed, starting from the general methodology exploited in the development of the whole work until coming to the introduction of Artificial Intelligence and, in particular, of Machine Learning.

3.1. Methodology

This study has originally been thought including an experimental part in order to collect AE signals related to crack propagation however, due to the SARS-CoV-2 global pandemic, it was not possible for the author to go to Clarkson University (NY, USA). For this reason, numerical simulations have been enforced for the data collection process after having analysed two different kind of software. More specific information about the simulations are given in the next section (3.2) and especially in Chapter 4.

The main idea driving this study, as mentioned in section 1.2, is to exploit Machine Learning algorithms to identify, as final aim, the size of a small discontinuity, thus performing the so-called *damage quantification*. The purpose is to create a set of samples for every simulation that, theoretically, the algorithms are able to recognize, and to validate them all with the *"leave one out"* cross-validation process. The choice of this validation procedure lies in the impossibility of creating a large enough dataset just through numerical simulations, due to reasonable time limitations.

The first part of the thesis was developed on a thin pristine aluminium plate where input signals, made of Hanning windows applied to five cycle sinusoidal functions with different central frequencies[1], were numerically generated at position 152.4 x 152.4 x 1.6 mm [2]. The outputs were then employed in order to get confident with wave propagation simulations, verify the suitability of a numerical model in terms of wave's characteristics but, most importantly, to understand which software should be employed to carry out the copious simulation campaign as efficiently as possible. The choice was indeed between the famous Abaqus CAE^{TM} and

---

[1] Refer to section 4.1 for a detailed explanation about their usage.

OnScale™; a comprehensive explanation about this topic is given in section 4.3.

The subsequent and key part of this study was then developed on the same metal plate but with a through-thickness discontinuity placed at the centre. This time, the output signals are expected to vary according to the dimensions of the defect they are interfering with, and actually the ultimate goal of the thesis is to demonstrate that some size-related features inside the signals can be automatically recognized. Three sensors were placed at various positions on the plate in order to collect the raw signals with which to train the Machine Learning algorithms, while having also a ToA difference between them. The sensor's position followed the non-interaction recommendation given by Hamstad et al. in [31]:

$$\frac{D}{s} \geq 7 \tag{3.1}$$

where *D* is the distance from any other similar device or from the actuator/source and *s* symbolizes the actuator/source's size.

	Coordinates from the bottom-left corner of the plate [mm]	Distance from the central discontinuity [mm]
Sensor 1	292.8 x 253.3 x 1.6	152.4
Sensor 2	508 x 152.4 x 1.6	203.2
Sensor 3	381 x 20.4 x 1.6	101.6

Table 3.1: Sensors coordinates[2] and distance from the discontinuity

Because of the theoretically unknown defect's position, the incident GLW usually impinge on it obliquely. Moreover, the lack of external factors contributing to the variation of experimental results compared with a numerical simulation environment, imposed the creation of a fictitious way to give a sort of randomness to the collected results. Eventually, it was decided to consider some orientations' variation, together with the different sizes, in order to accomplish this latter task.

All these considerations led the author to choose the aforementioned sensors' positions and four different angles as varying orientations.

---

[2] In section 4.4.1 the axis' origin is displayed in a figure summing up the overall model's geometry.

The full factorial design process for the simulation campaign was eventually made of twenty runs, resumed in the following table:

Factors	Units	Type	Levels	Minimum	Maximum	Range
Size of the discontinuity	mm	Numeric	5	2	10	+ 2
Orientation with respect to the longitudinal axis	deg	Numeric	4	5	90	+28

Table 3.2: Full factorial design process

It is important that the defect is detected well below becoming visible, in the early stages of its growth. As already reported in section 1.2, the common NDE techniques have a resolution of a couple of millimetres (e.g. eddy currents) or even less, with a corresponding PoD value ranging from 70 to 80 %.

The aim of this novel automated technology is to be comparable with the existing ones, thus seeing whether also small defects are detectable with sufficient accuracy. Indeed, in most of the literature researches involving GLW interacting with damages, only large discontinuities (above 10 mm) are studied.

After the raw signals were collected, they were firstly examined and later processed. Then, many Machine Learning algorithms were trained in order to evaluate if they were able to give satisfactory predictions about the discontinuity's size.

Resuming what has been explained above, the proposed methodology aims to simultaneously identify different damage-related signals linked to the discontinuity's characteristics, and to exploit them to train pre-built algorithms rather than working directly on every single feature extracted from the received signal, as most of the contemporary methods do. The final purpose, together with the futures studies that will come, is indeed the development of a *damage-size classifier / predictor.*

3.2. Simulation outlines: the choice of an active method

There are usually two fundamental steps to be followed in order to generate an acoustic emission through crack growth: firstly, the material's failure accompanied by the formation of new crack surfaces, and secondly, the propagation of the resulting displacement field as a transient wave. As an example, Lysak proposed in [32] an analytical approach involving the fracture mechanics and the wave propagation theories in order to address the acoustic waves generated by the material's breaking. The model has some lacks in the propagation phase, where the complexity of analytically describing the wave guide problem arises. Therefore, numerical FE simulations and actual experiments are usually exploited in the majority of the studies.

Throughout the recent years, researchers have been trying to make use of various numerical methods in order to model AE sources.

To accomplish this task, numerical simulations exploit mainly two different paths: one generating a signal which is more related to the reality but difficult to be properly characterized, and the other where the source signal has some common characteristics with an hypothetical real wave but it is actually an explicitly user defined force-time curve.

In between there could be the method proposed by Lee et al. in [33] where the pre-cracked model is firstly subjected to a static load and then the nodes around the crack tip are released simulating the crack growth through an element and exciting the wave modes. However, this approach may be too simplistic because most of the major phenomena affecting a real crack growth are missing.

Simulations belonging to the aforementioned first path make often use of experimental parameters like loads, material's constitutive equations and failure criteria based on the mechanics in order to define the proper conditions for the defect to grow. A criteria may be, for example, the degradation of the stiffness vector evaluated using as reference the geometrical position of the fracture surfaces [34]. In this paper by Markus G. R. Sause, the simulation is performed in three steps: first, a static load is applied to the structure, then the crack propagation occurs eventually followed by the signal propagation.

Simulations belonging to the second path are based on point-like AE sources where concentrated forces act. The dynamic law of these forces acting as AE source is usually found by fitting experimental data or by assumptions deriving from fracture mechanics. Various step functions describing the displacement of the crack surfaces exist, and a famous one is reported in [31] by Hamstad et al. This particular technique involving dipoles (point-like couples placed on opposite side of the crack surfaces) exploits a user-defined cosine bell function (reported in figure 3.1) and it was used to find crack length related features [17], [18], [35].

An important parameter in the definition of the source dynamic is the step functions' rise-time because it has to represent as closely as possible the crack opening displacement time.

In [35] the authors state that there are no literature evidences of rise-time actual measurements, therefore this parameter is usually estimated by using the elastic material properties.

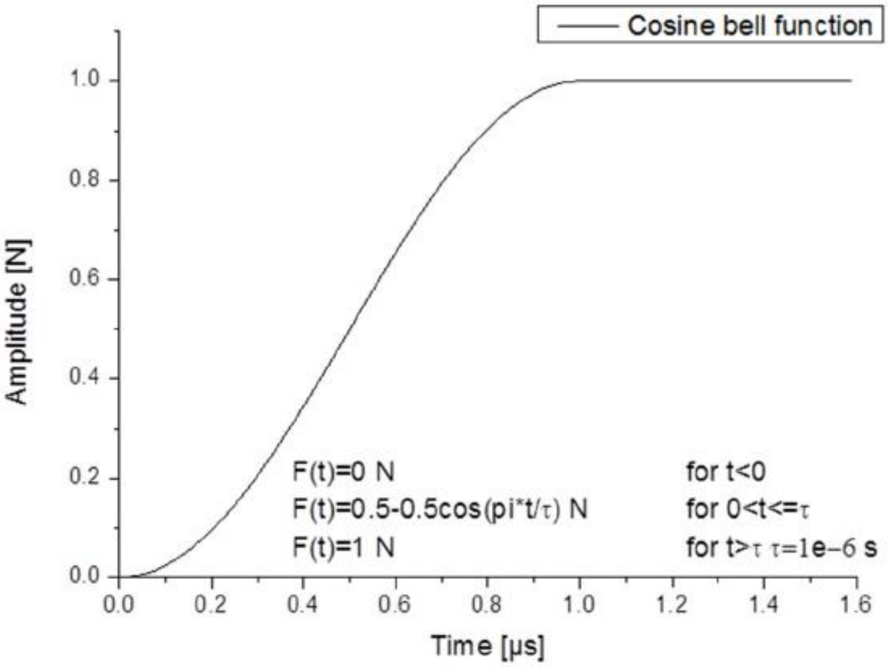


Figure 3.1: Cosine bell function

As mentioned before, the data collection process was supposed to be performed experimentally, therefore, to maintain the same work flow, the original aim of the thesis was to simulate a passive signal from the crack opening that could well resemble the experimental one.

In light of what stated until now throughout this section, the method to pursue this aim would have been to exploit the *Crack Tip Opening Displacement* [3] (CTOD) applied to the crack surface nodes. This displacement would have been theoretically able to generate a signal starting from the sudden movements of the nodes.

The issue of creating a crack growth model generating a realistic signal lies in the precise definition of the material's failure criteria and, most importantly, of the plastic zone developed around the crack.

A very precise modelling is required if large plastic deformations are expected prior to failure and, considering the ductile material under examination (aluminium), it is clear that plasticity would greatly affect the crack growth behaviour. Crack closure's phenomena and all the stresses induced in the nearby area are examples of this problem.

---

[3] Many notions regarding this parameter can be found in [64]-[65].

A proper evaluation and representation of all these factors is crucial, since they have a direct effect on the opening load and on the CTOD which, in turn, have consequences on the generated signal. Moreover, there could be also the possibility of having a generated high frequency signal, thus exciting the higher order modes and complicating enormously the successive signal analysis.

To resume, the following features have to be properly assessed and modelled in order to have the realistic generation of an AE signal within a numerical environment:

- Plasticity must be reasonably modelled in order to have a correct crack growth and therefore a correct crack surface’s displacement leading to a consequent realistic wave propagating through the material;
- The material’s proper failure criterions must be defined in order to have a realistic growth of the defect. These derive from multiple testing campaigns performed on specimens, where the measured parameters are employed, through some mathematical models, to create and improve the numerical one [36].
- The dynamic load must be defined, based on some realistic criteria (e.g. peak, stress ratio), and the number of cycles corresponding to a specific crack length must be assessed. Moreover, when dealing with dynamic loads, it is important to distinguish between signals generated from the crack opening (labelled as “primary”) and from the crack closure (labelled as “secondary”), meaning when the two surfaces go back in contact. Both could be possibly exploited to get complementary results [37].

The impossibility of performing the experimental tests necessary to validate and refine the numerical model reduced the chances that a so-developed passive methodology could give realistic results. As an example, the method used in [38], where the local degradation of the material properties is considered as the growth driving factor, could be a viable path. However, no experimental confirmations were specified, therefore it is not clear whether the plasticity modelling through a strain-hardening function was sufficiently realistic. Moreover, it was also evident that the difference between considering or not this aspect led to non-negligible changes in the results.

Because of the aforementioned reasons and due to the primary goal of this thesis, which is not related to the correct modelling of a crack growing inside the material but rather to determine whether some crack-related signal’s patterns could be automatically recognized, it was decided to exploit an *active* generation technique having an input signals interfering with discontinuities of different size directly drawn into the plate. The simulations were therefore performed exploiting the so-called *pitch-catch* methodology.

3.3. Techniques of signal analysis

Since the damage characterization is practically subjected to the interpretation of the recorded signals, before looking at the mathematical methods usually employed for signal analysis it is also important to state some characteristics of a time domain signal.

When an experimentally recorded signal is received, it is vital to discriminate it from the background noise and interferences (that are practically always present and varying) before analysing it. This process aims to identify the so-called *hits* of the signal, which are defined as transient wave packets isolated from the acquired waveform.

There are many techniques involved in detecting those hits. A detailed description of them all is outside the scope of this thesis, but the interested readers can find some of them in reference [20]. Conventionally, the most used technique consists in comparing the received signal with a specific threshold. The latter can be fixed or continuously adjusted in order to always stay just above the noise amplitude level present at that time. When the signal's amplitude overcomes the set threshold, a single hit is detected and it lasts until there are no more threshold crossing. Since the signals collected in this thesis actually come from numerical simulations (hence from an *"ideal"* environment), there is no need to perform the aforementioned discrimination. Background noise will not be present, therefore the amplitude recorded before the arrival of the desired signal should be null.

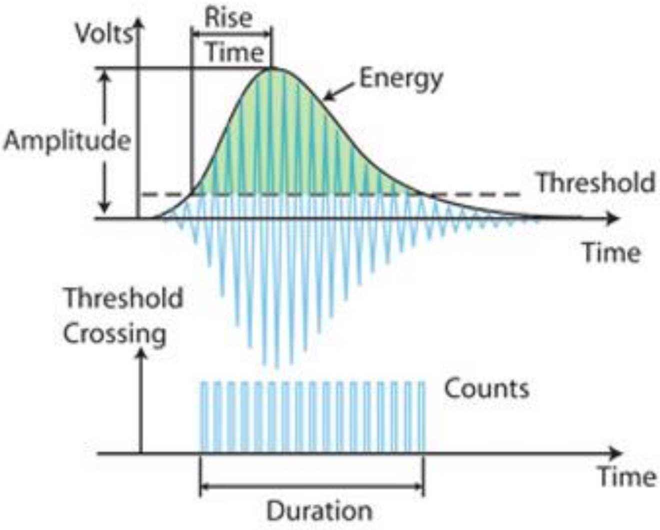


Figure 3.2: Some features of a hit

Some of the key characteristics associated to each hit, highlighted in figure 3.2, are:

- The *peak amplitude*, reached after the so-called rise time;
- The *counts*, which are the number of recorded peaks overcoming the threshold;
- The *duration*, determined by the time difference between the first and the last counts;
- The MARSE energy (Measured Area under the Rectified Signal Envelope), that does not have to be confused with the True energy (the area under the envelope squared).

Simple time domain analysis of signals have been used to detect damages, for example by comparing the recorded signal with an undamaged benchmark (i.e. a baseline) [39]. However, the most common way to examine a signal is usually through its frequency domain (via the Fourier transform), even if this transformation leads to a complete loss of the temporal history. There are also other techniques aimed at keeping as much information as possible from both the domains: they are labelled as “time-frequency representations”.
In the following two sections, some signal processing techniques are explained.

3.3.1. Hilbert transform

Considering a time domain signal *x(t)*, this transform allows to construct its mutual complex signal, also called *analytic* signal, through the following expression (3.2) [40]:

$$x_A(t) = x(t) + jH[x(t)] = x(t) + j\,\frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{x(\tau)}{t-\tau}\,d\tau = e(t)\,e^{j\,\varphi(t)} \tag{3.2}$$

The analytic signal is made of a real part coinciding with *x(t)*, and an imaginary part representing the signal *x(t)* phase-shifted by 90° (in other words, the initial signal is quadrature filtered).

The envelope *e(t)* and the instantaneous phase *φ(t)* can be derived exploiting respectively the equations (3.3) and (3.4) [40]:

$$e(t) = \sqrt{x(t)^2 + H[x(t)]^2} \tag{3.3}$$

$$\varphi(t) = arctan\frac{H[x(t)]}{x(t)}\ \ [rad] \tag{3.4}$$

Equation (3.4) is obtained applying the definition of *argument* of a complex number. Moreover, the argument’s time derivative allows to calculate the instantaneous frequency:

$$f_{inst} = \frac{1}{2\pi}\frac{d\,\varphi(t)}{dt}\ \ [Hz] \tag{3.5}$$

3.3.2. Time-frequency representation of a signal

As already mentioned, to retain as much information as possible from both the signal's domains, a time-frequency representation is necessary. This kind of analysis is especially employed when there are non-stationary signals of short duration and it is necessary to characterize their frequency content as a function of time.

Many signals related to practical applications have changing frequency characteristics, hence it is important to find also a correlation between the time evolution and themselves. The time-frequency representation comprises techniques studying a signal in both the domains at the same time: instead of focusing on one particular domain, these methods create a two-dimensional domain in which the signal is analysed. Indeed, the time-frequency representation may be considered as a refinement of the Fourier analysis.

In general, the switching from time domain to frequency domain is realized thanks to the Fourier transform[4]. The overall frequency information is obtained thanks to the this mathematical tool which reveals every frequency present along the entire time domain.

However, the Fourier transform presents a drawback: it gives a good *frequency resolution* but the *time resolution* is poor (more properly it is not defined) since the operation is applied to the whole time domain.

A signal represented only in the time domain is considered as a perfect time resolution. On the other hand, a signal solely represented by its Fourier transform is considered as a perfect frequency resolution, since this mathematical tool conveys every frequency component but it fails reporting when these occurred.

---

[4] It is worth spending some words also on the Fourier transform because it still remains a very useful tool for characterizing signals in the frequency domain and it has been used in countless applications. [16], [41], [42] are just some examples where the Fourier transform is respectively exploited to find frequency peaks related to specific crack length, to identify damage in composite materials and to classify frequency peaks related to different damage's types in aluminium alloy specimens.

The continuous Fourier transform is a mathematical operator defined in this way: $X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft}dt$, where *x(t)* is the continuous time domain signal that, due to the Fourier series theorem, can be represented by an infinite sum of sinusoidal functions with different frequencies and amplitudes. The Fourier transform is bijective, this means that it is possible to go back from the frequency domain to the time domain applying the following relation: $x(t) = \int_{-\infty}^{+\infty} X(f)e^{j2\pi ft}df$. To reduce the computational cost required, the Fast Fourier transform algorithm has been introduced and is widely employed for Lamb wave analysis, having a cost in the order of N*logN (where N indicates the size of the discrete sampled dataset).

Time-frequency representations provide a link between the previous two, therefore they are exploited when there are signals containing multiple time-varying frequencies.

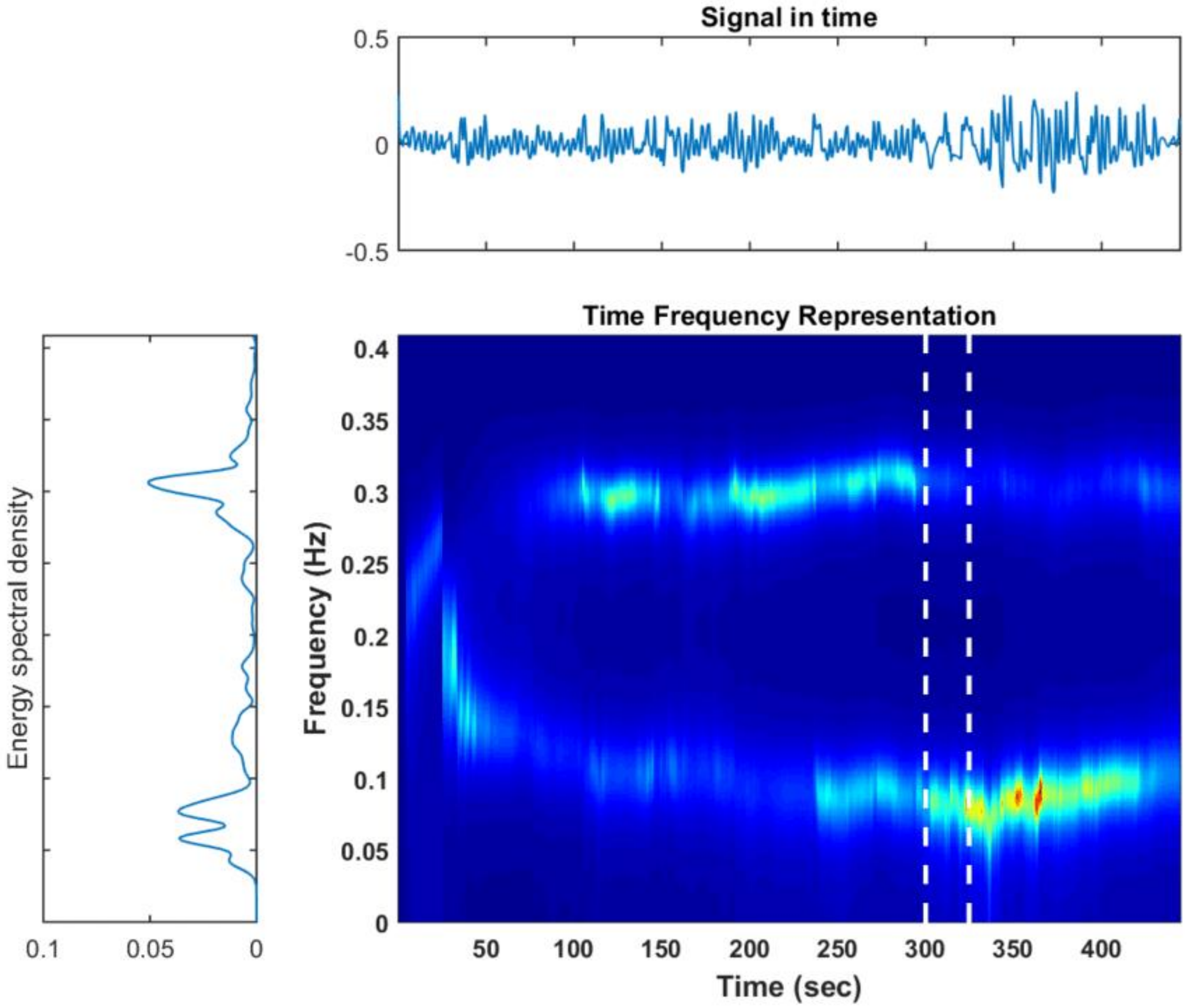


Figure 3.3: Three representation techniques of the same signal

Most of the time-frequency algorithms may be generalized under the following expression [28]:

$$P(t,\omega) = \frac{1}{2\pi} \iiint e^{-j\theta t - j\tau\omega - j\vartheta\delta} \varphi(\theta,\tau) x^{*}\left(\delta - \frac{\tau}{2}\right) x\left(\delta + \frac{\tau}{2}\right)\, d\theta\, d\tau\, d\delta \tag{3.6}$$

where *P(t,ω)* is the energy intensity at time *t* and frequency *ω*, *x** denotes the complex conjugate of the signal *x*, and *φ(θ,τ)* is a function depending on *x(t)*.

Actually, there are many techniques available to perform time-frequency representations of a signal [43]. These are all variants of equation (3.6), and some of them are:

- Short Time Fourier Transform;
- Gabor Transform;
- Wigner-Ville Transform;
- Choi-Williams Transform;
- Cone-shaped Transform;
- Wavelet Transform.

3.3.2.1. Short Time Fourier Transform

The STFT is an algorithm for the time-frequency representation introduced in 1946 to improve the Fourier transform potential and see how the spectrum changes over time.

The time domain signal is processed through a sliding time window which divides it into several sections (which may overlap) then individually subjected to the Fourier transform analysis for the identification of spectral components. Time windows of different shapes can be exploited in order to smooth abrupt signal's variations at both ends of the section.

The STFT of a signal function $x(t)$ is defined by relation (3.7):

$$X(\tau, \omega) = \int_{-\infty}^{+\infty} x(t)\varphi(t-\tau)e^{-j\omega t}\, dt \tag{3.7}$$

where $\varphi(t)$ is the window function, positioned in (τ,ω) on the time-frequency plane, that is shifted along the signal.

The knowledge of the original signal function $x(t)$ is thus required only within the analysed interval defined by the window function, and not on the entire domain.

An important feature to be underlined is that the product of the parameter describing the window function's width in both the domains is constant, therefore improving the resolution in one domain inevitably ruins it in the other. For example, exploiting a narrow time window gives a poor representation in the frequency domain and a relatively appreciable one in time. For this reason, selecting the window's width is always a compromise.

This is consistent with the *Heisenberg's uncertainty principle*, which indeed states that it is impossible to have a simultaneous perfect time and frequency resolution.

3.3.2.2. Gabor Transform

This time-frequency representation (named in this way due to its inventor) exploits a time window function having a specific Gaussian shape defined by relation (3.8):

$$\varphi(t) = \frac{1}{2\pi\alpha} e^{-t^2/4\alpha} \ \ with\ \alpha > 0 \tag{3.8}$$

The Gabor transform is then defined by the following expression:

$$G(\tau, \omega) = \int_{-\infty}^{+\infty} x(t)\varphi(t-\tau)e^{-j\omega t}\, dt \tag{3.9}$$

The particularity of this representation is that its window function allows to reach the lower limit defined by the uncertainty principle, meaning it is considered the best in terms of decomposition and possible reconstruction of the original signal.

3.3.2.3. Wigner-Ville Transform

The transforms previously described are able to give straightforward information only about the signal’s amplitude. However, exploiting them it is possible to define the so-called *spectrogram* of the signal (where the time-frequency components are associated with their respective power intensities) through the squaring operation of the spectral amplitudes.

To get the power spectrum of the signal, it might be convenient to directly calculate the energy over a finite and symmetrical time interval placed around the considered time instant. This idea is exploited in the Wigner-Ville transform, defined by equation (3.10):

$$WV(t, \omega) = \int_{-\infty}^{+\infty} x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) e^{-j\omega\tau}\, d\tau \tag{3.10}$$

This non-linear representation has a better time-frequency resolution with respect to the previous two however, the only use of this transform might not be sufficient to perform a satisfactory analysis. Indeed, this algorithm usually requires a very high sampling rate in order to work properly and avoid aliasing[5]. Moreover, when complicated signals (such as the ones generated by the combination of different signals) are analysed, some interferences may arise due to the non-linear form of the representation that leads to a cross linking between different components: each time-frequency element present in the initial signal couples with another one.

To reduce the aforementioned interferences and improve the raw signal characterization, a filter is added to the transformation process.

These new transformations belong to the Cohen's class, where one of the most famous is the *Choi-Williams* transform: this one suppresses the interferences generated by signal components at different times and frequencies, however the ones occurring at the same time or possessing the same frequencies remain unchanged.

The *Cone-shaped* transform is another time-frequency representation aiming at reducing the interferences created by the Wigner-Ville transform. However, despite the decrease of these detrimental interferences, the time-frequency resolution also deteriorates.

3.3.2.4. Wavelet Transform

The continuous wavelet transform (CWT) of a signal function *x(t)* is defined by the following relations:

$$W_{coeff}(a,\tau) = \int_{-\infty}^{+\infty} x(t)\varphi_{a,\tau}^{*}(t)\, dt \tag{3.11}$$

$$\varphi_{a,\tau}(t) = \frac{1}{\sqrt{a}}\varphi(\frac{t-\tau}{a}) \tag{3.12}$$

where $\varphi(t)$ is the wavelet function, * denotes the complex conjugate, a is a scale parameter and τ a time delay (representing the wavelet location along the time axis).

Wavelet functions, whose name derives from the fact that they actually resemble a piece of waveform, are localized in both time and scale (the equivalent of frequency), hence the

---

[5] The Nyquist-Shannon sampling theorem states: given a function whose Fourier transform is zero outside a certain frequency range (i.e. a signal with a limited bandwidth), in its analog-to-digital conversion the minimum sampling frequency necessary to avoid aliasing, and therefore loss of information in the reconstruction of the original analog signal, must be greater than the double of the signal's maximum frequency of interest.

During the analog-to-digital conversion the signal is subjected to two distinct discretization: the discretization of times (sampling) and the discretization of amplitudes (quantization). On the other hand, the digital signal is transformed back into the analog one by interpolation processes.

If the aforementioned theorem is not fulfilled, i.e. there is a sub-sampling of the analog signal in the time domain, there is a production of frequencies unrelated to the original signal in the frequency domain, and this generation process is called "alias". Vice versa, from the frequency domain to the time domain, significant distortions of the signal arise and the latter is no longer representative of the original one.

transformation of a signal through this method may lead to a good representation in the time-scale plane because the waveform is scaled and translated to catch many different signal features. Then, a proper transformation *(1/a)* leads to the actual time-frequency representation. A larger scale associated to the wavelet function (higher *a*) represents a more stretched window, hence a larger signal section is analysed and its coarser features (more frequency components) are characterized. On the other hand, decreasing *a* and translating the wavelet in the time domain thanks to the parameter τ, it is possible to get a poorer frequency resolution while having a very good time representation. This is the reason why this specific time-frequency analysis is labelled as *"multi-resolution"*, which is actually its major strength.

As already seen, the STFT is defined in both time and frequency but it has some resolution problems and often wavelets offer a better representation of the signal thanks to the use of the aforementioned "multi-resolution" analysis. Indeed, looking at the time-frequency plane representation of the wavelet's window function (figure 3.4), it is possible to see the advantage of this particular analysis: its shape is a function of the window's position, this means that the resolution changes over the time-frequency plane. This is opposite to the STFT, where the window's size is fixed and therefore the resolution is constant for every position on the time-frequency plane.

As last, the wavelet transform is less complex, requiring a computational time in the order of N, while the Fast Fourier transform, exploited in the STFT algorithm, requires a computing time in the order of N*logN (where N indicates the size of the discrete sampled data).

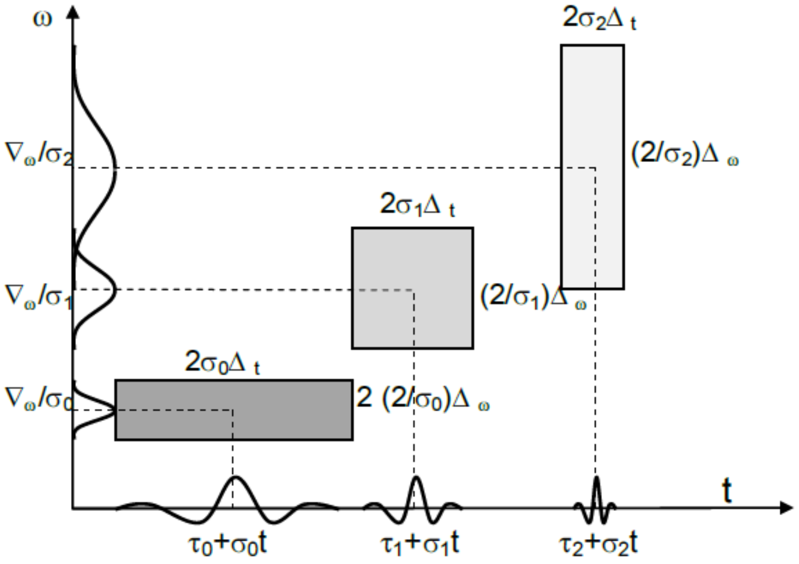


Figure 3.4: Wavelet window across the time-frequency plane (σ is actually the scale parameter named *a*) [43]

The wavelet function may be considered as the window function defined for the previous time-frequency representation methods (e.g. STFT). However, the *“mother wavelet”* has to submit some criteria in order to be properly exploited for this purpose: mainly, it has to be a time limited, oscillating and decaying function, whose average amplitude equals zero.

Calculating the wavelet coefficients for every scale point and at every time sample (since the context is referred to discrete signals) may be computationally expensive. However, once this procedure is completed, a 3D plot called *scalogram* is obtained where the three axes represent respectively the signal’s time domain, the wavelet scale *a* and the resultant wavelet coefficient. The signal energy concentrations in the time-scale domain are represented by the scalogram peaks, i.e. by the highest values of the wavelet coefficient.

