Numerical Evaluation of Classification Techniques for Flaw Detection

by

Suriyapriya Vallamsundar

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Civil Engineering

Waterloo, Ontario, Canada, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Suriyapriya Vallamsundar

Abstract

Nondestructive testing is used extensively throughout the industry for quality assessment and detection of defects in engineering materials. The range and variety of anomalies is enormous and critical assessment of their location and size is often complicated. Depending upon final operational considerations, some of these anomalies may be critical and their detection and classification is therefore of importance. Despite the several advantages of using Nondestructive testing for flaw detection, the conventional NDT techniques based on the heuristic experience-based pattern identification methods have many drawbacks in terms of cost, length and result in erratic analysis and thus lead to discrepancies in results.

The use of several statistical and soft computing techniques in the evaluation and classification operations result in the development of an automatic decision support system for defect characterization that offers the possibility of an impartial standardized performance. The present work evaluates the application of both supervised and unsupervised classification techniques for flaw detection and classification in a semi-infinite half space. Finite element models to simulate the MASW test in the presence and absence of voids were developed using the commercial package LS-DYNA. To simulate anomalies, voids of different sizes were inserted on elastic medium. Features for the discrimination of received responses were extracted in time and frequency domains by applying suitable transformations. The compact feature vector is then classified by different techniques: supervised classification (backpropagation neural network, adaptive neuro-fuzzy inference system, k-nearest neighbor classifier, linear discriminate classifier) and unsupervised classification (fuzzy c-means clustering). The classification results

show that the performance of k-nearest Neighbor Classifier proved superior when compared with the other techniques with an overall accuracy of 94% in detection of presence of voids and an accuracy of 81% in determining the size of the void in the medium. The assessment of the various classifiers' performance proved to be valuable in comparing the different techniques and establishing the applicability of simplified classification methods such as k-NN in defect characterization.

The obtained classification accuracies for the detection and classification of voids are very encouraging, showing the suitability of the proposed approach to the development of a decision support system for non-destructive testing of materials for defect characterization.

Acknowledgements

I cannot thank, in any adequate measure, the first teachers I had in my life, my dear parents who brought me into this world with great amounts of love and affection and taught me the first most important lesson “Patience, endurance and hard work always pays you back in the end”. I am indebted as ever to my father, Dr. Vallam Sundar for having instilled in me the passion to follow his foot-steps as a research scientist in the field of Civil Engineering and for having treated me as his research colleague through proving me with plentiful enriching research-related discussions. No measure of gratitude would be adequate for my very beloved mother, Poornima Sundar, my constant support system who saw me through each and every day of grueling research work and gave me all her love and blessings unconditionally. I would like to thank my cute twin sister, Banupriya, who was with me in all times of happiness and sorrows giving me constant encouragement and support, but for her I would have never managed to live here.

Most sincere thanks are owed to my supervisors, Dr.Giovanni Cascante and Dr. Kumaraswamy Ponnambalam for their kindness, generosity and patience in guiding and seeing me through each day of my research life. Their valuable feedback and enthusiastic questions gave me a better insight into the quality of this work. I acknowledge and appreciate their efforts for having given me their valuable time so generously.

I would like to extent my heartfelt thanks to all my friends, Gowri Shanthi, Sathyan.R.M, Papia Sultana, Arun Veeramany, Sanjay Kalyanasundaram, Bharathi Sambasivan, Sonika Soor, Subam Basuthkar, Guru Prakash, in no particular order of preference, for having encouraged and cheered me up whenever things got hard and life

looked bleak. Thanks a lot to all of you for having made my experience in Canada a wonderful memory and I will carry all these precious memories in my heart all through my life.

Special thanks to my NDT group members, Simon Berube, Fernando Tallavo, Yanjun Yang, Ali Nasseri Moghaddam, Zhiyong Jiang, Leigh Davis for all their research-related help along with numerous rewarding conversations that I would cherish for a long time to come. I owe a huge thank you to Marguarite Knechtel, administrative co-coordinator of graduate studies.

Last, however not the least, my grateful thanks go out to all the numerous teachers who have had a great share in shaping me and my career. I would not be what I am today without their presence in my life and each footprint they have left behind in my life will be treasured forever.

To my beloved parents, grandparents
and my teachers

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Forward

While *destructive testing* usually provides a more reliable assessment of the state of the test object than *nondestructive testing* (NDT), destruction of the test object usually makes this type of test not suitable, for example, of existing structures. The American society for NDT defines it as an examination of an object or material in a manner that will not impair its future usefulness. Non-destructive techniques have been recently gaining more importance due to the rapid development in technology and due to several types of materials and systems that could be tested without causing any damage or destruction. The number of NDT methods is large and continues to grow and can be used for

- Detecting presence of any internal or external imperfections or anomalies
- Determine structure, composition or material properties
- Measure geometric characteristics

Destructive testing is also inappropriate in many circumstances, such as forensic investigation. Although there is a tradeoff between the cost of the test and its reliability (cheaper NDT usually are not as reliable as the more expensive destructive testing), practical situations favor a strategy in which most test objects are inspected nondestructively; destructive testing, on the other hand, may be performed on a sampling of test objects that is drawn randomly for the purpose of characterizing the testing reliability of the nondestructive test.

During their service lives, many industrial components need regular nondestructive tests to detect damage that may be difficult or expensive to find by everyday methods.

These NDT techniques are very valuable in many applications including the assessment of the conditions of structures like pavements, bridges, quality monitoring in manufacturing, and in testing of nuclear reactors and transducers embedded in biomedical systems, etc. The following are a few examples:

- Detection of cracks in buildings, bridges, aircrafts and even space shuttles, implementation of effective and appropriate repair work and also for continuous monitoring of their performances;
- Integrity monitoring of onshore, port, coastal and offshore structures
- Status of underground onshore as well as offshore pipelines are often subjected to corrosion and stress corrosion cracking can be monitored which enables planning and undertaking appropriate protection measures
- Monitoring the status of pipes in industrial plants that may be subjected to erosion and corrosion from the products they carry
- Monitoring the properties of concrete structures that may be weakened if the inner reinforcing steel has corroded
- Condition assessment of wire ropes in suspension bridges that are often subjected to vibrations due to heavy loads and weather conditions so that appropriate repair work can be undertaken for broken wires and other damages.

1.2 Statement of the Problem

Detecting defects in materials is one of the most important applications of the nondestructive testing. A variety of imperfections can arise in engineering materials due to their manufacturing. Depending upon final operational considerations, some of these may be critical and their detection and classification is therefore of importance. The term 'imperfection' refers to any flaw, fault or irregularity in the structure of the material that may cause weakness or failure in the functioning of the product or system with which it is associated. Researchers continue to find novel approaches of applying physics and other scientific disciplines to develop NDT methods. Some of the widely used NDT methods include ultrasonic testing, visual inspection, electromagnetic or eddy-current testing, radiography, magnetic particle testing and penetrate testing.

Ultrasonics is one of the mostly used NDT techniques for the detection, localization and measurement of the imperfections present in engineering materials under inspection. Ultrasonic testing is based on the transmission of high frequency sound waves into materials to detect imperfections or to locate changes in the material properties. The reflected echoes from the defects or the imperfections are received by the ultrasonic transducers and are recovered for further processing.

Despite the advantages of using ultrasonics for flaw detection (ultrasonic has a high probability of detection and a low number of false results) the detection and classification of defects is frequently questionable, the reason being identification of types of the defects that depend exclusively on the experience and the knowledge of the operator. The human eye is unparalleled in its ability to recognize significant patterns after a period of suitable training and experience. However, an overwhelming flow of data often reduces

the effectiveness of the human eye in extracting relevant information for decision making. Thus, the conventional NDT techniques based on the heuristic experience-based pattern identification methods have many drawbacks in terms of cost, length and result in erratic analysis and thus lead to discrepancies in results.

Various modern signal processing tools, statistical and soft computing techniques have come into practice to overcome these problems. The use of these processing tools for the evaluation and classification operations should overcome the above mentioned drawbacks by enhancing the measurement accuracy and reducing the work load on the operators, thereby, enabling them to concentrate more efficiently on the evaluation of the results. The main advantages of using these techniques for computational purposes is the development of an automatic decision support system for the defect characterization that offers the possibility of an impartial, standardized performance.

1.3 Research Objectives

The objective of the present study was to develop a classification scheme for defect characterization using ultrasonics in a homogeneous half-space medium. The first step consists of constructing numerical models using commercial finite element code (LS-DYNA) to simulate the propagation of Rayleigh waves in a homogeneous half-space medium. The next step consists of introducing rectangular voids of varying sizes to simulate defects. A sufficient number of samples were generated by changing the input loading conditions in order to maintain enough variance in the dataset. A total of 37 features were extracted from the samples by applying several transformations.

Two types of learning algorithms were employed for defect characterization; supervised/classification and unsupervised/clustering using both soft computing and

statistical methods. The results are assessed in form of the confusion matrix which gives information about the predicted and actual classification. A comparison between the performances of various techniques used for classification was performed.

1.4 Thesis Organization

This thesis contains six chapters and is organized in the following order:

Chapter one provides a brief introduction to the problem addressed in this study and outlines the research objectives.

Chapter two provides a review of seismic wave theory, including body and surface waves. The concepts of wave velocity and attenuation are introduced, and different types of attenuation and damping are discussed. A review of the different signal processing techniques used in this study was presented. Lastly, an overview of the commonly employed ultrasonic testing techniques used for flaw detection was presented. Each method is introduced and a brief summary of the method was provided.

Chapter three provides an overview of the soft computing methods which includes Fuzzy Logic and Artificial Neural Networks. A detailed review of the different classification and clustering techniques using both soft computing and statistical methods applied for detection and classification of defects in this study was provided. Literature review of the classification/clustering techniques employed for defect characterization was presented at the end.

Chapter four reviews the different numerical methods used for modeling wave propagation problems. The principles of the finite element technique and the basic concepts of the numerical finite element code (LS-DYNA) that is used in this study are described. A literature review of the numerical work for flaw detection was presented.

This chapter continues with the description of the basic numerical model developed using LS-DYNA. The set-up, calibration, and validation of the model with theoretical solution are discussed. The finite element modeling of the voids along with a discussion on the generation of the sample data set was discussed. Lastly, contour plots of the surface responses along the surface of all the models in the presence and absence of voids are presented.

Chapter five discusses the feature extraction procedure and the various features extracted from the samples. The feature preprocessing tools and assessment of the classification results are discussed. The results obtained from supervised (classification) and unsupervised (clustering) techniques in defect detection and classification are presented in detail. Related concepts, such as sensitivity analysis, optimal selection are also discussed. Lastly, the critical features which give the best classification results are presented.

Chapter six summarizes the conclusions of this study. Further, recommendations for future studies are provided in this chapter.

Chapter 2

Theoretical Background

2.1 Introduction

The chapter begins with a brief introduction to wave propagation, followed by a review of the seismic wave theory which includes body waves and surface waves. The second section provides a review of the different signal processing techniques used in this study. The final section briefly discusses some of the commonly used ultrasonic testing techniques used for flaw detection in materials.

2.2 Seismic waves

2.2.1 Background

Mechanical waves are generated by the oscillatory motion of particles in a material media. The velocities at which these waves propagate depend on the material and inertial properties of the medium. These waves exist in various modes of propagation that are defined by the type of motion involved. The different types of waves can be broadly classified into body waves and surface waves. Other modes of wave propagation exist, but they are comparatively of minor importance.

Body waves

Body waves that can travel through the interior of the solid media are of two types: longitudinal and shear waves. Longitudinal waves also known as P, primary, pressure or compressional waves consist of alternate compressions and dilations along their propagation paths. These alternate compressions and dilations are directed in the direction of wave propagation. Thus, the direction of wave propagation is parallel to the direction of particle motion as shown in Figure 2.1. They are also called volumetric

waves because the volume of the solid fluctuates as they propagate. They are the fastest propagating waves in solids and can travel through both solids and liquids. These waves attenuate at a rate of 1/r where, r is the distance from the origin of the wave. Their velocity is a function of the elastic modulus E, Poisson ratio υ and the density ρ of the medium and is given by:

$$V_p = \sqrt{\frac{E(1-\upsilon)}{\rho(1+\upsilon)(1-2\upsilon)}} \tag{2.1}$$

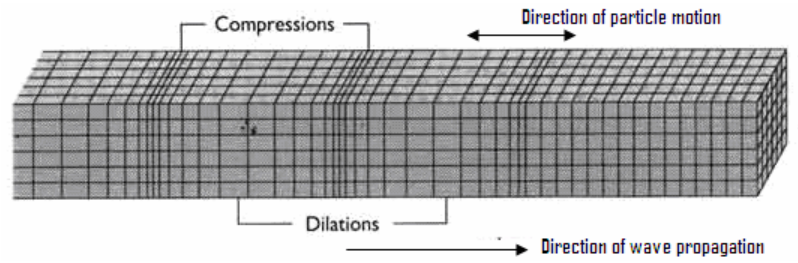


Figure 2.1: Propagation of the P-waves (Nisee, 1998)

Shear Waves

Shear waves also known as S, secondary or transverse waves have their particle motion perpendicular to the direction of wave propagation as shown in Figure 2.2. The main restoring force for these waves comes from shearing effects of the medium which, implies that a solid material medium is required for their effective propagation. Because liquids do not offer resistance to shear, they cannot sustain a shear wave. These waves are slower than the longitudinal waves but they attenuate at the same rate due to the effect of geometrical spreading. Unlike longitudinal waves, shear waves cause rotation in the media without any volume change as they propagate. Their velocity is a function of the density ρ and shears modulus G of the medium and is given by:

$$V_s = \sqrt{\frac{G}{\rho}} \tag{2.2}$$

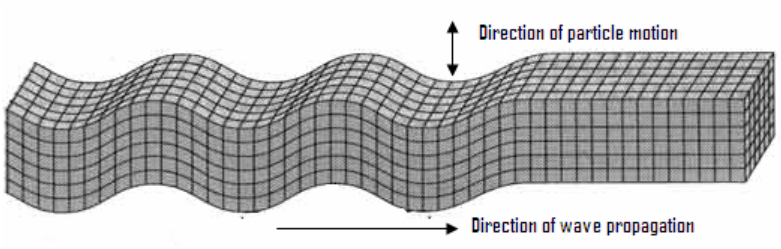


Figure 2.2: Propagation of the Shear waves (Nisee, 1998)

Shear waves can be polarized. If the direction of propagation and particle motion is perpendicular to the incident surface, the wave is said to be vertically polarized and is called SV wave. However if the particle motion is parallel to the incident surface, the waves are said to be horizontally polarized and are called SH waves.

Surface waves

The theoretical prediction of the existence of the surface waves occurred a century ago by the profound contributions of Rayleigh, Lamb and Love. The observation of the propagation of seismic waves over the surface of the earth led to the development of the experimental evidence of the occurrence of surface waves. This was obtained by examining the unusual ability of the seismic waves to travel along curved surfaces and the phenomenon of the energy decay with increased depth (Rose and Joseph, 1999).

The surface waves are generated due to the interaction between the body waves and the surface of the medium and their energy is concentrated closer to the surface. Another important property of surface waves is their dispersion in a layered medium, which, implies that different frequencies travel with different velocities as these waves propagate through different layers. Shorter wavelengths are affected by the near surface properties

of the medium, whereas, the longer wavelengths are affected by both the near and deeper properties. This property of the surface waves makes them very useful in the non-destructive testing techniques in evaluating the properties of the material media. Surface waves travel slowly compared to the body waves.

Rayleigh Waves

The first kind of surface wave called Rayleigh wave is generated due to the interaction between the P and SV waves at the free surface of the medium which are shown in Figure 2.3. The existence of the Rayleigh waves was predicted in the year 1885 by Lord Rayleigh, in whose name such waves are named after. These waves travel along the surface of the medium penetrating to a depth of approximately one wavelength. Their deformation of the particle motion is retrograde at the surface that changes into pro-grade at deeper levels of the media. They have the greatest amplitude and attenuate at the rate of $1/\sqrt{r}$. Thus, the particle motion of this type of surface waves is confined to the near-surface region because of the rapid attenuation of the amplitude of particle motion with respect to depth below the free surface. At a depth equal to 1.5 times the wavelength, the vertical component of the amplitude is approximately equal to 10% of the original amplitude at the ground surface (Santamarina, 2001). In a homogeneous medium, R-wave velocity (V_R) is constant and is independent of frequency (f) (Nazarian, 1984). Thus, the velocity of R-wave is given by the following relationship:

$$V_R = f\lambda \tag{2.3}$$

where λ is the corresponding wavelength. However, in a layered medium, where, there is difference in the material properties at different depths, R-wave velocity changes with respect to frequency (wavelength). This phenomenon of different velocities propagating

with different frequencies is known as dispersion. An approximation of the Rayleigh wave velocity is given by the following empirical relation (Achenbach, 1973):

$$C_r = \frac{0.87 + 1.12\upsilon}{1+\upsilon} C_S \tag{2.4}$$

where υ is the Poisson's ratio and V_S is the shear wave velocity

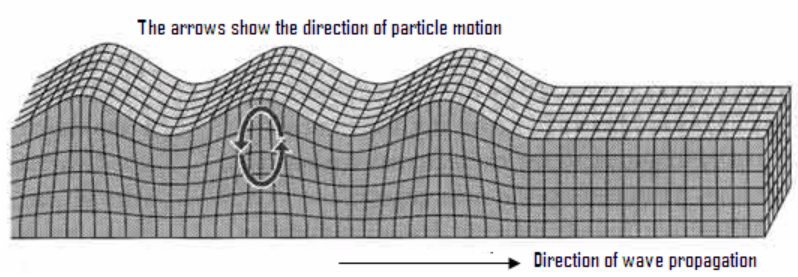


Figure 2.3: Propagation of the Rayleigh waves (Nisee, 1998)

In an isotropic, elastic half-space, the ratio of Rayleigh wave to shear wave velocity increases with an increase in the Poisson ratio. This ratio varies from 0.87 to 0.96 for values of Poisson ratio from 0.0 to 0.5.

Love Waves

The second type of surface waves called the Love waves was predicted by Love in the year 1911.These horizontally polarized waves are generated due to the interaction of the SH waves with the free surface of the medium and are shown in Figure 2.4. The velocity of the Love waves is slower than the body waves but faster than the Rayleigh waves. Love (1911) showed that these waves exist only in a layered media, in which, the shear wave velocity increases with an increase in depth. Their deformation is parallel to the surface and decreases exponentially with depth. Like Rayleigh waves, Love waves are also dispersive implying that the wave velocity is dependent on frequency, with low frequencies normally propagating at higher velocity. Depth of penetration of the Love

waves is dependent on frequency, with lower frequencies penetrating to greater depth of the media.

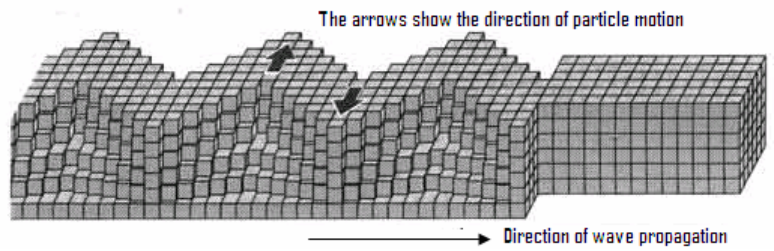


Figure 2.4: Propagation of the Love waves (Nisee, 1998)

The propagation of body waves and surface waves (Rayleigh waves) away from a vertically vibrating circular source at the surface of a homogeneous, isotropic, elastic half-space is shown in Figure 2.5.

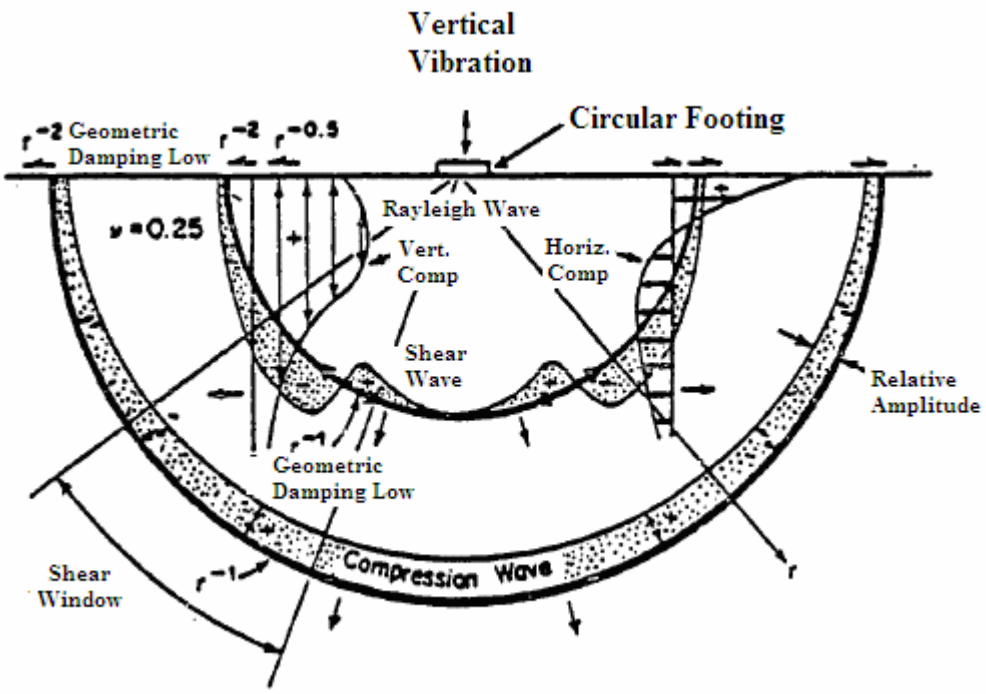


Figure 2.5: Distribution of Rayleigh, shear and compression wave displacement from a circular footing on a Homogeneous, isotropic, elastic half space (Richard et al. 1970)

It was found that approximately 67 percent of the input energy propagates in form of Rayleigh waves, while, the percentage of the input energy that propagates in form of

shear and compression waves is 26 and 7 respectively (Miller and Pursey, 1955). Body waves propagate radially outwards from the source and travel in a spherical wavefront. Rayleigh waves propagate along a cylindrical wavefront near the surface. At the surface of an elastic half-space, body wave attenuate at a rate of r^{-2} and Rayleigh waves attenuate at a rate of $1/\sqrt{r}$ where, 'r' is the distance from the source.

2.2.2 Velocities in wave motion

General

Mechanical waves propagate due to the simple harmonic motion of the particles about their equilibrium position within the material media. But these particles do not progress through the medium with the waves. The three velocities of wave motion namely particle, wave or phase and group velocities are quite different from one another although they are connected mathematically as explained in the following sections.

Particle velocity

The simple harmonic velocity of the particles about their equilibrium position is the particle velocity. All other kinds of waves are produced as a result of the superposition of longitudinal and shear wave particle velocity components.

Wave or Phase velocity V_{ph}

This is defined as the rate at which plane of equal phase, crests or troughs progress through a medium. It is also defined as the speed with which the shape of the wave moves is the phase velocity. A wave can be expressed as the function

$$\Psi = A\cos(\omega t - \kappa x) \tag{2.5}$$

The above equation is the solution to the one-dimensional wave equation where A is the amplitude of the wave, ω is defined as $2\pi f$, where, f is the frequency and κ is defined as $2\pi/\lambda$ where λ is the wavelength. The phase velocity is given by

$$V_{ph} = \frac{\omega}{\kappa} \tag{2.6}$$

Thus V_p represents the velocity of each frequency component in the propagating wave.

Group velocity V_g

A number of waves of different frequencies, wavelengths, and velocities may be superimposed to form a group. The speed at which the energy in the wave group is transmitted is referred to a the group velocity and is given by (Graff, 1991)

$$V_g = \frac{d\omega}{d\kappa} = V_{ph} + \kappa \frac{dV_{ph}}{d\kappa} \tag{2.7}$$

Thus group velocity is the rate at which the variations in the shape of the wave's amplitude propagate through space or the velocity of energy propagation is referred to as the group velocity. The velocity of the waves within a group may be different that of the velocity of wave group. When the wave velocity is independent of the frequency or in a non-dispersive media, i.e., when $\frac{dV_{ph}}{d\kappa} = 0$ the group velocity is the same as the phase velocity. When the wave velocity decreases with frequency, i.e., when $\frac{dV_{ph}}{d\kappa} < 0$ the group velocity is less than the phase velocity and the individual wavelets build up in front of the group and disappear in the rear end of the group. On the other hand when $\frac{dV_{ph}}{d\kappa} => 0$ the group velocity is greater than the phase velocity and the individual wavelets build up at the back of the group and disappear at the front.

2.2.4 Damping and attenuation in seismic waves

Background

Attenuation or damping of waves is a complex phenomenon resulting from the interaction of many processes which contribute to the reduction in the amplitude or the energy content of the wave form in time or space domain.

The attenuation of seismic waves occurs through the following processes:

- Spreading or focusing of the waves called the geometric attenuation
- Absorption of energy called material or intrinsic attenuation
- Dispersion
- Reflection and transmission at interfaces
- Mode conversion
- Scattering by interfaces, material inhomogeneities and defects

The first three processes are due to the interaction of the waves with the material media and the rest are as a result of the interaction between the waveforms.

Geometric or radiation attenuation

The geometric attenuation occurs due to the propagation of the waveforms over a greater area, which, results in the spreading of energy. This leads to a decrease in the energy per unit volume as the area of the spread increases. As mentioned earlier, body wave spread in a three dimensional spherical pattern, whereas, surface waves spread in a two dimensional cylindrical pattern. Geometric attenuation causes the amplitude of the body waves to attenuate at the rate of 1/r, whereas, that of the surface waves at the rate of $1/\sqrt{r}$ with increasing distance. Thus, surface waves geometrically attenuate more slowly than body waves because they propagate in two dimensional spaces. The geometric

attenuation changes only the amplitude or the energy content of the wave but not the wave speed.

Material or intrinsic attenuation

Material damping occurs through the conversion of elastic energy into other forms of energy such as heat accompanied by a reduction in the amplitude of the wave. This occurs through internal friction due to the non elastic response of the material. These are as a result of the molecular behaviour such as viscoelastic or non-elastic and plastic behaviour as well as due to friction at boundaries, between molecules and grains and in-homogeneities. These effects are dependent on the stress and strain history in contrast to the elastic behaviour which depends only on the current stress.

Dispersion

Dispersion is a phenomenon in which waves of different frequencies propagate with different velocities. Both Rayleigh and Love waves are dispersive. The dispersion occurs when the stiffness of a site varies with depth. The velocity of surface wave thus varies and becomes a function of wavelength (or frequency).

The frequency dependence of wave velocities results in the high frequency components propagating with different velocities when compared with the low frequency components. When the high frequency components propagate with high velocities, the ending phases of the waveforms becomes more smoothed or stretched. Similarly, the propagation of low frequency components with low velocities result in the ending phases of the waves becoming more compacted (Lempriere, 2002). Thus, dispersion changes the amplitude of the successive peaks but not the energy content. High frequencies (or shorter wavelengths) help in the investigation of near surfaces, and thus their wave

velocity depends on the properties of surface material only. Low frequencies (or long wavelengths) help in deeper investigation with their wave velocities affected by material properties of deeper layers. This phenomenon is utilized in the Spectral analysis of surface waves technique (SASW) to determine the thickness of different layers in a layered medium and to evaluate their elastic modules (Nazarian, 1984).

Reflection and transmission at interfaces

When a wave encounters an interface between two materials, reflection and transmission of the wave occurs across the interface. The nature and the distribution of energy between the reflected and transmitted waves can be determined by the theory of elasticity (Kramer, 1996). A change in the speed and the amplitude is observed in the waves produced at the interface. The speed of the wave is determined by the properties of the two media, characteristics and the direction of the incident wave, whereas, the amplitude is determined by the difference between the impedance of the two media. A wave incident on an interface at an angle not normal to the interface results in oblique reflection and transmission. Oblique transmission is referred to as refraction, since; the angle of transmission is not equal to the angle of incidence. The fraction of the incident wave intensity that is reflected and transmitted can be determined by the reflection and transmission coefficients respectively given by:

$$R = \frac{(Z_2 - Z_1)}{(Z_2 + Z_1)} \tag{2.8}$$

$$T = 1 - R \tag{2.9}$$

where Z_1 and Z_2 denote the acoustic impedance (which is the product of the density and the velocity of the material) of the two materials.

Mode conversion

When the wave encounters an interface between materials of different acoustic impedance at an incident angle not normal to the interface, other types of waves in reflection or transmission are generated. This process is referred to as the mode conversion.

Scattering by material inhomogeneities and defects

When a wave encounters a discontinuity in form of an inhomogeneity in the medium, scattering of waves occur. Discontinuity is referred to a region with properties different form those of the surrounding material medium. These inhomogeneities extract energy from the advancing wave front thereby, resulting in reflection and transmission in multiple directions in form of scattered waves. The amount of energy scattered depends on the size of the inhomogeneity and the wavelength of the propagating wave. Scattering is in form of reflection, refraction and mode conversion when the wavelength of the propagating wave is smaller than the size of the in-homogeneity, whereas, if the wavelength is greater than the size, then, the wave propagates in the medium with properties reflecting the combined properties of the medium and the inhomogeneity. When the size of the inhomogeneity is held a constant, the proportion of the energy scattered is inversely proportional to the wavelength and is greater at any given wavelength for shear waves than that for p-waves. This is due to the fact that shear waves has a shorter wavelength due to their lower velocity than p-waves for any given frequency and are therefore attenuated more than the p waves.

2.3 Signal Processing

2.3.1 Introduction

A wide variety of physical phenomena can be described by the variation of a function with respect to one or more independent variables. The desired part of this function is referred to as the signal and the undesired part as noise. Although, signals can be represented in many ways, in all cases, the information in a signal is contained in a pattern of variation of some form. The process of extracting information contained in a signal is known as signal processing.

Domain of signal analysis

In this dissertation, signals would be analyzed in the following five domains:

- Time domain analysis consists of decomposing the signal into shifted and scaled impulses (or step functions).
- Frequency domain analysis consists of decomposing the signal into different frequency components given by a sum of sinusoidals.
- Wavelet transformation through which the signal is represented as a summation of wavelets.
- Cepstrum coefficients are obtained by applying inverse Fourier transforms to log magnitude Fourier spectrum.

2.3.2 Time domain analysis

General concepts

The most frequently used independent variables to represent the variation of the information contained in a signal are the time or spatial coordinates. In the time domain, signals can be recorded in two types: continuous or discrete time signals. In continuous

time signals, the independent variable is continuous and thus, these signals are defined for a continuum of values of the independent variable. They can also be referred to as the uninterrupted observation of the signal in time or space and are denoted by x(t). On the other hand, intermittent observation of the signal in time or space gives rise to discrete time signals. These time signals are defined only at discrete time and thus, for discrete signals the independent variable takes on only a discrete set of values. An important branch of discrete time signals is produced by sampling the continuous time signals at regular intervals. The discrete time signal is denoted by:

$$x(i\Delta t) = x(t)\big|_{t=i\Delta t} \tag{2.10}$$

$x(i\Delta t)$ is the sampled signal of the continuous signal x(t). Δt is called the sampling interval and i is called the time index. Figure 2.6 shows a continuous signal and its discrete counter part.

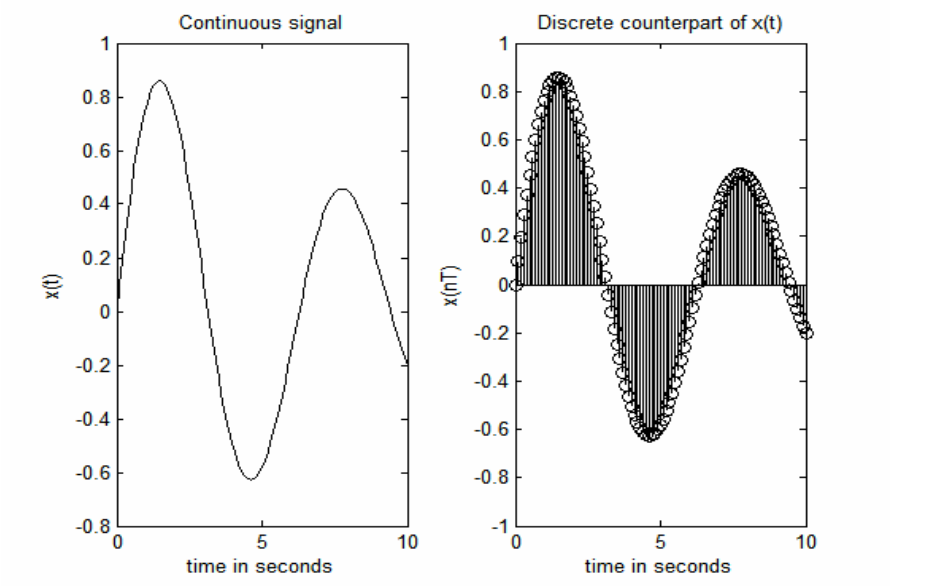


Figure 2.6: A continuous signal and its discrete counter part

Sampling

Sampling is the process of converting a continuous time signal into a discrete time signal. During sampling, the continuous time signal is defined at discrete instants of time and the time interval between two subsequent sampling instants is called sampling interval. This process of sampling is based on the sampling theorem.

The sampling theorem ensures that during sampling of a continuous signal, the sampling rate is kept sufficiently small so that the original signal can be completely recovered from its samples. This is essential in order to avoid the effect of under sampling or aliasing of the signals. Aliasing occurs when the sampling interval is greater or equal to half period $\Delta t \geq T/2$ and it takes place either in the time or in the spatial domain. The effect of undersampling or aliasing of the signal is shown in Figure 2.7.

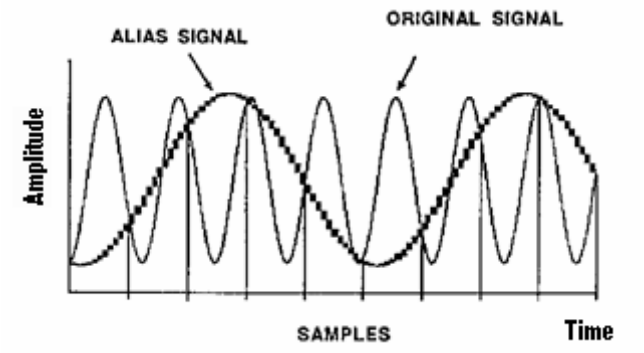


Figure 2.7: Aliasing of the signal (Nasseri, 2006)

It can be seen that the original signal is undersampled and appears aliased into a signal of low frequency content. According to the sampling theorem, the sampling rate (Δt) or the sampling frequency (f_s) should satisfy the Nyquist criterion:

$$f_s = \frac{1}{\Delta t} > f_{ny} = \frac{2}{T} = 2\Delta t \qquad (2.11)$$

where *T* is the largest period present in the signal. That is, to avoid aliasing, the sampling frequency must be greater than twice the highest frequency present in the signal. In practice, a minimum of *~10* points per cycle is recommended (Santamarina, 2001). Antialiasing filters are used to remove the high frequency components greater than $1/(2\Delta t)$ before it is sampled to reduce the effect of aliasing.

2.3.3 Frequency domain analysis

Fourier series and Fourier transforms

Analysis in the frequency domain has proven to be a helpful analytical tool for studying linear systems for many decades. In addition to the representation of the signal in a functional form or by a set of sample values in time, signals can also be characterized in their frequency domain in terms of their frequency content or frequency spectrum. The frequency content of a signal shows what frequencies are present in the signal and can be obtained by decomposing the signal into frequency components given by a sum of sinusoids.