To simplify the previous method, equation (3.11) can be computed at some discretised scale and time points, leading to the discrete wavelet transform (DWT).

A method to effectively implement the DWT was proposed by Mallat in 1989. In his work the term “approximation” was used to describe the low frequency components of the signal, while the term “detail” to highlight the high frequency ones. In simple words, the algorithm is based on a cascade of filtering operations which include two types of filters: a low-pass, which splits the “approximation” from the analysed signal, and a high-pass, which separates the “detail”. The filters’ properties are a consequence of the chosen wavelet function: the shape identifies the low-pass filter, and the scaling function the high-pass one.

At each step of the algorithm, the “approximation” coming from the previous step goes through both filters in a decomposition process labelled as “wavelet decomposition tree”.

The original signal may then be reconstructed exploiting an inverse algorithm with respect to the aforementioned decomposition.

A basically unlimited wavelet library exists (Haar, Daubechies, Meyer, Morlet, Ricker or Mexican Hat, etc.). The best one depends on the intended application, since the choice of the time-frequency representation method may have important consequences on the results extrapolated from the subsequent analysis.

As an example, CWT has been used in [5] for the precise determination of the signal arrival times, thus helping to refine a damage localization algorithm based on Lamb waves responses. Moreover, Su et al. successfully located delamination in carbon fiber/epoxy resin composite laminate exploiting WT analysis [44].

Due to the aforementioned versatility and properties they possess (mainly the ability to describe signals with limited number of coefficients and good time-location), the time-frequency representation and subsequent analysis exploiting wavelets is getting an increasing attention within the signal processing field.

3.4. The Shazam algorithm

The author would like to give also a brief introduction to this algorithm, since it is the one that inspired the very first work by N. Facciotto.

The Shazam^{TM} smartphone application is a very famous and used tool allowing the user to recognize songs even from a short and noisy sample. To exploit this software, the listener is just required to capture a short excerpt of a song with the smartphone’s microphone, and in some seconds the app will report the song title and the artist.

How does the application work? One of the developer of the algorithm, Avery Li-Chun Wang, reported in [45] the idea behind the software and its basic operative principles.

The algorithm was born in 2000 with the idea of allowing people to quickly recognize many songs in every environment, simply using their mobile phones. Because of the multitude of ambient distortions that could be present, the algorithm was required to work even with heavy ambient noise, reverb, foreground voices and signal drops, while having a low number of false positives. At the beginning the database was composed by two million tracks and it is now being regularly updated to keep track of new released songs.

The working principle is the following: each audio file, both database and sample, is *“fingerprinted”*, this means that reproducible hashes (short numerical values) are extracted and labelled to a specific file, then the ones from the unknown sample are compared with the large database in order to look for a possible match. Fingerprints are not casually chosen, but they must possess some characteristics; some of the most important are:

- Temporal localization: each calculated hash is localized in time so that distant events do not affect it.
- Translational constancy: each hash is independent from the position within the audio file.
- Robustness: hashes generated from the original “clean” database’s song should be able to match with disturbed hashes coming from the recorded sample. This property highlights the required insensitivity to noise and distortions.

- Sufficient entropy: the fingerprints should have the right characteristics in order to minimize the probability of false positives. An insufficient entropy would produce excessive positive matches, while an exuberant entropy would lead to the inability of detecting positive matches in noisy conditions.

Each fingerprint is built through the spectrogram's peaks of the signal. Points in the time-frequency plane having higher energy content with respect to the neighbouring ones are eligible as song-marks because higher amplitude peaks are more likely to survive to the distortions already mentioned.

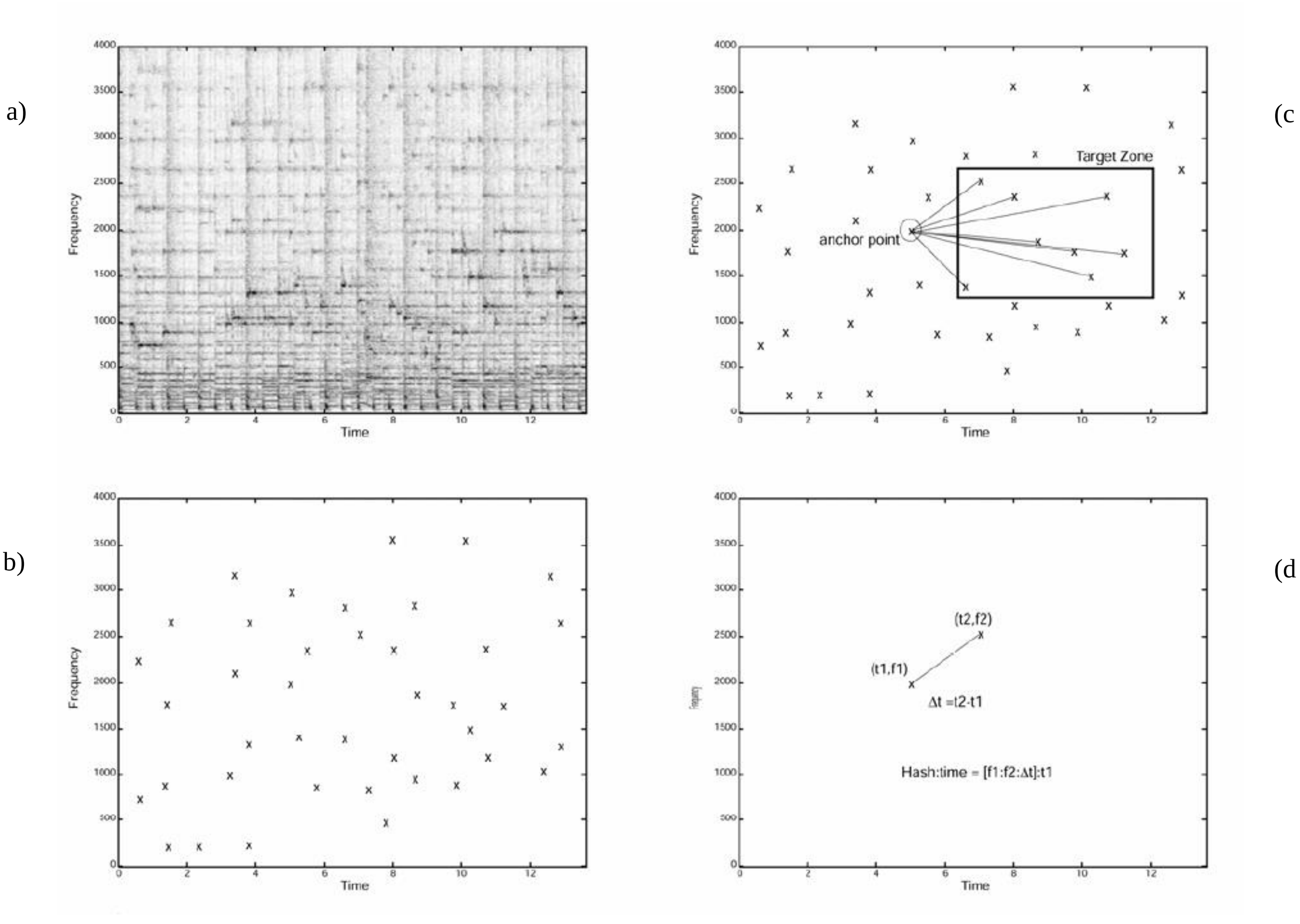


Figure 3.5: Database creation sequence of the Shazam algorithm: (a) Spectrogram, (b) Constellation map, (c) Hash generation, (d) Hash details [45]

In figure 3.5 is displayed the operative sequence the algorithm performs in order to create the final table reporting the hashes parameters.

The spectrogram reported in the upper left side of the picture is reduced to a sparse set of coordinates called "constellation map" since the plot resemble a star field. To create it, each amplitude-time plot is processed to get its spectrogram. Then, the peaks are selected considering the amplitudes overcoming a certain selected threshold.

The subsequent hashes are formed combining two time-frequency points: first, the so-called "anchor point" is chosen, then all the other points taken from the anchor point's "target zone" are coupled with it. The process is repeated for different anchor points eventually producing a sort of table where each pair (highlighted in figure 3.6) has the two frequencies and the time difference registered in a 32-bit integer. Because of the aforementioned peaks selection criteria, these hashes are usable even in presence of strong noise.

To create a database index, the above described operations are carried out on each song in order to generate a corresponding hashes list with the associated offset times.

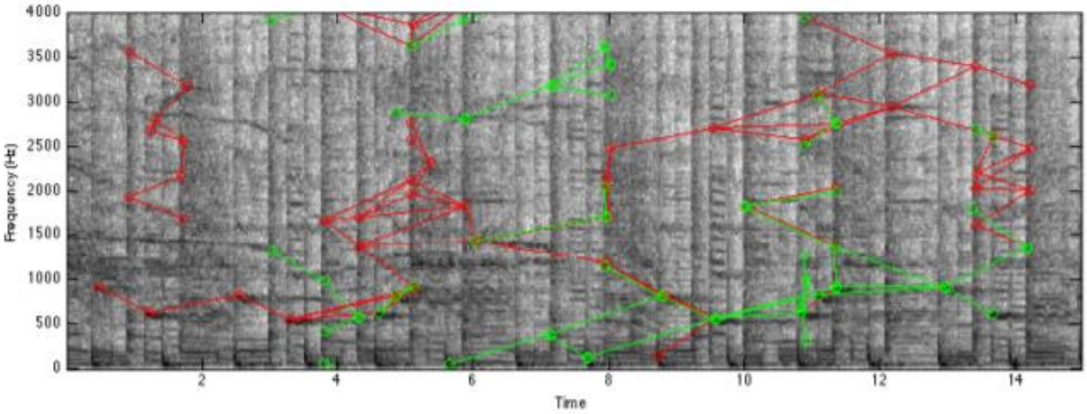


Figure 3.6: Map of hash pairs for a 15 seconds sample

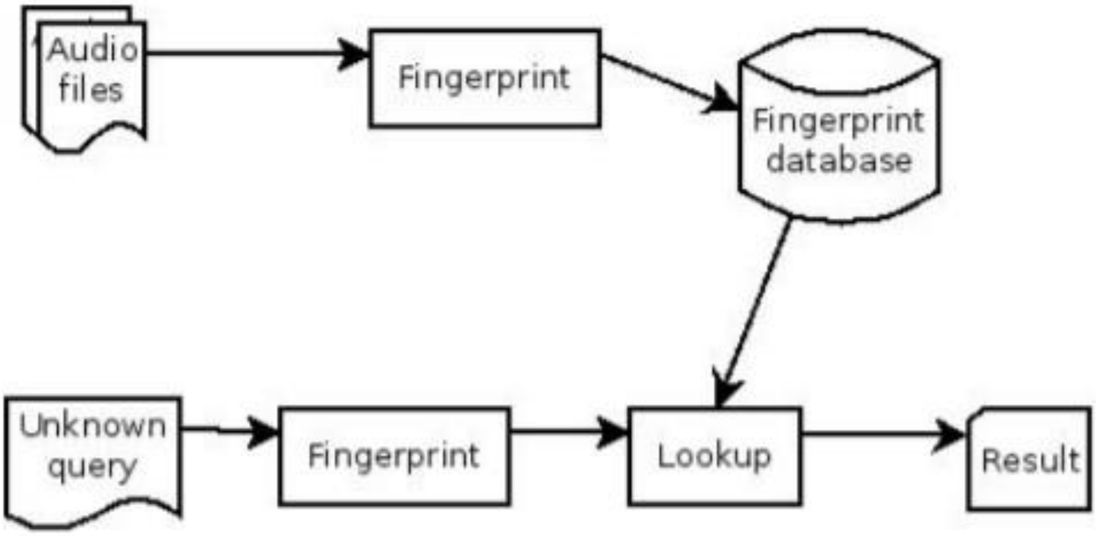


Figure 3.7: General working scheme of a pattern recognition algorithm

In simple terms, this pattern recognition algorithm tries to look for a match between the constellation maps in the database and the one derived from the unknown sample; when the proper time offset is found, a significant number of constellation points will coincide and the algorithm will return a positive match. The Fourier analysis is the key mathematical tool that powers the algorithm, however also wavelet-based analysis were performed in some cases.

In the recent years, a Shazam-like algorithm coupled with Machine Learning has been proposed for digital auscultation and detection of cardiological valve diseases since it requires much less expertise: primary care physicians would be able to assess problems without specifically asking to a cardiologist [46].

Seyoum et al. also proposed in [47] an approach to detect water leakages through sound recordings and later signal processing by an identification software comparing the records with a database library. This opposes to the more expensive and laborious conventional method where flow sensors are placed within the piping systems to perform mass balance calculations.

3.5. Artificial Intelligence and Machine Learning techniques

When dealing with the processing and interpretation of GLW signals, two methods' categories usually exist: *forward* and *inverse*.

As the name suggests, the former analysis is referred to the drawing of conclusions directly from the properties of the received signal, and in most cases the solution is unique. On the other hand, an inverse problem is more difficult to be solved and the solution might be ambiguous.

Some techniques belonging to the first category are related to [28]:

- The Time of Flight, employed to locate damages through some triangulation techniques.
- The Time-Reversal imaging methods [48].
- Benchmarking: hence studying some damage's characteristics through comparisons of signals and the definition of a properly constructed damage index.

To resume, direct analysis on waves scattered by a defect can be labelled as *forward*, while the analysis to fulfil damage's quantification trying to find adequate features from the scattered waves can be named as *inverse*.

The quantitative damage identification this thesis is trying to address is indeed known as a typical inverse problem, and it is in such situations that Machine Learning techniques show their strength.

Many efforts have been directed to the assessment of damages throughout the years, and one of the emerging approaches is the exploitation of Artificial Intelligence (AI). Within this branch, Machine Learning algorithms are of particular interest.

Machine Learning is that branch of computer science teaching the computers how to *learn from experience*: this means to automatically find recognizable patterns directly from the acquired data without relying on existing constitutive equations, take proper decisions, and improve the predictive performance on future inputs as the amount of available data increases.

Two different broad techniques are usually employed in Machine Learning [66]:

- *Unsupervised* learning, which finds hidden patterns and discover internal representations based only on input data. Clustering of data within the multidimensional feature space is the most common technique of this category. There are studies confirming the effectiveness of this analysis, performed on signals' frequency content, for damage identification in composite materials [49].

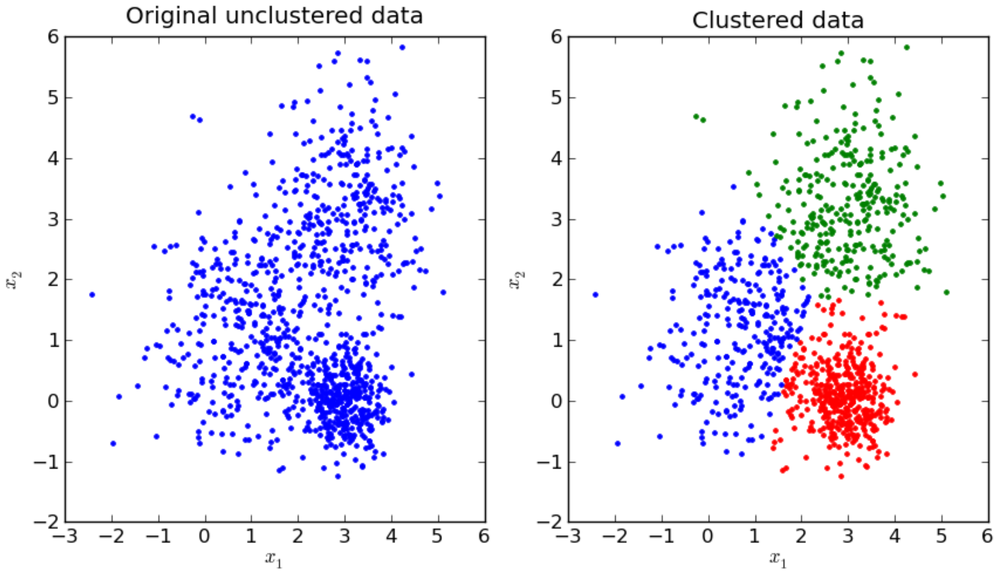


Figure 3.8: Result of a clustering process

- *Supervised* learning, which exploits models trained from both input and known output data to find a sort of mapping relation and make predictions about future inputs.

  More precisely, these algorithms take a training set of input data (which in this thesis are signals collected from the numerical simulations) and known responses (correct and related discontinuity's length) in order to train a model and generate reasonable estimates for the responses to new input data. Depending on the nature of the response (discrete or continuous), the supervised learning techniques can be split in:

  - *Classification* techniques, where discrete data categories are assigned to different classes through the definition of a mapping function.
  - *Regression* techniques, where patterns are assigned to real continuous variables. Future values of these continuous parameters (e.g. temperature changes) are addressed starting from a discrete number of training samples and a function the algorithm came up with during the training phase.

An example of a family of algorithms belonging to the regression category is represented by the so-called *Artificial Neural Networks* (ANN). They are computational models exploring, for a given system, the connections between series of inputs and outputs. A well-trained ANN may predict the outcomes deriving from an unknown stimulus according to previously gained knowledge.

As already mentioned in previous sections, Lamb waves have a much higher sensitivity to damages with respect to other techniques linked to structural responses, hence their combination with ANN and other supervised Machine Learning techniques may lead to a more precise damage characterization.

As the reader may see referring to the following figure 3.9, there are plenty of algorithms available in both categories and it is very difficult to state which is the best, because the choice mainly depends on the size and type of data, and the required responses. A comprehensive explanation of each algorithm is out of this thesis' scope; moreover, most of the times, the proper one for a specific purpose is found simply by enforcing a trial and error evaluation.

Most of the problems involved in the application of Machine Learning techniques are related to the data handling and finding of a proper model. Indeed, pre-processing the data requires knowledge directly related to the working field (signal analysis, image processing, etc.) and finding the right model is always a trade-off between speed and accuracy.

Despite the difficulty in finding the proper algorithm, there are some hints that may help addressing this topic [67]. First of all, if the goal is to train a model to make predictions or classifications based on existing data, the supervised algorithms should be chosen. Otherwise, if the aim is to explore the data to find correlations between them, the unsupervised algorithms have to be selected.

In this thesis, supervised algorithms were exploited since the work requires a classification of signals and a prediction of the damage size through the recognition of size-related patterns within different received signals.

Classification is an easier procedure with respect to regression since the latter accounts for a continuous changing value of the chosen response, therefore it is better to address the problem first with a classification and only later with the regression approach.

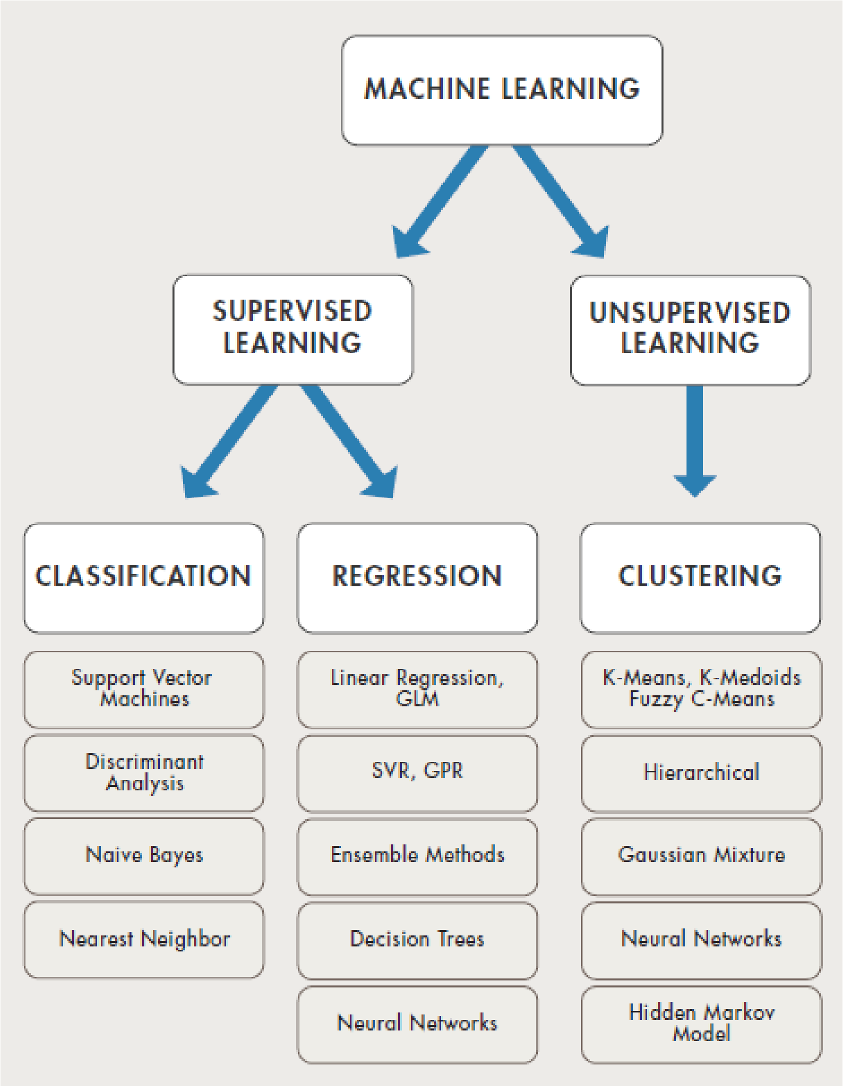


Figure 3.9: Machine Learning algorithms [66]

After the proper macro-class of algorithms has been identified, these are the operations necessary to successfully implement the best possible Machine Learning model:

1. *Collect* and *load* the data making sure that there are no information missing.

2. *Pre-process* the data to remove noise and interferences if present, normalize them, get rid of the outliers, etc.

3. *Data Mining:* extract the characteristic features using the pre-processed data. This step turns the raw data into useful *high-level* information for the algorithm. The number of extractable features is theoretically unlimited.

   Sometimes, in different branch of the so-called *Data Science* [6], the researchers have to deal with "big data": this expression underlines the fact that the amount of data is so huge that a manual process is no longer feasible. In this study, the number of simulations results was not so high to be defined as above, however the use of automatic procedures could well improve the analysis capabilities.

   It is important to underline that, within MATLAB^{TM}, the extracted features and the desired response should be put in a table format before using them as training or test sets for the algorithms.

4. *Split* the data in three sets: the training set, the validation set (which is sometimes comprehended in the former), and the test set. The latter is used to assess the performances of the chosen model. Usually, the larger is the training set, the better are the performances of the algorithms.

   In the validation procedure employed in this thesis, however, the test set was not present since it would have required another set of simulations to be run. Instead, the author preferred to use one of the training samples as the validation set and the other nineteen samples as the training set in a procedure called, as already mentioned in section 3.1, *"leave one out"* cross-validation.

5. *Train* different models, starting from the simpler and moving onto complex ones, using the previously derived features. Batch learning was exploited in this study: this means that the training was done only once, and then the system was switched to the working mode. There exist other two types of learning options (Incremental and Natural), they can give better results but are also more prone to the loss of generalization (the so-called *"overfitting"* phenomena).

---

[6] It is the scientific sector that deals with different methodologies for the extraction of useful information from raw datasets. Techniques for the managing and manipulation of data, their exploration, fitting operations and statistical analysis are within this branch.

6. Find the best model through the evaluation of the *performances*. The performances of classification algorithms are based on accuracy percentages, the ones highlighted inside the so-called *confusion matrices*. These are tables representing the probability of correct classifications according to the dataset; they basically compare the classifications made by the supervised algorithm with respect to the assigned know output classes.

   On the other hand, for regression problems the algorithms' performances are based on the *Root Mean Square Error* (RMSE).

   The more are the classes, the higher is the required complexity of the model.

   If the results are not satisfactory there is the possibility to improve the model through simplifications or adding complexity (e.g. merging simpler models or augmenting the data sources). The first option, which is also the preferable one, is reached instead through the reduction of the number of features.

   This can be done exploiting:

   - Correlation matrix: it tells which variables are less mutually related so that those can be removed.
   - Principal Component Analysis (PCA): it linearly combines and maps existing features into new ones, eliminating the redundancy and finding a single feature that captures the important distinctions between the originals.

     The fact that this analysis finds linear relations between features is a limiting factor for this particular work: GLW propagation phenomena is indeed linear, however the interactions with defects give rise to non-linear effects that would not be consequently captured.
   - Neighbourhood Component Analysis (NCA): it finds the importance each feature has in predicting the output and the less useful are discarded.
   - Sequential feature reduction: it reduces the features in the model until there is no more a performance increment.

7. *Integrate* the model into a larger system for its application.

The following figure shows the general workflow described above:

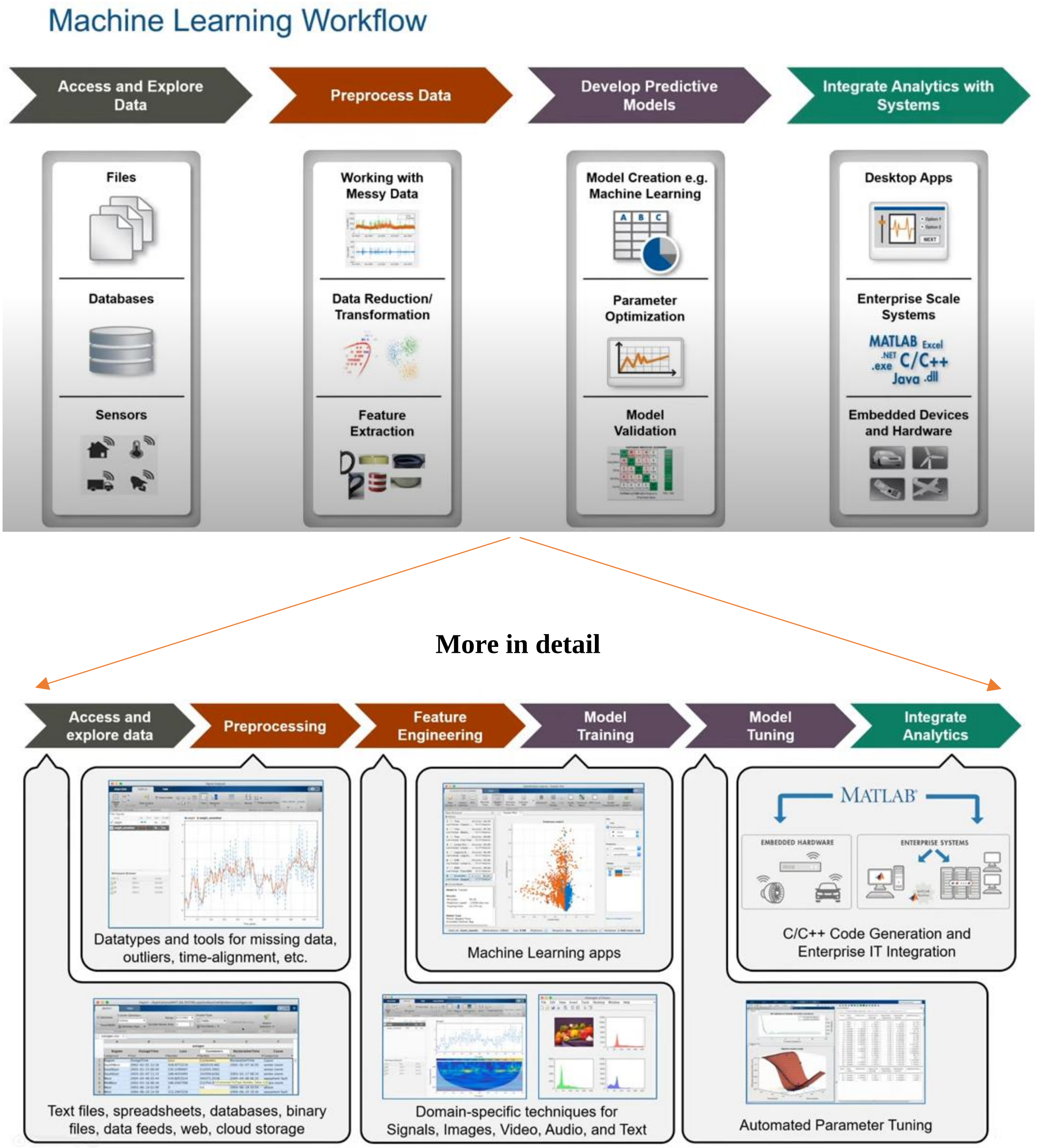


Figure 3.10: Machine Learning workflow [68]

CHAPTER 4

4. NUMERICAL SIMULATIONS

The first forms of finite element methods (FEM) belongs to A. Hrennikoff and R. Courant and they were used for purposes related to civil and airline engineering [50]. Nowadays, the accuracy of computer simulations have significantly improved, hence they are becoming more and more employed in the scientific research and industrial design.

As mentioned within the methodology section 3.1, in this thesis they were mainly exploited to collect the amount of raw data necessary to train the Machine Learning algorithm aimed at pursuing size classification and prediction. However, prior to those, other simulations were actually performed in order to become familiar with the wave propagation phenomena and to decide on the most time-efficient FE software between the available two.

4.1. Hanning windowed tone burst

Usually, narrowband signals are exploited when dealing with active techniques because of the aforementioned Lamb waves' dispersive nature: indeed, each frequency component introduced in the material will propagate changing both its frequency and velocity, eventually leading to an intricate signal being received. However, for the researchers is vital to separately identify every behaviour the signal may show, and this means reducing as much as possible the frequency bandwidth. This would avoid further challenges when implementing the successive signal analysis resulting in a more spot on damage location, detection and characterization methods.

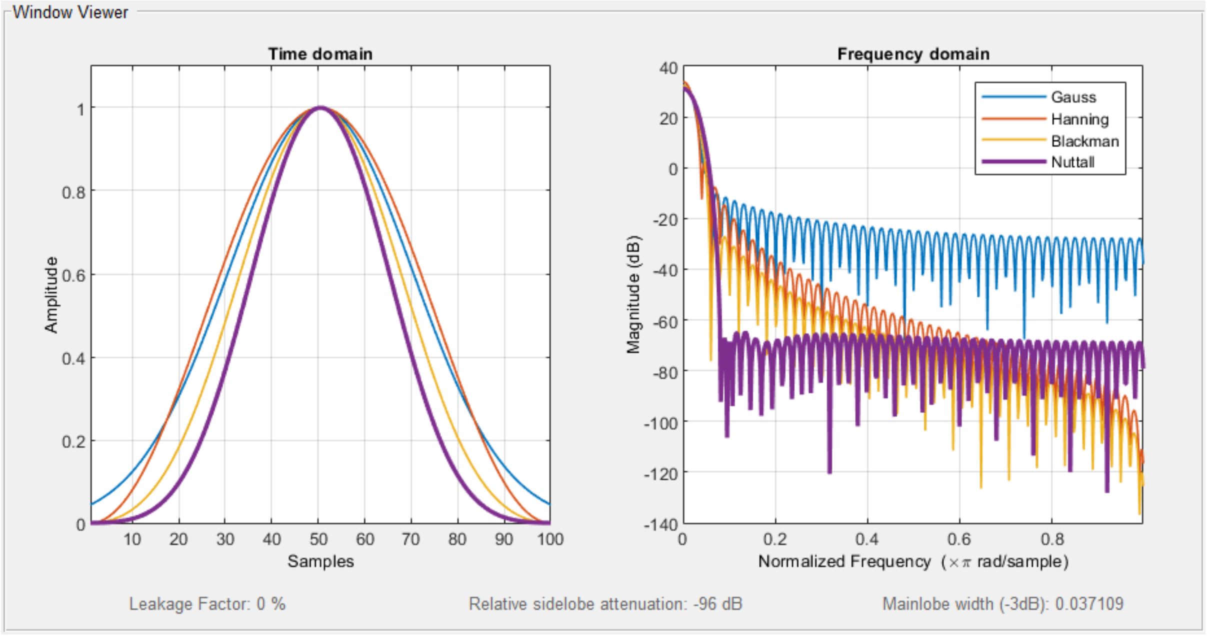
Moreover, there is another thing the researchers have to care about even if they actuate a single frequency signal: the excitation of the secondary modes. This happens when the chosen frequency overcomes the threshold mentioned in section 2.2.1. Observing the dispersion curves, like the one reported in figure 2.7, a "safe" frequency range for actuating only the fundamental modes can be established for each case study, thus avoiding once again a chaotic situation within the structure under analysis.

In light of the aforementioned reasons, the choice of a narrowband signal centred around a single frequency is then desirable, and the availability of the dispersion curves is paramount when selecting this latter parameter.

To construct the narrowband signal, a window modulation applied to a sine function (where the frequency of the sine function corresponds to the central frequency of the generated waves) is usually chosen, leading to the generation of the so-called *windowed tone bursts*. This type of diagnostic signal has become a current practice for the majority of the researchers studying this subject.

Actually, when windowing a sine function, there are some new spectral components arising and this phenomena is called *spectral leakage* (refer to figure 4.1): the choice of the window function is mostly dependent on the minimization of the "side frequencies" in order to consider the actual narrowband signal a sort of single frequency, as it was the original sine wave.

Several time windows functions exists: one well known and used in the SHM field is the Hanning Window, that is usually preferred to others like Nuttall and Gaussian. The following figure, obtained exploiting the function *"wvtool"* in MATLAB^{TM}, shows that Hanning and Blackman windows are the best in terms of spectral leakage and they are also very similar.



Leakage Factor: 0 %

Relative sidelobe attenuation: -96 dB

Mainlobe width (-3dB): 0.037109

Figure 4.1: Comparison of window functions

The Hanning window function is represented by the following expression:

$$h(t) = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi f_0 t}{N}\right)\right) \tag{4.1}$$

where f_0 is the frequency of the sine wave and N is the count number in the tone burst.

In order to perform the initial part of this study, relatively low frequencies were adopted (50 and 200 kHz) to lower the computational cost, as will be better understood in section 4.4.2 where the mesh size and time steps are assessed.

However, due to the interaction rule stating that the minimum defect size detectable in a pitch-catch configuration should correspond to at least half of the minimum wavelength associated with the incoming Lamb wave [12], the central frequency associated to the raw data collection simulations shifted up to 600 kHz because of the initial 2 mm discontinuity's size.

Actually, some detectable results could be seen even for a 200 kHz Hanning windowed tone burst interacting with a defect but, shifting up the frequency, the interactions and the differences in the received signal should be more evident [51].

The results of the windowing operation, performed in MATLAB^{TM}, for a five cycle sine wave having frequency equal to 600 kHz is reported in the next figure, following a step-by-step representation of the process: (a) sine wave, (b) Hanning window, (c) final result.

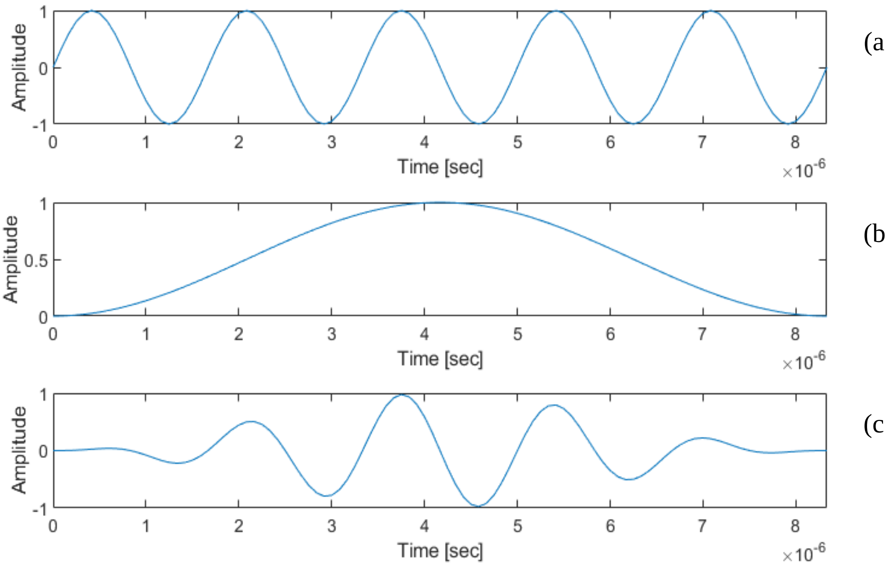


Figure 4.2: 5 cycle 600 kHz Hanning windowed tone burst

4.2. Actuation and sensing of a signal through PZT transducers

There are different excitation methods available in literature for triggering GLW in a numerical environment, and some of them are reported in [30], [40], [52]. According to those, the exploited solver for the solution of the differential equations can be explicit or implicit.

The fact that in some techniques (e.g. concentrated nodal forces and uniform pressure distribution over an area) it is not required to invert matrices, reduces the calculation time and makes the explicit finite element method very efficient for modelling the wave propagation. It is generally employed when the excitation method does not involve piezoelectric actuators.

On the other hand, the most used and famous method for actuating GLW in a real structure is through piezoelectric actuators. When these devices are involved in simulations it is generally necessary to exploit the implicit solver, e.g. contained in the Abaqus/Standard package.

However, a sort of hybrid solver is instead employed in OnScale^{TM} allowing for a great reduction of the computational time even when piezoelectric transducers are involved, as it is discussed in section 4.3.

4.2.1. Characteristics for their selection

Transducers are primarily selected based on the operating frequency response (which must suit the application's range), but also sensitivity to damage-related features and characteristics of resistance to atmospheric agents play an important role.

Usually, the frequency range for the selection of a specific device have to be determined experimentally since many factors contribute to its definition [69]:

- *Attenuation* is frequency dependent: indeed, considering the same material, the higher is the frequency, the stronger is the attenuation per unit distance. Lower frequencies are used for materials having high attenuation (e.g. CFRP) in order to better distinguish the signals [53].
- *Background noise* is usually affecting more the lower frequencies, therefore false triggering can be avoided when moving to higher frequencies.
- *Source mechanism and the medium* through which the waves travel also affect the frequency content. The material in which an elastic wave propagates has a very inhomogeneous effect on the frequency distribution, indeed it attenuates certain frequencies more than others.

  Moreover, wave propagation is also affected by *macroscopic geometric features* of the tested object: dispersion (i.e. different wave velocities at different frequencies) in plate like structures is a striking example already mentioned in section 2.2.1.

The features that can be extracted from the received signal are, to some extent, linked to the frequency characteristics (mainly peak and range) of the chosen piezoelectric sensor. Therefore, selecting the appropriate device for a specific application is crucial for the final success of the measurement campaign.

Each piezoelectric device has its own *characteristic response vs frequency* curve, where the frequency range giving the best response in terms of amplitude is identified.

Sensors responding uniformly to a very large band of exciting frequencies, thus presenting the wider usable frequency range, are referred to as *"broadband"* or *"wideband"*. These sensors with a large and almost flat response curve are usually preferred either when the frequency of interest is still unknown or when many frequencies within the same signal should be analysed. In order to optimize the device with respect to the working frequency, a general rule is to excite the piezoelectric elements as near as possible to its natural frequency [54]. Most sensors are indeed of resonant type, which means that they are more sensitive near their resonance frequencies and possess other frequency bands where their sensitivity is lower.

The *size* of the piezoelectric elements also affects the resonance frequency of the device: in general, the resonance frequency is higher for smaller elements and the lower is the desired frequency range, the larger should be the chosen size. This operation of changing the dimensions according to the operating frequencies is called *"tuning"*; Santoni et al. reported in [55] that, in order to achieve the maximum theoretical tuning effect, the Piezoelectric Wafer Active Sensor (PWAS) dimension should equal half the wavelength of the desired Lamb wave mode.

Considering the fact that throughout all the simulations performed in this thesis the author wanted to keep the same dimension for the piezoelectric actuator and sensors, the further analysis of the received signal's frequency content, and the lack of experiments to properly assess the transducer's selection, the final conservative choice fell on a "broadband" type with a diameter dimension of 6.5mm.

Looking at some datasheets available online it was indeed recognized that broadband sensors operating at high frequency bands (> 400 kHz) and having their peak frequency near 600 kHz (which is the key frequency of this study), have a disc's diameter around 7mm.

The relations constituting the proper definition of a piezoelectric material are given by the following system [56]:

$$\begin{bmatrix} D \\ \varepsilon \end{bmatrix} = \begin{bmatrix} e^{\sigma} & d^{d} \\ d^{c} & s^{E} \end{bmatrix} \begin{bmatrix} E \\ \sigma \end{bmatrix} \tag{4.2}$$

where the vector D (3x1) is the electric displacement [Coulomb/m^2], ε is the strain vector (6x1), E is the applied electric field vector (3x1) [Volt/m], and σ is the stress vector (6x1) [N/m^2]. The piezoelectric constants are the dielectric (electrical permeability) e_{ij}^{σ} (3x3) [Farad/m], the piezoelectric charge strain coefficients d_{im}^{d} (3x6) and d_{jk}^{c} (6x3) [Coulomb/N or m/Volt], and the elastic compliance S_{km}^{E} (6x6) [m^2/N].

Looking at this latter statement, it is clear that piezoelectric materials require the definition of both mechanical and electrical characteristics.

The choice of the piezoelectric material to be assigned to all the discs fell on the PZT 5A. This is a Navy type II material used primarily in devices like phono pickups and sound detectors. The geometric and material features eventually assigned to the piezoelectric discs are reported in table 4.1, using the OnScale^{TM} software notation for sake of clarity. Because of the advantages reported later in the development of this thesis, this was indeed the chosen software to carry out the numerical simulations with varying defects' geometries.

PZT 5A Navy II material:

	Diameter and Thickness of the disc	6.5 x 0.25 mm
Mechanical Stiffness constants	**c11 = c22**	1.203 *10^{11} Pa
	c12	7.52 *10^{10} Pa
	c13 = c23	7.51 *10^{10} Pa
	c33	1.109 *10^{11} Pa
	c44 = c55	2.11 *10^{10} Pa
	c66	2.26 *10^{10} Pa
	Density	7750 kg/m^3
Piezoelectric Stress constants	**e31 = e32**	-5.4 C/m^2
	e33	15.8 C/m^2
	e24 = e15	12.3 C/m^2
	Poling	along Z^+
Rayleigh Damping parameters	**Bulk attenuation**	75 Q units
	Shear attenuation	75 Q units
	Loss frequency	1 MHz
Dielectric Strain constants	**EpsX = EpsY**	916
	EpsZ	830

Table 4.1: PZT 5A Navy II material properties

4.2.2. Actuation

In order to actuate the signal, a potential difference between the upper and lower surfaces of the actuator disc is applied: the desired actuation signal modulates the voltage amplitude on the upper surface (10 Volt in this study) while the lower surface remains grounded, hence it possesses zero voltage.

According to the central frequency chosen for the Hanning windowed tone burst shown in figure 4.2, only the two *fundamental modes* (i.e. S_0 and A_0) were excited.

Throughout the years, there has been a great debate on which of these two fundamental modes is actually more adapt to highlight sensitive defect-related features.

In [5], [40], A_0 is reported being more sensitive to smaller damages (even if the mesh size must be inevitably refined) because of its lower wavelength compared to an S_0 mode at the same frequency; moreover, its out-of-plane displacement allows this mode to better detect surface defects with respect to S_0. However, in the latter paper it is also stated that the anti-symmetrical mode exhibits more dispersion and energy leakage due to its substantial and characteristics out-of-plane vibrations while, besides exhibiting lower dispersion, the S_0 mode possess also a lower attenuation. In contrast to A_0, the S_0 mode is simpler to detect and the stresses are almost uniform throughout the thickness of the plate. This explains why it is sometimes employed when studying the interactions with through-thickness cracks.

It is also reported that the S_0 mode can provide much better reflections than the A_0, thus resulting in a stronger signal for effective post-processing [40].

In [9] it is said that being S_0 faster, it is consequently more distinguishable (considering boundary reflections and other disturbing phenomena). This enables a more clear extraction of the desired data from the received signal since the temporal window is easier to define. The drawback is that enough space should be present in order for the two modes to separate thanks to their different group velocities: if the travelled distance is increased, the S_0 mode becomes more separated from the A_0 mode, and this reduces the inaccuracy of the subsequent calculations.

Considering fixed the plate’s thickness, it is important to state that up to 300-400 kHz the S_0 mode possesses a smaller amplitude with respect to A_0 while, as the frequency goes up, the S_0 amplitude overcomes the other mode. This is to say that, if the actuation is made with a piezoelectric actuator mounted on one of the plate surfaces (as actually done in this thesis in order to study the interaction of both modes with the defect), the two modes are contemporary present but may possess very different amplitudes.

The choice of staying below 1 MHz with the actuation frequency was also due to the purpose of not exciting higher order modes that may complicate the propagation and, consequently, the received signal, as mentioned in section 4.1. This value was taken from the dispersion curves calculated according to the plate's material and thickness considered also in this thesis, and reported in figure 4.3 [13].

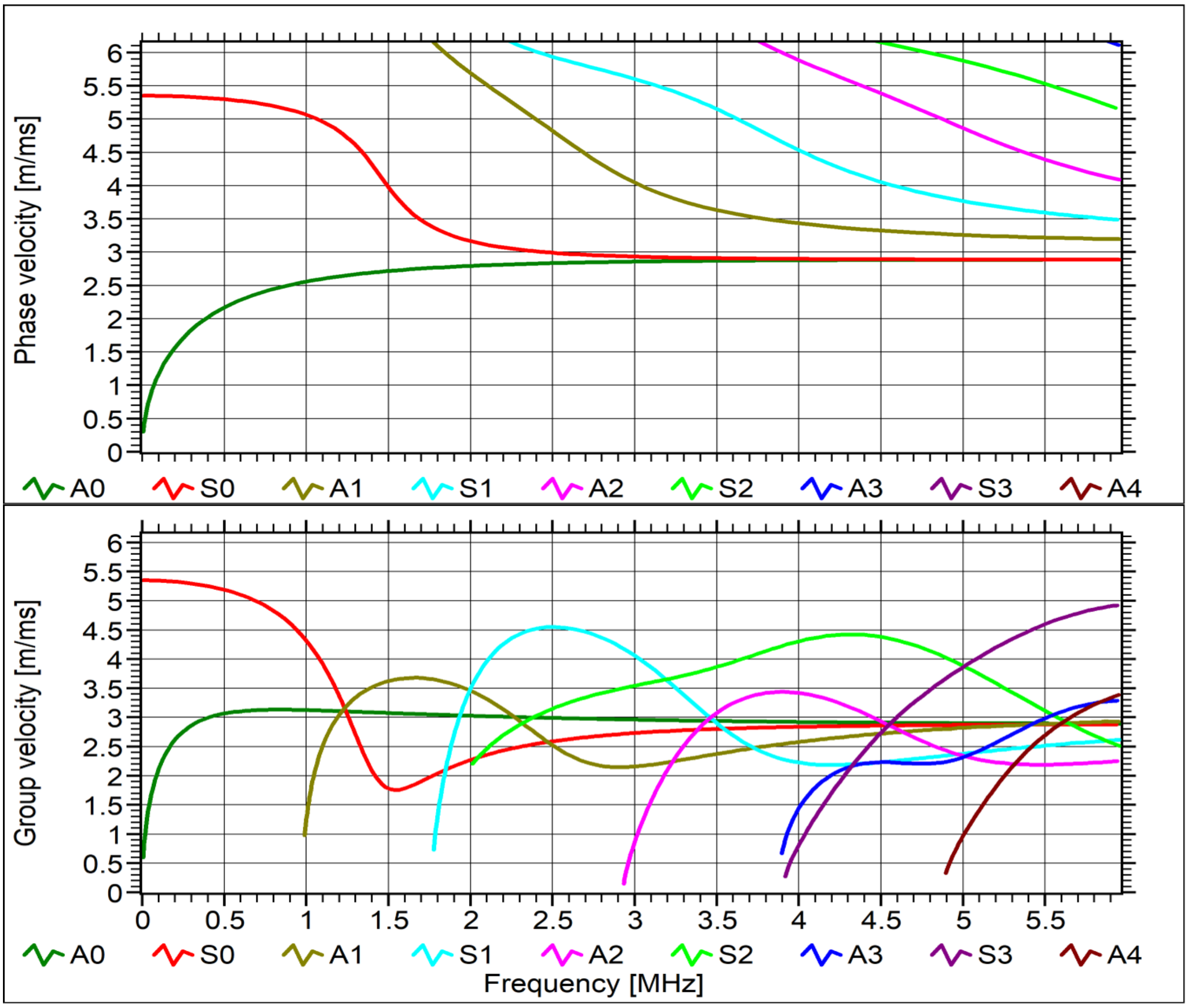


Figure 4.3: Dispersion curves provided by the Vallen Systeme software for the 1.6mm A7075-T651 plate

Continuing the discussion about the actuation of the 600 kHz Hanning window as voltage function onto the piezoelectric disc's top surface, an interesting phenomena was encountered: even if the employed input signal had a null value from time ≃ $0.86*10^{-5}$ [sec] on, this was not reflected on the last part of the generated function. Conversely, the latter showed some further abnormal and non-negligible oscillations before reaching asymptotically the zero voltage value. This phenomena is clearly visible in the following figure 4.4, where the arrow indicates the theoretical point in which the signal should have become null.

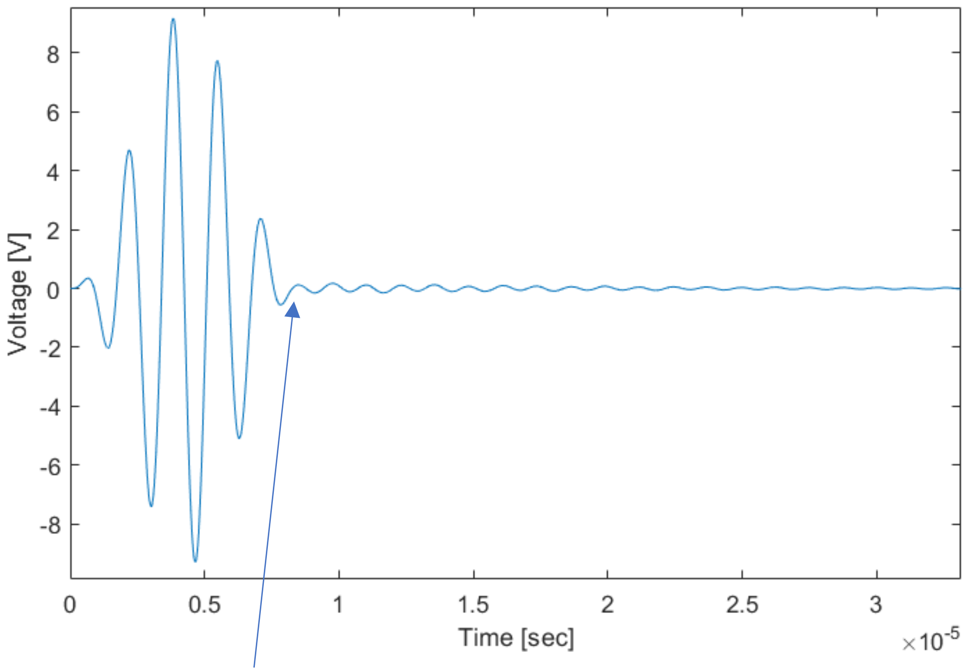


Figure 4.4: Oscillations at the end of the actuated 600 kHz Hanning window signal

These oscillations were actually introducing unwanted wave packets in the plate, thus complicating the received signals.

Based on the knowledge acquired up to this point, the author thought that the problem was connected to either the mesh quality of the disc or the chosen material. Both of them were changed accordingly but the results kept on showing some oscillations, of varying amplitude, at the end of the actuated signal.

The last remaining setting was linked to the value of the resistor applied, as a circuit, in series with the actuating function onto the upper surface of the disc.

The resistance's value is actually related to the impedance of the piezoelectric material that is in turn connected to the frequency. Some experimental studies performed at the National Research Council (NRC) of Canada by the Co-Supervisor of this thesis showed that the mean resistance value is around 50 ohm, therefore this was the value exploited in the definition of the aforementioned resistor.

Theoretically, as reported on the OnScale^{TM} website, the resistor is there to help smoothing out the voltage when the actuating signal is ending, however this time it was behaving exactly in the opposite way. Indeed, after its complete removal, the aforementioned problem did not occur anymore.

The reasons behind this issue are not clear, hence it is impossible for the author to state why this happened. It is just possible to observe that the resistor's removal fixed the inconvenience.

Not considering this resistance is actually meaning a complete absence of impedance associated to the piezoelectric material. This is not realistic, even if also within the Abaqus CAE^{TM} environment is not possible to define such circuits.
In the final model, the resistors were eventually employed only on the three sensors.

4.2.3. Detection

In order to exploit the piezoelectric disc as a sensor, the voltage induced by the material strain on the upper surface is requested as output by the command “open voltage”. The elements’ displacement should indeed be proportional to the voltage itself.
Many different outputs could be requested besides the former, e.g. acoustic pressure and displacements in the three directions. Since it is not possible to compare voltage with the other two quantities because of their different nature and associated properties, the author decided to stick to voltage outputs, which is also the quantity varying more slowly.

A further sensing location can be added employing the actuator disc also as a sensor (thus working in a pulse-echo scenario). At first glance, from a technical and functional point of view it does not seem possible because the user is actually requesting the same device to contemporary behave according to two different properties however, due to the easily accessible OnScale^{TM} symbolic language, this option results viable.
To perform this operation, it is indeed necessary to split the very same simulation in two time sections: the first one (from the beginning to the end of the actuated signal) where the voltage is modulated by the assigned input function, and the subsequent one (going to end of the simulation time) where the upper surface’s options are manually changed to “open” to receive any signal, as done for the setting of every sensor working in a pitch-catch configuration.

The coding lines necessary to perform this operation are posted here below:

```
c User defined runtime for the model (input part)
symb #get { step } timestep
symb ttime = … value representing the end of the actuation phase …
symb nexec = $ttime / $step
symb nloops = … as many as desired …
symb nexec2 = $nexec / $nloops

… here there is the plotting set-up function
   and the running plot procedure …
```

```
c Run the model for the above specified amount of time, then wait
proc plot $nloops

c CHANGE THE TOP ACTUATOR ELECTRODE TO "OPEN" TO RECEIVE THE SIGNAL AND ASSOCIATE THE RESISTOR CIRCUIT:
piez conn load_1 circuit_1 open

c User defined runtime for the model (listening part)
symb ttime = ... value representing (total sim time - end of the actuation phase) ...
symb nexec = $ttime / $step
symb nloops ... as many as desired ...
symb nexec2 = $nexec / $nloops

c Run model
proc plot $nloops

term            /* End of the running phase
```

These lines should be added to the final part of the symbolic OnScale^{TM} code (dedicated to the running of the model) before the submission for the analysis. There is no way of doing this operation during the graphical aided construction of the model.

4.3. Abaqus CAE^{TM} vs OnScale^{TM}

Abaqus CAE^{TM} is a finite element software very spread and well known in the engineering industries since it can fulfil an enormous number of different numerical analysis: static, dynamic, linear, non-linear, thermal, etc.

Moreover, it has a user friendly graphical user interface (GUI) allowing for a complete personalization of the model: the geometry design, the different meshing options, the constraints, the definition of the simulation’s steps, are all features that can be accurately modelled inside this sole software exploiting its GUI.

OnScale^{TM} (previously called PZFlex) is instead a software that was born in symbolic language and that was updated with a graphical user interface only in the recent years, therefore it is not able to fulfil the degree of personalization the former software has. This is particularly observable in the mesh construction tool that, compared to the other package, allows only much coarser definitions, and where the refinement can only be achieved by working on the script input file. This topic is comprehensively addressed in the next section 4.3.1.

Moreover, this latter package is made specifically for acoustic and electric simulations: it can efficiently simulate wave propagation phenomena, together with *active* excitation mechanisms. Unfortunately, here it is not possible to resolve the crack growth phenomena in order to find the associated parameters, as instead feasible in the former software.

OnScale^{TM} is able to run either on the local machine on which is installed and licensed or on the Cloud, thus exploiting the powerful servers available and paying only for the core hours required by the simulations. Also, it allows the user to parameterize some variables and the software will then sweep through the assigned range in order to perform parallel simulations on the same model while varying the inputs.

The fact that the latter package is more adapt for being exploited in acoustic simulations can be seen in some of its settings.

As an example, superposition of the reflections coming from the boundaries and the main signals may invalidate what is received by the sensors. They make impossible distinguishing between the different incoming modes, especially their final part, and reduce a lot the analysis capabilities.

In experimental studies, the researchers have to deal with such reflections exploiting proper time windows by which analysing the signal. However, in FE simulations there are two ways of reducing their effect:

- *Enlarging the geometry*, so that the waves take more time to travel back;
- Employ the so-called *“non-reflective boundaries”.*

The first method increases a lot the computational time since enlarging the geometry means creating also more mesh elements. This is the reason why the second option was born and is usually employed [57].

In OnScale^{TM}, the non-reflective boundaries are set just by selecting *“absorbing”* in the option window referring to the domain boundaries, and the only requirement to make them working properly consists in the edges being at least three wavelengths away from the source.

On the other hand, the same operation in Abaqus CAE^{TM} would have required a precise modelling of an area around the test piece by exploiting, for example, the so-called Stiffness Reduction Method (SRM) shown in figure 4.5. In [13] it was proved that this is indeed the best method to reduce the boundary reflections when dealing with GLW.

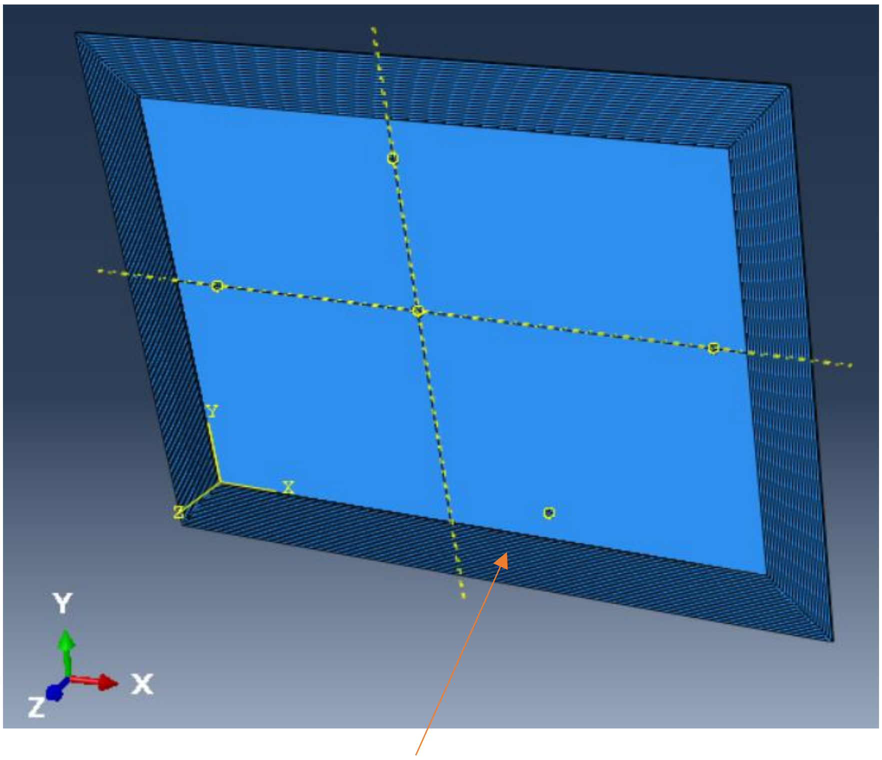
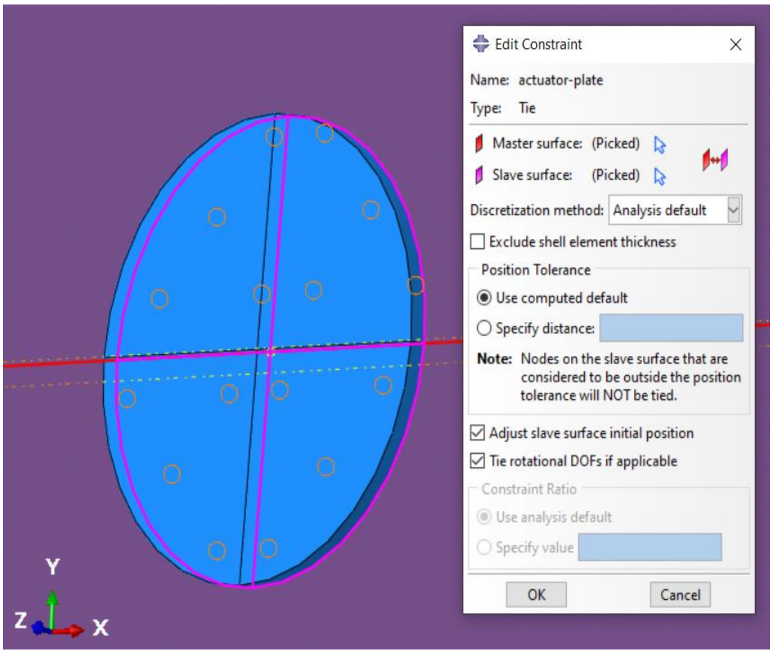


Figure 4.5: SRM subdivided area around the test piece

Another positive feature OnScale^{TM} possesses consist in automatically recognizing the interface between the sensor's bottom surface and the underneath geometry, so that the user has just to set the mesh size capable of broadly reproducing the sensor's shape. In Abaqus CAE^{TM}, the interface should instead be modelled by a so-called *"tie constraint"* between the two surfaces, as shown in figure 4.6.


Edit Constraint

Name: actuator-plate

Type: Tie

Master surface: (Picked)

Slave surface: (Picked)

Discretization method: Analysis default

☐ Exclude shell element thickness

Position Tolerance

● Use computed default

○ Specify distance:

Note: Nodes on the slave surface that are considered to be outside the position tolerance will NOT be tied.

☑ Adjust slave surface initial position

☑ Tie rotational DOFs if applicable

Constraint Ratio

● Use analysis default

○ Specify value

Figure 4.6: Tie Constraint between plate and actuator

The main difference between the two mentioned software lies, however, in the *solver* exploited to resolve problems involving time variations, i.e. dynamic ones.

Abaqus CAE^{TM} is able to solve them in two distinct ways: explicit or implicit. The first one is much faster than the other because it does not require to invert matrices. However, when dealing with piezoelectric elements, the implicit solver must be necessarily employed for the whole model: this instead requires matrix inversion operations and, due to this reason, it comes out much slower. Increasing the working frequency slows accordingly the resolution process even more because of the smaller time step required for accuracy purposes: in the case of an actuated 200 kHz simulation, the software took around thirtysix hours to simulate the entire wave propagation along the plate.