In general, the frequency spectrum is complex values and can be represented in terms of real and imaginary components or in terms of magnitude and phase spectrum. In case of periodic signals, the frequency spectrum can be generated by computing the Fourier series. Fourier series is named after the French physicist Jean Baptiste Fourier (1768-1830) who was responsible for predicting that any periodic waveforms could be represented by a sum of sinusoids (Kamen, 2000). The Fourier series of any periodic signal x(t) of period T is given by (Oppenheim and Wilsky,1983):

$$x(t) = a_0 + \sum_{n=1}^{\infty}\left[a_n \cos\left(n\Omega_0 t\right) + b_n \sin\left(n\Omega_0 t\right)\right] \qquad (2.12)$$

Therefore any periodic signal can be represented as an infinite sum of sine and cosine functions which are themselves periodic signals of angular frequencies *0, Ω_0....$k\Omega_0$*.....This series of sine and cosine terms are referred as trigonometric Fourier series. The coefficient a_0 is called the DC component and is given by;

$$a_0 = \frac{1}{T}\int_{t_0}^{t_0+T} x(t)\,dt \tag{2.13}$$

coefficients a_n and b_n are given by:

$$a_n = \frac{2}{T}\int_{t_0}^{t_0+T} x(t)\cos\left(n\Omega_0 t\right)dt \tag{2.14}$$

$$b_n = \frac{2}{T}\int_{t_0}^{t_0+T} x(t)\sin\left(n\Omega_0 t\right)dt \tag{2.15}$$

[$a_1$cos (Ω_0t) + $b_1$sin (Ω_0t)] is the first harmonic, *[$a_2$cos (2 Ω_0t) + $b_2$sin (2 Ω_0t)]* the second harmonic and *[$a_n$cos (n Ω_0t) + $b_n$sin (n Ω_0t)]* the nth harmonic.

A periodic signal x(t) can be represented by Fourier series only if satisfies the Dirichlet conditions (Oppenheim and Wilsky,1983) given by

1. *x(t)* has at most a finite number of discontinuities in one period.
2. *x(t)* has at most a finite number of maxima and minima in one period.
3. *x(t)* is bounded
4. Over any period, *x(t)* must be absolutely integrable; that is,

$$\int_T \left|x(t)\right| dt < \infty \tag{2.16}$$

In case of non-periodic signals, the frequency spectrum is generated by computing the Fourier transform. A non-periodic signal can be obtained from a periodic signal whose period tends to infinity. As the period becomes infinite, the frequency components form a

continuum and the Fourier series sum becomes an integral. Thus Fourier stated that the Fourier transforms of a continuous non-periodic signal can be obtained from the Fourier series sum of a periodic function and then extending the period T to infinity. The general definition of the Fourier transforms of a continuous signal x(t) is given by:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(j\omega) e^{j\omega t} d\omega \tag{2.17}$$

$$X(j\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt \tag{2.18}$$

The above two equations are referred to as the Fourier transform pair in which ω is the angular frequency, the function *X(jω)* is referred to as the Fourier transforms or Fourier integral of the signal *x(t)* and first equation is referred to as the inverse Fourier transform of *X(jω)*. As stated earlier Fourier transform will be a complex value given by:

$$F(\omega) = |F(\omega)| e^{i\Phi(\omega)} \tag{2.19}$$

where the amplitude spectrum is given by:

$$|F(\omega)| = \sqrt{\mathrm{Re}|F(\omega)|^2 + \mathrm{Im}|F(\omega)|} \tag{2.20}$$

and the phase spectrum is given by:

$$\Phi(\omega) = \tan^{-1}\left(\frac{\mathrm{Im}|F(\omega)|^2}{\mathrm{Re}|F(\omega)|^2} \right) \tag{2.21}$$

The Fourier transforms for a discrete signal can be computed as follows:

$$x_i = \frac{1}{N} \int_{u=0}^{N-1} X_u e^{j\left(u \frac{2\pi}{N} i \right)} \qquad i=0,\ 1 \ldots\ N\text{-}1 \tag{2.22}$$

$$X_u = \sum_{i=0}^{N-1} x_i e^{-j\left(u\frac{2\pi}{N}i\right)} \qquad u=0, 1 \ldots N\text{-}1 \qquad (2.23)$$

The above two equations are referred to as the discrete-time Fourier transform pair. X_u is the discrete-time Fourier transforms of x_i and x_i is inverse discrete-time Fourier transform of X_u. For each value of u, computation of X_u will require N multiplications. As stated before, X_u is complex valued and hence the computation of an N-point DFT or the inverse DFT requires N^2 complex multiplications. In practice, a collection of efficient numerical algorithms are used to compute the discrete Fourier transform known as the fast Fourier transforms (FFT) (Walker, 1996). In the FFT algorithm the number of available points in time should be a power of two.

Properties of Fourier transforms

Properties of Fourier transforms have been summarized below. The proof of these properties can be found in reference Papoulis, (1962). In these properties, x(t) and y(t) are time functions, X(ω) and Y(ω) are the corresponding Fourier transforms, a and b are constants, F and $F^{-1}$are the Fourier and inverse Fourier operators respectively.

- **Linearity:** $F[ax(t)+by(t)] \leftrightarrow aX(\omega)+bY(\omega)$ (2.24)
- **Scaling:** $F[x(at)] \leftrightarrow \frac{1}{|a|}X\left(\frac{\omega}{a}\right)$ (2.25)
- **Time shifting:** $F[x(t-t_0)] \leftrightarrow X(\omega)e^{-j\omega t_0}$ (2.26)
- **Frequency shifting:** $F^{-1}[X(\omega-\omega_0)] \leftrightarrow x(t)e^{-j\omega_0 t}$ (2.27)
- **Time reversal:** $F[x(-t)] \leftrightarrow X(-\omega)$ (2.28)
- **Derivative:** $F\left(\frac{dx(t)}{dt}\right) = i\omega X(\omega)$ (2.29)

- **Convolution:** $x(t) * y(t) \leftrightarrow X(\omega)Y(\omega)$ (2.30)

- **Parseval's theorem:** $\int_{-\infty}^{\infty} |x(t)|^2 \, dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(\omega)|^2 \, d\omega$ (2.31)

Thus this theorem relates the energy content of the signal in time and frequency domains.

- **Duality:** $X(t) \leftrightarrow 2\pi x(-\omega)$ (2.32)

- **Fourier Transform of a signal + reflection:**

$$F[x(t) + ax(t - t_0)] = X(\omega)(1 - ae^{-i\omega t_0})$$ (2.33)

This property explains that the presence of reflection in time domain is seen as successive peaks in frequency domain though the general trend of frequency response is not changed.

2.3.4 Wavelet transforms

According to Fourier theory, a signal can be expressed as a sum of infinite number of sines and cosines. The main drawback in Fourier analysis is the loss of time information during the transformation from time to frequency domain. When looking at a Fourier transform of a signal, it is impossible to infer when a particular event took place. If the signal properties do not change much over time as in case of a stationary signal -this drawback is not very important. However, in practice, most of the signals contain numerous non-stationary or transitory characteristics in them. These characteristics are often the most important part of the signal, and Fourier analysis is not suited to detecting them. In order to have resolution in both the frequency and time domain, a representation of the signal simultaneously in both the domains is essential.

This can be achieved by dividing the signals into a number of segments, small enough such that the signal is stationary in these segments. For this purpose, a window function

whose width is equal to the segment of the signal is multiplied with the segment. This results in the time- frequency representation of the signal. Several tools (Cohen, 1989) for time-frequency analysis have been proposed such as the short time Fourier transforms, Wigner distributions etc. These tools have certain limitations in terms of width of the window to be chosen, thereby, resulting in a compromise between the time-frequency resolutions of the signal. The reason for this limitation could be explained by Heisenberg's uncertainty principle (Chan, 1995), which states that it is impossible to know at which frequency what instant of time exist, but it is possible to know at which frequency band exist at what instant of time exist. To have a good resolution in the frequency domain, a wide window has to be chosen which results in poor resolution in time domain and visa versa. To bridge the gap between time and frequency resolution, wavelet transform technique was introduced which overcomes these drawbacks inherent in the other tools. Wavelet analysis allows the use of long time intervals, where, one would need more precise low-frequency information, and shorter regions, or one would need information in the high-frequency domain.

The definition of continuous wavelet transforms is given by:

$$WT(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t)\psi^* \left(\frac{t-b}{a} \right) dt \qquad (2.36a)$$

Where x(t) is the time signal, ψ(t) represents the window known as mother wavelet, parameter b is used to time shift the window ψ(t), parameter a is used to define the center frequency of ψ(t), * denotes the complex conjugate and the coefficient $\frac{1}{\sqrt{|a|}}$ ensures that all dilated versions of ψ(t) used to measure the time signal have the same energy. In discrete form, the wavelet transforms is computed as

$$W_{k,m} = \frac{1}{\sqrt{k}} \sum_{n=0}^{N-1} x_n \psi^* \left(\frac{n-m}{k} \right) \quad (2.36b)$$

Where x_n denotes the discrete time signal over a time period given by N.Δt, Δt is the sampling time interval, m is an integer counter giving a shift time m.Δt and k is an integer counter giving the center frequency of the wavelet ($f_0 = \frac{1}{2k\Delta t}$). A number of wavelets are available for implementation of wavelet transformation (Qian, 2002). Mallat's pyramid algorithm (1989) is used for implementing wavelet transforms. In wavelet transforms, Mallat's algorithm is used for decomposing the signal into a number of components called levels with each level starting at -1 and onwards. The total number of levels is determined by number of sampling points, i.e if $N=2^n$ then the number of levels is equal to n+1. Summation of all the levels helps in reconstructing the signal.

2.3.5 Cepstrum Analysis:

Cepstral analysis is a nonlinear signal processing technique that is applied most commonly in speech processing and homomorphic filtering. The cepstrum has been highly effective in automatic speech recognition and in modeling the frequency content of audio signals. Cepstral coefficients c(k) are a very convenient way to model spectral energy distribution and are computed by taking the inverse Fourier transforms of the complex logarithm of the magnitude of the Fourier transform given as (Malcolm Slaney, 1998)

$$c(k) = IDFT\left\{\log\left|DFT\{x(n)\}\right|\right\} \quad (2.37)$$

where DFT denotes the Fourier-transform and IDFT its inverse.

2.4 Previous related work: Ultrasonic testing techniques for flaw detection

Most of the current non-destructive testing techniques (NDT) were invented in the late 1930's. Sokolov (1929) in Russia conducted experiments using ultrasonic waves for detecting metal objects. The primary purpose of using these techniques during the earlier days was for defect detection. In view of this need, several NDT techniques using ultrasonics, eddy currents, X-rays, magnetic particles, and other forms of energy were developed. Over the years, rapid development made in the field of computer technology for data acquisition and subsequent signal processing led to significant improvements in these techniques which are now being applied in different fields with different degrees of success.

Ultrasonics is one of the mostly used NDT techniques for the detection, localization and measurement of the defects present in engineering materials under inspection. Ultrasonic testing is based on the transmission of ultrasonic signals to detect imperfections or to characterize the materials. The reflected echoes from the defects or the imperfections are received by the ultrasonic transducers and are recovered for further processing. Four commonly used techniques used in ultrasonic testing are impact echo (IE), pulse velocity (PE), spectral analysis of surface waves (SASW) and multiple analysis of surface waves (MASW).

2.4.2 Impact-Echo Method

Impact-Echo method has been successfully employed for flaw detection in concrete and masonry structures (Carino and Sansalone 1986; Carino and Sansalone 1990; Lin 1996; Poston et al., 1995). This method consists of applying a light impact on the surface of the material which results in a stress wave propagating radially outwards from the source in

form of compression, shear and surface waves. The stress wave pulse undergoes multiple reflections between the top surface and any internal defect or the bottom of the material. These multiple reflection arriving at the surface produces periodic displacements which are monitored with a transducer placed adjacent to the source. The time signals are analyzed in frequency domain for further interpretation of the test data. The large peak in the frequency spectrum as a result of the multiple wave reflections is known as the resonant frequency. To interpret the frequency spectrum, the basic relationship used in IE method is

$$h = \frac{V_p}{2f} \tag{2.38}$$

where V_p is the compression wave velocity, f is frequency of compression wave reflections, h is the distance to the reflecting interface. The basic principle of IE method is shown in Figure 2.8.

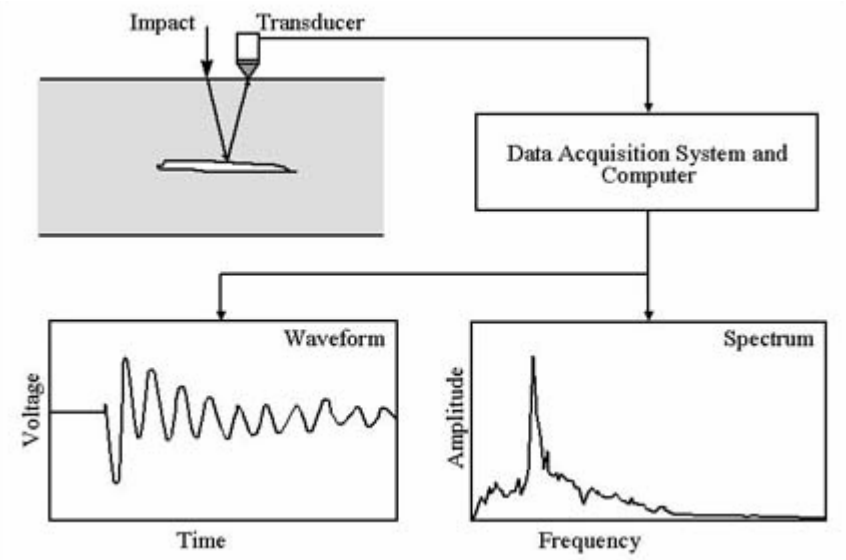


Figure 2.8: Principle of IE Method (Sansalone and Streett, 1997)

The main advantage of IE method is that access to only one surface of the material is required. This method is very effective for defects oriented parallel to the surface. However this method can be difficult to interpret especially for complex geometries and it is not sensitive for small defects (smaller than half the pulse wavelength). Due to the type of input source (short duration impact) results in scattering of high frequency waves limiting this method to be used only for relatively thin structures (Popovics and Rose, 1994).

2.4.3 Pulse Velocity Method

In the late 1940's application of pulse velocity through concrete began and even today this method is very popular among NDE techniques being used for evaluation of quality of materials and for detection of flaws in materials (Carino and Nicholas 1986). The basic principle of the pulse-velocity method is the measurement of compressional wave velocity through a solid which are indicative of its material properties. Presence of internal cracks, voids and other defects causes a change in the wave velocity due to the fact that compressional waves are reflected at air or water interfaces. This method consists of placing two piezoelectric transducers on two sides of the test object. There are three possible transducer configurations (Figure 2.9) in which the transducers may be arranged; direct, semi-direct and indirect transmission.

In one of the sensors, pulses are generated and the time taken by the pulse to travel through the object is measured by the other sensor. Knowing the travel time and length of the specimen, the corresponding wave velocity can be determined. Conversely, knowing the wave velocity, the travel time can be used to determine the depth to a given flaw or thickness of the specimen.

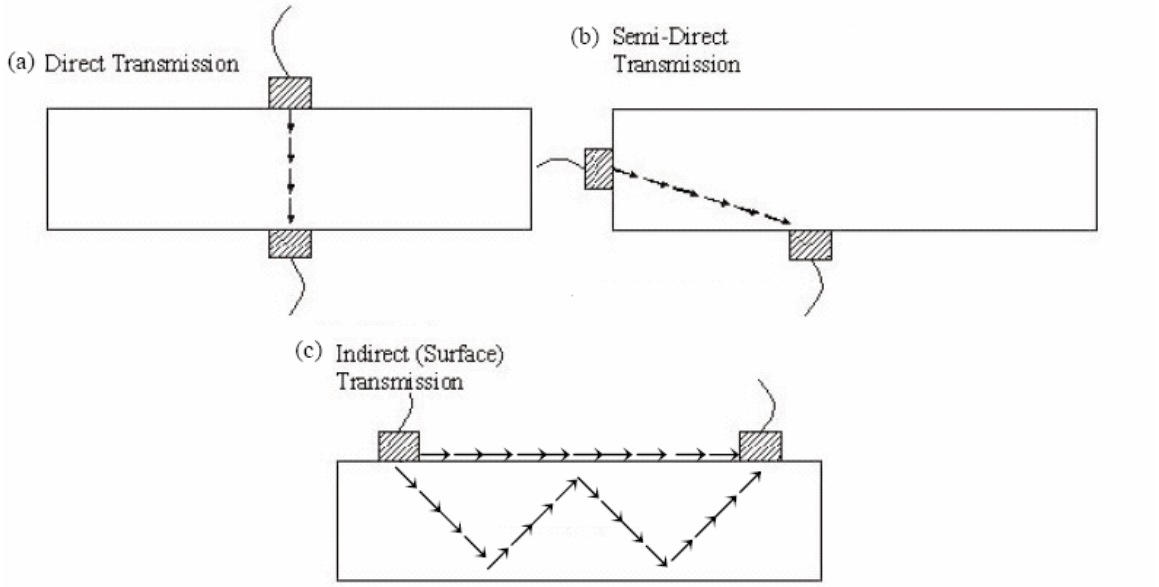


Figure 2.9: Transducer Configurations in pulse velocity method (Olson, 2004)

Presence of a defect causes a delay in the arrival time which results in lower wave velocities. There is also a reduction in the signal amplitude which occurs due to the attenuation of the wave energy by scattering/ reflection at internal flaws.

This method is easy to perform and mathematically simple. However the transducers must be in full contact with the test medium which might be problematic in case of materials like wood. Only defects larger than the transducer contact face will cause a measurable reduction in velocity and therefore this method is not sensitive for small defects (Suaris et al., 1987).