On the other hand, what makes OnScale^{TM} much faster than Abaqus CAE^{TM} is the exploitation of a hybrid solver allowing the contemporary use of explicit methods to resolve the wave propagation along geometries not made of piezoelectric materials (the aluminium plate in this study), and of implicit methods just to resolve the so-called "Electric Window" made of the elements constituting the piezoelectric material disc. The same mesh and model exploited for the previous Abaqus CAE^{TM} simulation were created also on OnScale^{TM} and the registered running time, on the same machine[1], dropped to around four hours giving as output very comparable results, as the reader may appreciate in the following figure showing a comparison between two S_0 modes captured by one sensor.

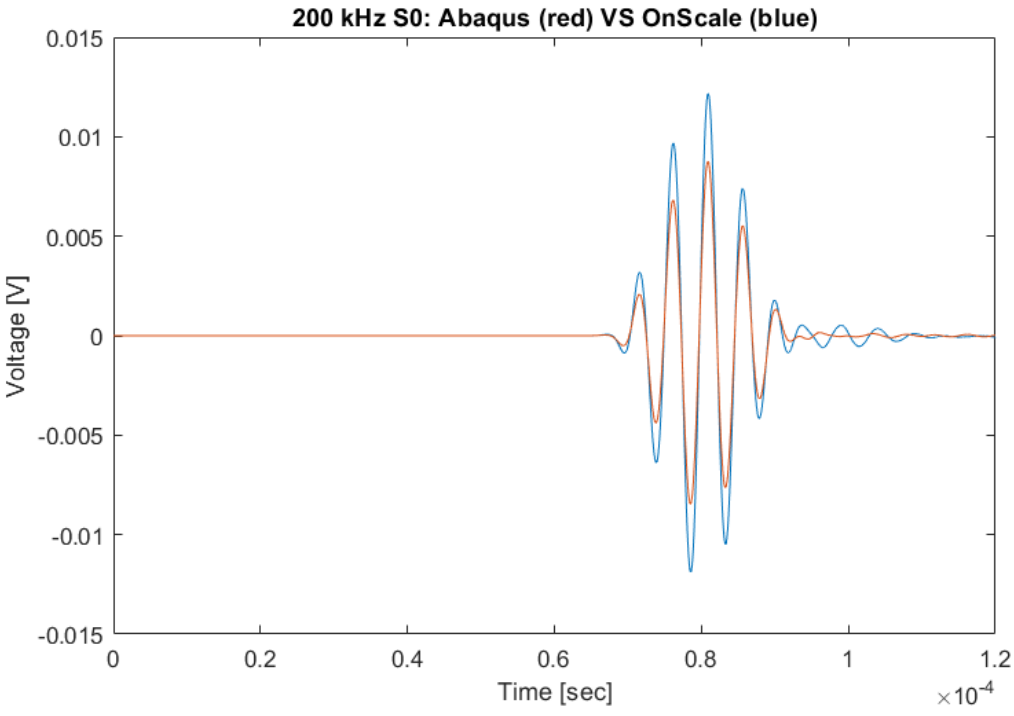


Figure 4.7: 200 kHz S_0 comparison between Abaqus CAE and OnScale

Both the signals presented in figure 4.7 are in phase, the only difference comes with the mode's amplitude. However, this can be straightforwardly explained with the different piezoelectric material involved in the Abaqus CAE^{TM} simulations (i.e. a Navy type I material).

To prove the fact that the implicit method is the one rising the resolution time, it was observed that the simulation time of a mesh refined plate was not substantially different with respect to the one of a coarser plate, while refining the piezoelectric discs required much more time to complete the simulation having the same excitation frequency. In the final model, the mesh elements through the thickness of the discs were eventually set to two, as a good compromise between resolution speed and accuracy.

In conclusion, on the available local computer platform[1], Abaqus CAE^{TM} could be exploited for simulations involving frequencies of maximum 200 kHz, since it runs only implicit solvers. Just one of the required simulations at 600 kHz would have needed even weeks to run, whereas exploiting OnScale^{TM} the resolution time was in the order of one day.

From what it is stated in this section, the reader can evaluate him/herself why the choice of OnScale^{TM} was almost mandatory, and how advantageous it is in solving this particular kind of problems from a time saving perspective. However, creating the correct mesh on this new software represented a great challenge.

4.3.1. Mesh construction and evaluations

OnScale^{TM} mainly uses 2D and 3D *Cartesian* elements. The standard elements belonging to the first category are Quadrilateral and possess four nodes, while the elements for the second category are Hexahedral possessing eight nodes. However, other types of elements such as Shell, Bar, Tetrahedral (for unstructured mesh), etc. are also available [70].

This software exploits mainly a *Structured* mesh: this means that the grid actually resembles the structure of a graph paper. Using the so-called *keypoints*, the user is able to define axis along the main directions and to select the least number of elements between each of the axis (layers) previously defined, thus allowing the grid to assume an irregular distribution, as shown in the following figure.

---

[1] All the numerical simulations mentioned in this thesis were performed on a local machine at Clarkson University, remotely controlled by the author from Bologna. Characteristics: Windows 10 workstation with 2 Intel(R) Xeon(R) CPU E5-2620 v3 (24 cores) @ 2.40 GHz, 128 Gb of RAM and Nvidia Quadro M4000 GPU.

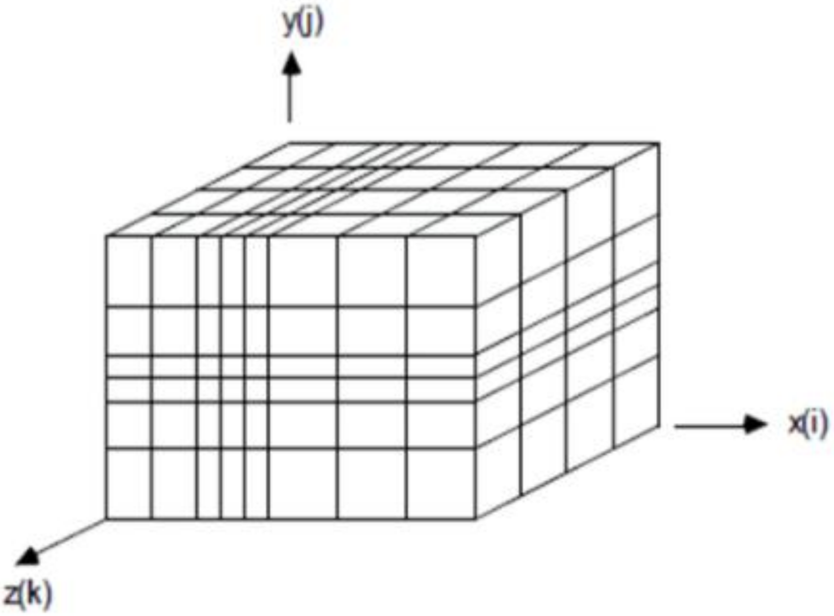


Figure 4.8: Example of a standard mesh with the use of keypoints [70]

This grid is able to efficiently solve small deformation models and is well recommended for wave propagation problems.
The keypoints are also mandatory to define the already mentioned "Electric Window" region, that should be minimized as much as possible.

Despite these advantages, OnScale^{TM} also provides *Skewed* (still structured) and *Non-structured* grids. Figure 4.9 resumes the three mesh categories:

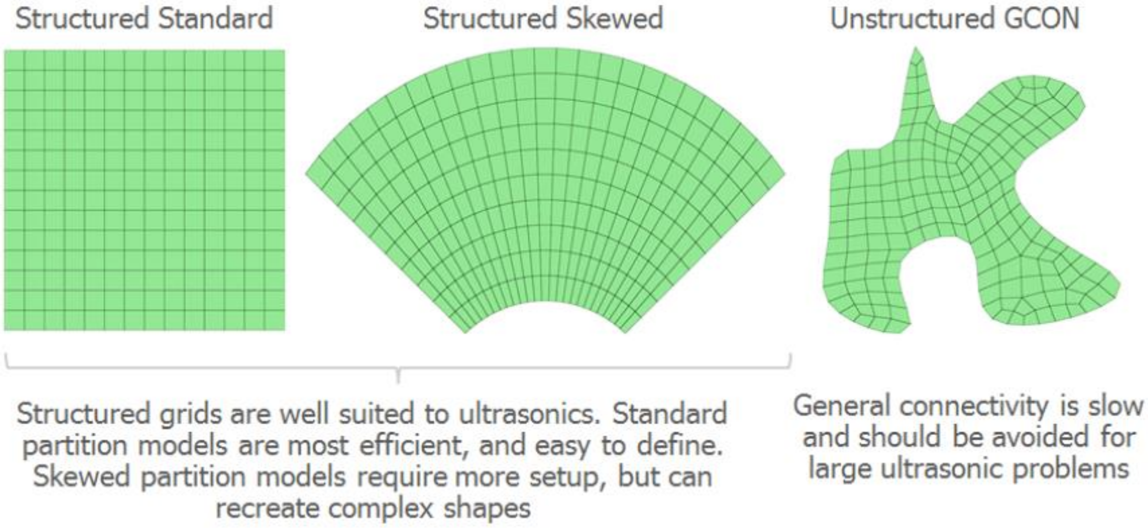
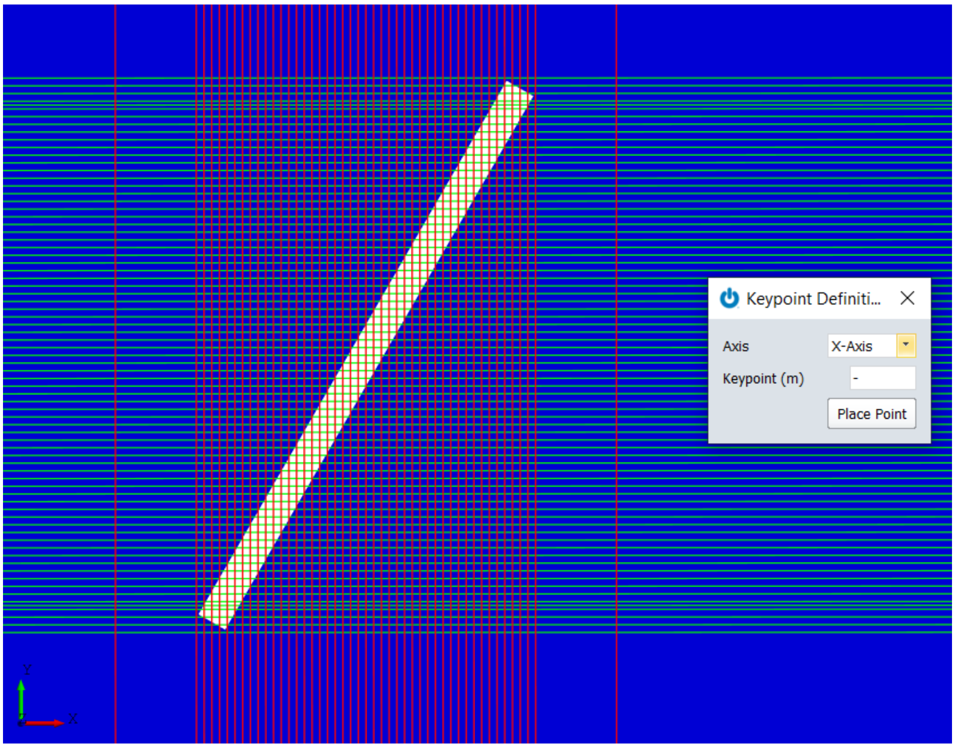


Figure 4.9: 2D example of the three mesh categories [70]

The biggest advantage of the unstructured grids is the ability to take shape around any geometry, while the structured grids does not possess this property and leads to the stair stepping of the angled and curved geometries. If this conformation functionality is not required, an unstructured mesh is not necessary, therefore the computational time and memory requirement can be reduced exploiting the usual structured grid [71].

The refinement exploiting *keypoints* is useful and interesting for those models involving thin layers within their geometry. Because of that, some elements of the cartesian grid may fall between these layers due to a choice, for the overall mesh size, which is not adequate with respect to the thickness of the layers [71]. The drawback of this method is that the refinement is performed for the entire plane along that particular axis.



Axis X-Axis

Keypoint (m) -

Figure 4.10: Example of a model with keypoints along x and y axis

Some simulations were performed with the configuration visible in figure 4.10 and considering the input function presented in figure 4.2, however the generated mesh led to an abnormal and undesired reduction (damping) of the A_0 mode along the mesh refining direction as the simulation time increased. The circle in figure 4.11 highlights this phenomena.

This undesired reduction of the anti-symmetric mode also compromised its interaction with the discontinuity, leading to a consequently even more damped scattered signal.

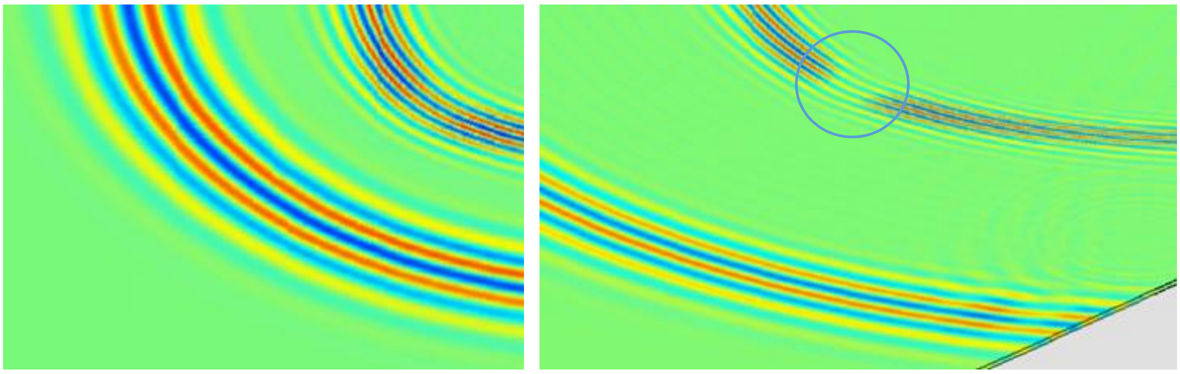


Figure 4.11: A_0 abnormal damping as the sim goes on

It was found that this phenomena was due to the stretching of the mesh elements along the larger propagation direction just because of the keypoints' configuration. This resulted in a considerable enlargement of the *mesh aspect ratio* [2] which apparently affected more the wave mode with the shortest wavelength: this is reasonable since that mode is the more critical in terms of elements' size. As a further confirmation, in mechanical problems the aspect ratio should indeed be as close as possible to 1 (ideal value) in order for the simulations to give reasonable results. However, this is not valid in, for example, heat transfer problems, where the mesh aspect ratio can be higher.

Moreover, since the discontinuity is the only part that has to be refined in order to resemble its geometry as close as possible, the method reported in figure 4.10 may not be the appropriate one as it drastically increases the solving time. This is due to the smaller time step required by the new mesh elements, as it is better explained in section 4.4.2.
One idea could be to *refine* the grid elements in a small box around the defect.

OnScale^{TM} does have a refined grid method allowing to merge fine and coarse grids together (as can be seen in the next figure 4.12), however the setup process is way more complex with respect to Abaqus CAE^{TM}. Indeed, this task can be accomplished only within the script file and not through the graphical interface.
Moreover, this mesh refinement can be fulfilled only when the geometry is designed directly inside OnScale^{TM} (i.e. "primitive") and not when it is imported, like in this study.

---

[2] The mesh aspect ratio is defined as the ratio between the elements' length along each of the main axis: x and y in a 2D case, x, y and z in a 3D scenario. The unity value represents the perfect, ideal case.
In order to get comparable results between different simulations, the mesh elements' size should remain almost constant.

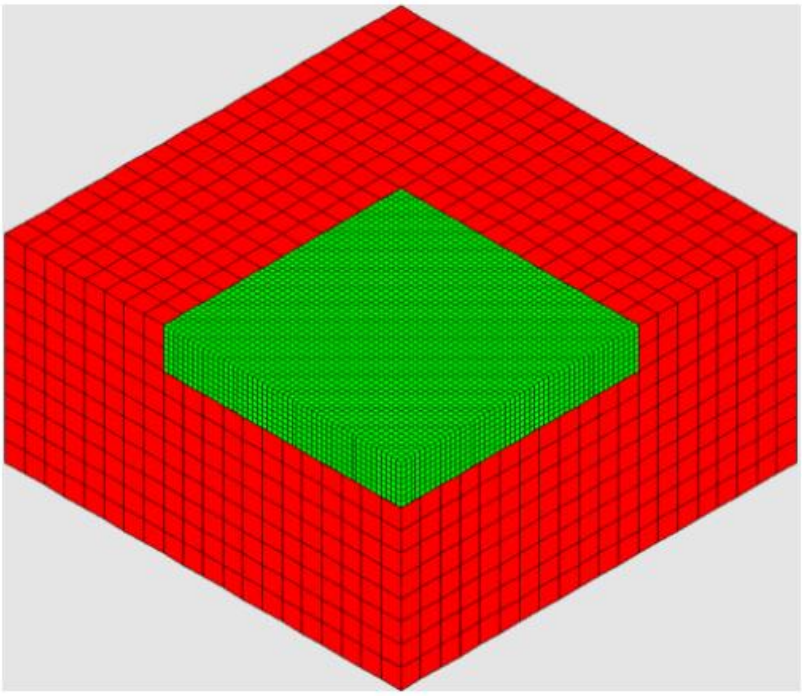


Figure 4.12: Example of a 3D refined structured mesh [71]

The solution is to eventually construct a so-called *hybrid mesh.*

As previously mentioned, Abaqus CAE^{TM} offers the possibility of constructing a hybrid mesh through its simple graphical user interface.
However, when dealing with holes, the software does not allow to exploit a structured mesh on the entire domain; therefore, the best meshing method consists in creating a partition between the external geometry and the region around the hole itself, and to exploit a structured grid for the outer part and an unstructured mesh (called *Sweep* inside this software) for the inner.

Since in this thesis there is a discontinuity positioned at the plate's centre, the author decided to exploit a hybrid mesh made of a structured grid on the majority of the plate and an unstructured mesh (with tetrahedral elements) only in a 11 x 11 mm through-thickness box around the defect, in order to reproduce in OnScale^{TM} what would have been done inside Abaqus CAE^{TM}. The purpose was indeed to ensure a precise matching between the grid and the discontinuity's contours, thus to avoid the stair stepping representation of the latter otherwise present when using a structured mesh.
This feature of combining structured and unstructured grids within a hybrid mesh is only allowed for CAD geometries, hence not for geometries directly drawn inside the OnScale^{TM} designer platform. However, this is not an issue since the geometries employed in this thesis were designed in Solidworks^{TM} and imported as *.step* files inside the OnScale^{TM} environment.

Eventually, these two meshes must be glued together to allow the transmission of the propagating wave from one to the other. Similarly to the “tie constraint” function available in Abaqus CAE^{TM}, OnScale^{TM} provides the *“glue”* function which basically fulfils the same purpose. The next figure shows the hybrid mesh and the glued surfaces of a final model:

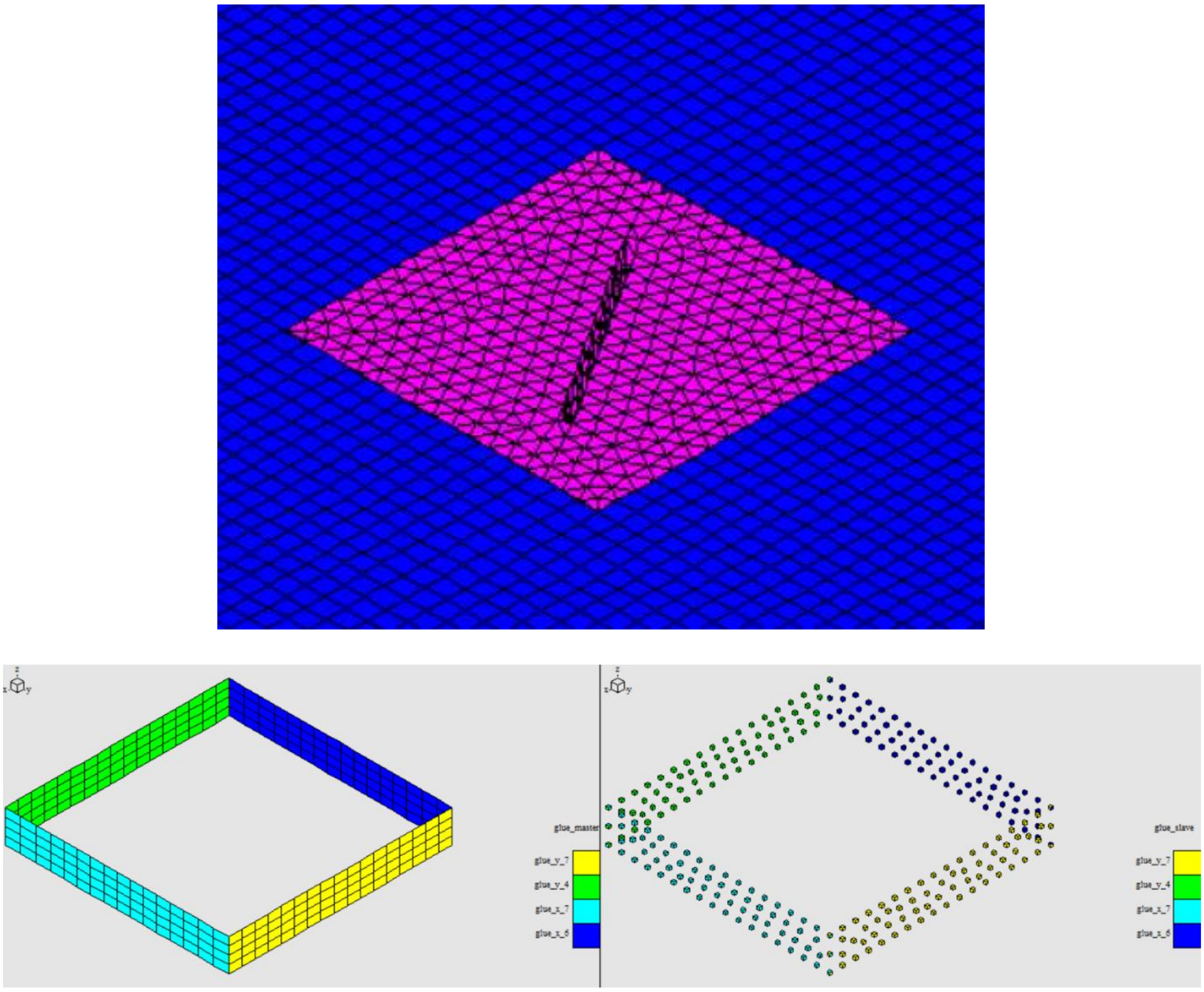


Figure 4.13: Hybrid mesh and glued surfaces of the model

From the upper picture it is clearly visible the perfect matching, as intended, between the unstructured grid and the discontinuity. The bottom picture highlights, on the other hand, the master (outer) surfaces and slave (inner) elements that have to be glued together in order for the wave to propagate from the blue (structured) into the pink (unstructured) section and vice versa.

The mesh refinement localized only within the 11 x 11 mm through-thickness box around the discontinuity allowed to avoid any change in the aspect ratio of the structured mesh and to guarantee a correct interaction with the defect, since also the unstructured mesh aspect ratio was set close to the unity.

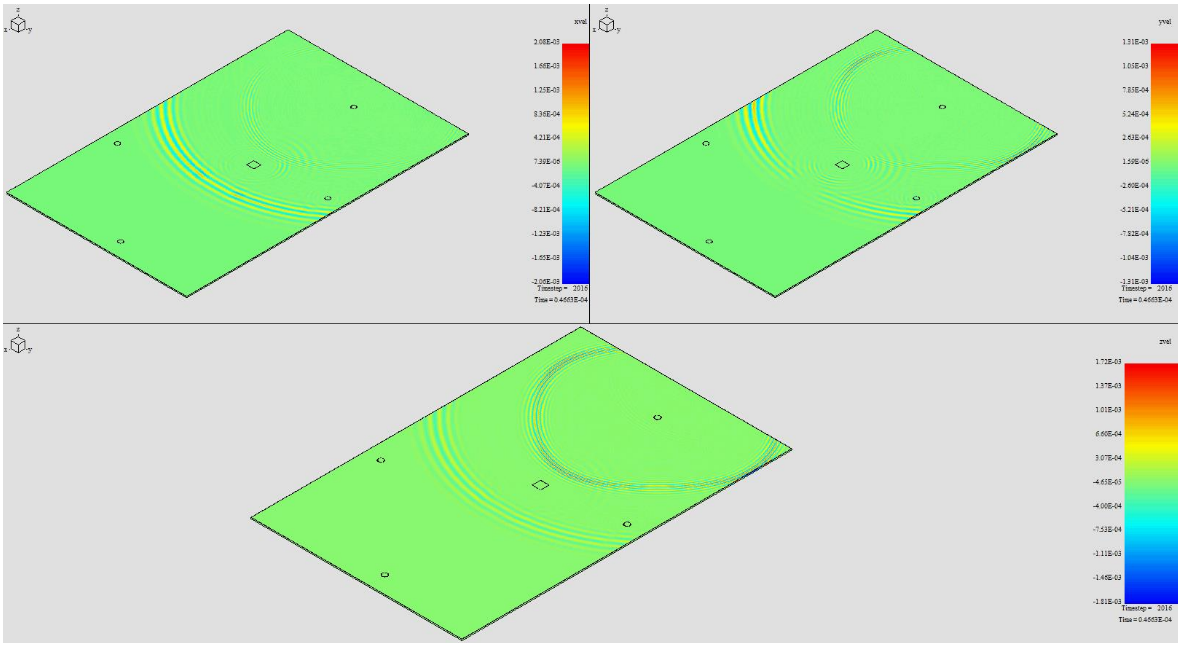


Figure 4.14: A_0 normal propagation (x, y and z velocity graphs)

As can be seen from figure 4.14, the new hybrid mesh configurations solved the A_0 propagating issue. Unfortunately, when dealing with unstructured grids, OnScale^{TM} has a plotting bug that does not allow to visualize the acoustic pressure as output (like it is done in figure 4.11), hence the velocities along the three orthogonal directions have been displayed.

However, even with this layout, it is anyway visible the lack of any abnormal damping along each propagating direction.

The author would like to thank the OnScale Support Team, since they provided a valuable help in the creation and validation of the hybrid mesh grid (which is a quite recent and still developing feature added to their package).

4.4. Model Description

In this section, the whole OnScale^{TM} modelling of the aluminium plate comprehensive of geometry, material characteristics, loads and boundary conditions is finally described. Moreover, the damage's geometry is introduced and reasonably explained, considerations on the mesh elements' size and time steps are presented and, eventually, the signal attenuation phenomena is briefly discussed.

4.4.1. Geometry, material, loads and boundary conditions

As mentioned at the beginning of this study, the reference plate used in these simulations has a rectangular geometry with dimensions taken from the one available at the HolSIP Laboratory at Clarkson University and equal to 609.6 x 304.8 x 1.6 mm. The actuator and sensors' positions reported in the methodology section 3.1 are indeed referred to these measures and to the origin displayed in figure 4.15.

However, due to the computational time-related benefits of lowering the number of grid elements and the employment of the absorbing boundaries along the x and y edges (meaning that such a large outer space around the transducer was no longer required), the actual dimensions were shrunk to 410 x 280 x 1.6 mm. This allowed to fulfil the primary objective of maintaining the overall devices' configuration exploiting the provided the dispersions curves (since they are related to the thickness), while saving as much elements as possible and keeping the required three wavelength distance between the source and the boundaries.

The former and current plate geometries are reported here below:

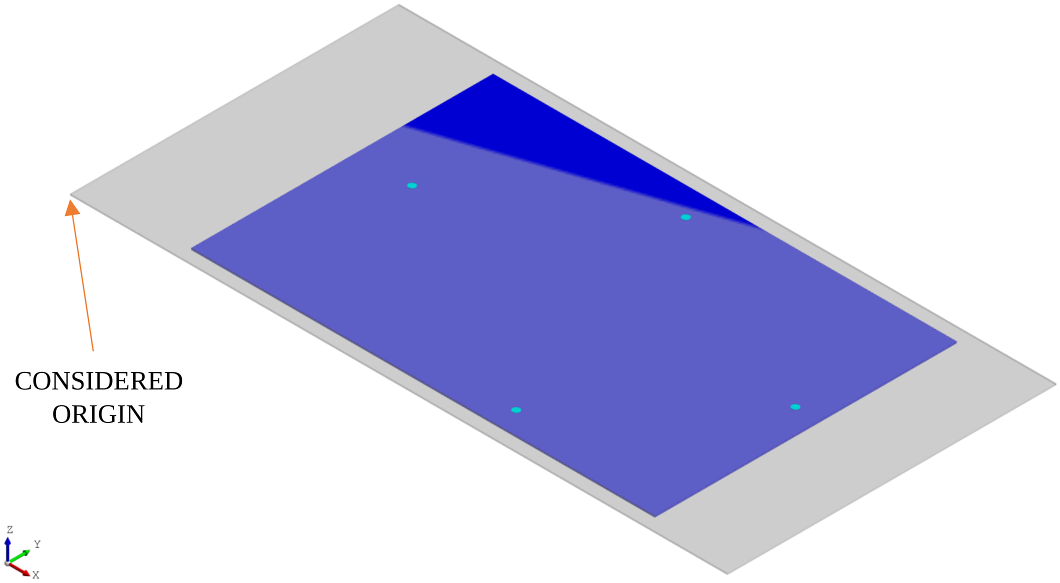


Figure 4.15: Plate geometries

The plate is made by aluminium 7075-T651 with the following homogeneous properties:

- Density ρ: 2810 kg/m^3
- Modulus of elasticity (Young modulus) E: 71.7 *10^9 Pa
- Poisson ratio μ: 0.33
- Bulk modulus K: 70.3 *10^9 Pa
- Shear modulus G: 26.9 *10^9 Pa

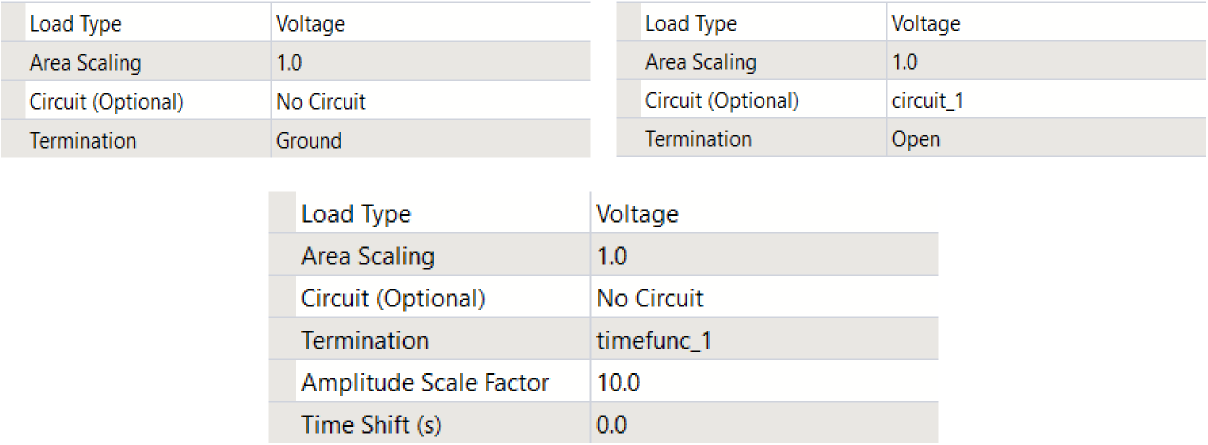
The last two parameters should be inserted into the moduli table of the OnScale^{TM} material window, and they were calculated from the previous exploiting the following equations:

$$K = \frac{E}{3(1-2\mu)} \quad (4.3) \qquad\qquad G = \frac{E}{2(1+\mu)} \quad (4.4)$$

Alternatively, it is also possible to define the longitudinal and shear velocities for the propagating bulk waves instead of the moduli. This can be done exploiting equations (4.5) and (4.6), and the results are respectively equal to 6148 and 3097 m/sec:

$$C_L = \sqrt{\frac{E(1-\mu)}{\rho(1+\mu)(1-2\mu)}} \quad (4.5) \qquad\qquad C_T = \sqrt{\frac{E}{2\rho(1+\mu)}} = \sqrt{\frac{G}{\rho}} \quad (4.6)$$

Regarding the voltage loads, as mentioned at the beginning of section 4.2.2, all the lower surfaces of the transducers were set to "ground", while the upper surfaces of the sensors were set to "open" and the actuator's one was set accordingly to the input time function (figure 4.2) with an amplitude of 10 volts.