2.4.4 Spectral Analysis of Surface Waves (SASW)

The SASW method introduced by Heisey et al., 1982 is based on the dispersive properties of Rayleigh waves to evaluate the low strain (in the order of .001%) elastic properties of different layers (Sheu and Rix et al., 1988; Roesset et al., 1989). Dispersion is the property wherein each frequency component of the Rayleigh wave propagates with different velocities called the phase velocities. When a seismic source is created on the

surface of an elastic medium, majority of the energy is in the form of surface waves, known as Rayleigh waves and the rest goes into body waves (Miller and Pursey 1955, Woods 1968). Further, Rayleigh waves attenuate at a slower rate than body waves due to the effect of geometrical damping. Thus Rayleigh waves can be generated with relatively small energy and have the potential of traveling long distances. All these characteristics make Rayleigh waves very attractive for non-destructive testing.

Basically the SASW method involves three steps (Nazarian, 1984):1) Data gathering at the surface, 2) construction of experimental dispersion curve, 3) inversion of the dispersion curve. The basic set-up of SASW method is shown in Figure 2.10. In SASW investigation, the surface of the medium is subjected to an impact to generate surface waves with various frequencies. Two receivers placed on the surface are used to monitor the propagation of the waves. Heisey (1981) suggested that the distance between the receivers should be less than two wavelengths and greater than one-third of a wavelength. This is expressed as

$$\frac{\lambda}{3} < \Delta x < 2\lambda \tag{2.39}$$

where λ denotes wavelength of surface wave and Δx denotes the receiver spacing that is commonly equal to the distance between the source and first receiver. The signals are recorded in a digitized form using a data acquisition system and are analyzed in the frequency domain using a spectral analyzer.

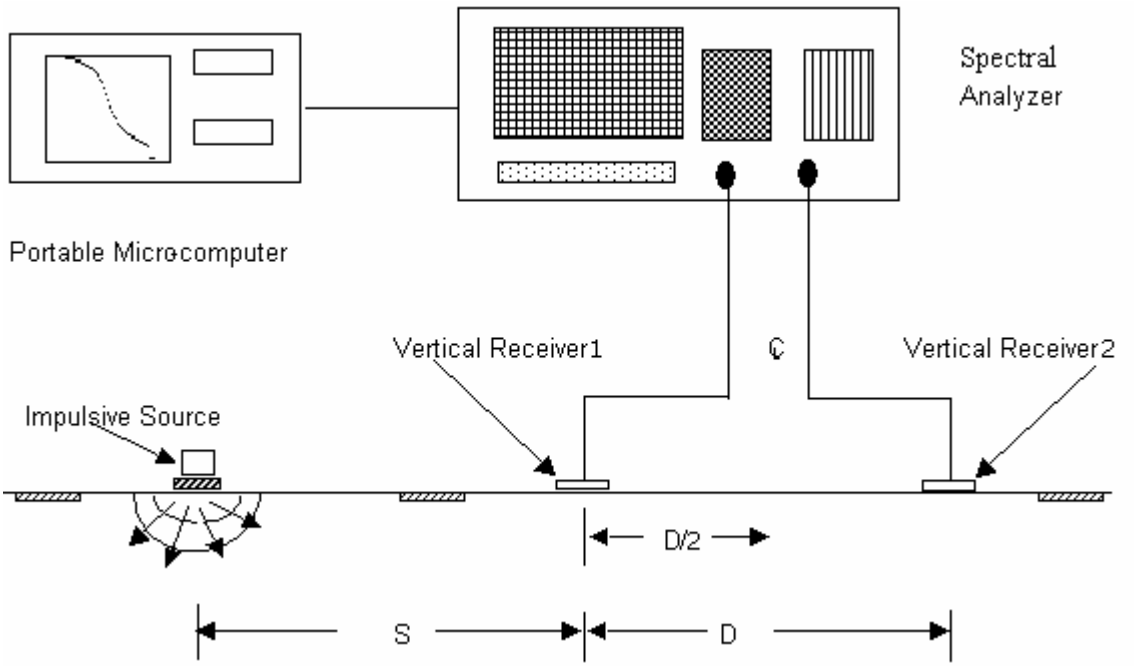


Figure 2.10: SASW test configuration. S is the distance from the source to the first receiver and D is the distance between the two receivers. (Roesset et al 1989)

The phase difference between the receiver pair for each frequency is determined from the cross power spectrum. The cross power spectrum is expressed as

$$C(f) = Y_1(f)Y_2(f)^* = A_1(f)A_2(f)e^{i\Delta\varphi(f)} \tag{2.40}$$

where $Y_1(f)$ and $Y_2(f)$ are the Fourier transforms of the two signals, $A_1(f)$ and $A_2(f)$ are spectral amplitudes, i is the imaginary unit, * denotes the complex conjugate, $\Delta\varphi = \varphi_2 - \varphi_1$ represents the phase difference as a function of frequency. The phase differences obtained are stated as an angle in the range $\pm\frac{\pi}{2}$ (wrapped phase). The correct phase is obtained by adding an appropriate number of cycles to the wrapped phase which gives the unwrapped phase.

The phase velocity of the Rayleigh wave V_{ph} at any frequency f is related to distance Δx and the phase difference $\Delta\varphi$ by the following relationship

$$V_{ph} = 2\pi f \frac{\Delta x}{\Delta \varphi} \tag{2.41}$$

The wavelength λ as function of frequency f and phase velocity V_{ph} is given as

$$\lambda = \frac{V_{ph}}{f} \tag{2.42}$$

The dispersion curve is constructed by plotting the V_{ph} against λ. An appropriate frequency range should be chosen for analysis to ensure that reliable information is obtained for construction of the dispersive curve. Inversion of the dispersive curve is accomplished iteratively by matching the theoretical (constructed by assuming a shear wave velocity profile) with the experimental dispersive curve from which shear wave velocity profile can be determined (Bullen, 1963). The process of inversion is accomplished by a least-squares approach (Nazarian, 1984; Lai, 1998). Layer thickness and elastic properties of each layer are readily obtained from the shear wave velocity profile.

Advantages of SASW:

This method requires access to only one side of the specimen. The stiffness profile can be determined without knowing the layer thicknesses. By varying the range of frequencies emitted by the source, the depth of penetration of surface waves can be changed thus making deep sounding possible. Surface waves are very sensitive to anomalies close to the surface which makes this method very effective in detecting near surface flaws (Curro 1983; Dravinski 1983). Belesky and Hardy (1986), reported that the amplitude of surface waves are more affected by presence of obstacles than the travel time which makes this method more efficient than conventional seismic methods in defect detection.

Disadvantages of SASW:

The two-receiver arrangement is sensitive to ambient noise. Surface waves have a limited penetration depth which ranges between 12 to 30 m (Drossaert, 2001). Since only two receivers are used, this method requires performing the test with several testing configurations to sample the desired frequency range which is time consuming.

2.4.5 Multiple Analysis of Surface Waves (MASW)

Multiple Analysis of Surface Waves was developed to overcome the limitations associated with SASW method. The main difference between the two methods is that the SASW method uses only two receivers whereas the later uses several receivers (the use of 12 to 48 receivers are reported in literature) Socco and Strobbia (2004). MASW offers the advantages of more effective noise removal process, faster data collection, more reliable shear wave velocities, identification of higher modes (Park et al., 1999). The basic set-up of the MASW method is shown in Figure 2.11.

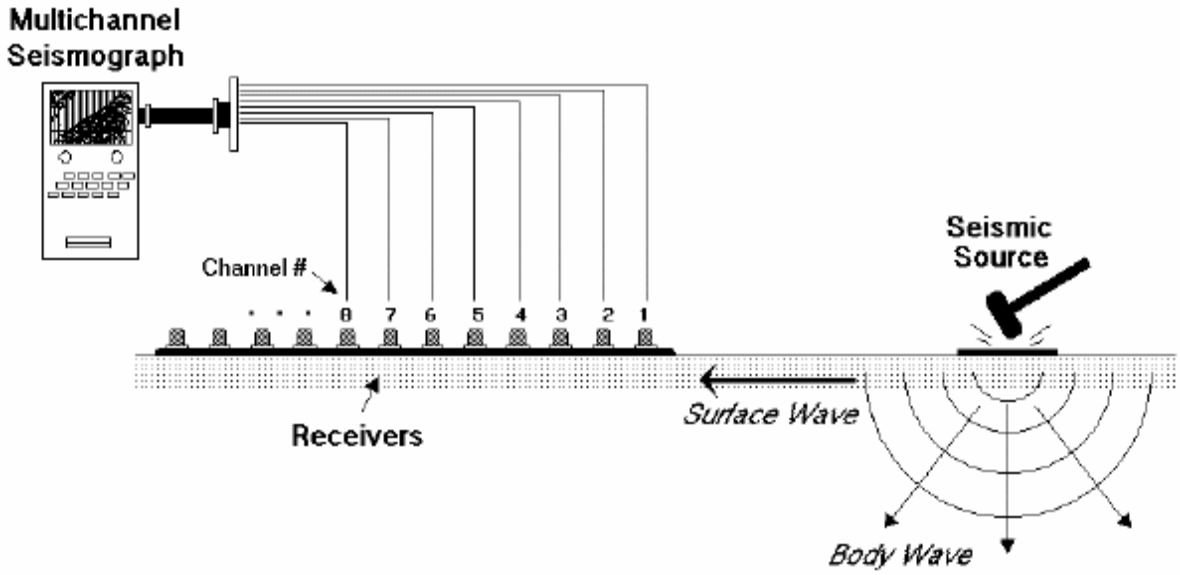


Figure 2.11: MASW testing configuration (Park et al., 1999)

The choice of offset value D (distance between the source and the first receiver) and receiver spacing Δx has significant effect on the data collected. The selection of the offset distance depends on the stiffness of the material to be tested, attenuation properties of the medium, desired investigation depth, wave velocity of the medium, the frequency range used (Hiltunen and Woods, 1989). The offset distance should be large enough to allow the surface waves to be fully developed (near field effects). A number of criteria have been proposed that relates the receiver spacing with wavelength (Lysmer, 1965; Sanchez-Salinero et al. 1987, Hiltunen and Woods 1990, Al Hunaidi 1993). The commonly used criteria is given by Heisey (1981) Equation (2.39). The basic procedure of MASW is similar to SASW with the exception of having multiple receivers instead of two. The basic steps are 1) Data gathering at the surface with multiple receivers deployed in a line at a distance from the source which record the responses 2) Construction of dispersion curve which consists of measuring the phase difference for each frequency between various receivers. In SASW where there are only two receivers this process is simple, but in MASW this process might be different due to multiple receivers which therefore require more rigorous techniques (Phillips et al., 2003). In a homogeneous medium, the phase angle of each frequency component varies linearly with distance. However in case of non-homogeneous medium, the phase angle will not vary linearly with distance. The variations in the phase angle with distance can be utilized detect areas where there are changes in the elastic properties of the medium or presence of an obstacle can be identified from the variations in the phase angle with distance. 3) Inversion of the dispersive curve to obtain the shear wave velocity profile from which layer thickness and elastic properties of the medium can be determined.

The spatial resolution of the results depends on the number of receivers, the more the number; greater is the ability to detect changes in the horizontal properties of the medium. Successful case study reports using MASW include Xie et al. (1999); Gang Tian et al. (2003); Chih-Ping Lin et al. (2004); Chaoqiang and Stephen (2005).

Advantages of MASW:

The results obtained from MASW are highly reliable even under the presence of noise and higher modes of surface waves. The process of data collection and processing is very fast. Compared to SASW, this method has the capacity of recognizing and differentiating between different events such as reflections and refractions.

Disadvantages of MASW:

This method is only suitable for horizontal layers and does not take into account lateral inhomogeneities. Difficulties in differentiating between different modes of Rayleigh waves lead to complications in data interpretation. A standard criterion is not available for the choice of the offset distance and receiver spacing (Nasseri, 2006).

Chapter 3

Classification and Clustering

3.1 Introduction

Due to rapid development in the field of defect detection using non-destructive testing techniques, the growth in the amount of data available, and the speed with which it can be collected, the traditional manual data analysis has become inefficient and computer-based analyses are gaining momentum. Statistical methods, soft computing methods like fuzzy logic and neural network algorithms are being tested on a variety of problems related to defect prediction so as to provide a decision support system.

The term artificial intelligence was first coined by John McCarthy who referred it to as the "science and engineering of making intelligent machines". "Artificial Intelligence" can be defined as the *simulation of human intelligence on a machine, so as to make the machine efficient to identify and use the right piece of "Knowledge" at a given step of solving a problem* (Amit Konar, 2000).

The most basic requirement of any system in recognizing and classifying objects is to possess a detailed knowledge about the characteristic features of those objects; the next stage in selecting a classification method involves choosing one of the two basic paradigms: supervised learning and unsupervised learning. In supervised learning scheme, the system is provided with training examples and is trained to recognize them. Unsupervised learning involves simply allowing a system to cluster samples together based on similarities that it perceives in the feature space. This form of learning is also referred to as clustering. In this study, analysis of data for fault detection and classification was done using these two basic paradigms:

Under supervised learning, classification was performed using

1. Soft computing methodology using Fuzzy logic and Neural Networks
2. Statistical methods

Under unsupervised learning, the following soft computing method was utilized

1. Fuzzy C-means Clustering

This chapter begins with an overview of soft computing methods which includes Fuzzy Logic and Artificial Neural Networks. The second section briefly discusses the supervised and unsupervised classification techniques. The final section provides a literature review of the classification/clustering techniques used for flaw detection.

3.2 Soft Computing Methods

3.2.1 Fuzzy Logic

Introduction

The fuzzy logic is a form of logic in which, the underlying modes of reasoning are approximate instead of exact and can be thought of as an extension of conventional Boolean logic to handle the concept of partial truth. It emulates the ability to reason out and use approximate data to find solutions. It was introduced by Lotfi Zadeh of UC/Berkeley in the 1960's. The following section provides a brief summary of evolution of Fuzzy logic.

Historical Background

1965: Zadeh introduced his seminal idea in a continuous-valued logic called fuzzy set theory.

1970s: Zadeh illustrated a possibility theory resulting from special cases of fuzzy set.

1988: The investigations of Klir and Folger showed a strong relationship between evidence, probability and possibility theories with the use of fuzzy measures

1980s: East Asian countries particularly the Japanese have adopted Fuzzy Logic in industry.

1990s: Americans and Europeans have adopted it with much reluctance.

2000s: In China about 1000 researchers are believed to be working on Fuzzy Systems.

For a long time, most of the Western scientists have been reluctant to use fuzzy logic as they felt that it threatens the integrity of scientific thought. Other scientists feel that it even expands the possibilities of computer programming. Eastern philosophy and scientists have embraced fuzzy logic. In over the three decades, since its inception by Zadeh, fuzzy set theory has undergone tremendous growth.

Fuzzy Logic

Fuzzy sets, fuzzy operators and the knowledge base are building blocks of fuzzy logic theory. Fuzzy sets are represented by membership functions. A particular element of the fuzzy set will have grade of membership which gives the degree to which a particular element belongs to a set. A membership function is a curve that provides the degree of membership within a set of any element that belongs to the universe of discourse, X. If X is the universe of discourse and the elements in X are denoted by x, then a fuzzy set A in X is defined as a set of ordered pairs

$$A = \{x, \mu_A(x);\ x \in X,\ \mu_A(x) \in [0, 1]\} \tag{3.1}$$

where $\mu_A(x)$ is a membership function of x in A shown in Figure 3.1

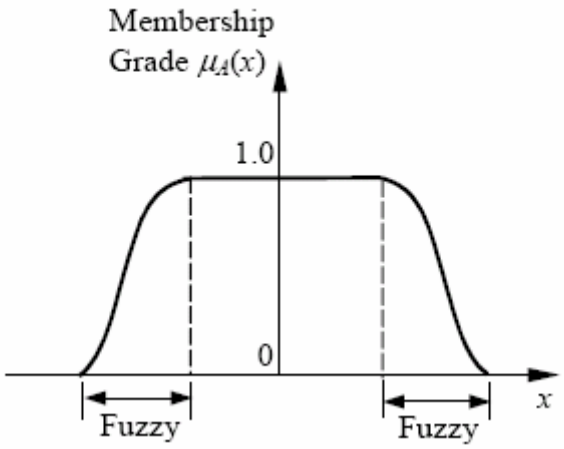


Figure 3.1: The membership function of a fuzzy set (Karray and C.De.Silva, 2004)

Boolean operators such as 'AND', 'OR' and 'NOT' operators exist in fuzzy logic as well. These operators are defined as the minimum, maximum and complement. The knowledge base in fuzzy logic is represented by if-then rules of fuzzy descriptors. A single 'IF-THEN' rule assumes the following format:

if x is A then y is B

A and B are linguistic variables and are defined by fuzzy sets on the ranges X AND Y (where X and Y are the universes of discourse). The '*if* 'part of the rule *"x is A"* is the antecedent or premise and the *then* part of the rule '*y' is B* is the consequence or conclusion. One of the most important applications of fuzzy logic is fuzzy control which offers a formal methodology in controlling systems based on human's heuristic knowledge. The fuzzy control essentially consists of four components

- Fuzzification of inputs to the system so that they can be compared and interpreted with the rules in the rule base
- Knowledge rule base which consists of a set of if-then rules of fuzzy descriptors provides information as how best to control the system

- Inference mechanism which determines the relevant control rules at the current time and decides what the input to the system should be
- Defuzzification of the fuzzy conclusions reached by the inference engine into the inputs to the system

The fuzzy logic controller ensures the performance objectives of the system are met, by comparing the system output data with the reference input which helps in deciding what the system input should be. Thus the use of fuzzy logic in applications involving complex knowledge-based decision making is appealing and valuable particularly because of its ability of incorporating approximate reasoning.

3.2.2 Artificial Neural networks

Introduction

Artificial neural networks were born of the interest in mimicking the immense capabilities of the human brain in processing information and making instantaneous decisions under extremely complex situations and even in unknown environments. This extremely powerful learning and decision making capability of the human brain is attributed to the way the brain processes the information using a massive network of parallel and distributed computational elements called neurons tied together by weighted connections called synapses. This efficient computational biological model has inspired research for the last few decades in developing computational systems which can process information in a similar manner. Such systems are called artificial neural networks (ANN) and are composed of a parallel information processing array based on a network of interconnected artificial neurons (also called cells, units, nodes, or processing

elements). These neurons are tied together by weighted connections similar to the brain synapses.