Load Type	Voltage
Area Scaling	1.0
Circuit (Optional)	No Circuit
Termination	Ground

Load Type	Voltage
Area Scaling	1.0
Circuit (Optional)	circuit_1
Termination	Open

Load Type	Voltage
Area Scaling	1.0
Circuit (Optional)	No Circuit
Termination	timefunc_1
Amplitude Scale Factor	10.0
Time Shift (s)	0.0

Figure 4.16: Assigned loads

Eventually, the discontinuity was modelled with a rectangular through-thickness cut centred at the former plate's coordinates 304.8 x 152.4 mm. Its length and orientation varied according to table 3.2, and the constant width was set to 0.4 mm. This latter choice was made since the author did not want to study the influence of this parameter, as the main focus is on the length.

4.4.2. Mesh size and time steps

Most FE simulations require fine meshes in order to accurately resolve details within the geometry, edge effects and material interfaces. This means that a trade off between accuracy and resolution time is always involved in the decision of this first parameter. However, in order to ensure the numerical stability during the resolution, the time step must be chosen smaller than the time required for the fastest wave to propagate across the smallest element.

As the reader may appreciate, these two parameters are very correlated and, especially when dealing with explicit solvers, the second one is of paramount importance since the stability and convergence of each model is well linked to the chosen time step.

Indeed, the implementation of an explicit scheme is *"conditionally"* stable. This means that the time step Δt has to be smaller than a critical time step.

According to the CFL criterion there should be at least twenty time steps, during the cycle of a wave at the highest frequency, in order to prevent a wave from completely travelling through an element [9]:

$$\Delta t \leq \Delta t_{crit} = \frac{1}{20 * f_{max}} \tag{4.7}$$

However, an approximation often employed in wave propagation problems consists in identifying the critical time step as equal to the transit time of a dilatational wave through the smallest element in the model (Moser criterion) [50]:

$$\Delta t \leq \Delta t_{crit} = \frac{\Delta L}{C_L} \tag{4.8}$$

where ΔL is the smallest element size and C_L is the longitudinal wave speed.

The smallest value between the previous two is then used as the critical time step.

However, in OnScale^{TM} the calculation of the stable time increment is performed automatically at the beginning of the explicit solving process considering the imposed mesh size, therefore the user does not have to care about this aspect.

The size of the mesh elements is also important to avoid instability issues and to guarantee reasonable spatial resolutions. It must be chosen in order to satisfy the usual condition that the wavelength divided by the elements' size is greater than ten [50].

Moreover, a minimum number of four elements through the plate thickness were required in order to avoid other anomalies in the wave propagation [13].

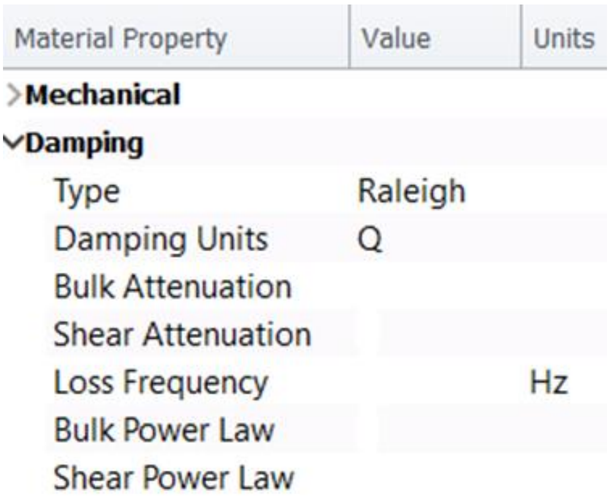
If the mesh elements' size is set accordingly to the aforementioned rule, the wave propagation would turn out being *circular*, as expected for an isotropic material. If the element size is instead too coarse, the propagation would tend to resemble a rhomboid.

4.4.3. Signal attenuation

When an acoustic wave travel through a medium, it undergoes a decrease of its intensity as the travelled distance increases. The signal's amplitude is reduced by a phenomena called *attenuation*, which is made of three different causes [72]:

- *Geometric spreading* $\frac{1}{\sqrt{r}}$, which is due to the principle of conservation of energy;
- *Material damping*, which is due to the absorption and conversion of the elastic and kinetic energy into heat. This is linked to the mechanical properties of the material constituting the structure;
- *Wave scattering*, which is due to the geometric discontinuities (e.g. grain boundaries, defects, non-metallic inclusions, etc.) and structural boundaries both reflecting the incoming waves into directions different from the original one, thus wasting some of their initial energy.

The assessment of these three phenomena is not simple since in many low dissipative materials, such as aluminium, the second one is somehow hidden behind the first [13].



Material Property	Value	Units
> **Mechanical**		
∨ **Damping**		
Type	Raleigh	
Damping Units	Q	
Bulk Attenuation		
Shear Attenuation		
Loss Frequency		Hz
Bulk Power Law		
Shear Power Law		

Figure 4.17: Required values to model the damping behaviour

Usually, a reliable value for the overall attenuation can only be achieved experimentally for the particular material being used, frequencies employed, and desired wave modes. However, due to the impossibility of performing experimental activities, the values reported in the left column of the following figure 4.17 could not be evaluated, hence the contribution related to the material damping could not be introduced in the numerical models.

Nevertheless, the lack of this parameter is not a big deal for aluminium structures; indeed, for this material, damping is not even modelled in the database available inside the OnScale^{TM} software.

This fact is especially true for GLW, and this is the reason why they are studied specifically to evaluate large structural portions in the aerospace industry (where aluminium is still the most spread material employed). In steel, for example, the material damping would be much more present and is indeed already modelled in the database.

On the other hand, the other factors contributing to the signal's attenuation are directly taken into consideration by the differential equations the software will resolve, since they are a simple consequence of the wave propagation and motion. For this reason, they do not require any further assumption before being implemented into the numerical model.

CHAPTER 5

5. SIGNAL ANALYSIS AND IMPLEMENTATION OF THE ALGORITHMS

In this chapter, the signals collected from the 600 kHz simulations are firstly pre-processed in order to get them normalized, then they are examined. Moreover, the subsequent features' choice is presented and motivated. Eventually, an entire set of Machine Learning algorithms is trained and validated.

All the operations of this chapter were performed exploiting MATLAB^{TM} scripts created by the author, built-in functions and applications available inside the package.

5.1. Signal Pre-processing

The first thing to do after the raw data collection is completed, is to *normalize* [1] the time domain amplitude signals in order to compare them getting rid of the attenuations directly modelled inside the numerical scheme (presented in section 4.4.3).

To accomplish this task, the whole signals' sets were loaded into the MATLAB^{TM} environment and each amplitude was normalized considering the S_0 maximum amplitude captured by sensors 1 and 3 (for the signals received by these sensors), and the maximum amplitude captured in the pristine plate's simulation by sensor 2 (for the signals belonging to this sensor). These choices were made since the symmetric mode possesses the larger amplitude and arrives first at sensors 1 and 3, therefore it is always present and superposable in any received signal regardless of the discontinuity.

Moreover, the signal arriving on the second sensor in the pristine plate's simulation is inevitably the largest in terms of amplitude, since it did not interfere with any defect. Indeed, all the other simulations contains a discontinuity that, albeit small, reduces the energy of the signal crossing it, thus reducing also the mode's amplitude.

In other words, the normalization was done with respect to the maximum amplitude captured by each sensor.

---

[1] In statistics, normalization can have a large range of meanings. In this study it means adjusting (i.e. scale) the values measured on different scales to a notionally common one, here between -1 and 1.

The sensors' location, reported in table 3.1, is also posted in the following figure for sake of clarity:

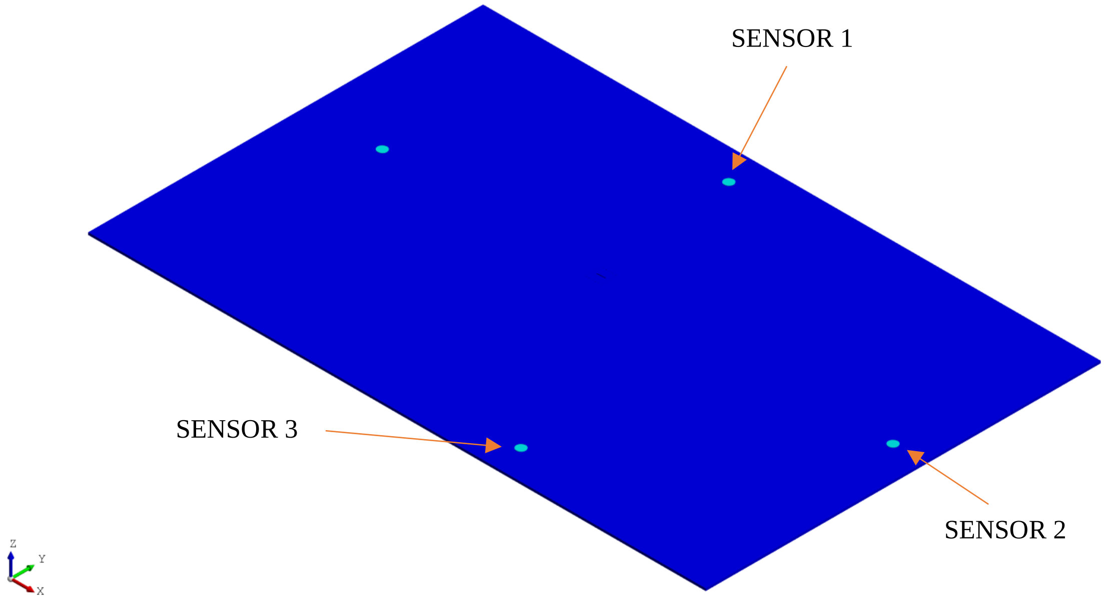


Figure 5.1: Sensors' positions on the 3D model

A *resampling* of five sets of signals was also necessary due to the different time steps automatically chosen by the OnScale^{TM} software. This was probably caused by the changing orientation of the defect between different simulations that led to a variation of some elements' size within the unstructured grid part. These sets were resampled according to the time vector of the other fifteen simulations, in order to eventually have all the signal vectors of the same size to perform any comparing operation.

The *interp1* function was exploited for this purpose, since the other built-in function applies a further filtering operation that is here unnecessary:

```
interp1(old_sample,signal_vector,new_sample,'spline');
```

5.2. Examination of the signals and interesting features

Once the signals have been pre-processed, they were examined to identify some possible features to be extracted for the subsequent analysis through the Machine Learning algorithms. However, due to the small differences between each of the discontinuities modelled in this study, the outputs resulted being very similar to each other in both the domains (time and frequency).

As an example, the correlation[2] index between two signals referring to the discontinuities 2mm_5deg and 10mm_90deg is reported in the next table, respectively for the time domain and the frequency domain[3]. The reader may evaluate him/herself the degree of similarity between these two signals, which are the polar opposites of the chosen range for the simulations.

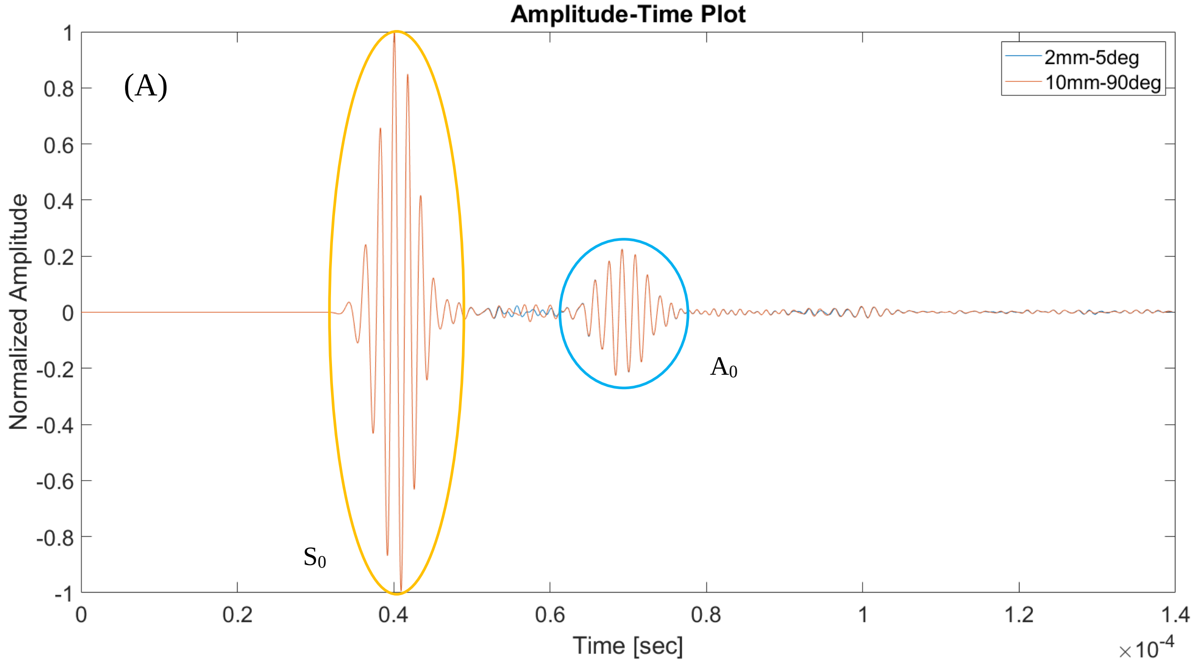
	Time domain	*Frequency domain*
Sensor 1	0.9994	0.9997
Sensor 2	0.9815	0.9903
Sensor 3	0.9995	0.9998

Table 5.1: Correlation index between 2_5 and 10_90

The similarity between these two signals can be directly seen also in the following figures representing, for each of the three sensors, the received:

- Time domain amplitudes (A);
- Frequency domain amplitudes (B);
- Time-Frequency representation (e.g. spectrogram contours[4] (C)).

Sensor 1:



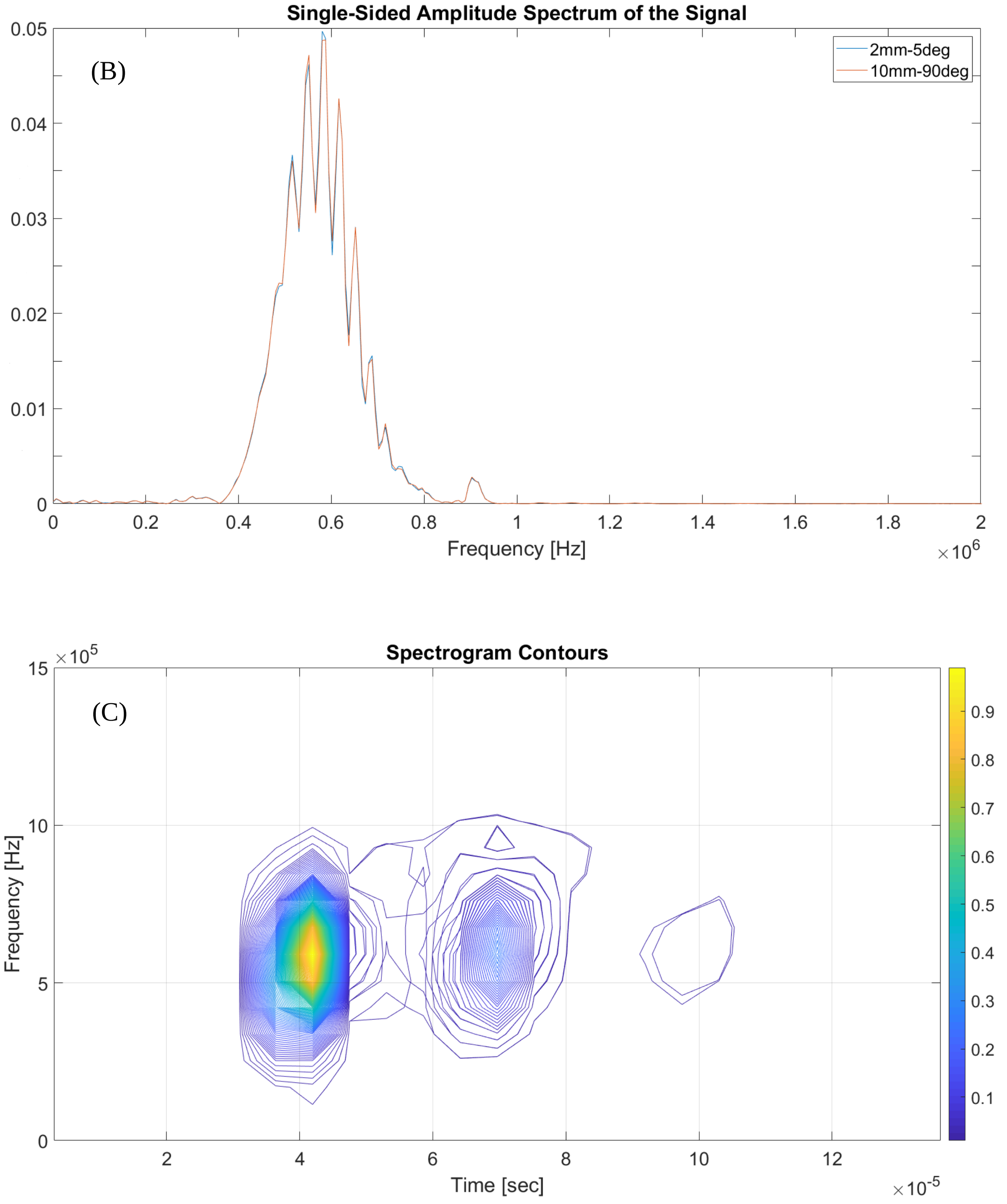


Figure 5.2: 2_5 and 10_90 at sensor 1: time domain (A), frequency domain (B) and spectrogram (C)

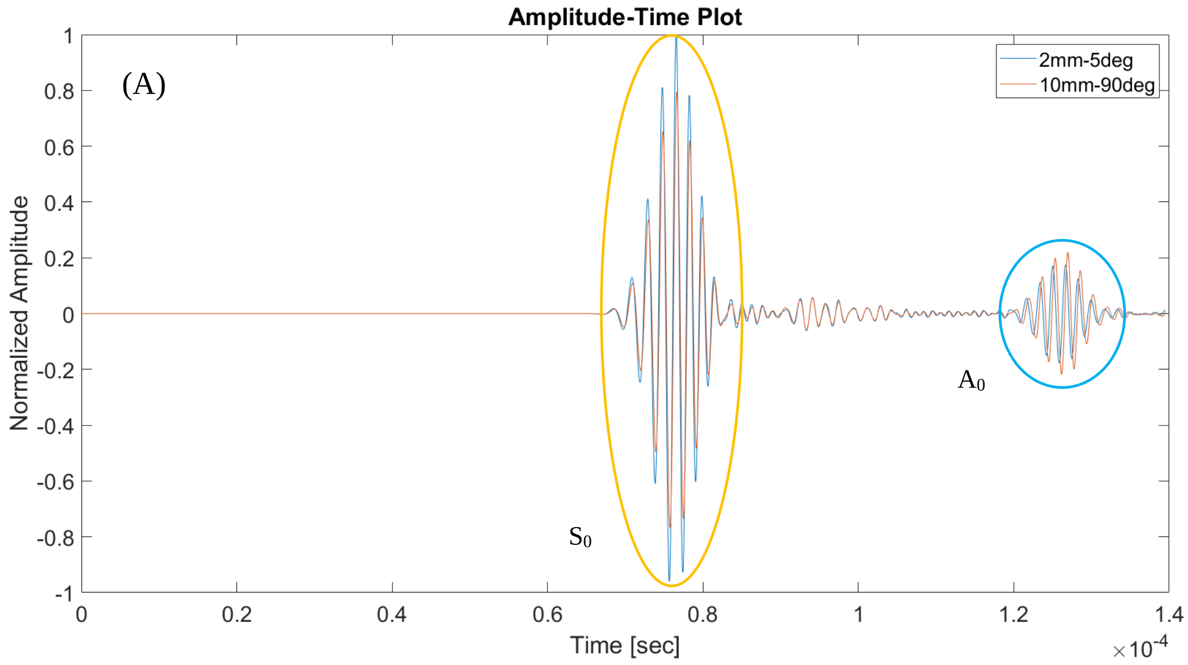
---

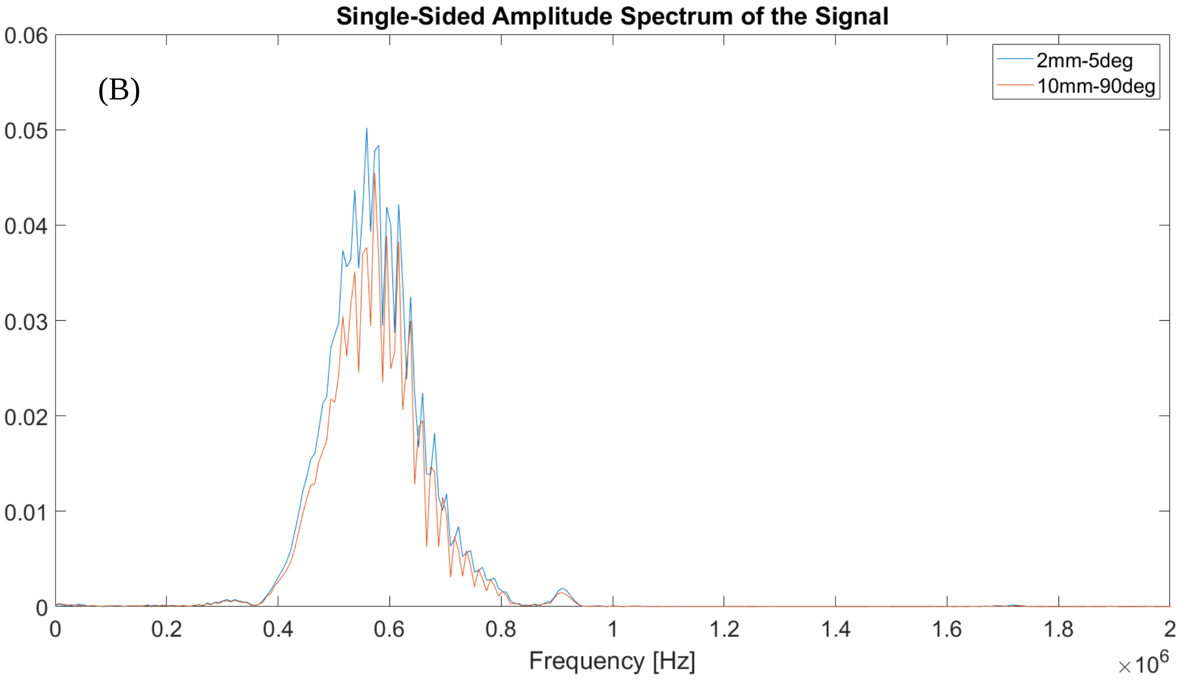
[2] The scalar representing the correlation index between two signals was derived by applying the cross-correlation function to both signals and taking its maximum. The value ranges from 0 to 1 and, the closer to the unity the more similar are the signals. The unity value corresponds to the perfect correlation, it means that the two signals are identical.

[3] The signals' amplitudes in the frequency domain were calculated applying the Fast Fourier Transform algorithm.

[4] The spectrogram was calculated with the Short Time Fourier Transform algorithm (refer to section 3.3.2.1).

Sensor 2:


Amplitude-Time Plot
(A)
2mm-5deg
10mm-90deg
Normalized Amplitude
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
Time [sec]
×10-4
S0
A0


Single-Sided Amplitude Spectrum of the Signal
(B)
2mm-5deg
10mm-90deg
0.06
0.05
0.04
0.03
0.02
0.01
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
Frequency [Hz]
×10^6

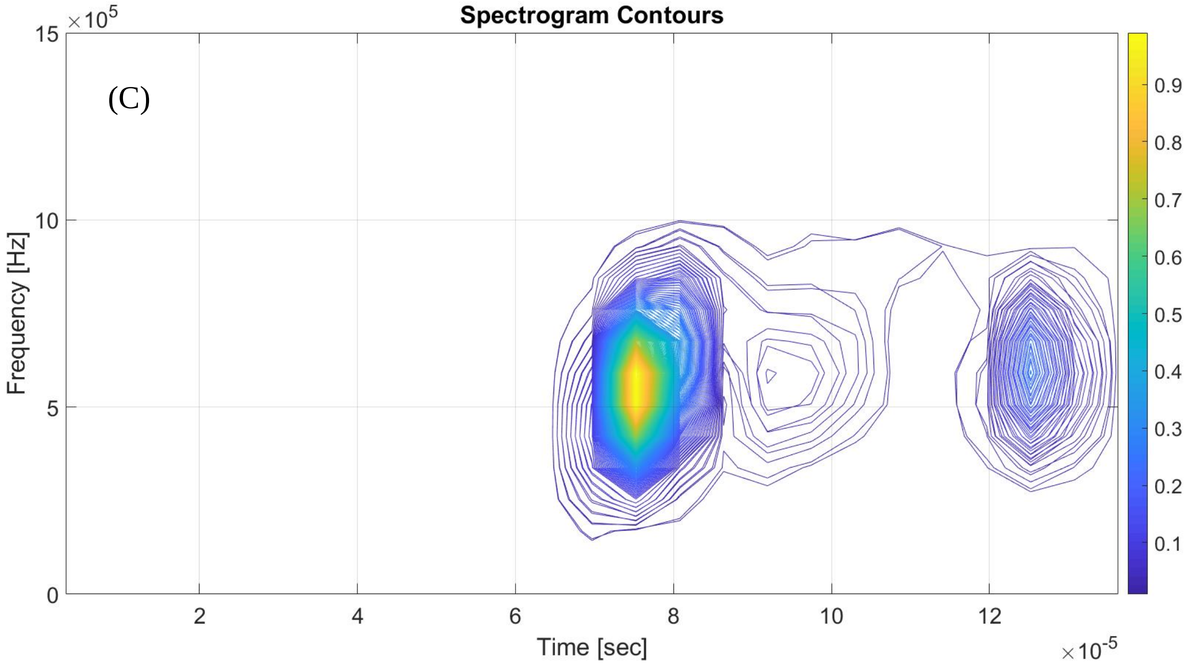
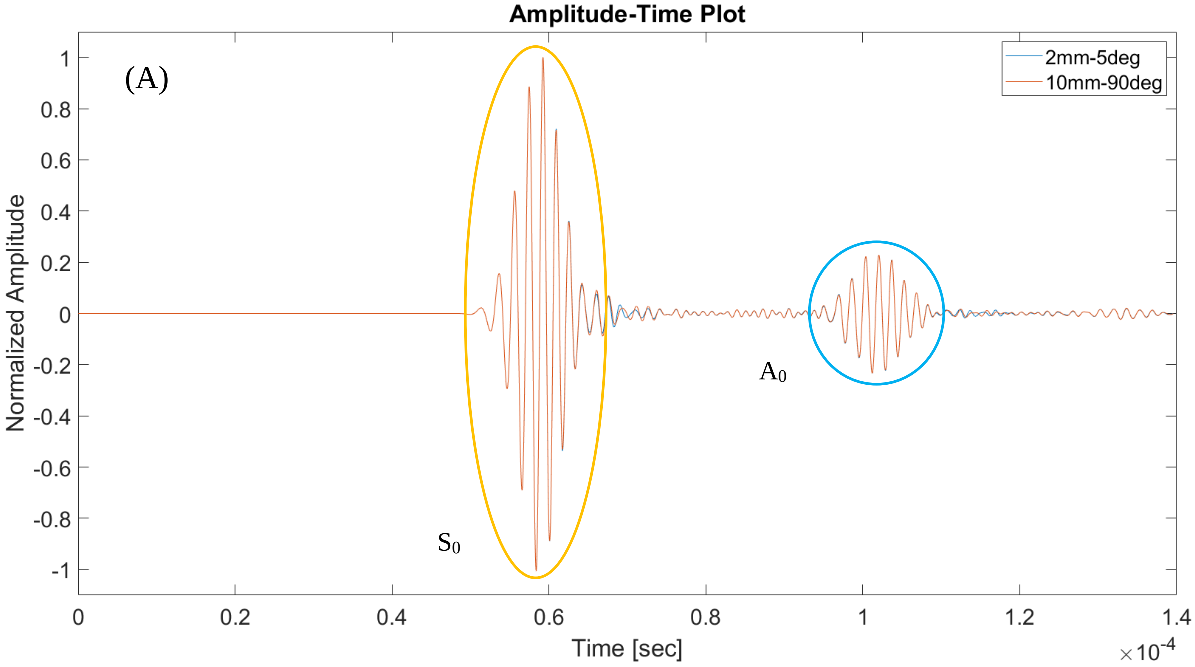


Figure 5.3: 2_5 and 10_90 at sensor 2: time domain (A), frequency domain (B) and spectrogram (C)

Sensor 3:



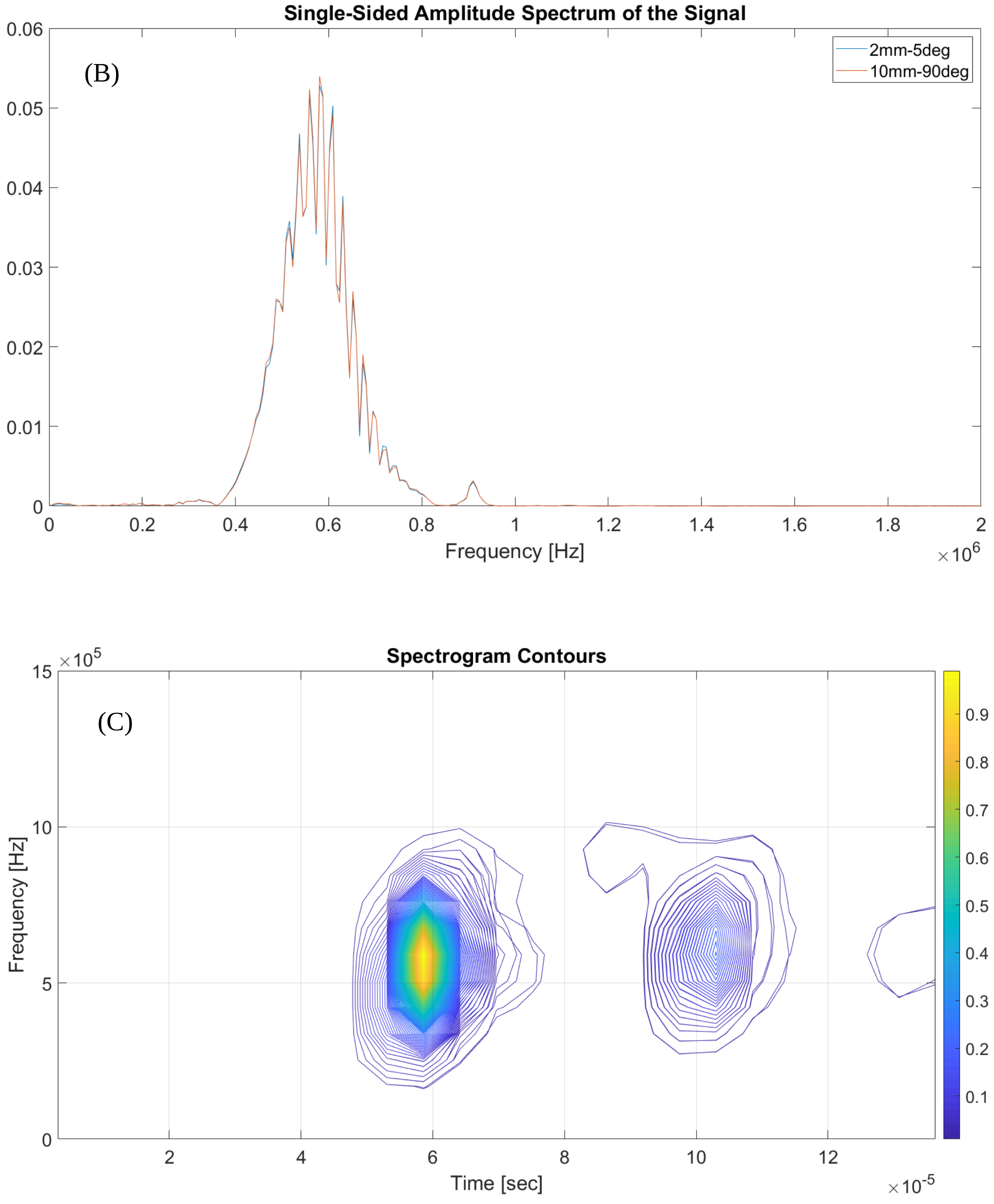


Figure 5.4: 2_5 and 10_90 at sensor 3: time domain (A), frequency domain (B) and spectrogram (C)

In all the three diagrams related to the frequency domain (B) it is clear that new frequency components did not arise, but only the amplitudes of the already existing components changed. This was however expected due to the narrowband nature of the input signal impinging on the discontinuity: all the spectrums are centred around the 600 kHz frequency and smooth out moving away from it. Therefore, this graphs could not give any further useful information.