Historical background

1943: McCulloch and Pitts introduced the first neural network model (Pitts and McCulloch, 1947).

1949: Donald Hebb published a book called "The Organization of Behaviour" which formed the basis of 'Hebbian learning' which is now regarded as an important concept in the field of ANN.

1950 to 1956: The development of digital computers led to further development and investigation in the field of artificial intelligence. In 1956, Dartmouth Summer Research Project on artificial intelligence stimulated further research in this area.

1958: John Von Neumann proposed modeling the neuron functions using items of computer hardware.

1960: Frank Rosenblatt developed neuron models in hardware called the Perceptron used for classifying linearly separable sets into one or two classes.

1962: Bernard Widrow and Marcian Hoff developed neuron models called the ADALINE (ADAptive LInear Neuron) and MADALINE (multiple ADALINE). The MADALINE was the first model to be applied to a real world problem.

1969: Minsky and Pappert published a book called "Perceptrons" in which they brought out the limitations of the Perceptrons. The main limitation was its inability to handle the 'exclusive or' function. The training of the network was another limitation. All these limitations resulted in dampening the expectations set in the field of ANN which led to the diminishing interest in this field. This period lasted till 1981. Despite the reduction in

research funding, a number of researchers continued working in ANN. Werbos in 1974 developed the back-propagation learning algorithm (Werbos, 1974) to overcome the limitations of the earlier models.

1982: Hopfield presented a paper in which the advantages of ANN was brought out. The training algorithms for multilayered networks were introduced and hence his contributions marked a revived interest in ANN. Since then, significant progress took place in the field of neural networks which has resulted in ANN being successfully applied across an extraordinary range of problem domains.

Basic Structure of a neural network

The basic structure of a neural network consists of a set of parallel and distributed processing units called the neurons. A single neuron receives a number of inputs $x = (x_1, \ldots, x_I)$ (either from the original data or from the output of other neurons in the neural network) and determines the optimal connection weights $w = (w_1, \ldots, w_I)$ that are appropriate to each input through learning which describes the corresponding strength of input. Each neuron also has a threshold value (bias effect) which is intended to occasionally inhibit the activity of some neurons (Karray and De Silva, 2004). The weighted sum of the inputs is formed, and the threshold subtracted to compose the activation of the neuron. This value is passed through a mapping (not necessarily linear) called the transfer (or activation) function F to produce the output of the neuron, Figure 3.2. The output of a typical neuron is given as:

$$O = F\left(\theta + \sum_{i=1}^{I} w_i x_i\right) \tag{3.2}$$

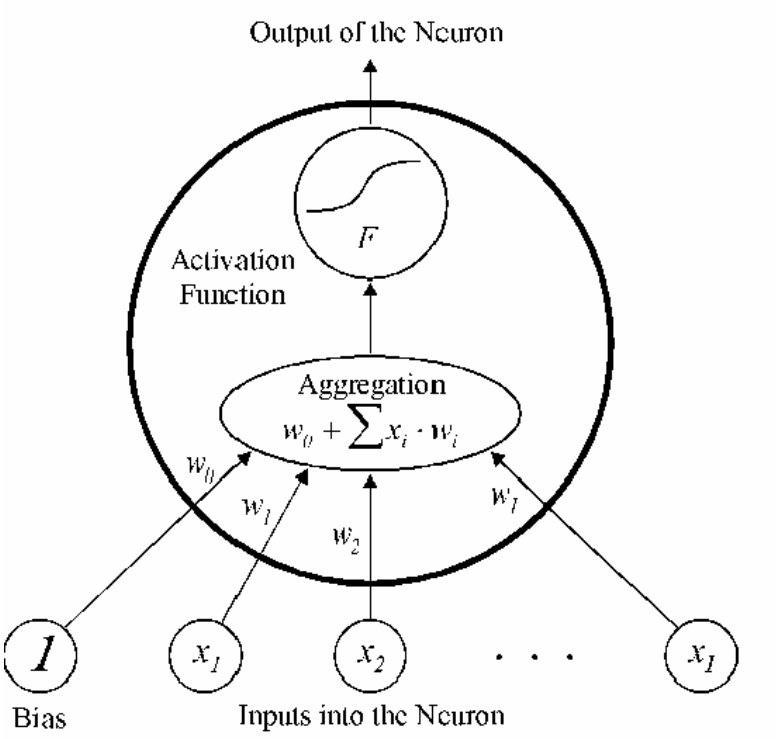


Figure 3.2: Basic structure of a Neuron (Wolfgang and Heiko (2004))

These neurons are usually arranged in different layers, appropriately connected by weighted connections as shown in Figure 3.3. The neural network shown in Figure 3.3 has its nodes hierarchically arranged in three layers starting with the input layer and ending with the output layer. In between the two layers is the hidden layer which provides most of the network computational power. The input layer takes the input and distributes it to the hidden layer which performs all the necessary computations and outputs the results to the output layer. The processing activity within each layer is done simultaneously which provides the neural network with a powerful capability of parallel computing.

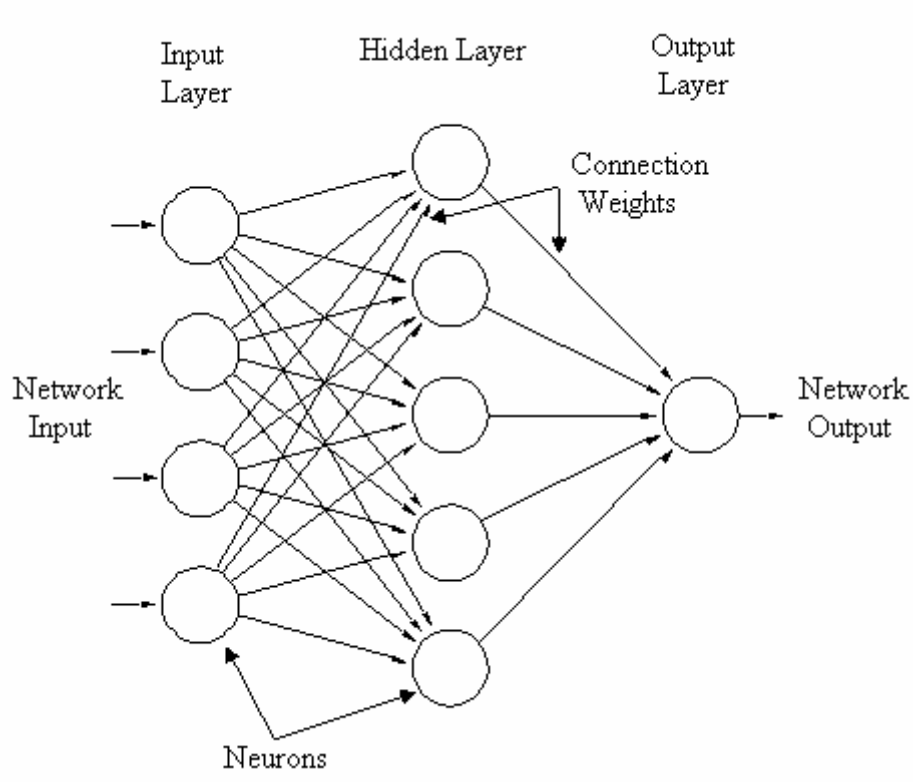


Figure 3.3: Basic Structure of a Neural Network (Steward, 2002)

In general, there are two principal functions for artificial neural networks.

• The input–output mapping or feature extraction.

• Pattern association or generalization.

The mapping of input and output patterns is estimated or learned by the neural network with a representative sample of input and output patterns. The generalization of the neural network is an output pattern in response to an input pattern, based on the network memories that function like the human brain. Therefore, a neural network can learn patterns from a sample data set and determine the class of new data based on previous knowledge.

Classification of ANN

General

Different types of neural networks have evolved based on the neuron arrangement, their interconnections and training paradigm used. ANN can be broadly classified according to the following:

- Learning paradigms
  - Supervised
  - Unsupervised
  - Hybrid
- Architecture
  - Feed forward
  - Recurrent
- Activation functions

Detailed description of the different types of ANN can be found in Karray and De Silva (2004).

3.3 Previous related work: Evaluation of Classification/Clustering methods for flaw detection

In the application of non-destructive testing techniques for material inspection, researchers have always tried to develop a decision support system for the automatic defect classification. In spite of several modern automated inspection methods, defect classification still is a difficult task. Much work published in this area is based on artificial intelligence methods using artificial neural networks and conventional statistical classification algorithms. The most popularly employed artificial neural network for

defect characterization reported in literature is the multilayer perceptron (MLP). On the other hand, K-nearest neighbor is most commonly used amongst statistical classifiers.

Rose (1977) used pattern recognition algorithms and tried to separate 23 flaw geometries into as many classes as possible. Shankar et al. (1978) developed an adaptive learning network for non-destructive testing classification problem that combined a statistical approach and the application of a neural network like model. Burch and Bealing (1986) using pulse echo method tried to classify smooth and rough cracks from more benign volumetric flaws such as porosity and slag. Qualitative physical models were developed for the interaction of ultrasound with these defects to determine three uncorrelated features (amplitude ratio, waveform kurtosis and sphericity). Weighted minimum distance pattern recognition algorithm was used for defect classification which gave an accuracy of 100% for 40 buried defects in ferritic steel.

Researchers have reported the use of advanced signal processing techniques based on time-frequency analysis, wavelet transforms, Hilbert transforms and power spectrum for processing transient signals for characterization of materials and to improve the probability of defect detection (Flandrin 1988, Drai et al 2000 and Legendre et al., 2001).

Baker and Windsor (1989) used the Hopfield network for classification of processed ultrasonic data from various classes of defects (cracks, rough cracks, pores and slag) within steel test welds. They reported a classification accuracy of 100% which is comparable to the conventional minimum distance classification algorithm. Chiou and Schmerr (1991) using the quasi pulse echo ultrasonic classification technique tried to distinguish between smooth edged and sharp edged geometries. To test the practicality of this approach, experiments were conducted on cylindrical cavities and surface breaking

fatigue cracks. Their results showed a good consistency in the classification of smooth and sharp edged flaws. Self learning backpropagation neural networks for classification of ultrasonic inspection data was performed in both time and frequency domain. It was found that most of signals with flaws could be correctly classified (Raja Damarla et al., 1992). Song and Schemrr (1992) tried to classify flaws in weldments from their ultrasonic scattering signatures using probabilistic neural networks and they reported a high classification performance than the other types of neural networks.

Neural network was used to distinguish crack-like defects from volumetric defects by directly analysis the images collected from ultrasonic scanning, (Windsor, 1993). Masnata and Sunseri (1995) developed a methodology for automatic characterization of weld defects detected by a P-scan ultrasonic system. Fischer linear discrimination analysis was used to reduce the number of features from 24 to 14 uncorrelated features which were classified using a three-layered multiple layer Perceptron.

Margrave et al. (1999) evaluated performance of different types of neural networks in accurate flaw detection in steel plates. The networks employed were 3 layers Multilayer Perceptron, Self Organizing maps using Kohonen learning rule and Linear vector quantization (LVQ) networks. Neural networks were trained to classify between six classes of defects; no defective, side drilled holes, slots, inclusions, porosities and cracks. Signals in time domain obtained directly from the ultrasonic scans and in frequency domain were given as inputs to the neural networks with little or no preprocessing. They reported the performance of MLP to be the best among all three, the self organizing maps performed the least satisfactorily among the considered networks and performance of LVQ was faster than MLP but requires the classes to be clearly discriminated. In

comparison with the time domain analysis, frequency domain analysis results in better classification.

Santos and Perdigao (2001) using pulse-echo technique tried to discriminate between defects with three different shapes (cylindrical, spherical, planar). A total of four features were extracted from the signals, three in time domain, namely, pulse duration, pulse decay rate and peak-to peak relative amplitude of the third cycle, frequency for maximum amplitude was extracted in the frequency domain. They reported a classification accuracy of 100% using the nearest neighbor approach. Obaidat et al. (2001) developed a methodology to detect defects obtained from ultrasonic-based NDT using the multilayer perceptrons (MLP). They found that results obtained by using discrete wavelet transform (for feature extraction) and neural networks were superior over the classification of NDT signals using only neural networks. Drai et al. (2002) tried to distinguish between a planar and volumetric flaw based on the calculation of wavelet coefficients, time and frequency domain parameters to characterize the defects. Classification was performed using K nearest neighbor, Bayesian statistical method and artificial neural network. They reported higher classification accuracy of 97% with features from wavelet transforms associated with ANN.

Francesca Cau et al. (2005) developed an ANN model for fault detection in not accessible pipes. Fault classification was based on the depth and width of the faults and the signal database for the training, validation and test set were obtained using finite element simulations based on propagation of guided ultrasonic waves. A total of 46 features were extracted: 39 in time domain and 7 in frequency domain. Out of the two types of data reduction strategies employed, namely, Garson's method and Principle

component analysis, the later gave better classification accuracy. Their results showed that the percentage error of ANN for fault width classification to be less than 5% and less than 7% for fault depth classification.

In addition to experimental works, numerical simulations were also used to generate ultrasonic signals for flaw detection using neural networks. Rhim and Lee (1995) used the back propagation neural networks in conjunction with finite element method to detect the existence and to identify the characteristics of damage in laminate composite beams with various imperfections. Liu et al. (2002) simulated the pulse-echo method by developing numerical models using finite element method combining with boundary integral equation to investigate the effects of cracks in a medium. Back propagation neural network was utilized which gave an accuracy of 94% in classifying between three distinct classes; without cracks, surface cracks and sub-surface cracks.

3.4 Classifiers used in this study

3.4.1 Supervised Learning: Soft Computing Methodologies

Back Propagation Neural Network

Among the various types of neural networks, the multi-layer perceptron trained with the back-propagation learning algorithm has proved to be the most useful in engineering applications (Upda 1990, Damarla et al. 1991, Obaidat et al 2001). The back-propagation network is given its name due to the way that it learns by back propagating the errors in the direction from output neurons to input neurons. The multi-layer perceptron network comprises of an input layer, output layer and a number of hidden layers. The presence of hidden layers allows the network to represent and compute more complicated associations between patterns. Many researchers (Margrave et al., 1991; Obaidat et al.,

2001; Amitava Roy et al., 1995) proved that the multi-layer perceptron with three layers can perform arbitrarily complex classification, while, the complexity depends on the number of neurons in the hidden layer. The number of neurons in each layer may vary and is dependent on the problem. The basic structure of a feed-forward, back-propagation network based on the multi-layer perceptron is shown in Figure 3.4

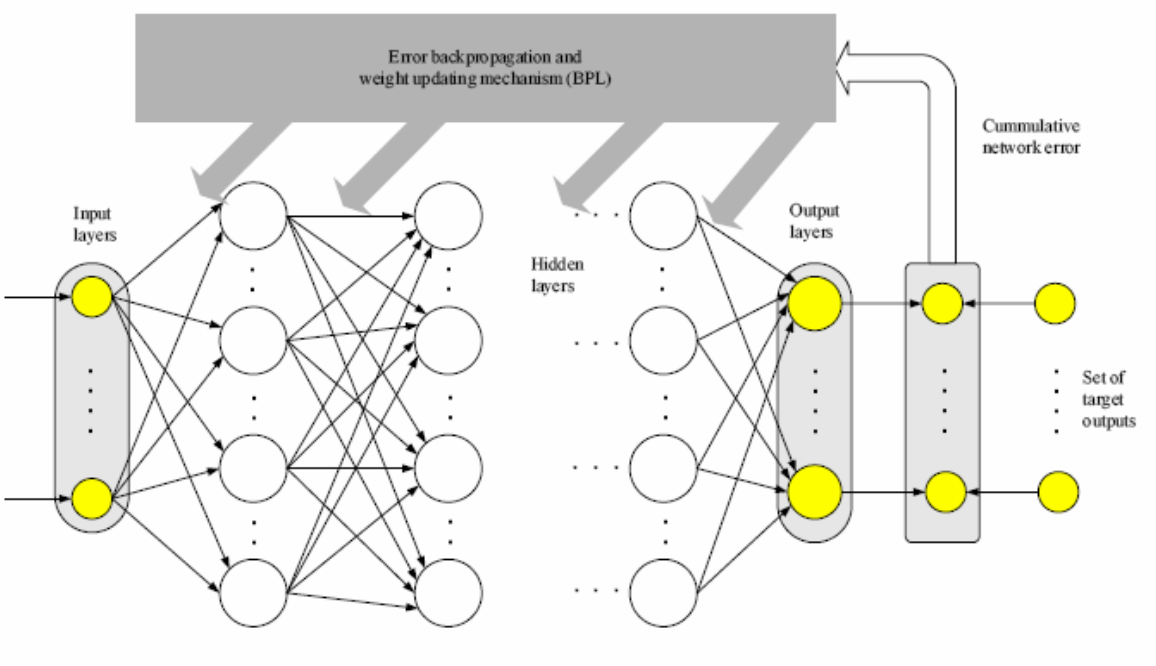


Figure 3.4: Schematic representation of the BPNN (Karray and De Silva, 2004)

Propagation of data takes place from input layer to the output layer. There is no connectivity between neurons in a layer. This type of neural network is trained using a process of supervised learning in which the network is presented with a series of matched input and output patterns and the connection strengths or weights of the connections automatically adjusted to decrease the difference between the actual and desired outputs. To begin with, patterns are presented to the network and a feedback signal which is equal to the difference between the desired and actual output is propagated backwards through

the network for the adjustment of weights of the layers' connections according to the backpropagation learning algorithm. Typical neuron employed in back-propagation learning is shown in Figure 3.5

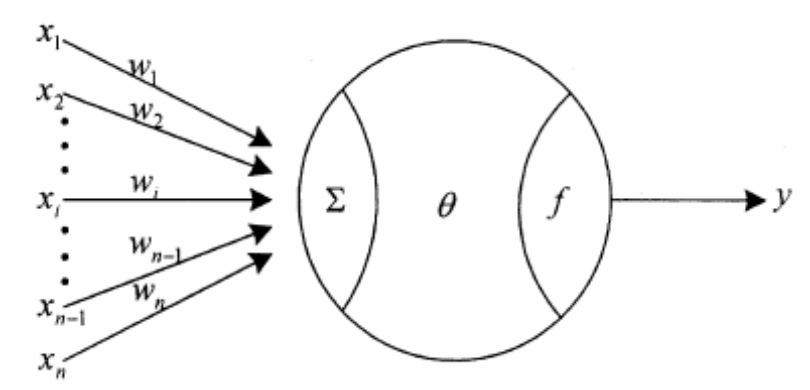


Figure 3.5: Structure of single artificial neuron (Liu et al., 2002)

The basic function of a single neuron consists of collecting all the incoming signals x_1, $x_2, \ldots, x_l$ multiplies them by corresponding weights w_1, $w_2, \ldots, w_l$ and compares the result with a predetermined bias θ before applying some sort of activation mapping function f resulting in the output o expressed as

$$o = f\left[\theta + \sum_{i=0}^{l} w_i x_i\right] \tag{3.3}$$

The activation function is required to transform inputs from minus infinity to plus infinity and scale it into the range of either -1 to 1 or into 0 to 1 interval. The most commonly used activation function is the sigmoid nonlinearity function (Figure 3.6) because of the continuity of the function over a wide range.