Regarding the spectrograms (C), they are theoretically able to provide a more complete set of information. However, herein the differences are only seen at very low amplitudes, hence they cannot constitute once again the required set for a proper automatic understanding of the relations between signals and discontinuities.

It is possible to infer that the “conventional” features that could be extracted from a spectrogram diagram, like for example the employed by N. Facciotto in his work [21], namely:

- Mean and standard deviation of the frequency;
- Minimum, maximum and range of frequency;
- Mean normalized amplitude;
- Number of peaks, with the corresponding time and frequency values;
- Time-frequency area;

could not be used in this study, since the signals were so similar that these values would have resulted being practically identical for every simulation.

In this respect, the work of N. Facciotto was indeed drastically different, since he performed experiments with four rather distinct kind of signals (both broadband and narrowband).

Due to the aforementioned reasons and the key absence of multiple time-varying frequencies, it was decided to stick to the complete time domain amplitude diagrams, trying to extrapolate as many information as possible from them.

5.3. Signal processing

The first option that the author tried was to take the amplitude vector *as it is*, in order to basically keep every possible information in the time domain, in light of the relatively small amplitude’s variations between one signal and another.

Since, in this case, the features themselves are made of each element of the amplitude vector constituting the signal, this solution could be feasible only when signals are relatively short and therefore there is less risk of having a huge database in terms of features’ number.

Because the signals are all equally sampled, it is like saying that each feature is associated to a specific time-sample, and the same number of features is extracted from each signal vector.

Figure 5.5 shows the point extracted from an amplitude-time plot:

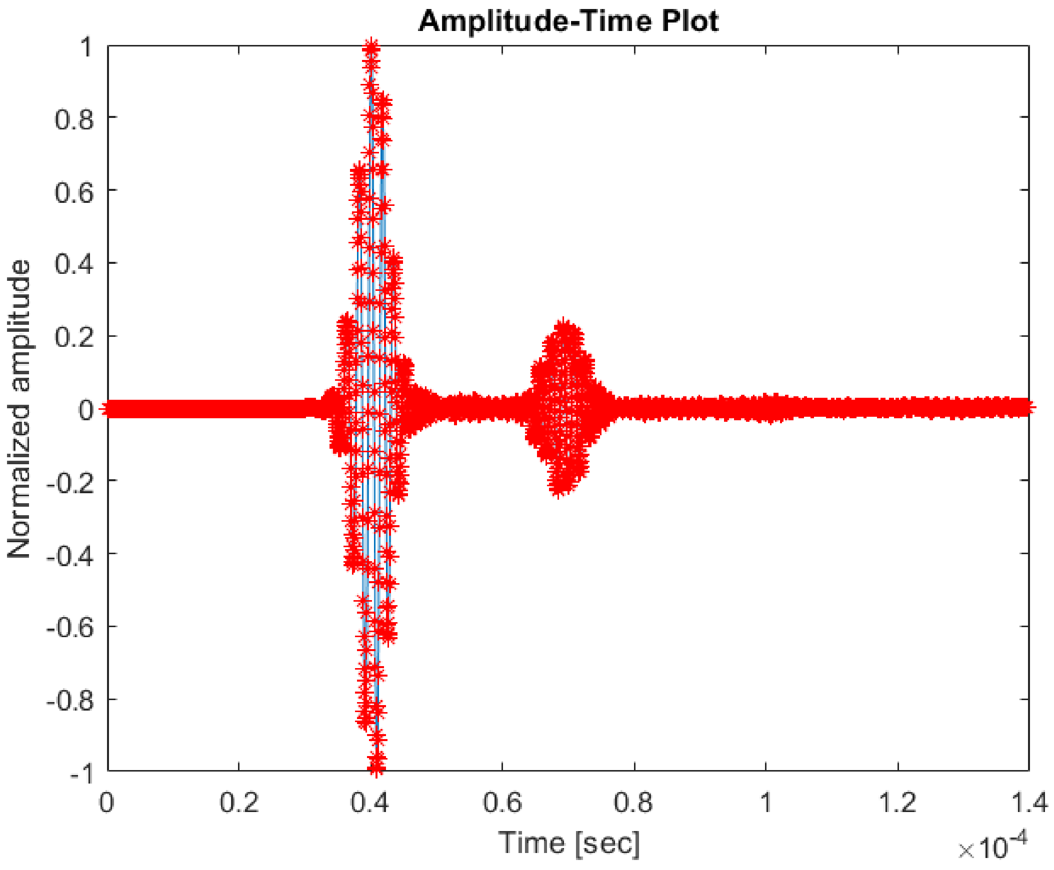


Figure 5.5: Elements of an amplitude vector

The number of extracted features is, as the reader may observe, quite large, even if it could be theoretically unlimited. As a matter of fact, the larger is the number of features the better should be the performances of the algorithms, but also the longer is the required training time and the overall computational process.

Another option consisted in extracting only the *peaks* of the time domain amplitude signals in order to accomplish a more efficient data reduction process, while trying to keep the most peculiar information.

Here, the peaks were taken selecting only the elements having an amplitude, in absolute value, which is larger with respect to the one of both the previous and the following vector’s elements. The so chosen values were eventually saved in a new vector, whereas the others were ignored in order to reduce the number of features.

As already mentioned in section 3.5, the extracted features and the desired responses should be put in a table format before exploiting them through the algorithms.

NaN values were inserted inside the table in order to equalize the number of elements in each row. Indeed, each signal may have a different number of peaks (i.e. of extracted features), thus resulting in a disparity between successive elements along the rows. This is not accepted within the MATLAB^{TM} table format: the number of elements per rows and columns must necessarily be the same. These *NaN* values are then interpreted by the Machine Learning algorithms as missing data, thus doing without them during the training phase.

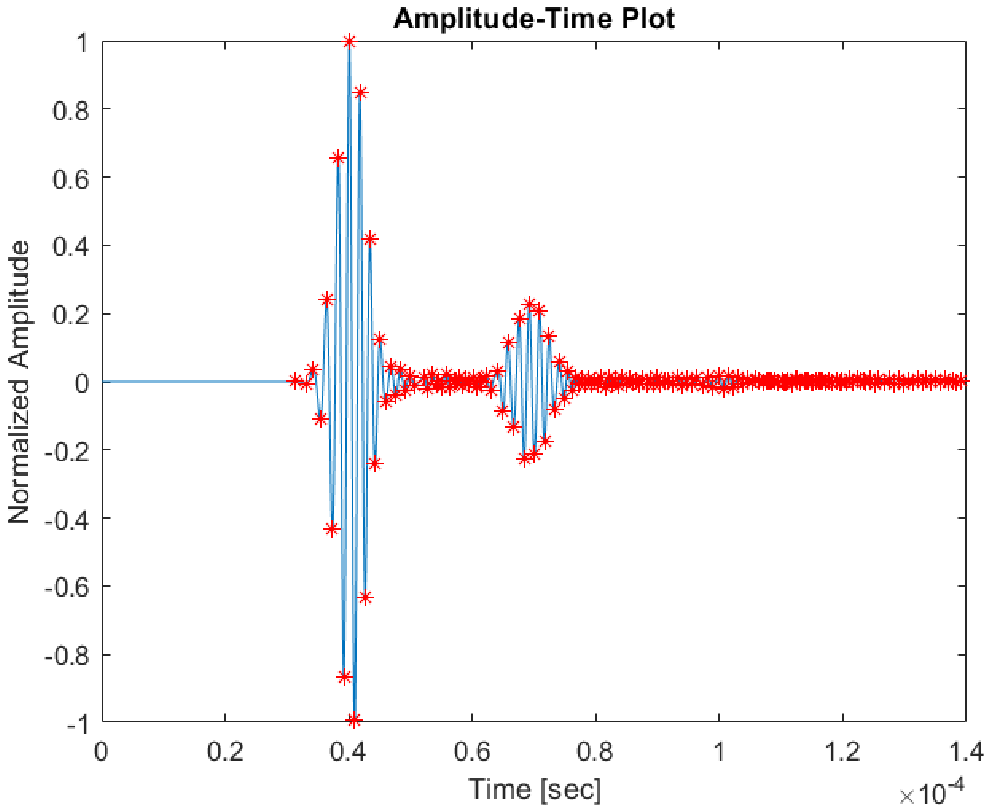


Figure 5.6: Peaks extracted from an amplitude vector

The same strategy, consisting of the two aforementioned methods, was adopted also on the *Hilbert transform* of each amplitude-time signal, in order to see whether the energy distribution (represented by the envelope-time plot) could give some extra and, most importantly, clearer information to be extracted. These graphs were constructed by exploiting the equation (3.3).

Regarding the first option, the envelope's elements were however taken considering a larger time step, since the time variation of this diagram is not as high as the former, and the employment of the sampling used previously would have resulted in a too high number of, in fact, useless features.

The following two figures show these methods applied to the envelope-time plot derived from the same signal constituting the former amplitude-time graphs (in figures 5.5 and 5.6).

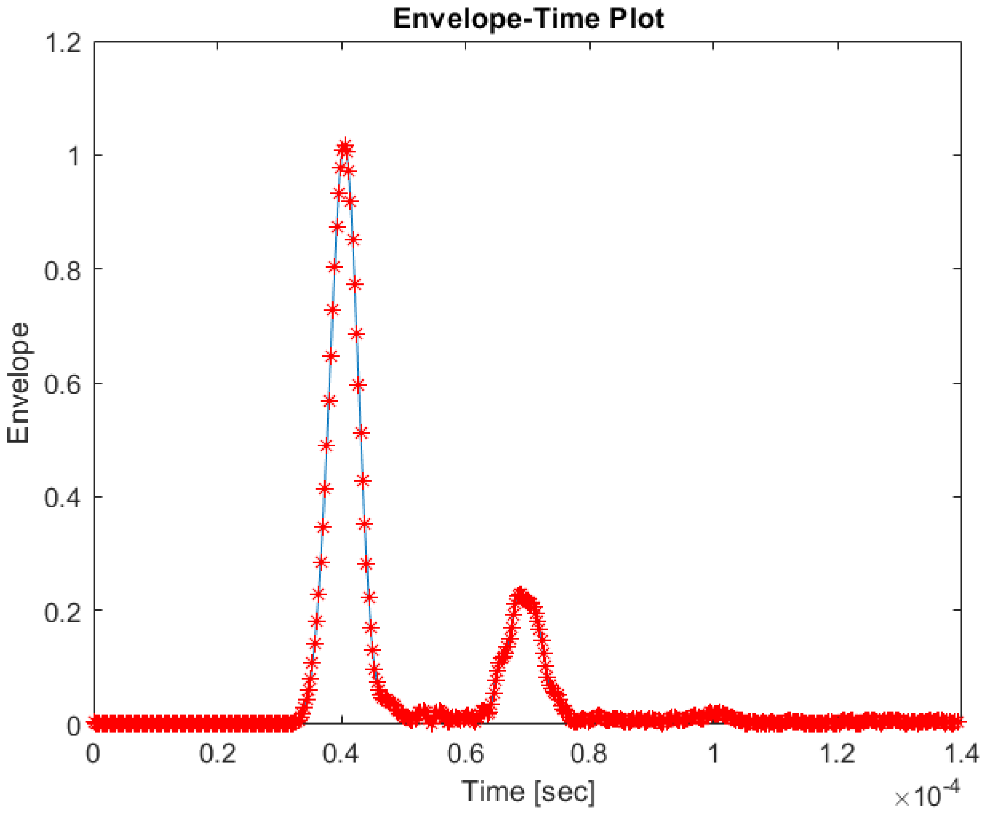


Figure 5.7: Elements extracted from an envelope vector

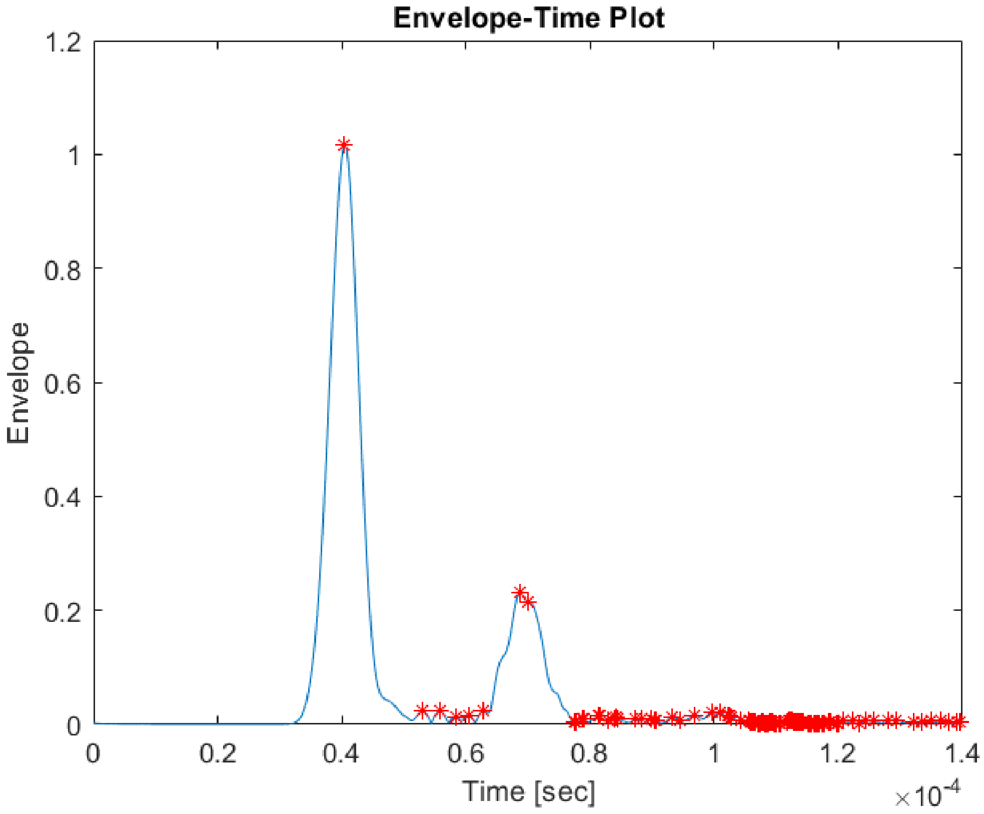


Figure 5.8: Peaks extracted from an envelope vector

The processing procedures explained up to this point involved separated signals: this means that each of the three signals coming from the same simulation (indeed recorder from the three sensors) was considered separately, and the successive analysis through the Machine Learning algorithms was done accordingly.

However, it could be also interesting to evaluate the algorithms' responses with respect to signals *merged* before the analysis: in this respect, the three signals were actually summed and then used inside the same algorithms.
In this latter case, the employed features extraction's methods were again the same as the ones mentioned for the individual signal scenario.

5.4. Machine Learning algorithms

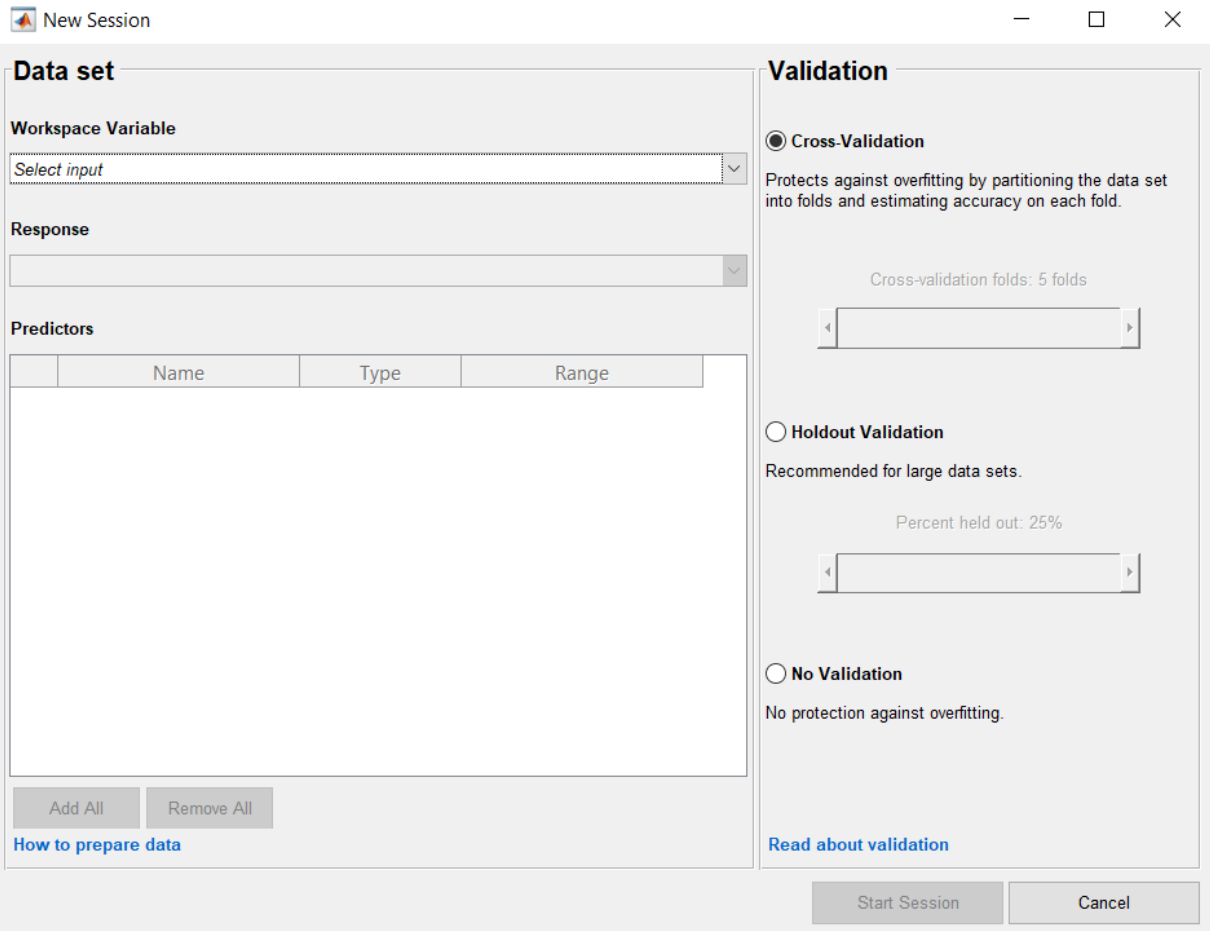
To perform the training and validation of all the algorithms employed in this study, the *Classification Learner App* and the *Regression Learner App* were used. As the name suggests, these two MATLAB^{TM} applications contain many of the most famous algorithms employed respectively for classification and regression of features, in order to assign classes and predict future response values. These applications eventually allow to export the chosen models in order to use them for the evaluations on future input data.
It is important to remember that, with these algorithms, it is not possible to use more than one response at the same time: in case of a multiple response problem, different models should be built in order to assess different responses. Artificial Neural Networks (ANN) may be instead employed to construct a model identifying more responses at the same time. In this study, this is however not a concern since the only important response is the discontinuity's length.

The feature's tables constructed during the previous processing phase were made of as many rows as the number of simulations and as many columns as the number of features. Then, it was added a last column representing the known output responses.
In case of the classification problems, the known responses were represented by *char* variables (e.g. *"2 mm"*). On the other hand, the regression problems needed to possess a *numeric* variable response as output (e.g. *2*).

The so-derived tables were then imported into the applications as Workspace Variables, the last column was selected as the one corresponding to the desired Responses, and the rest of the columns were addressed as Predictors.

The creation of a test set (the so-called “Holdout Validation” option in figure 5.9) is recommended only for large datasets, where a user defined percentage of the table’s rows can be excluded from the training phase in order to form a different set used later to test the algorithms’ performances.

However, since this is not the case of large datasets, a Cross-Validation procedure was applied. In this respect, as already mentioned in section 3.5, the so-called *“leave one out”* cross-validation was enforced. This method partitioned the datasets into twenty folds and estimated the performances on each of them, meaning that one of the training samples was in turn employed as the validation set while the other nineteen samples represented the training sets.



Select input

● Cross-Validation

Protects against overfitting by partitioning the data set into folds and estimating accuracy on each fold.

Cross-validation folds: 5 folds

○ Holdout Validation

Recommended for large data sets.

Percent held out: 25%

○ No Validation

No protection against overfitting.

Figure 5.9: Starting window for a new learning session

After these aforementioned procedures were properly carried out, the training and validation of the entire set of algorithms contained within the aforementioned applications could be performed. All the features were selected for the analysis, and the PCA was not implemented. In order to speed up the process, Parallel Training was employed.

5.4.1. Classification's results

The whole set of algorithms exploited for classification purposes is made of twentyfour elements which, grouped by family, are:

- Fine, Medium and Coarse Tree;
- Linear and Quadratic Discriminant;
- Gaussian and Kernel Naive Bayes;
- Linear, Quadratic, Cubic, Fine Gaussian, Medium Gaussian and Coarse Gaussian SVM;
- Fine, Medium, Coarse, Cosine, Cubic and Weighted KNN;
- Ensemble Boosted Trees, Bagged Trees, Subspace Discriminant, Subspace KNN and RUSBoosted Trees.

They were all trained and validated since there is not an objective way of choosing, a priori, which is the best for this purpose: each algorithm may give different accuracies depending on the problem and on the chosen dataset of features. Some of them even failed during the training process or gave very low accuracies.

In the next table, the best percent accuracy and the respective algorithm is reported for each of the training sessions:

Signal's amplitudes at each time step:

	Accuracy	**Corresponding algorithm**
Sensor 1	80 %	Linear Discriminant
Sensor 2	35 %	Ensemble RUSBoosted Trees
Sensor 3	75 %	Ensemble Subspace Discriminant
Merged signals	70 %	Linear Discriminant

Signal's peaks in time order:

	Accuracy	**Corresponding algorithm**
Sensor 1	55 %	Ensemble RUSBoosted Trees
Sensor 2	25 %	Ensemble RUSBoosted Trees
Sensor 3	65 %	Ensemble Bagged Trees
Merged signals	50 %	Ensemble Bagged Trees

Envelope's amplitudes at regular time steps:

	Accuracy	**Corresponding algorithm**
Sensor 1	85 %	Linear Discriminant
Sensor 2	30 %	Ensemble Subspace KNN
Sensor 3	70 %	Ensemble Subspace Discriminant
Merged signals	60 %	Ensemble Subspace Discriminant

Envelope's peaks in time order:

	Accuracy	**Corresponding algorithm**
Sensor 1	45 %	Ensemble RUSBoosted Trees
Sensor 2	20 %	Ensemble RUSBoosted Trees
Sensor 3	55 %	Ensemble RUSBoosted Trees
Merged signals	40 %	Ensemble RUSBoosted Trees

Envelope's peaks in descending order:

	Accuracy	**Corresponding algorithm**
Sensor 1	40 %	Ensemble RUSBoosted Trees
Sensor 2	45 %	Ensemble RUSBoosted Trees
Sensor 3	50 %	Ensemble RUSBoosted Trees
Merged signals	40 %	Ensemble RUSBoosted Trees

Table 5.2: Classification's results

The first thing to notice is that the overall trend of the second sensor is to give poorer results with respect to the other two and with respect also to the merged signals.

This behaviour is due to the extreme similarities between signals linked to defects of different length but with the same low degree of inclination. As an example, the signal recorded by this sensor with a discontinuity of 2 mm in length was not much different from a signal of 10 mm in length, both with an orientation of 5°. This could be associated to the section facing perpendicularly the incoming narrowband signal: this parameter, as the discontinuity's length increases, remains almost constant when the orientation goes towards an horizontal setting, while its value and its increment rate grows moving towards a vertical setting.

The former case may have led to similar interactions and consequent absorption of energy between the 2 and 10 mm scenarios. On the other hand, when the discontinuity was oriented more and more towards a vertical direction, the occurred interaction phenomena turned out being more evident.

The other important aspect is that the consideration, as features, of either the entire signal's amplitudes or energy envelope, led to quite promising classification accuracies of 70-80% in some cases (e.g. for sensors 1 and 3).

In the following, the confusion matrices reporting the best results for each of the three sensors and the merged signals are posted. The elements on the diagonal represent the percentage of correct matching, therefore, the more are the elements outside of the diagonal, the higher is the error. On the right hand side of each matrix is also reported the resume of each row: the first column reports the so-called *true positives* or *false negatives*, while in the second column the percentage sum of the errors is represented.

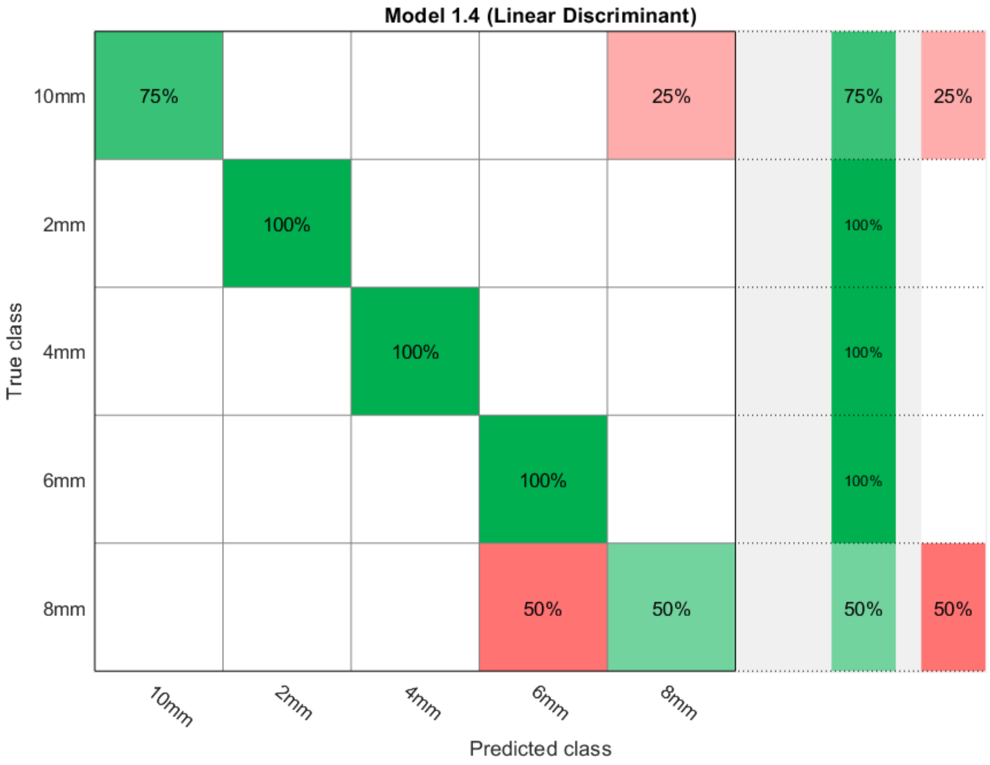


Figure 5.10: Best Confusion Matrix for sensor 1

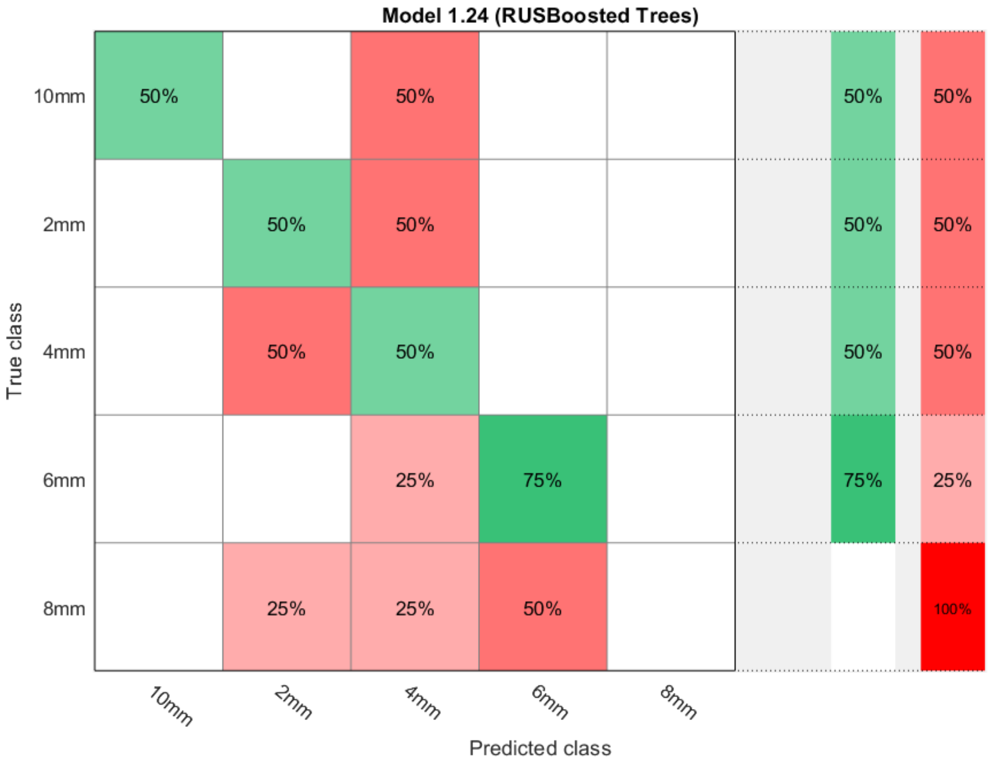


Figure 5.11: Best Confusion Matrix for sensor 2

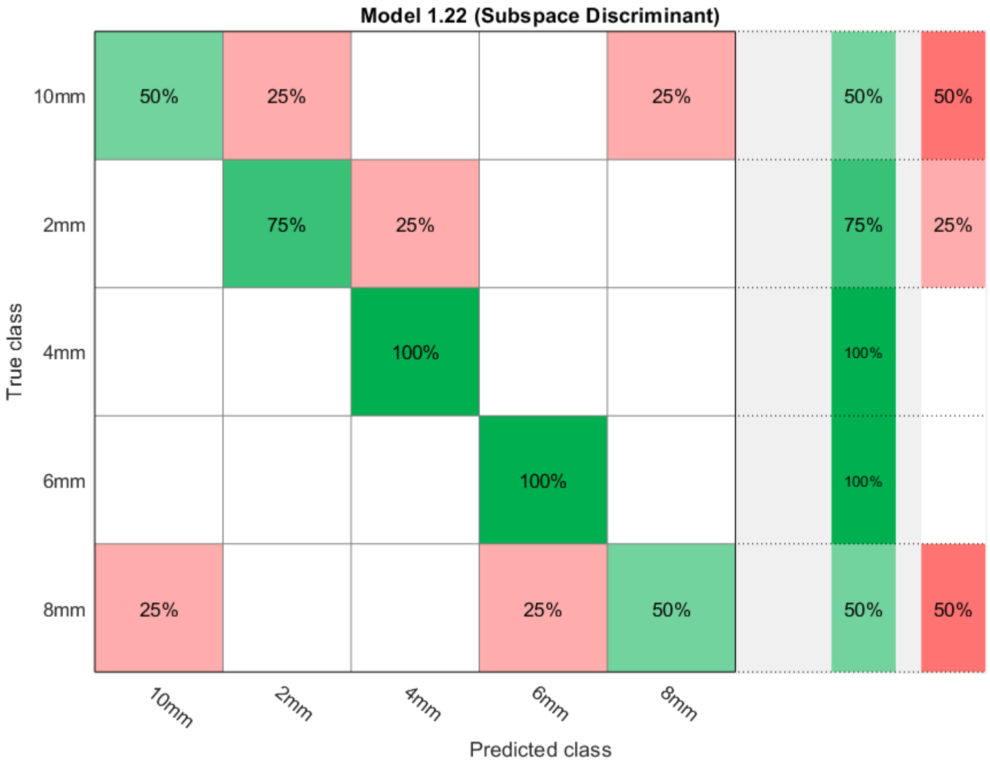


Figure 5.12: Best Confusion Matrix for sensor 3

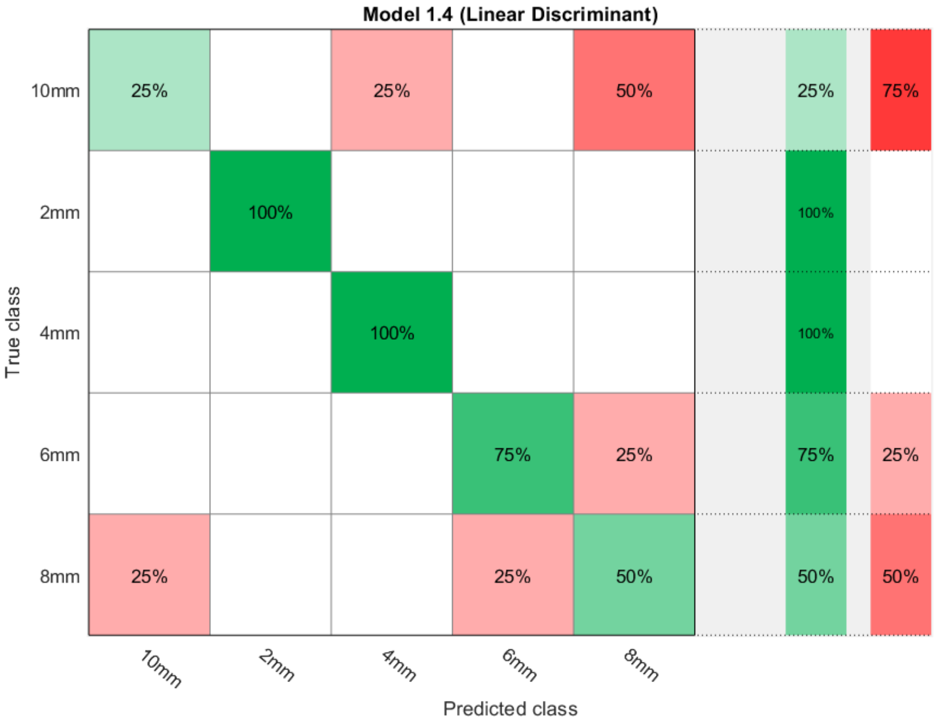


Figure 5.13: Best Confusion Matrix for the merged signals

It is noticeable that, especially for the sensors 1 and 3, the results are somehow well grouped along the diagonal, meaning that even if the length was not precisely classified, the algorithms still assigned the neighbouring classes.

5.4.2. Regression's results

Looking mainly at the future continuation of this work, where numeric defect-related parameters will continue to be the studied outcomes, the author decided to exploit also regression algorithms: they can interpolate the results inside of the training range, and therefore they are theoretically able to return correct predictions even outside of the known results' set. On the other hand, classification algorithms can just label new signals within the assigned categories they were trained with, even if the correct response may belong to a different one.

The whole set of algorithms employed for regression purposes is made of nineteen elements which, grouped by family, are:

- Linear, Interactions Linear, Robust Linear and Stepwise Linear Regression;
- Fine, Medium and Coarse Tree;

- Linear, Quadratic, Cubic, Fine Gaussian, Medium Gaussian and Coarse Gaussian SVM;
- Ensemble Boosted and Bagged Trees;
- Squared Exponential, Matern 5/2, Exponential and Rational Quadratic Gaussian Process Regression.

Once again, they were all trained and validated since there is not an objective way of choosing, a priori, which is the best: each algorithm may give different RMSE results depending on the problem and on the chosen dataset of features. Some of them even failed during the training process or gave very high errors. In the next table, the smallest RMS error and the respective algorithm is reported for each of the training sessions:

Signal's amplitudes at each time step:

	RMSE	**Corresponding algorithm**
Sensor 1	1.0608	Linear Regression
Sensor 2	1.8654	Cubic SVM
Sensor 3	1.0246	Rational Quadratic Gaussian Process
Merged signals	1.531	Rational Quadratic Gaussian Process

Signal's peaks in time order:

	RMSE	**Corresponding algorithm**
Sensor 1	1.5898	Ensemble Boosted Trees
Sensor 2	2.685	Ensemble Bagged Trees
Sensor 3	1.518	Ensemble Boosted Trees
Merged signals	1.3154	Fine Tree

Envelope's amplitudes at regular time steps:

	RMSE	**Corresponding algorithm**
Sensor 1	1.3086	Ensemble Boosted Trees
Sensor 2	2.0316	Linear SVM
Sensor 3	1.3438	Linear SVM
Merged signals	1.3838	Quadratic SVM

Envelope's peaks in time order:

	RMSE	**Corresponding algorithm**
Sensor 1	2.8719	Ensemble Boosted Trees
Sensor 2	2.7628	Ensemble Bagged Trees
Sensor 3	1.5811	Fine Tree
Merged signals	2.8065	Ensemble Bagged Trees

Envelope's peaks in descending order:

	RMSE	**Corresponding algorithm**
Sensor 1	2.9773	Medium Tree
Sensor 2	2.2879	Fine Tree
Sensor 3	1.5119	Ensemble Boosted Trees
Merged signals	2.3298	Fine Tree

Table 5.3: Regression's results

When the RMSE is close to 3 (e.g. equal to 2.9773), it means that all the predicted responses are around the centre of the given range, hence very close to 6 mm. Smaller values are desirable, since it would mean that the predictions are going towards the actual values.

Again, it is noticeable that the overall trend of the second sensor is to give poorer results with respect to the other two and with respect also to the merged signals.

The other important aspect is that the consideration, as features, of the entire signal's amplitudes led one more time to reasonable predictions, especially for sensors 1 and 3. This will be clearer in the next two pages where the predicted responses are shown.

Actually, the location may have played an important role in the best overall results of sensor 1, since its position allowed to catch also some of the reflections originating from the interference with the discontinuity, and not only the signal's part related to the pure transmission.

In the following, the best *predictions* for each of the three sensors and the merged signals are posted. The vertical axis shows the discontinuities' lengths (in millimetres) with which the algorithms were trained.

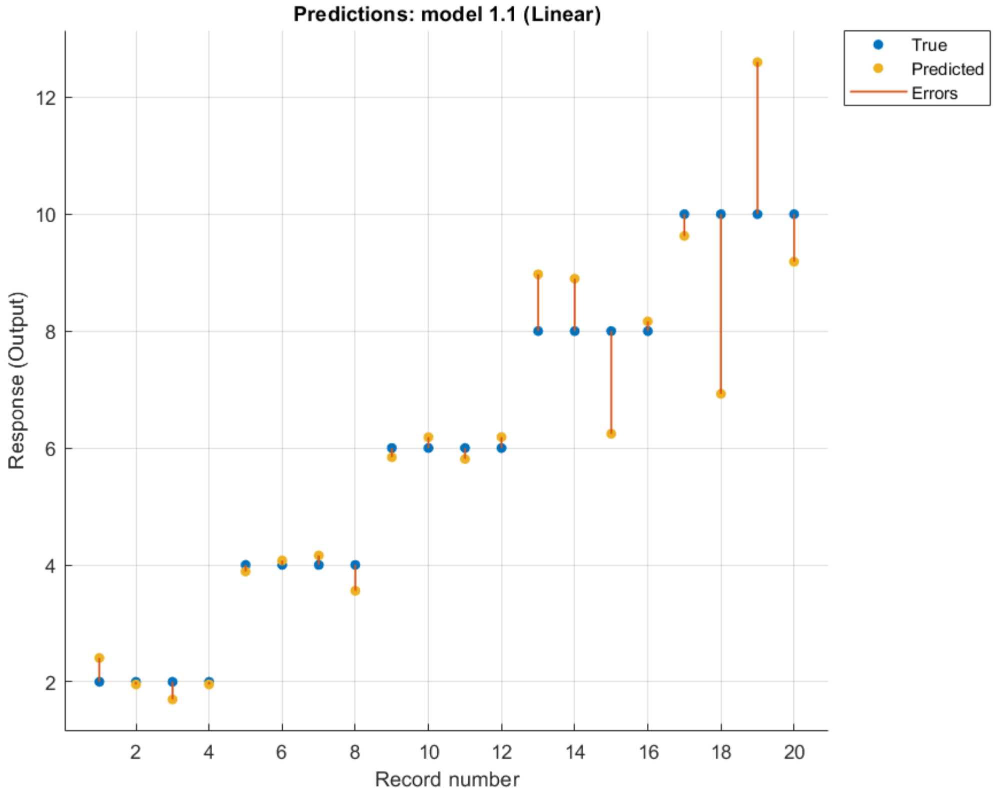


Figure 5.14: Best predicted responses for sensor 1

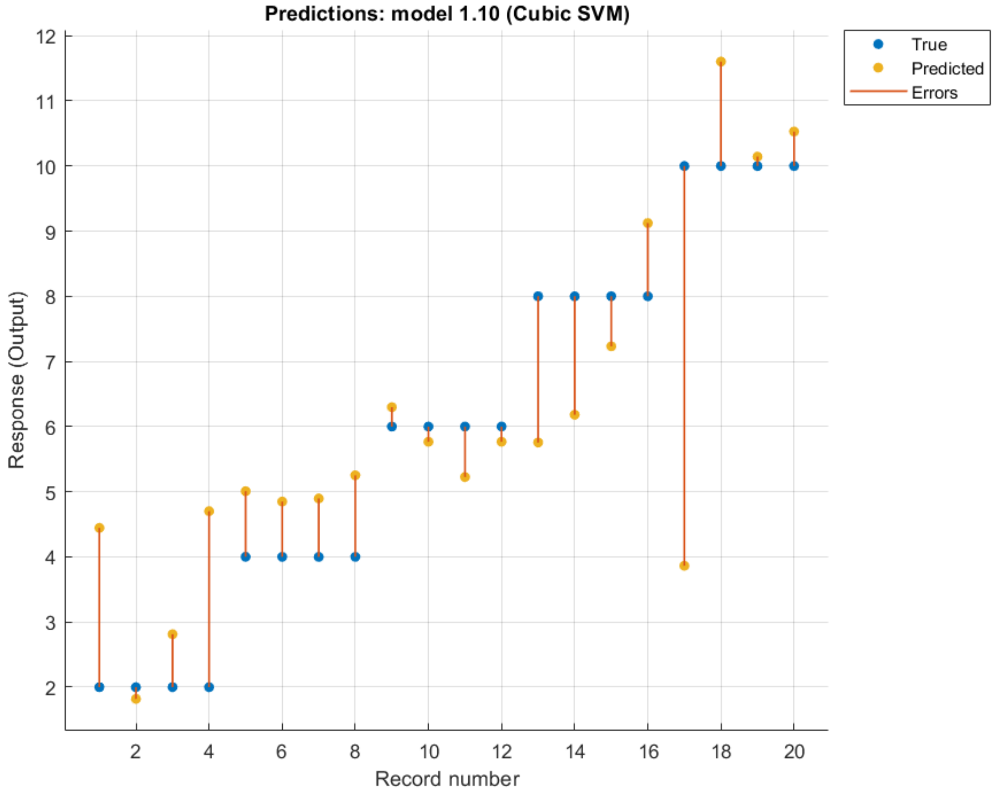


Figure 5.15: Best predicted responses for sensor 2

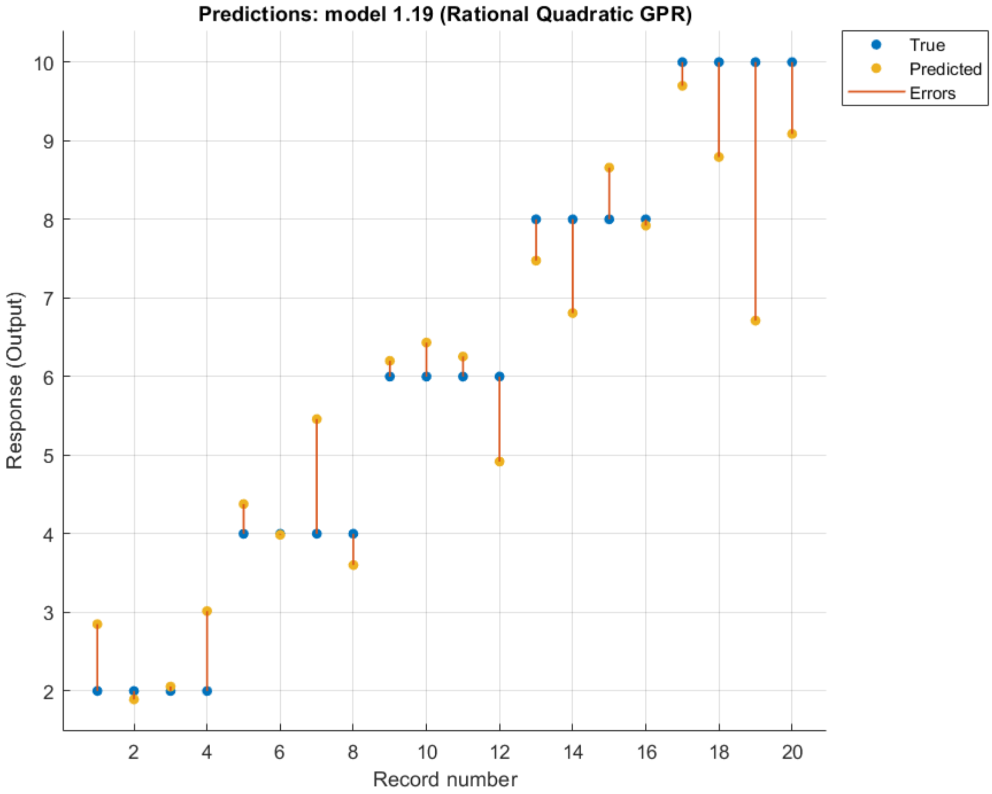


Figure 5.16: Best predicted responses for sensor 3

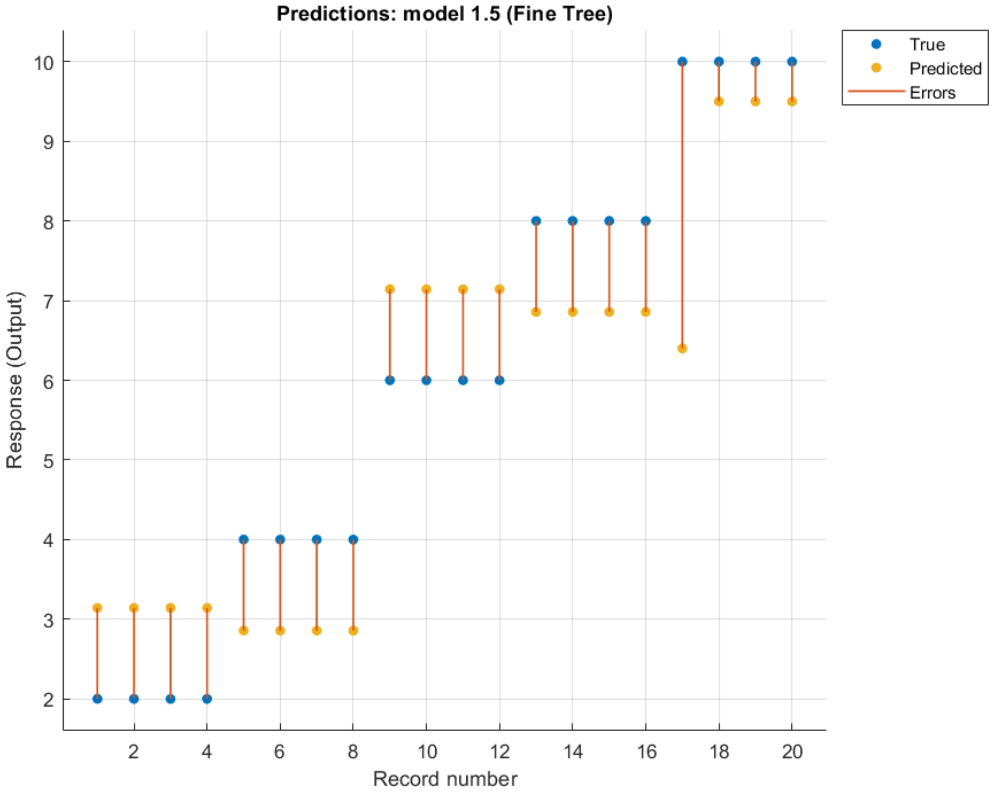


Figure 5.17: Best predicted responses for the merged signals

A detailed explanation of all the supervised algorithms exploited in this study is given in [73]. Regarding the Ensemble Methods, they make a contemporary use of a multitude of algorithms in order to increase the overall performance compared to the ones individually obtainable from the same algorithms. This may be the reason why, in a relevant part of the examined cases, they actually gave the best results.

The other exploited algorithms, however, proved to be effective in some cases, thus confirming the fact that each dataset may be more suitable for a specific algorithm, and that a trial and error procedure is usually necessary to find the best one.

There are a couple of considerations the author would like to make regarding the performances of both sets of exploited Machine Learning algorithms:

1) More simulations could probably increase the correct results because of the larger training set. Indeed, within section 3.5 on the introduction to Machine Learning techniques, it was mentioned how usually the increased size of the training set goes hand in hand with an improvement in the performances, just due to the augmented number of samples and the inherent learning improvements coming from the more information available.
2) An increased number of sensors, or a better layout, could caught every nuance the signals may present after its interaction with the defect, perhaps improving the resulting overall performances. For example, reflections and diffractions along particular directions could enhance the diversity within the set of collected signals. Despite exploiting a sort of ad-hoc layout may raise the positive results, it may also lead to the loss of generality with respect to the many possible positions the defect can have within the geometry.

CHAPTER 6

6. FINAL CONSIDERATIONS

6.1. Recap and future works

Throughout this thesis, two commercial software were comprehensively evaluated and an entire numerical setup was studied, implemented and run. The raw data collected were then imported, pre-processed and analysed into the MATLAB^{TM} environment.

However, the problem of taking as features just the amplitudes of the time domain signal may lead to some problems when dealing with the real working environment. Indeed, when noise is present, there is the need to insert a threshold to discard it and, if the amplitudes variations are at a small order of magnitude (as in most of the case studied herein), they can be overwritten by noise and therefore vanish. This could mean having no more differences between the signals. In order to see whether these small differences would still be present also in a real environment, *experimental studies* must be performed. During these studies, actual cracks could also be initiated and grown within the loaded structure in order to directly generate the related signals and evaluate a proper passive technique, as originally planned.

A further and future developing possibility regarding the automatic processing of signals recorded either by numerical simulations or experiments was suggested by Dr. Julio J. Valdés[1]. He proposed to exploit the so-called *Deep Learning*, instead of the more classical Machine Learning, to directly recognize the images deriving from a set of signal transformations.

In this other approach, the user is able to either construct his own network or to exploit pre-built ones that possess the innovative possibility of extracting and learning features through an automatic process, repeatedly applying a convergence study to find the most suitable. Therefore, the need of a user who tries to find them by chance, as for Machine Learning, is not necessary anymore, and the most adequate features for each specific case may be found.

Some of the most famous Deep Learning networks for the recognition of images are the so-called Convolutional Neural Networks (CNN) like AlexNet, ImageNet, ResNet, etc., and they

---

[1] He is a mathematician and a researcher at the National Research Council (NRC) of Canada. He works mainly within the Artificial Intelligence scientific field and is an expert on these subjects. The author had the opportunity to have a video-call with him, thanks to the co-supervisor and his colleague Shashank Pant.

can be used either pre-trained or re-trained with new inputs.

As a downside, these algorithms require a huge computational power and resources, because of their architecture, and the need of large input databases. They are indeed efficient only when working with big datasets (as the one shown in [74]), whereas traditional Machine Learning algorithms are also suitable for smaller ones.

The Deep Learning networks differ from the others because they are made of a much higher number of intermediate internal levels between the input and the output.

Summing up, the suggestion consists in exploiting a Deep Learning pre-trained algorithm to automatically extract the features considered as the most suitable in highlighting the differences between various sets of input images associated to a specific signal. Then, for example, train only the successive classifier with the previous extracted features.

The set of input images he proposed is made of three time-frequency representations of the same signal:

- Spectrogram;
- Continuous Wavelet Transform;
- Wigner-Ville Transform.

According to his experience, they provide a good characterization of any frequency change together with a good time representation.

The following figure resumes this new workflow:

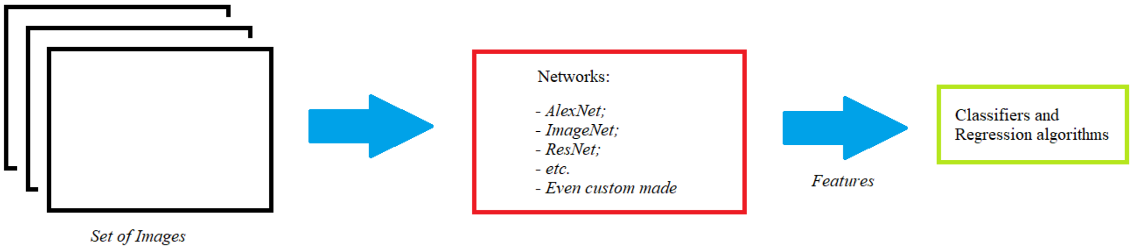


Figure 6.1: An interesting workflow for future AI studies

If there had been the opportunity of carrying out experiments, more raw data would have been collected and such a path could have been followed. Unfortunately, the fact of having to deal with simulations which in this case are longer than experiments, and models that took a fair amount of time to effectively be ready on the new software employed herein, meant it was not possible to collect the necessary large number of results in order to train a Deep Learning network. This task is left to future studies.

Moreover, this thesis has only focused on problems involving a single discontinuity. Future investigations could well concentrate on structures containing *multiple* defects.

Future studies may also work on comparing signals (e.g. subtracting a signal with the baseline taken from a pristine structure), in order to extrapolate a *damage index* linked to the crack's state. However, when dealing with this comparison, researchers should pay attention to the loading conditions, environmental conditions, residual stresses induced on the structure, etc. They all contribute to changing the wave propagation's velocities, therefore the signals and the baseline should be taken respecting the same working conditions in order to get comparable results.

As already mentioned throughout this chapter, future works should investigate more deeply into many fields: from the exploitation of passive AE techniques, going through the improvement of the extracted signal's features, to the creation of ad-hoc automated regression procedures capable of giving satisfactory results, and the consequent drafting of PoD curves.

In this regard, the extrapolation of these curves for NDE techniques requires at least from 60 to 120 experimental specimens and, when this is not possible, numerical physics-based models (experimentally validated) should be enforced. However, when dealing with the SHM field, the PoD studies are a little different: indeed, the former required independent measurements while, in the latter, the measures are actually coming from a stream of data. This means that other external factors (e.g. sensor's degradation, load variations, etc.) could affect the results more significantly; as a consequence, the definition of a proper PoD study within the SHM field is still an ongoing research activity.

Inevitably, the current model did not consider all the variability the real situation can bring and did not include any experimental validation, therefore it was not possible to compare it directly with the available PoD curves of some NDE techniques. However, the continuation of the presented preliminary study incorporating the aforementioned factors could eventually lead to the definition of these mandatory graphs.

6.2. Conclusions

Detection and localization techniques based on GLW, which are already more developed within the scientific community (even on wing-box structures and with the influence of temperature and load variations), may be further completed by damage quantification automated procedures. This work does represent a breakthrough in the field of inverse quantification of small damages by means of GLW, but it is certainly not exhaustive. Nevertheless, considering the preliminary nature of this study intrinsic in all the simplifications adopted and in the lack of an experimental confirmative campaign, the results previously reported in sections 5.4.1 and 5.4.2 are somewhat encouraging.

Indeed, supported by these outcomes, the author believes that AI procedures exploiting sensing networks embedded into even complex structures should be intensively studied in order to have a future industrial employment on existing frames. This will eventually bring the benefits mentioned within the introductory chapters: namely the development of an intelligent structure capable of self-assessing its own state and providing useful information about future condition-based maintenance and repairing operations.

BIBLIOGRAPHY - SITOGRAPHY:

[1] Z. Sharif Khodaei and M. H. Aliabadi, ‘A Multi-Level Decision Fusion Strategy for Condition Based Maintenance of Composite Structures’, Materials, vol. 9, no. 9, p. 790, Sep. 2016, doi: 10.3390/ma9090790.

[2] V. Memmolo, N. Boffa, L. Maio, E. Monaco, and F. Ricci, ‘Damage Localization in Composite Structures Using a Guided Waves Based Multi-Parameter Approach’, Aerospace, vol. 5, no. 4, p. 111, Oct. 2018, doi: 10.3390/aerospace5040111.

[3] S. Fu, L. Shi, Y. Zhou, and Z. Fu, ‘Enhancement of Lamb Wave Imaging Resolution by Step Pulse Excitation and Prewarping’, Shock Vib., vol. 2015, pp. 1–8, 2015, doi: 10.1155/2015/185148.

[4] J. E. Michaels and T. E. Michaels, ‘Guided wave signal processing and image fusion for in situ damage localization in plates’, Wave Motion, vol. 44, no. 6, pp. 482–492, Jun. 2007, doi: 10.1016/j.wavemoti.2007.02.008.

[5] F. A. Masurkar and N. P. Yelve, ‘Optimizing location of damage within an enclosed area defined by an algorithm based on the Lamb wave response data’, Appl. Acoust., vol. 120, pp. 98–110, May 2017, doi: 10.1016/j.apacoust.2017.01.014.

[6] W. Wang, Y. Bao, W. Zhou, and H. Li, ‘Sparse representation for Lamb-wave-based damage detection using a dictionary algorithm’, Ultrasonics, vol. 87, pp. 48–58, Jul. 2018, doi: 10.1016/j.ultras.2018.02.011.

[7] M. Dziendzikowski, P. Niedbala, A. Kurnyta, K. Kowalczyk, and K. Dragan, ‘Structural Health Monitoring of a Composite Panel Based on PZT Sensors and a Transfer Impedance Framework’, Sensors, vol. 18, no. 5, p. 1521, May 2018, doi: 10.3390/s18051521.

[8] A. Muller, B. Robertson-Welsh, P. Gaydecki, M. Gresil, and C. Soutis, ‘Structural Health Monitoring Using Lamb Wave Reflections and Total Focusing Method for Image Reconstruction’, Appl. Compos. Mater., vol. 24, no. 2, pp. 553–573, Apr. 2017, doi: 10.1007/s10443-016-9549-5.

[9] J. He, Y. Ran, B. Liu, J. Yang, and X. Guan, ‘A Fatigue Crack Size Evaluation Method Based on Lamb Wave Simulation and Limited Experimental Data’, Sensors, vol. 17, no. 9, p. 2097, Sep. 2017, doi: 10.3390/s17092097.

[10] Z. Su, C. Zhou, M. Hong, L. Cheng, Q. Wang, and X. Qing, ‘Acousto-ultrasonics-based fatigue damage characterization: Linear versus nonlinear signal features’, Mech. Syst. Signal Process., vol. 45, no. 1, pp. 225–239, Mar. 2014, doi: 10.1016/j.ymssp.2013.10.017.

[11] X. Zhao et al., ‘Active health monitoring of an aircraft wing with embedded piezoelectric sensor/actuator network: I. Defect detection, localization and growth monitoring’, Smart Mater. Struct., vol. 16, no. 4, pp. 1208–1217, Aug. 2007, doi: 10.1088/0964-1726/16/4/032.

[12] Hameed, Li, Chen, and Qi, ‘Lamb-Wave-Based Multistage Damage Detection Method Using an Active PZT Sensor Network for Large Structures’, Sensors, vol. 19, no. 9, p. 2010, Apr. 2019, doi: 10.3390/s19092010.

[13] F. Falcetelli, ‘Modelling of Pencil-Lead Break Acoustic Emission Sources using the Time Reversal Technique’, University of Bologna, 2018.

[14] A. Eremin, M. Golub, E. Glushkov, and N. Glushkova, ‘Identification of delamination based on the Lamb wave scattering resonance frequencies’, NDT E Int., vol. 103, pp. 145–153, Apr. 2019, doi: 10.1016/j.ndteint.2019.03.001.

[15] V. Pavelko, ‘Application of the Fatigue Crack Opening/Closing Effect for Aircraft SHM’, p. 8.