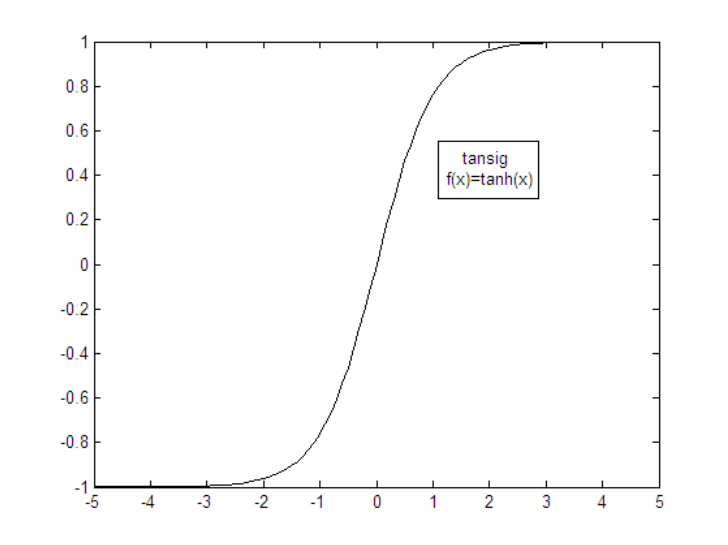


Figure 3.6: Hyperbolic tangent sigmoid function.

The steps involved in training a multilayer perceptron with backpropagation learning essentially consists of the following steps (Karray and De.Silva, 2004):

Step 1: The weights and threshold or bias are initialized to small random values

Step 2: An input-output pattern from the training dataset is presented to the network

Step 3: The output values for all *i* neurons at every layer *(l)* are calculated using Equation 3.3

Step 4: The error between the actual and predicted output is calculated using

$$E = \frac{1}{2}\sum_{i=1}^{q}[t_i(k) - o_i(k)]^2 \tag{3.4}$$

E represents the square of the difference between the *k*-th target output vector *t(k)* and the *k*-th actual output vector *o(k)* of the network and index *i* represents the *i*-th neuron of the output layer composed of a total of *q* neurons.

Step 5: The weights are updated according to

$$\Delta w^{(l)} = -\eta \frac{\partial E(k)}{\partial w^{(l)}} \tag{3.5}$$

where $\frac{\partial E(k)}{\partial w^{(l)}}$ is the gradient of the error E with respect to the vector $w^{(l)}$ corresponding to all interconnection weights between layer *(l)* and the preceding layer *(l-1),*η is the learning rate parameter indicating the convergence and stability behavior of the learning algorithm and is a small positive number usually between 0 and 1. $\delta w^{(l)}$indicates the difference between the vectors $w^{(l)}(k+1)$ and $w^{(l)}(k)$. In other words it represents the interconnection weights leading to neurons at layer *(l)* after and before the presentation of the training pattern k.

Step 6: The above process is repeated with another input-output pattern. One epoch or iteration is completed once after all the patterns from training dataset are presented to the network

Step 7: The process of training the network is carried out through a large number of training sets and training cycles. As the training process progresses, the RMS error (root mean square error between the target and the actual output) decreases until it reaches a minimum threshold value. The training phase stops if any one of the following conditions occurs: when the designated number of epochs is reached, designated amount of time has been executed or the RMS error reaches the threshold value or goal.

Adaptive Neuro-Fuzzy Inference System (ANFIS)

Neuro-fuzzy system is a combination of neural network and fuzzy system in such a way that neural network learning algorithms are used to tune parameters of the fuzzy system. ANN is usually considered as a black box that is able to provide a correct matching in the form of output data for a set of previously unseen input data. Leaning mechanism of ANN does not rely on human expertise. ANN are capable to learn from the scratch by adjusting interconnections between layers, but due to homogenous structure of ANN it is

difficult to extract structured knowledge from the weights or the configuration of an ANN.

Fuzzy Inference System (FIS) is a popular computing framework based on the concept of fuzzy set theory. The knowledge base of a FIS with its rule base containing a number of if-then rules and database which defines the membership functions of the fuzzy sets used in the fuzzy rules is capable of utilizing human expertise. Although it is very handy to express the knowledge as a set of if-then rules, it is impossible to extract knowledge stored in the form of numerical data. Consequently, the association ANN and FIS lead to mutually beneficial relationship that eliminates mentioned shortcomings. This is embodied through integrated Neuro-Fuzzy (NF) architectures that share data structures and knowledge representations. Adaptive Neuro-Fuzzy Inference System (ANFIS) is a particular type of NF architectures proposed by Jang (1993). ANFIS architecture is presented in Figure 3.8, where two fuzzy if-then rules based on a first order Sugeno model are considered:

Rule 1: If x is A_1 and y is B_1 then $f_1 = p_1 \times x + q_1 \times y + r_1$

Rule 2: If x is A_2 and y is B_2 then $f_2 = p_2 \times x + q_2 \times y + r_2$

where x and y are inputs, A_i and B_i are fuzzy sets, f_i are the outputs within the fuzzy region specified by the fuzzy rule and p_i, q_i and r_i are linear parameters which are tuned during the training process.

The first layer consists of adaptive neurons. The fuzzy membership grade of the inputs are the outputs of this layer and are given by

$$O_i^1 = \mu_{A_i}(x) \text{ , i=1, 2, or} \tag{3.6}$$

$$O_i^1 = \mu_{B_{i-2}}(y) \text{ , i=3, 4,} \tag{3.7}$$

where $\mu_{A_i}(x)$, $\mu_{B_{i=2}}(y)$ can adopt any fuzzy membership function. Gaussian membership function is used in this study (Figure 3.7) given by

$$\mu_{A_i}(x) = e^{\frac{-(x-c_i)^2}{2\sigma_i^2}} \tag{3.8}$$

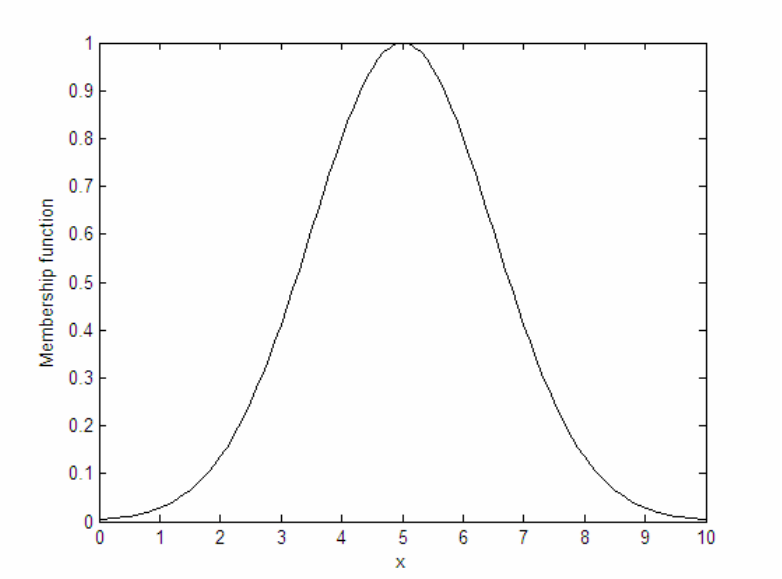


Figure 3.7: Gaussian Membership Function

where $\{\sigma_i$ and $c_i\}$ is the parameter set of the membership function (MF), governing the shape of the MF. The parameters in this layer are referred to as premise parameters.

The second layer consists of fixed neurons. Each node in this layer estimates the firing strengths of a rule by applying the AND operator. Firing strengths implies the degree to which the antecedent part of a fuzzy rule is satisfied. The output of this layer can be represented as the firing strengths of the rules

$$O_i^2 = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \quad i = 1,2 \tag{3.9}$$

The third layer consists of fixed nodes. Each node in this layer calculates the ratio of the *i*th rule's firing strength in inference layer to sum of all the rule's firing strengths. The output of each node is given by

$$O_i^3 = \overline{w} = \frac{w_i}{w_1 + w_2}, \quad \text{i=1, 2} \tag{3.10}$$

The fourth layer consists of adaptive nodes. The output of each node in this layer is computed as

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i), \quad \text{i=1, 2} \tag{3.11}$$

Here p_i, q_i and r_i are the modifiable parameters in this layer. They are referred to as consequent parameters.

The single node in the fifth layer computes the overall output as the summation of all incoming signals.

$$O_i^5 = \sum_{i=1}^{2} \overline{w}_i f_i = \frac{\sum_{i=1}^{2} w_i f_i}{w_1 + w_2} \tag{3.12}$$

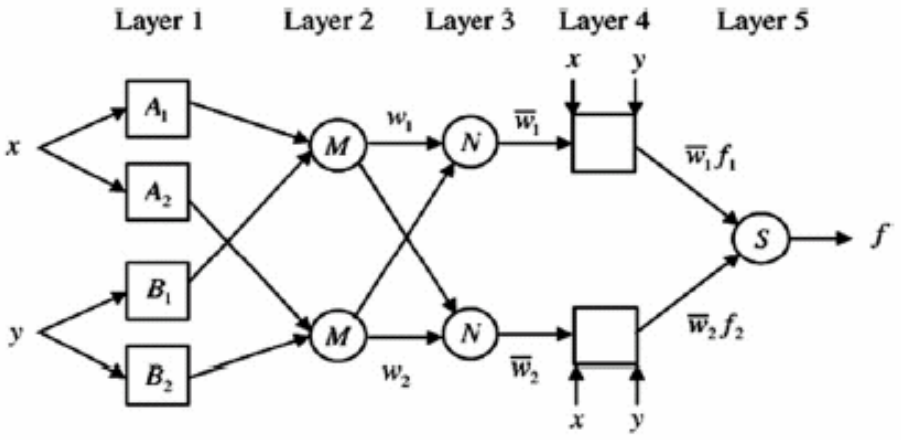


Figure 3.8: ANFIS Structure (Jung, 1993)

From the ANFIS architecture, we observe that the values of the premise and consequent parameters have to be tuned to make the ANFIS output match training data. When the values of the premise parameters of the MF's are fixed, the overall output can be expressed as a linear combination of the consequent parameters.

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \tag{3.13}$$

$$= \overline{w}_1 (p_1 x + q_1 y + r_1) + \overline{w}_2 (p_2 x + q_2 y + r_2)$$

$$= (\overline{w}_1 x) p_1 + (\overline{w}_1 y) q_1 + (\overline{w}_1) r_1 + (\overline{w}_2 x) p_2 + (\overline{w}_2 y) q_2 + (\overline{w}_2) r_1$$

The learning algorithm for ANFIS is a hybrid algorithm which is a combination between gradient descent and least squares method. More specifically, in the forward pass of the hybrid learning algorithm, node outputs go forward until layer 4 and the consequent parameters are identified by the least-squares method. In the backward pass, the error signals propagate backward and the premise parameters are updated by gradient descendent. The consequent parameters are identified optimal under the condition that the premise parameters are fixed. Accordingly, the hybrid approach converges much faster since it reduced the search space dimensions of the original pure backpropagation method.

3.4.2 Supervised Learning: Statistical Methods

k- Nearest Neighbour Classifier

The k-nearest neighbour classifier is a powerful and simple method of classification that has proved successful in a wide variety of applications like medicine, face recognition, signature recognition, handwriting recognition and food industry. k-NN classifier belongs to the category of supervised classifiers used for classifying objects based on training samples in the feature space. Classification of an object in the feature space is based on the maximum number of neighbours, with the object being assigned the most frequent class amongst its surrounding k nearest neighbours.

Algorithm of k-NN classifier

The algorithm consists of two phases: training and testing. In the training phase, the training samples represented with a set of attributes or features are stored in a multidimensional features space. The multidimensional space is divided into regions by locations and labels of the training samples.

Training set: $(x_1, y_1),(x_2, y_2),....,(x_n, y_n)$

Assume $x_i = (x_i^{(1)}, x_i^{(2)},..., x_i^{(d)})$ is a d-dimensional feature vector of real numbers for all i and y_i is a class label in $\{C_1, C_2...C_n\}$ for all i, where n stands for the number of classes.

In the testing phase, the new sample x_{new} whose class is not known is represented as a vector in the feature space. Distances from the new sample x_{new} to all samples in the training set are calculated. It is common to use the Euclidean distance, though other types of distance measures like the Manhattan distance could be used instead.

The Euclidean distance between two points is $X = (x_1, x_2,..., x_n)$ and $Y = (y_1, y_2,..., y_n)$ is defined as

$$D(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3.14}$$

Based on the calculated distances the number of k nearest samples to the new sample is selected. The class of the new sample is predicted based on the majority of the class of the k nearest neighbours. The modules of k-NN classifier are shown in Figure 3.9. The first and second modules consist of the training and testing set. The measurement of distance is considered as the third module. The fourth module consists of finding the class that has the maximum number of neighbors closest to the sample from the testing

set and the decision is taken in the fifth module. No training is required as the training set already represents the features of the samples to be learnt.

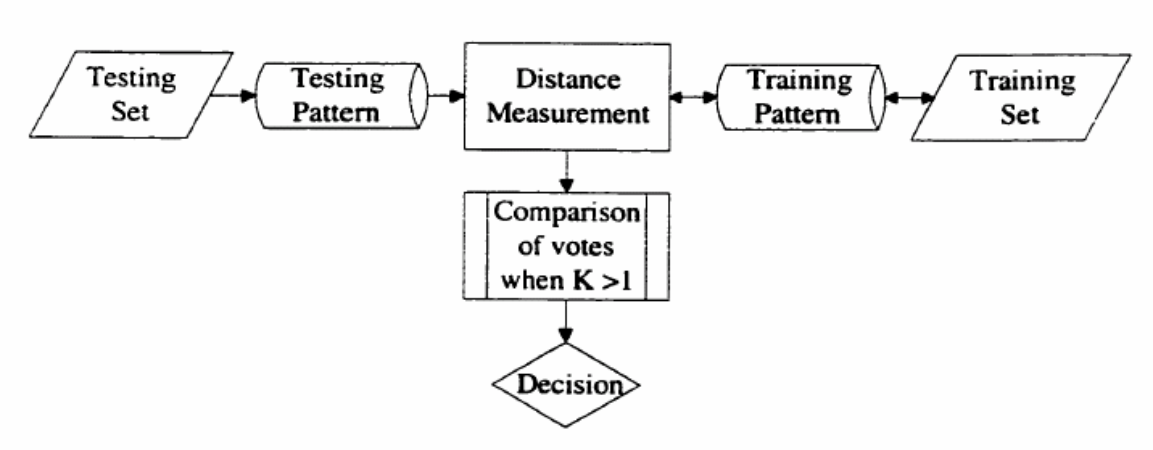


Figure 3.9: Modules of k-NN classifier (Kandir, 2000)

Due to the fact that no training is required k-NN classifiers can be used to solve any classification problem because they do not have to spend additional effort and time for classifying additional classes. The new training samples can be added to the training set and are considered for classification only if they are among the nearest neighbours of the new sample to be classified. However, k-NN classifiers have some drawbacks because there is no training, every time a new classification is performed the Euclidean distances has to be examined for the entire training data set. The efficiency of this approach decreases with increase in the number of training samples making them more efficient for problems with limited training data set.

Classify: Linear Discriminate Analysis

Linear Discriminate analysis is a standard statistical method used for classifying samples of unknown classes based on training samples of known classes. LDA is used for data classification, although it is commonly used for data reduction prior to classification. In LDA, the original data is transformed into a new feature space in which class

separability can be carried out more effectively. LDA maximizes the ratio of between-class variance to within-class variance (Fukunuga, 1990). The main purpose is to maximize this ratio so that adequate class separability is achieved.

Let the training set consists of a set of N samples $\{f_1, f_2, ..., f_N\}$. Each sample belongs to one of M classes $\{C_1, C_2, ..., C_M\}$. The within-class scatter matrix S_w is defined as

$$S_w = \sum_{j=1}^{M} \sum_{i=1}^{n_j} \left(f_i^j - m_j\right)\left(f_i^j - m_j\right)^T, \qquad m_j = \frac{1}{n_j} \sum_{i=1}^{n_j} f_i^j \tag{3.15}$$

where f_i^j is the *i*th sample of the *j*th class, n_j the number of samples for the *j*th class and m_j the mean of the *j*th class. The between-class scatter matrix S_b is defined as

$$S_b = \sum_{j=1}^{M} \left(m_j - m\right)\left(m_j - m\right)^T, \qquad m = \frac{1}{N} \sum_{i=1}^{N} f_i \tag{3.17}$$

where m is the mean of the entire data set. In LDA, within-class and between- class scatter are used to formulate criteria for class separability. The optimizing criterion for class separability is the ratio of between-class scatter to within-class scatter. The optimizing criterion is given by

$$\max imize = S_w^{-1} S_b \tag{3.18}$$

The transformation function found by maximizing the ratio of between-class scatter to with-in class scatter rotates the axes of the original data so that when the classes are projected on the new axes, the differences between the classes are maximized. After transformation, Euclidean distance (Equation 3.13) is used to classify data points. In the testing phase, test vectors are transformed and the Euclidean distance of the test vectors from the class means is calculated. The test vector is classified as belonging to the class which has the shortest distance.

3.4.3 Unsupervised Learning

Fuzzy C-means Clustering

Fuzzy C-means Clustering (FCM) belongs to the category of unsupervised learning wherein the algorithm identifies patterns and classifies samples based on the patterns present in them. Clustering is accomplished by grouping of patterns based on similarities between the individual patterns. The patterns that are similar to the highest extent are assigned to the same cluster (Pedrycz, 1997). Thus a collection of samples is partitioned into a number of clusters, where the samples inside a cluster show a certain degree of similarity. The clustering starts with an initial guess for the cluster centers, which are intended to mark the mean location of each cluster. In the next step, the samples are assigned a membership grade for each cluster which describes the degree of membership to that cluster. As with fuzzy sets, each sample can belong to more than one cluster with varying degrees of membership. Based on the membership grades, a sample is classified as belonging to the cluster for which it has the highest membership grade.

The FCM algorithm determines the following steps with a data set x_j, where j=1,…n , n being the number of data points. The first step in FCM consists of randomly selecting the cluster centers from among all the data points. The second step consists of determining the membership matrix U according to Equation 3.19

$$\sum_{i=1}^{c} u_{ij} = 1, \forall j = 1,..., n \qquad (3.19)$$

wherein u_{ij} is the membership grade (between 0 and 1) of j_{th} data point belonging to i_{th} cluster, c_i is the centroid of cluster i, d_{ij} is the Euclidian distance between i_{th} centroid(c_i) and j_{th} data point, m ϵ [1,∞] is a weighting exponent. The third step consists of computing

the dissimilarity function given in Equation 3.20 which gives the distance between the data points and centroids

$$J(U, c_1, c_2, ..., c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j=1}^{n} {u_{ij}}^m {d_{ij}}^2 \tag{3.20}$$

This process is stopped if its improvement over previous iteration over the previous iteration is below a threshold. The fourth step consists of updating the cluster centroids and membership matrix according to

$$c_i = \frac{\sum_{j=1}^{n} {u_{ij}}^m x_j}{\sum_{j=1}^{n} {u_{ij}}^m} \tag{3.21}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)}} \tag{3.22}$$

Detailed algorithm of fuzzy c-mean clustering can be found in the paper by Bezdek, (1973). By this iterative process, FCM moves the cluster centers to the right location within a dataset by iteratively updating the membership functions and cluster centers for each data point. The termination criteria for the iterations occur after a specified number of iterations or, if less than a specified number of objects change clusters. The performance of the algorithm depends on the initial positions of centroids which are randomly selected. So the algorithm gives no guarantee for an optimum solution. This can be overcome by running FCM several times with different initial centroids.

Chapter 4

Numerical Investigations

4.1 Introduction

For the objectives of this work, either experimental or numerical investigations can be used. The later method is fast and economical through which a detailed parametric study can be carried out in less time, once the model is calibrated and validated. However, considerable effort is needed for the calibration and validation of the model. In the case of experimental investigation, the incorporation of all the parameters dictating the problem chosen is quite complicated, laborious and sometimes may not be possible.

This chapter explains the details of the numerical models constructed using LS-DYNA, finite element software to simulate the propagation of Rayleigh waves in a homogeneous elastic half-space. The principles of the finite element technique and the basic concepts of the numerical finite element code (LS-DYNA) that is used in this study are described in the first section of this chapter. A literature review of the numerical work for flaw detection is provided in the second section. In the third section, the details of the numerical models developed for the present study, the details of the boundary conditions, loading conditions and surface responses obtained are presented. These 2-D ax-symmetric finite element models simulate the MASW (Multiple Analysis of Surface Waves) test method that facilitates the investigation of the behaviour of Rayleigh wave in the presence and absence of anomalies in the medium. To simulate anomalies, voids of different sizes were introduced in the medium. Sufficient numbers of samples were generated by changing the input loading conditions inorder to maintain enough variance in the dataset. A total of 2400 samples were generated.

This chapter in addition, outlines the parameters associated with the numerical simulation of surface waves which include model boundaries, damping parameters, temporal and spatial discretization parameters, source configuration and model calibration to Lamb's theoretical solution. The effect on the surface responses due to the presence of voids of different sizes has also been presented.

4.2 Numerical modeling of Rayleigh wave propagation

4.2.2 Numerical methods

Background

Physical problems result in differential equations. To obtain the exact solution in closed form is difficult. Therefore, numerical methods are the only other alternative to obtain approximate solution. Many numerical methods such as Rayleigh-Ritz method, Galerkin method, the least square method, finite difference and finite element methods are often used to obtain approximate solutions.

Numerical simulation methods to model the propagation of Rayleigh waves have greatly improved in the last two decades due to the rapid progress in computers. The earlier approach to model, wave propagation problems consists of analytical methods solving the governing equations of motion and their boundary conditions. However, this approach can be utilized for specimens with simple geometries and for perfect specimens (having no defects). The main advantage of these numerical methods lies in their ability to model specimens with complicated geometries and for imperfect specimens having defects in them.

A number of different numerical methods to solve elastic wave propagation problems have been reported in the literature. Alterman et. al, (1981) and Berthholf, (1967)

provide a review of these different methods. Among these methods, finite element (FEM) and finite difference methods (FDM) are extensively used in this field (Boore, 1972).

The main difference between the two methods is that: finite difference method is an approximation to the differential equation, whereas the finite element method is an approximation to its solution. Table 4.1 illustrates the main differences between FDM and FEM. The primary advantage of FEM is its versatility in handling complex domains and availability of numerous commercial FE codes overcomes the need to develop the actual code. These commercial software provide several advantages of being user friendly and providing sophisticated pre- and post- processing options.

4.2.2 Finite Element Method

History of FEM

The origin of finite element analysis dates back to the Ritz method of numerical analysis in 1909, however, the term finite element was first developed by Courant in 1943 who used the Ritz method of numerical analysis and calculus of variation in 1943 to obtain approximate solutions to problem of vibration systems. A major turning point in the development of FEA occurred in 1956 with a paper titled “stiffness and deflection of complex structures” by Tumer et al. In 1960, the term “finite element” was coined by Courant and around this time engineers began applying method of approximate solutions to problems in stress analysis, fluid flow and aerospace industry. By the late 1960s and early 1970s FEA had become established as a general numerical technique for solving any system of differential equations. However, the usage of this method was limited owing to the requirements of powerful computers and thus was used only in the aeronautical, automotive, defense and nuclear industries.

Table 4.1: Difference between FEM and FDM

FDM	FEM
FDM defines the domain as an assemblage of grid points.	FEM models the domain as a set of piecewise continuous sub-domains called finite elements bounded by finite number of nodes.
This method was developed for solving the system of ordinary differential equations, ODE's for problems having simple geometries with simple boundaries	This method was designed to model inhomogeneous, nonlinear materials bounded by irregular boundaries
FDE approximates the derivatives in the differential equations by difference equations between grid points. Therefore it requires rectangular grids.	FEM approximates the unknown in term of linear independent interpolating functions called the shape functions and a set of unknown parameters.
FDM is restricted to solve one-dimensional and two dimensional domains with simple geometries and with boundaries parallel to the coordinate axes.	FEM can be used to solve one, two and three dimensional domains and allows usage of irregular meshes of varying density and allows using elements of varying geometry, order of approximation thereby enabling to achieve different degrees of accuracy in different parts of the domain.
The quality of approximation between grid points is low.	Since it uses shape functions that approximate the distribution of the unknown function over the domain, FEM has a good quality of approximation. The quality of the approximation increases by either decreasing the size of the elements or increasing order of the interpolating functions.
FDM is mostly used for the analysis in the area of computational fluid dynamics.	FEM is mostly utilized for analysis in areas of structural dynamics and structural mechanics.

The advent of micro-computers (pc's and workstations) in the 1980's has made this method widely applied to solve a wide variety of problems. The development in mainframe computers and availability of powerful microcomputers has made this method used in a variety of applications.

Principles of FEM

The key idea of the finite element method is to discretize the actual domain of the object into an assemblage of sub-domains called finite elements which are connected by a finite number of boundary nodes. Thus, the continuum having 'infinite' degrees of freedom is discretized into a number of finite elements having 'finite' degrees of freedom defined at a set of nodal points. The grid of finite elements is assigned the material and structural properties which defines how the real structure would respond to the loading condition. The forces actually distributed in the real structure are transformed to act at the nodal points. Thus, the response of the real structure is approximated from the response of the discretized object obtained from the assembly of finite elements. The response of each element is expressed in terms of degrees of freedom characterized by unknown functions defined at a set of nodal points. For structural problems, these nodal degrees of freedom are unknown values of displacements or translations of the material defined in each coordinate direction. It has been found that the accuracy of the response progressively increases as the density of mesh increases or the mesh size becomes finer which in turn would be computationally very expensive. This in turn calls for a trade off between accuracy and computational resources. The selection of optimum number of elements should be based on (Bhatti, 2005).

1. The required subdivision of the continuum is to be based on past experience

2. In absence of past experience the optimum subdivision should be based on convergence study for various mesh grading.

Principles of FEM in wave propagation problems:

General equation for wave propagation is

$$Lu = F \tag{4.1}$$

where *F, u* represent force and displacement vectors and L is a linear operator of the form

$$L = [M]\frac{\partial}{\partial t^2} + [C]\frac{\partial}{\partial t} + [K] \tag{4.2}$$

where *[M], [C], [K]* represent the mass, damping and stiffness matrices respectively.

One approach for solving the above equation of motion is to analytically solve the governing differential equations and their associated boundary conditions. However, this procedure can be utilized only for problems having simple geometries. For complex geometries and geometries having imperfections in them, numerical techniques are often utilized.

In case of steady state dynamic problems, the time variable in equation (4.2) is eliminated and proper consideration is given only to spatial discretization (Valliappan et al., 1977). In case of transient dynamic problems, Equation (4.2) is solved following an integration scheme in time domain in addition to the spatial discretization. This requires proper consideration to be given in the selection of both time and spatial discretization parameters. Thus the temporal and spatial resolution of the finite element model is critical for the accuracy of finite element results. The implications of improper discretization of each of these parameters are discussed below:

In the spatial domain, if the element size is too big, the mesh causes removal of high frequency (short wavelength) energy. This removal results in two adverse effects. Firstly

the low pass filtering of mesh producers spurious oscillations called Gibb's phenomenon (Valliappan and Murti, 1984).Secondly the wave velocity obtained is higher than the actual velocity which occurs as a result of premature arrival of the waves. On the other hand, a smaller element size can lead to numerical instability. A larger time step leads to numerical instability and affects the accuracy of the results. On the other hand a smaller time step produces spurious oscillations (Gibb's phenomenon).

Convergence, stability and consistence

The basic property that a scheme must possess is that its numerical solution converges to the exact solution and this convergence improves as the discretization parameters in both spatial and time domain decreases or tends to zero. This property is called convergence of a scheme.

In practice, direct effort to prove that a given method is convergent is difficult, so this has been replaced by evaluating the stability and consistency of the method. A scheme that satisfies both completeness and compatibility requirements is said to be consistent. These conditions are briefly described as follows:

Completeness: The finite elements must have enough power to approximate the analytical solution in the limit of a mesh refinement process.

Compatibility: The shape functions must provide displacement continuity between elements.

Stability: Stability is the requirement that a scheme must satisfy in order to have the same solution uniqueness properties as the analytical solution.

Marfurt, 1984 used the following stability conditions for the determination of spatial (Δx) and temporal (Δt) discretization parameters

$$\alpha = V_p \frac{\Delta t}{\Delta x} \tag{4.3}$$

where

$\alpha = 1$ for finite element solutions of the scalar and elastic equations

$\alpha = \frac{1}{\sqrt{2}}$ for finite difference solutions of the scalar and elastic equations

$\alpha = \frac{V_P}{\sqrt{V_P^2 + V_S^2}}$ for finite difference solutions of the elastic equations.

V_P and V_S correspond to the compressional and shear wave velocities. The following expressions have been recommended in References (Valliappan and Murti, 1984) & (Zerwer et al., 2002) to determine the spatial discretization parameter. The selection of element size is based on the highest frequency (f_{max}) of the lowest velocity wave (V_R) and is computed as

$$g \leq \chi \lambda_{\min} \tag{4.4}$$

where $\lambda_{\min}$ is the minimum wavelength and is given by

$$\lambda_{\min} = \frac{V_R}{f_{\max}} \tag{4.5}$$

and constant χ must be less than 0.5 because of the Nyquist limit and further depends on whether the mass matrices are consistent (χ =0.25) or lumped (χ =0.2). This formulation is based on the assumption that the elements have square dimensions.

Alleyne and Cawley, 1998 used the following equation for the determination of element dimension which is based on 10 nodes per wavelength.

$$g = \frac{\lambda_{\min}}{10} \tag{4.6}$$

Marfurt,1984 used a much higher condition for the element size determination based on 20 nodes per wavelength.

$$g = \frac{\lambda_{\min}}{20} \tag{4.7}$$

where g is the element length and λ_{min} is the shortest wavelength of interest.

Calculation of the time step depends on the element dimension computed with the following expression (Zerwer et al., 2002):

$$\frac{1}{10}\frac{g}{V_p} \le \tau \le \frac{g}{V_p} \tag{4.8}$$

where τ = characteristic time; g=element dimension; V_p compressional wave velocity. This time step calculation incorporates the spatial Nyquist limit. The time step must also incorporate the temporal Nyquist limit given as

$$f_{\max} < \frac{1}{2\Delta t} \tag{4.9}$$

Mesh grading:

As far as possible, uniform grading of finite elements are always preferred throughout the model. This uniformity is essential in order to maintain the relative importance of each frequency components in wave propagation problems. Mesh grading is acceptable only when the grading is mild.

4.2.3 LS-DYNA

General

LS-DYNA is a general purpose non-linear finite element code for analyzing large deformation dynamic response of structures. The origin of LS-DYNA dates back to DYNA3D, a public domain software which was developed by Lawrence Livermore

National Laboratory in the mid-seventies. Later, it underwent several revisions in the following years in terms of new capabilities being added. In 1989, Livermore Software Technology Corporation was founded to continue the development of DYNA3D as a commercial version called LS-DYNA3D which was later shortened to LS-DYNA (Hallquist, 1991-1998). The present LS-DYNA has a huge library of sophisticated material models, an extensive element library with both under-integrated and fully-integrated element formulations and allows different contact conditions (Hallquist, 1992-2005). All these features make it suitable to investigate large deformations thereby making it capable of simulating many complex real world problems.

Structural module of LS-DYNA

In general, the structural module of LS-DYNA has three phases.

- Pre-processing
- Processing or analysis
- Post-processing

Pre-processing

The first step consists of constructing the finite element model of the structure to be analyzed. The geometric shape of the structure is constructed followed by the discretization of the model which consists of specifying the number, size, type and the arrangement of the finite elements within the model. The next stage consists of defining the loading, material and boundary conditions. The structure is represented by a grid of finite elements having the same properties as the structure. LS-DYNA is designed to operate with a number of commercial pre-processing software like TRUEGRID, FEMB,

ANSYS, and HYPERMESH. It also has its own built in preprocessor LS-INGRID with limited capabilities.

Processing

The next stage of the FEA process is the analysis. The FEM conducts a series of computational procedures taking into account the effect of the applied forces, and the element properties which produces the model solution. This analysis results in the determination of effects such as deformations, strains, and stresses which are caused by applied structural loads such as velocity, force, pressure and gravity.

Post-processing

The last step in FEA is post processing the results obtained from step 2. This stage involves presentation of the distribution of stresses and strains, deformed configuration of the structures, mode shapes etc. During post-processing contour plots, time histories, fringes and animation of large number of quantities may be interactively plotted on the meshes of actual model.

4.3 Previous related work: Numerical NDT work for flaw detection

Numerical simulation methods to model the propagation of Rayleigh waves have greatly improved in the last two decades. Alterman et. al, (1981) and Berthholf (1967) provide a review of the different numerical methods to solve elastic wave propagation problems. Among these methods, Finite element (FEM) and Finite difference methods (FDM) are extensively used in this field (Boore, 1972).

Lamb (1904) calculated the vertical and horizontal displacement histories generated by a point source acting on the free surface of a homogeneous half-space and found that geometric damping is smaller for Rayleigh waves than for body waves. Watkins et al

(1967) and Rechtien & Stewart (1975) conducted numerical studies for the detection of near surface cavities. They reported distances and concentration of energy in the time domain signals in the vicinity of the cavity. Hirao and Fukuoka (1982) developed numerical models using finite difference method to investigate the scattering of Rayleigh waves by surface edge cracks. The depth of the cracks was determined from the increasing Rayleigh wave arrival times. The numerical results were in good agreement with the experimental observations. Al-Hunaidi (1993) justified the use of axi-symmetric or plane numerical models to simulate wave propagation problems and the numerical results had a good agreement with the field data. Imran (1995) developed numerical models using FEM to recommend the best field-test configuration to be used to detect and size defects with Rayleigh waves. They reported the existence of standing wave energy in the vicinity of crack and suggested one receiver and source located close to the crack on one side and a second receiver located on the opposite side.

Gucunski et al (1996) developed finite element models to investigate the effect of underground obstacles on the dispersion curves obtained from SASW method. They reported reflections from the near and far faces of the void which resulted in strong fluctuations in the dispersion curves. Ganji (1997) detected obstacles from the fluctuations in the dispersion curve, due to reflections of Rayleigh waves from the obstacle. They developed finite-element models to simulate the SASW test and compared it with the experimental results. The application of the SASW to detect underground obstacles is limited to shallower depths. Different numerical studies have shown significant effects due to presence of voids on the surface responses of a medium in both time and frequency domain.