[16] V. Giurgiutiu and B. Poddar, ‘Can One Hear the Length of a Crack?’, Procedia Eng., p. 8, 2016.

[17] M. Y. Bhuiyan and V. Giurgiutiu, ‘Multiphysics Simulation of Low-Amplitude Acoustic Wave Detection by Piezoelectric Wafer Active Sensors Validated by In-Situ AE-Fatigue Experiment’, Materials, vol. 10, no. 8, p. 962, Aug. 2017, doi: 10.3390/ma10080962.

[18] M. Y. Bhuiyan, J. Bao, B. Poddar, and V. Giurgiutiu, ‘Toward identifying crack-length-related resonances in acoustic emission waveforms for structural health monitoring applications’, Struct. Health Monit., vol. 17, no. 3, pp. 577–585, May 2018, doi: 10.1177/1475921717707356.

[19] B. Rocha, C. Silva, and A. Suleman, ‘Guided Lamb Waves Based Structural Health Monitoring Through a PZT Network System’, p. 8.

[20] R. Unnorsson, ‘Hit Detection and Determination in AE Bursts’, in Acoustic Emission - Research and Applications, W. Sikorski, Ed. InTech, 2013.

[21] N. Facciotto, ‘Source Differentiation and Identification of Acoustic Emission Signals by Time-Frequency Analysis’, University of Bologna, 2016.

[22] C. R. Farrar and K. Worden, ‘An introduction to structural health monitoring’, Philos. Trans. R. Soc. Math. Phys. Eng. Sci., vol. 365, no. 1851, pp. 303–315, Feb. 2007, doi: 10.1098/rsta.2006.1928.

[23] D. Balageas, ‘Introduction to Structural Health Monitoring’, in Structural Health Monitoring, ISTE, 2006.

[24] Giurgiutiu Victor, ‘Structural health monitoring with piezoelectric wafer active sensors – predictive modeling and simulation’, INCAS Bull., vol. 2, no. 3, pp. 31–44, Sep. 2010, doi: 10.13111/2066-8201.2010.2.3.4.

[25] Lamb Horace, ‘On waves in an elastic plate’, p. 15.

[26] J. L. Rose, Ultrasonic Guided Waves in Solid Media. Cambridge University Press, 1999.

[27] P. D. Wilcox, ‘LAMB WAVE INSPECTION OF LARGE STRUCTURES USING PERMANENTLY ATTACHED TRANSDUCERS’.

[28] Z. Su, L. Ye, and Y. Lu, ‘Guided Lamb waves for identification of damage in composite structures: A review’, J. Sound Vib., vol. 295, no. 3–5, pp. 753–780, Aug. 2006, doi: 10.1016/j.jsv.2006.01.020.

[29] Y. Zhao et al., ‘Generation mechanism of nonlinear ultrasonic Lamb waves in thin plates with randomly distributed micro-cracks’, Ultrasonics, vol. 79, pp. 60–67, Aug. 2017, doi: 10.1016/j.ultras.2017.04.004.

[30] V. Giurgiutiu, ‘Lamb wave generation with piezoelectric wafer active sensors for structural health monitoring’, San Diego, CA, Aug. 2003, p. 111, doi: 10.1117/12.483492.

[31] M. A. Hamstad, A. O’Gallagher, and J. Gary, ‘Modelling of buried acoustic emission monopole and dipole sources with a finite element technique’, Journal of Acoustic Emission, pp. 97–110, 1999.

[32] M. V. Lysak, ‘Development of the theory of acoustic emission by propagating cracks in terms of fracture mechanics’, Eng. Fract. Mech., vol. 55, no. 3, pp. 443–452, Oct. 1996, doi: 10.1016/0013-7944(96)00026-4.

[33] C. K. Lee, B. Drinkwater, M. Friswell, J. Scholey, P. Wilcox, and M. Wisnom, ‘Acoustic emission during fatigue crack growth in aluminium plates’, p. 8.

[34] M. G. R. Sause and S. Richler, ‘Finite Element Modelling of Cracks as Acoustic Emission Sources’, J. Nondestruct. Eval., vol. 34, no. 1, p. 4, Mar. 2015, doi: 10.1007/s10921-015-0278-8.

[35] B. Poddar and V. Giurgiutiu, ‘Detectability of Crack Lengths from Acoustic Emissions Using Physics of Wave Propagation in Plate Structures’, J. Nondestruct. Eval., vol. 36, no. 2, p. 41, Jun. 2017, doi: 10.1007/s10921-017-0392-x.

[36] F. Ding, M. Feng, and Y. Jiang, ‘Modeling of fatigue crack growth from a notch’, Int. J. Plast., vol. 23, no. 7, pp. 1167–1188, Jul. 2007, doi: 10.1016/j.ijplas.2006.10.010.

[37] D. Gagar and P. Foote, ‘A Novel Closure-based Approach for Fatigue Crack Length Estimation Using the Acoustic Emission Technique in Structural Health Monitoring Applications’, p. 27.

[38] M. G. R. Sause, ‘Modelling of Crack Growth Based Acoustic Emission Release in Aluminum Alloys’, p. 8.

[39] H. Sohn and C. R. Farrar, ‘Damage diagnosis using time series analysis of vibration signals’, Smart Mater. Struct., vol. 10, no. 3, pp. 446–451, Jun. 2001, doi: 10.1088/0964-1726/10/3/304.

[40] Y. Lu, L. Ye, Z. Su, and C. Yang, ‘Quantitative assessment of through-thickness crack size based on Lamb wave scattering in aluminium plates’, NDT E Int., vol. 41, no. 1, pp. 59–68, Jan. 2008, doi: 10.1016/j.ndteint.2007.07.003.

[41] E. Maillet, C. Baker, G. N. Morscher, V. V. Pujar, and J. R. Lemanski, ‘Feasibility and limitations of damage identification in composite materials using acoustic emission’, Compos. Part Appl. Sci. Manuf., vol. 75, pp. 77–83, Aug. 2015, doi: 10.1016/j.compositesa.2015.05.003.

[42] T. Chuluunbat, C. Lu, A. Kostryzhev, and K. Tieu, ‘Influence of Loading Conditions during Tensile Testing on Acoustic Emission’, Key Eng. Mater., vol. 626, pp. 121–126, Aug. 2014, doi: 10.4028/www.scientific.net/KEM.626.121.

[43] A. Majkowski, M. Kołodziej, and R. J. Rak, ‘Joint Time-Frequency And Wavelet Analysis - An Introduction’, Metrol. Meas. Syst., vol. 21, no. 4, pp. 741–758, Dec. 2014, doi: 10.2478/mms-2014-0054.

[44] Z. Su, L. Ye, and X. Bu, ‘A damage identification technique for CF/EP composite laminates using distributed piezoelectric transducers’, Compos. Struct., vol. 57, no. 1–4, pp. 465–471, Jul. 2002, doi: 10.1016/S0263-8223(02)00115-0.

[45] A. L.-C. Wang, ‘An Industrial-Strength Audio Search Algorithm’, p. 7.

[46] F. Michard and A. Badheka, ‘Toward the “Shazam-Like” Identification of Valve Diseases with Digital Auscultation?’, Am. J. Med., vol. 132, no. 6, pp. e595–e596, Jun. 2019, doi: 10.1016/j.amjmed.2018.11.022.

[47] S. Seyoum, L. Alfonso, S. J. van Andel, W. Koole, A. Groenewegen, and N. van de Giesen, ‘A Shazam-like Household Water Leakage Detection Method’, Procedia Eng., vol. 186, pp. 452–459, 2017, doi: 10.1016/j.proeng.2017.03.253.

[48] R. K. Ing and M. Fink, ‘Time-reversed Lamb waves’, IEEE Trans. Ultrason. Ferroelectr. Freq. Control, vol. 45, no. 4, pp. 1032–1043, Jul. 1998, doi: 10.1109/58.710586.

[49] R. Gutkin, C. J. Green, S. Vangrattanachai, S. T. Pinho, P. Robinson, and P. T. Curtis, ‘On acoustic emission for failure investigation in CFRP: Pattern recognition and peak frequency analyses’, Mech. Syst. Signal Process., vol. 25, no. 4, pp. 1393–1407, May 2011, doi: 10.1016/j.ymssp.2010.11.014.

[50] N. Markovic, D. Stojic, R. Cvetkovic, V. Radojicic, and S. Conic, ‘Numerical modeling of ultrasonic wave propagation - by using of explicit FEM in ABAQUS’, Facta Univ. - Ser. Archit. Civ. Eng., vol. 16, no. 1, pp. 135–147, 2018, doi: 10.2298/FUACE170830011M.

[51] M. Martinez, S. Pant, M. Yanishevsky, and D. Backman, ‘Residual stress effects of a fatigue crack on guided lamb waves’, Smart Mater. Struct., vol. 26, no. 11, p. 115004, Nov. 2017, doi: 10.1088/1361-665X/aa8a0c.

[52] C. Sbarufatti et al., ‘Strain Wave Acquisition by a Fiber Optic Coherent Sensor for Impact Monitoring’, Materials, vol. 10, no. 7, p. 794, Jul. 2017, doi: 10.3390/ma10070794.

[53] K. Ono, ‘Review on Structural Health Evaluation with Acoustic Emission’, Appl. Sci., vol. 8, no. 6, p. 958, Jun. 2018, doi: 10.3390/app8060958.

[54] S. Grondel, C. Delebarre, J. Assaad, J.-P. Dupuis, and L. Reithler, ‘Fatigue crack monitoring of riveted aluminium strap joints by Lamb wave analysis and acoustic emission measurement techniques’, NDT E Int., vol. 35, no. 3, pp. 137–146, Apr. 2002, doi: 10.1016/S0963-8695(01)00027-5.

[55] G. B. Santoni, L. Yu, B. Xu, and V. Giurgiutiu, ‘Lamb Wave-Mode Tuning of Piezoelectric Wafer Active Sensors for Structural Health Monitoring’, J. Vib. Acoust., vol. 129, no. 6, pp. 752–762, Dec. 2007, doi: 10.1115/1.2748469.

[56] D. Samaratunga and R. Jha, ‘Lamb wave propagation simulation in smart composite structures’, p. 11, 2012.

[57] Y. Shen and V. Giurgiutiu, ‘Effective non-reflective boundary for Lamb waves: Theory, finite element implementation, and applications’, Wave Motion, vol. 58, pp. 22–41, Nov. 2015, doi: 10.1016/j.wavemoti.2015.05.009.

[58] https://www.britannica.com/science/elastic-wave

[59] https://gssc.esa.int/navipedia/index.php/Phase_%26_Group_Velocity

[60] https://www.nde-ed.org/EducationResources/CommunityCollege/Ultrasonics/Physics/modepropagation.htm

[61] https://www.nde-ed.org/GeneralResources/Formula/UTFormula/ultrasonicPrint.pdf

[62] http://sensor-works.com/how-do-piezoelectric-sensors-work/

[63] https://en.wikipedia.org/wiki/Piezoelectricity

[64] https://en.wikipedia.org/wiki/Crack_tip_opening_displacement

[65] https://www.wmtr.com/en.ctod.html#:~:text=The%20Crack%20Tip%20Opening%20Displacement%20or%20CTOD%20Test%20measures%20the,the%20essentials%20of%20the%20test.

[66] Mathworks, Introducing Machine Learning

[67] Mathworks, Getting Started with Machine Learning

[68] https://www.youtube.com/watch?v=hcLP-ZqQkT8&list=PLF0zuIDWcRPqLa9vdLN23cUfIWL_DP7dF&index=2

[69] Acoustic Emission Sensors and Preamplifiers (vallen.de)

[70] https://support.onscale.com/hc/en-us/articles/360015653612-The-Basics

[71] https://onscale.com/blog/meshing-in-fea-mesh-convergence/

[72] https://www.nde-ed.org/EducationResources/CommunityCollege/Ultrasonics/Physics/attenuation.htm

[73] Mathworks, Applying Supervised Learning

[74] https://www.ted.com/talks/fei_fei_li_how_we_re_teaching_computers_to_understand_pictures#t-1066202

APPENDIX A

In the following lines it is reported a script input file.

The author added this appendix in order for the interested readers to be able to look directly at the code and identify all the features addressed within Chapter 4 (e.g. hybrid mesh, keypoints, boundary conditions, gluing function, load definitions, etc.).

```
c mem 800 200 /* Allocate 800 megawords of memory - 3 GigaBytes (Not necessary for Windows Operating Systems).
c NOTE: MEM Command must be first command in file, if used.
rest no

c **************************************************************************
c
c                                   Generated Flex Input File
c
c **************************************************************************
c
c        DESIGNER                  : OnScale - Designer Generated
c        MODEL DESCRIPTION : This is a flex input file generated by the application
c        DATE CREATED            : 18 dic 2020
c        VERSION                    : 1.0
c **************************************************************************

mp
        omp * *                /* Number of CPUs to be used in the execution
        end

c TITLE TO CHANGE FOR EACH SIMULATION:
titl pristine_hybrid-Sim


c **************************************************************************
c
c                          Define User Variables
c
c **************************************************************************
c
c        These variables have been set by the user through the interface
c
c **************************************************************************

symb coordFactor = 1.0                              /* Coordinate conversion factor
```

```
symb timeFactor = 1.0                                /* Time conversion factor
symb dMassFactor = 1.0                               /* Mass conversion factor

c .STEP FILE TO CHANGE FOR EACH SIMULATION:
text geom_filename = 'pristine_hybrid_plate.STEP'          /* The source file for the model
symb geom_units_scale = 0.001                        /* Conversion from file units to meters
symb geom_scaleFactor = 1.0 * $geom_units_scale * $coordFactor
symb freqint = 600000.0                                    /* Defined Drive Frequency (Hz)

c ********************************************************************
c
c                         Define Meshing
c
c ********************************************************************
c
c         Set the variable for the approximate element size for the model. Must be
c         sufficient to represent the wavelengths of interest. Recommended that at least
c         15 elements per wavelength are used.
c
c ********************************************************************

symb freqdamp = $freqint
symb wavevel = 3100                  /* Group velocity of the slowest mode of interest (m/s)
symb wavelgth = $wavevel / $freqint                  /* Wavelength (m)
symb nmesh = 11                                      /* Elements per wavelength
symb box = $wavelgth / $nmesh


c ********************************************************************
c              Project Material List
c ********************************************************************

c -----------------------------------------------------------------
c Global variables used in all the material definitions
c -----------------------------------------------------------------

symb epvacm = 8.854e-12              /* dielectric constant for vacuum
symb freqdamp = 1.e6 if noexist      /* specified frequency for damping model
symb rmu0 = 1.2566e-6

symb #msg 5

*****************************************************
Damping matched at $freqdamp Hz
Redefine variable 'freqdamp' if device centre frequency
varies significantly from this value
*****************************************************
```

```
c ------------------------------------------------------------------------------------
c    Now define the axis transformation - only posx used in this file
c ------------------------------------------------------------------------------------

axis
    form vctr
    defn posx car2 0. 0. 0.    1.  0.  0.   0. 1. 0.
    defn negx car2 0. 0. 0.   -1.  0.  0.   0. 1. 0.
    defn posy car2 0. 0. 0.    0.  1.  0.   0. 0. 1.
    defn negy car2 0. 0. 0.    0. -1.  0.   0. 0. 1.
    defn posz car2 0. 0. 0.    0.  0.  1.   1. 0. 0.
    defn negz car2 0. 0. 0.    0.  0. -1.   1. 0. 0.
    end

c ---------------------------------------------------
c    Input material properties to program
c ---------------------------------------------------

matr

c --------------------------------------------------------------
c type : METAL
c name : Al7075-T651
c desc : Aluminium 7075-T6
c --------------------------------------------------------------

          wvsp off
          type elas
          prop Al7075-T651 2810.00 7.02941e+10 2.69549e+10 0.010000

c --------------------------------------------------------------
c type : PIEZO
c name : pzt5a
c desc : PZT5A Generic
c --------------------------------------------------------------

          symb rho  = 7750.00               /* density
          wvsp off
          type lean

c   define baseline stiffness coefficients (constant electric field)

          symb c11 = 1.203e+11
          symb c12 = 7.52e+10
          symb c13 = 7.51e+10
          symb c14 = 0
          symb c15 = 0
```

```
        symb c16 = 0
        symb c22 = 1.203e+11
        symb c23 = 7.51e+10
        symb c24 = 0
        symb c25 = 0
        symb c26 = 0
        symb c33 = 1.109e+11
        symb c34 = 0
        symb c35 = 0
        symb c36 = 0
        symb c44 = 2.11e+10
        symb c45 = 0
        symb c46 = 0
        symb c55 = 2.11e+10
        symb c56 = 0
        symb c66 = 2.26e+10

        prop pzt5a $rho
        $c11   $c12   $c13   $c14   $c15   $c16   $c22
        $c23   $c24   $c25   $c26   $c33   $c34   $c35
        $c36   $c44   $c45   $c46   $c55   $c56   $c66

c   define baseline dielectric coefficients

        symb epxx = 916
        symb epyy = 916
        symb epzz = 830

c   scale material properties as specified above

        symb aepxx = $epvacm * $epxx
        symb aepyy = $epvacm * $epyy
        symb aepzz = $epvacm * $epzz

        elec pzt5a $aepxx $aepyy $aepzz

c   define baseline piezoelectric coupling coefficients

        symb ex1 = 0                /* coupling constant
        symb ex2 = 0                /* coupling constant
        symb ex3 = 0                /* coupling constant
        symb ex4 = 0                /* coupling constant
        symb ex5 = 12.3             /* coupling constant
        symb ex6 = 0                /* coupling constant
        symb ey1 = 0                /* coupling constant
        symb ey2 = 0                /* coupling constant
        symb ey3 = 0                /* coupling constant
```

```
        symb ey4 = 12.3             /* coupling constant
        symb ey5 = 0                /* coupling constant
        symb ey6 = 0                /* coupling constant
        symb ez1 = -5.4             /* coupling constant
        symb ez2 = -5.4             /* coupling constant
        symb ez3 = 15.8             /* coupling constant
        symb ez4 = 0                /* coupling constant
        symb ez5 = 0                /* coupling constant
        symb ez6 = 0                /* coupling constant

        piez pzt5a 1 1 $ex1 1 2 $ex2 1 3 $ex3 1 4 $ex4 1 5 $ex5 1 6 $ex6 &
                   2 1 $ey1 2 2 $ey2 2 3 $ey3 2 4 $ey4 2 5 $ey5 2 6 $ey6 &
                   3 1 $ez1 3 2 $ez2 3 3 $ez3 3 4 $ez4 3 5 $ez5 3 6 $ez6

        rdmp pzt5a $freqdamp q 75 75 1e+06 * * *
        axis pzt5a posz                     /* relate materials local system to global system

c ----------------------------------------------------------------
c type : PIEZO
c name : pzt5a_2
c desc : PZT5A Generic (Copy of pzt5a material)
c ----------------------------------------------------------------
  copy pzt5a pzt5a_2

c ----------------------------------------------------------------
c type : PIEZO
c name : pzt5a_3
c desc : PZT5A Generic (Copy of pzt5a material)
c ----------------------------------------------------------------
  copy pzt5a pzt5a_3

c ----------------------------------------------------------------
c type : PIEZO
c name : pzt5a_4
c desc : PZT5A Generic (Copy of pzt5a material)
c ----------------------------------------------------------------
  copy pzt5a pzt5a_4

c -------------------------------------------------------------------------------
c type : METAL
c name : Al7075-T651_2
c desc : Aluminium 7075 T6 (Copy of Al7075-T651 material)
c -------------------------------------------------------------------------------
  copy Al7075-T651 Al7075-T651_2

        elec void $epvacm
        end
```

```
c **************************************************************************
c
c                               CAD Geometry Allocation
c
c **************************************************************************

c IMPORT STRUCTURED PARTS:
cad
        defn cadSTRUCT
        c .STEP FILE TO CHANGE FOR EACH SIMULATION:
        file 'pristine_hybrid_plate.STEP'
        scal $geom_scaleFactor
        mesh struct *
        useonly 1 2 3 4 5
        matr cad_part Al7075-T651 1
        matr cad_part pzt5a 2
        matr cad_part pzt5a_2 3
        matr cad_part pzt5a_3 4
        matr cad_part pzt5a_4 5
        surf surf_2_3 2 3
        surf surf_3_3 3 3
        surf surf_4_3 4 3
        surf surf_5_3 5 3
        surf surf_2_4 2 4
        surf surf_3_4 3 4
        surf surf_4_4 4 4
        surf surf_5_4 5 4
        dout * * 0 0 0
        end

c IMPORT GCON PART:
cad
        defn cadGCON
        c .STEP FILE TO CHANGE FOR EACH SIMULATION:
        file 'pristine_hybrid_plate.STEP'
        scal $geom_scaleFactor
        mesh gcon *
        useonly 6
        matr cad_part Al7075-T651_2 6
        surf surf_6_1 6 1 1                         /* left side boundary
        surf surf_6_2 6 2 1                         /* lower side boundary
        surf surf_6_3 6 3 1                         /* right side boundary
        surf surf_6_4 6 4 1                         /* upper side boundary
        symb tet_elem = $box / $geom_scaleFactor               /* size of the tetrahedral elements
        mopt_tet edgelen $tet_elem                             /* assign the tetrahedral elements
        dout * * 0 0 0
        end
```

```
c **************************************************************************
c
c                                   Geometry Locations (XYZ)
c
c **************************************************************************

c Scale Parameters
symb xmin = 0.0 * $coordFactor
symb xmax = 0.43 * $coordFactor
symb ymin = 0.0 * $coordFactor
symb ymax = 0.275 * $coordFactor
symb zmin = 0.0 * $coordFactor
symb zmax = 0.00185 * $coordFactor

c Determine lengths of the model
symb xlen = ( $xmax - $xmin )
symb ylen = ( $ymax - $ymin )
symb zlen = ( $zmax - $zmin )

c ***************************************************
c
c                                   Keypoints in the X-Direction
c
c ***************************************************

symb x1 = $xmin
symb x2 = 0.04915 * $coordFactor
symb x3 = 0.05565 * $coordFactor
symb x4 = 0.18955 * $coordFactor
symb x5 = 0.19605 * $coordFactor
symb x6 = 0.1993 * $coordFactor
symb x7 = 0.2103 * $coordFactor
symb x8 = 0.27775 * $coordFactor
symb x9 = 0.28425 * $coordFactor
symb x10 = 0.40475 * $coordFactor
symb x11 = 0.41125 * $coordFactor
symb x12 = $xmax
symb #get { idx } rootmax x

c ***************************************************
c
c                                   Keypoints in the Y-Direction
c
c ***************************************************

symb y1 = $ymin
symb y2 = 0.01715 * $coordFactor
```

```
symb y3 = 0.02365 * $coordFactor
symb y4 = 0.1469 * $coordFactor
symb y5 = 0.14915 * $coordFactor
symb y6 = 0.15565 * $coordFactor
symb y7 = 0.1579 * $coordFactor
symb y8 = 0.25005 * $coordFactor
symb y9 = 0.25655 * $coordFactor
symb y10 = $ymax
symb #get { jdx } rootmax y

c ***************************************************
c
c                              Keypoints in the Z-Direction
c
c ***************************************************

symb z1 = $zmin
symb z2 = 0.0004 * $coordFactor
symb z3 = 0.0008 * $coordFactor
symb z4 = 0.0012 * $coordFactor
symb z5 = 0.0016 * $coordFactor
symb z6 = 0.001725 * $coordFactor
symb z7 = $zmax
symb #get { kdx } rootmax z

c **************************************************************************
c
c                              Indices Locations (IJK)
c
c **************************************************************************

c Grid in I direction, using approximately element size of 'box' and at least 1 element
symb #keyindx i 1 $idx 1 $box 1
symb indgrd = $i$idx

c Grid in J direction, using approximately element size of 'box' and at least 1 element
symb #keyindx j 1 $jdx 1 $box 1
symb jndgrd = $j$jdx

c Grid in K direction, using approximately element size of 'box' and at least 1 element
symb #keyindx k 1 $kdx 1 $box 1
symb kndgrd = $k$kdx


c CALCULATE THE GRID EXTENT FOR THE HYBRID MESH:
symb #get { ngcon } gcon_nodescad 2 0 0
symb nnodes = int ( ( $ngcon ) / ( $i$idx * $k$kdx ) ) + 2
```

```
symb indgrd = $i$idx + $nnodes
symb igcon = $i$idx + 1
symb jndgrd = $j$jdx
symb jgcon = 1
symb kndgrd = $k$kdx
symb kgcon = 1


c **************************************************************************
c
c                              GCON Grid & Geometry Allocation
c
c **************************************************************************

grid $indgrd $jndgrd $kndgrd

c DEFINE THE GCON SECTION:
gcon
   defn cad cadGCON $igcon $jgcon $kgcon
   end

c DEFINE THE WHOLE GEOMETRY:
geom
        nchk
        keypnt $idx $jdx $kdx
        skew stnd
        skew cad
        end



c **************************************************************************
c
c                              Driving Conditions
c
c **************************************************************************
c
c       A number of predefined waveform functions can be accessed in OnScale. The DATA
c       HIST option is used below, other examples include wavelets, gaussians and step
c       functions. The manual details the function entries
c
c **************************************************************************

symb freqtimefunc_1 = 600000.0

data
c SELECT THE PROPER FOLDER DIRECTORY WHERE THE SIGNAL .dat FILE IS LOCATED:
```

```
        hist ud_timefunc_1 * 'C:\Users\Matteo Sarti\Documents\.flexfea\Thesis
        Simulations\Models to Run at 600 KHz\Version 3\hanning windowed burst at 600
        KHz.dat'
        end

func
        name timefunc_1
        hist ud_timefunc_1 *
        end

circ
        defn circuit_1
        elem rest sers 50.0
        prnt
        end


c **************************************************************************
c
c                         Primitive Definitions
c
c **************************************************************************

c CONSTRUCT THE "SITE":
site
        regn void
        cadmap cadSTRUCT 1 $i$idx 1 $j$jdx 1 $k$kdx
        end


c **************************************************************************
c
c                         Boundary Definitions
c
c **************************************************************************

c THE BOUNDARY CONDITIONS MUST BE DEFINED IN THIS WAY:
boun
        defn side_xmin absr
        node 1 1 1 $j$jdx 1 $k$kdx
        defn side_xmax absr
        node $i$idx $i$idx 1 $j$jdx 1 $k$kdx
        defn side_ymin absr
        node 1 $i$idx 1 1 1 $k$kdx
        defn side_ymax absr
        node 1 $i$idx $j$jdx $j$jdx 1 $k$kdx
        defn side_zmin free
```

```
          node 1 $i$idx 1 $j$jdx 1 1
          defn side_zmax free
          node 1 $i$idx 1 $j$jdx $k$kdx $k$kdx

          end

c GLUE THE BOUNDARIES OF THE TWO MESH ZONES TOGETHER:
glue
          defn glue_x_6 fuzz 0.01
          master $i6 $i6 $j4 $j7 1 $k5                /* keypoints delimiting the gcon part
          slave cad cadGCON surf_6_1

          defn glue_x_7 fuzz 0.01
          master $i7 $i7 $j4 $j7 1 $k5                /* keypoints delimiting the gcon part
          slave cad cadGCON surf_6_3

          defn glue_y_4 fuzz 0.01
          master $i6 $i7 $j4 $j4 1 $k5                /* keypoints delimiting the gcon part
          slave cad cadGCON surf_6_2

          defn glue_y_7 fuzz 0.01
          master $i6 $i7 $j7 $j7 1 $k5                /* keypoints delimiting the gcon part
          slave cad cadGCON surf_6_4

          end


c ************************************************************************
c
c                         Calculated Properties
c
c ************************************************************************
c
c         By default, Flex only calculates the minimum required data set, typically this
c         means only velocities. This is done for memory efficiency. Should other
c         properties be required (e.g. displacements, stresses, strains, pressure), then
c         these must be requested by the CALC command. The manual lists all these options
c
c ************************************************************************

calc

c ************************************************************************
c
c                         Piezoelectric Load Definitions
c
c ************************************************************************
```

```
piez
        wndo auto piez
        defn load_1 1.0
        cad cadSTRUCT surf_2_3 pzt5a * * * * * * *
        bc load_1 volt functimefunc_1 10.0 0.0

        defn load_2 1.0
        cad cadSTRUCT surf_3_3 pzt5a_2 * * * * * * *
        conn load_2 circuit_1 open

        defn load_3 1.0
        cad cadSTRUCT surf_4_3 pzt5a_3 * * * * * * *
        conn load_3 circuit_1 open

        defn load_4 1.0
        cad cadSTRUCT surf_5_3 pzt5a_4 * * * * * * *
        conn load_4 circuit_1 open

        defn load_5 1.0
        cad cadSTRUCT surf_2_4 pzt5a * * * * * * *
        bc load_5 grnd

        defn load_6 1.0
        cad cadSTRUCT surf_3_4 pzt5a_2 * * * * * * *
        bc load_6 grnd

        defn load_7 1.0
        cad cadSTRUCT surf_4_4 pzt5a_3 * * * * * * *
        bc load_7 grnd

        defn load_8 1.0
        cad cadSTRUCT surf_5_4 pzt5a_4 * * * * * * *
        bc load_8 grnd

        slvr drct
        end

time * * 0.8


c ***************************************************************
c
c                         Process Model
c
c ***************************************************************
c
```

```
c        Issue process (PRCS) command. Checks model integrity, and calculates stable
c        time step. NOTE: Process command must always be issued.
c
c ************************************************************************

symb #msg 1
Checking Model Integrity......
prcs

c MODEL THE PREVIEW GRAPHS:
grph
         nvew 1
         plot matr
         end
grph
         type stnd
         nvew 3 2
         line on
         plot matr
         zoom 15
         plot glue_master
         plot glue_slave
         end

term


c ************************************************************************
c
c                          Choose Time Histories To Store
c
c ************************************************************************
c
c        Save field (such as displacement or pressure) from a node or element for all
c        time steps with POUT command. Histories are referenced by order of
c        specification. Histories will be saved in the Flex History file (flxhst).
c
c ************************************************************************

pout
         form matlab
         rate 1
         hist functimefunc_1
         histname electrode vq all
         end
```

```
c ****************************************************************************
c
c                         Run the Model
c
c ****************************************************************************
c
c                 Specify the number of time steps to be run. Can be set to auto by using
c                 'Ringdown'.
c
c ****************************************************************************

c User defined runtime for the model (input part)
symb #get { step } timestep
symb ttime = 0.00014
symb nexec = $ttime / $step
symb nloops = 40
symb nexec2 = $nexec / $nloops

c PLOTTING SET UP:
grph
        type stnd
        line off
        nvew 3 1
        set imag avi
        end


c Create run plot procedure
proc plot save

c Run some timesteps
exec $nexec2

c PLOT THE RUNNING MODEL:
grph
        mirr off * *
        plot xvel
        plot yvel
        plot zvel
c       plot aprs aprst
        imag
        end

end$ proc

c Run model then wait
proc plot $nloops
```

```
term                                  /* End of the running phase

data
c .flxdato FILENAME TO CHANGE FOR EACH SIMULATION:
          file out 'pristine_hybrid-Sim.flxdato'
          out modl
          end


c ***********************************************************************
c
c                            Save symbol variables to file for later use
c
c ***********************************************************************

symb #get { labl } jobname
symb #save '$labl.symb'
stop                                  /* return to command prompt
```