Phillips et al., (2000, 2002) conducted laboratory and numerical along with field tests and reported that the presence of an underground void causes energy concentrations for certain frequencies over the location of the void. Nasseri et al., (2004) reported significant effects on the surface responses of a medium due to the presence of embedded anomalies. Nasseri (2006) developed finite difference numerical models to simulate MASW test in presence and absence of anomalies. The numerical results were verified with experimental and laboratory results. It was found that the anomaly starts to vibrate due to the interaction with the Rayleigh wave which causes partitioning of energy. A part of the energy is reflected in form of Rayleigh waves and another part is converted into body waves which get dissipated into the medium. A new technique was proposed to locate and determine the depth of the anomaly. Nasseri et al., (2006) conducted field and numerical experiments using finite difference models (MASW method) to investigate the effects of underground cavities on Rayleigh wave. The displacement time histories along the surface were recorded. They reported amplitude changes in the displacement time histories in the region over the void and concentration of energy in the Fourier spectra in the vicinity of the void.

4.4 Numerical Methodology

General

For several engineering applications, many elements of a structure can be approximated by a half-space with reasonable accuracy especially over short distances. In a half-space, the medium extends in all directions and there are no boundary effects except at the surface. In this study, the half space profiles are assumed to be homogeneous that extends to infinity in horizontal and vertical directions. In addition, it is assumed that the material

model to be elastic, isotropic and the waves are plane. The propagation of plane waves is independent of the properties of the medium in one direction. This condition reduces the wave propagation problem to a two dimensional problem. The coordinate system used in this study consists of the horizontal direction represented by x-axis and the vertical direction represented by y-axis. The numerical modeling activity in this research study consists of four stages. The first stage comprised of two dimensional finite element models constructed to simulate the propagation of Rayleigh waves through a homogeneous elastic half space. Model calibration was done in the second stage for the theoretical Lamb Solution. Rectangular voids with different sizes were incorporated into the medium in the third stage. The maximum penetration depth of Rayleigh wave is $2\lambda_{ch}$ (λ_{ch} is the characteristic wavelength), however as a rule of thumb, Rayleigh waves are very sensitive to the mechanical properties at depth of $\frac{\lambda_{ch}}{3}$ (Huch Doyle, 1995). Based on this concept, the sizes of the voids were chosen. The concept of characteristic wavelength is discussed in section on normalization and characteristic wavelength. Three different sizes of voids were chosen: small sized void which responds to $0.5(\frac{\lambda_{ch}}{3})$, medium sized void responds to $1(\frac{\lambda_{ch}}{3})$ and the large size void responds to $2(\frac{\lambda_{ch}}{3})$. The fourth stage consists of changing the type of input loading in order to generate sufficient number of samples. Table 4.2 gives a description of the different models adopted for the study. Herein, model Model_1 corresponds to a homogeneous model without any void, where as, the other models correspond to a homogeneous model with different degrees of void.

Table 4.2: Model types used in this study

Model Type	Description
Model_1	2-D Homogeneous elastic half-space
Model_2	2-D elastic half-space with a small void [0.5($\frac{\lambda_{ch}}{3}$)]
Model_3	2-D elastic half-space with a medium void [1($\frac{\lambda_{ch}}{3}$)]
Model_4	2-D elastic half-space with a large void [2($\frac{\lambda_{ch}}{3}$)]

Numerical Model Description

Numerical models were developed using commercial non-linear finite element software (LS-DYNA). All the numerical simulations were executed with double precision solver. The size of the model is 250 mm in both horizontal (X direction) and vertical (Y direction). The basic model consists of a mesh of 595 numbers uniformly graded finite elements each of which of size 0.42mm both in the vertical and horizontal directions consisting of 354,025 finite elements spread over the entire model. To record the surface responses, 75 recording locations were chosen along the surface of the model (X direction). The surfaces responses were recorded at a distance of every other 8 grid points. Therefore the distance between the recording locations is 3.36 mm. Both the horizontal and vertical displacements were recorded at these locations. All the numerical models were developed from the basic model shown in Figure 4.1.

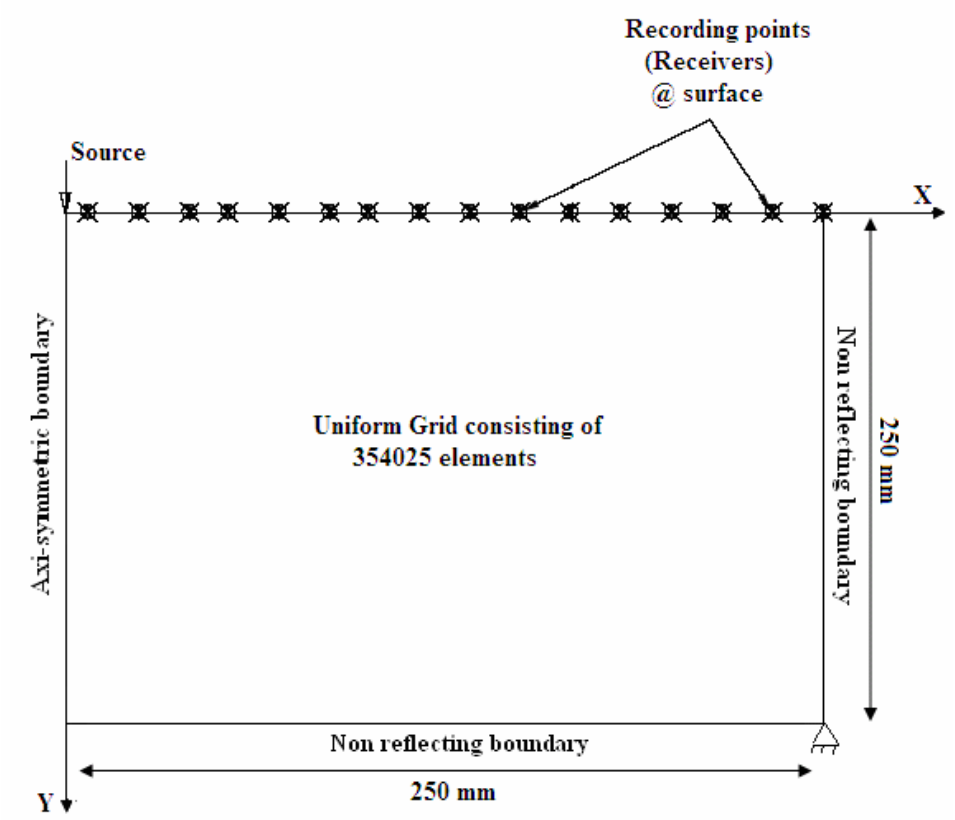


Figure 4.1: General geometry of the model

Test Material Properties

In LS-DYNA different material models can be incorporated into one structure, for example, by making some elements null or removing some elements, the effect of presence of voids can thus be simulated. In addition, it is assumed that the material of the model is elastic and isotropic. The material properties of the chosen model are given in Table 4.3.

Table 4.3: Material properties of the model

Material Properties	Values
Density (Kg/m^3)	2340
Poisson ratio (υ)	0.25
Modulus of Elasticity E (MPa)	45000
Shear Modulus G (MPa)	18000
P-wave velocity V_p (m/s)	4800
Shear wave velocity V_s (m/s)	2770
Rayleigh wave velocity V_R (m/s)	2550

These material properties represent typical values for sound concrete structure.

Boundary Conditions

Finite size of the numerical model is one of the most important problems associated with numerical simulation of wave propagation problems. The finite boundary of the finite element model causes the waves to be reflected from the mesh boundaries and superimposed with the progressing waves which finally contaminate the desired solution. This can be eliminated by increasing the mesh dimensions which in turn is expensive in terms of computer storage and execution time. Another way of overcoming this problem is by introducing absorbing or non-reflecting boundaries which is equivalent to making the grid boundaries transparent to the outgoing energy (Kenneth, 1985) , thereby, minimizing reflections from the boundary.

The non reflecting boundary type used by LS-DYNA is based on the viscous boundary method developed by Lysmer and Kuhlemeyer (1969). It applies viscous dampers in the normal and shear directions on the boundary elements. This boundary condition can be

expressed by the following equation which ensures that all energy arriving at the boundary is absorbed

$$\sigma = a\rho V_p \dot{w} \tag{4.10}$$

$$\tau = b\rho V_s \dot{u} \tag{4.11}$$

where V_p and V_s are the p-wave and shear-wave velocities; σ and τ are the normal and tangential stresses; a and b are constants, whereas, $\dot{w}$ and $\dot{u}$ are the normal and tangential velocities respectively. The quiet boundary conditions can be implemented by modifying the governing equation of motion ($M\ddot{u} + C\dot{u} + Ku = f$) to include dashpots oriented normal and tangential to the boundary. Numerical analysis of the absorbing boundary method shows that for a given choice of a and b the effectiveness of the absorption depends on the angle of incidence θ of the waves. Perfect absorption cannot be achieved over the whole range of incident angles by any choice of a and b. Nearly perfect absorption is attained for angles of incidence greater than 30 degrees for a = b = 1. For smaller angles of incidence, this method is not very effective; however some absorption of energy still takes place. The absorption characteristic of this method depends on the frequency of the wave and it can be applied to harmonic and transient waves.

In this study, the bottom and right boundaries of the model are free and non-reflecting boundary conditions were applied to them inorder to represent an infinite half space and to minimize the effect of reflections. The left boundary is fixed in the horizontal direction, thereby, representing the axes of symmetry. The free body motion of the model in X and Y direction is prevented by applying translational constraints in both the X and Y directions on the lower right corner of the model.

Calibration

To establish the validity of finite element models used in this study, they were first calibrated with a known theoretical solution. This is done in order to ensure that the finite element results provide a certain degree of confidence in their accuracy, since; exact analytical solutions are not available for complex problems like the one taken up under the present study. The results of this calibration were used to develop guidelines for the application of finite element analysis for more complicated geometries.

Input Loading: Lamb's Problem

Lamb (1904) calculated the vertical and horizontal displacement time histories generated by point source acting on the free surface of a semi-infinite half-space. The choice of the Lamb source is justified for calibration as the source transfers 2/3rd of its energy to the medium in the form of Rayleigh waves (Barker, 1997), since the focus of this study is based on propagation of Rayleigh waves. Loading is simulated by point source acting vertically on the surface of an infinite half-space given by

$$F(t) = \frac{F_b}{\pi\left(t^2 + \Psi^2\right)} \tag{4.12}$$

wherein, F_b alters the force amplitude, ψ controls the width of the pulse (or the frequency content) and t represents time. In this study the Lamb source with the following parameters is chosen:

- $F_b = 0.75$
- Time shift = .0075 msec
- $\psi = .001$ msec

The graphical representation of Lamb source in time and frequency domain is shown in Figure 4.2. The source has energy spread over wide range of frequency of up to 1000 KHz (Figure 4.2 (b)) and is concentrated a short duration of about 0.01ms (Figure 4.2 (a)).

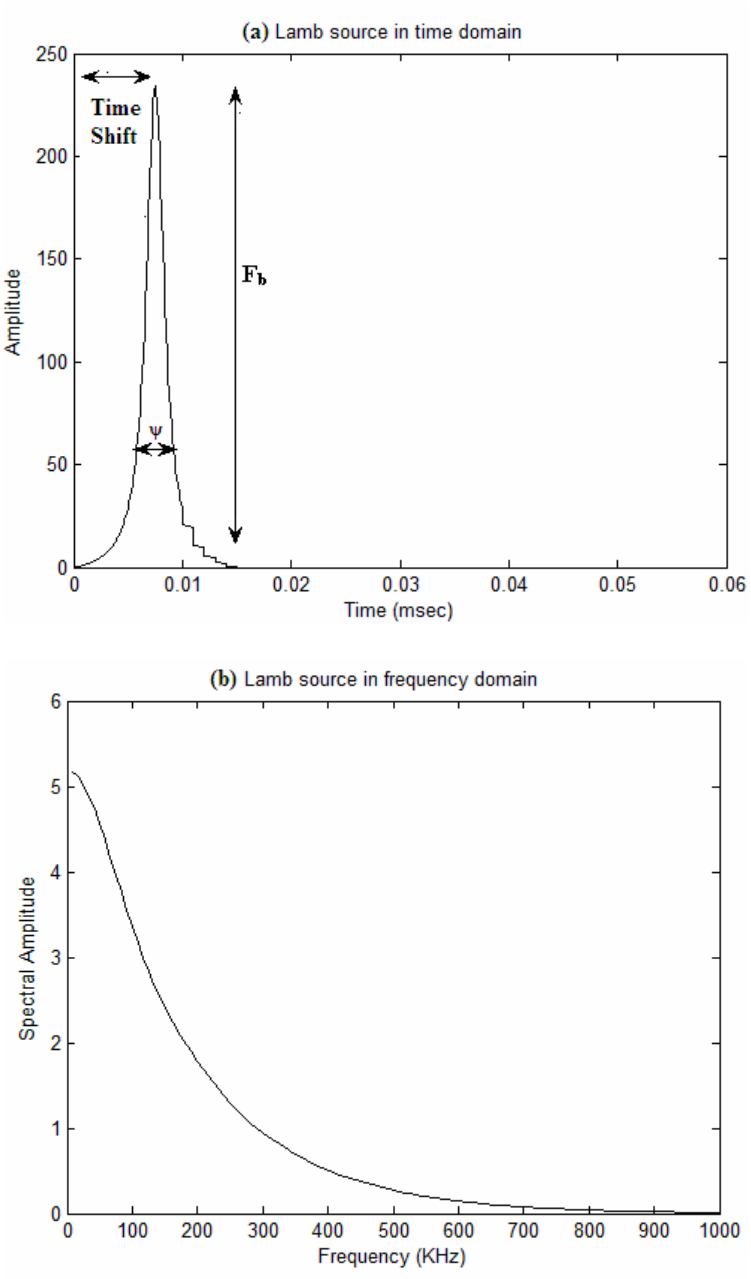


Figure 4.2: Representation of the used Lamb source in time (a) and frequency (b) domains

The cumulative distribution of energy as a function of frequency shown in Figure. 4.3. Almost the entire energy is concentrated within frequencies smaller than 300 KHz.

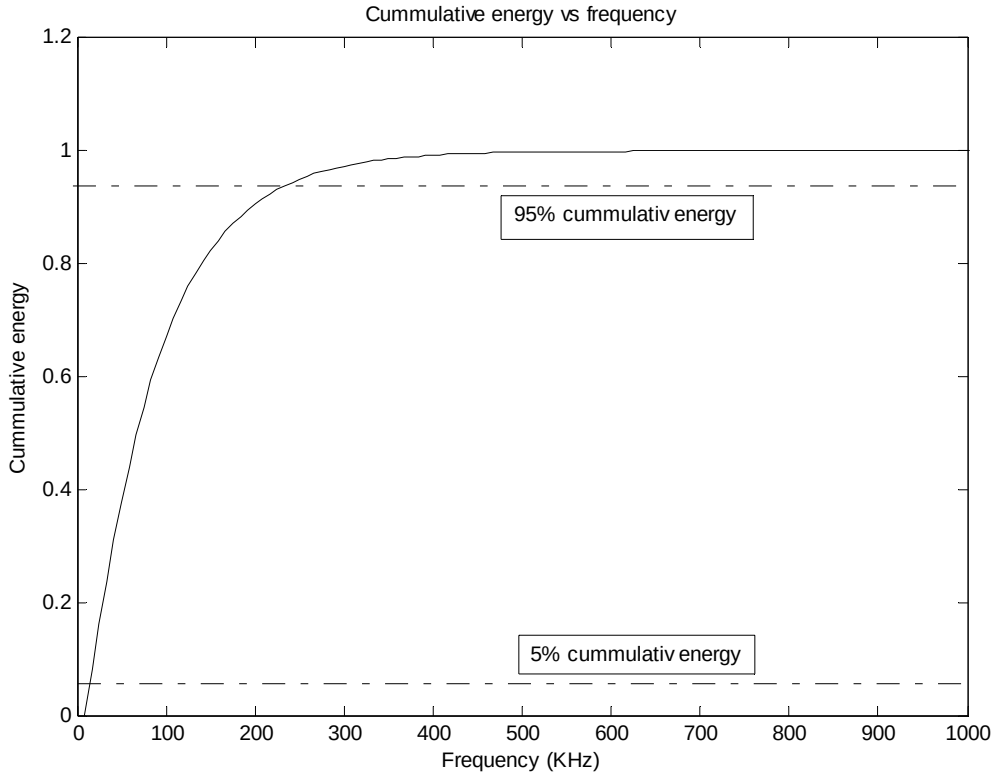


Figure 4.3: Cumulative energy versus frequency of the source

Information regarding the frequency content of the input source is given in Table 4.4, which is important in the selection of finite element discretization and damping parameters.

Table 4.4: Information about the frequency content of the input source

Source	**F_{cog} (KHz)**	**F_{max} (KHz)**	**$F_{5\%}$ (KHz)**	**$F_{95\%}$(KHz)**
Lamb(1904)	169.6	8.33	14	251

where F_{cog} = frequency corresponding to center of gravity of frequency spectrum

F_{max} = frequency corresponding to maximum spectral amplitude

$F_{5\%}$, $F_{95\%}$ = frequency corresponding to 5% and 95% of cumulative energy respectively

Damping

Damping occurs in dynamic systems due to dissipation of energy as a result of geometric spreading of wave fronts and inelastic nature of the medium. To simulate material damping in numerical models, the frequency dependent Rayleigh damping is adopted. Rayleigh damping has two components: mass-proportional and stiffness proportional and is given by (Liu and Gorman, 1995) as

$$C = \alpha M + \beta K \tag{4.13}$$

where α and β are proportionality constants for mass and stiffness respectively. Mass proportional damping is dominant at lower frequency ranges, while, stiffness damping is dominant at higher frequencies. The relationship between damping ratio and Rayleigh damping for a single degree of freedom system is given by

$$D = \frac{\alpha}{2\omega} + \frac{\beta\omega}{2} \tag{4.14}$$

The constants α and β are determined by (Leger and Dussault 1992; Woodward and Griffiths 1996)

$$\alpha = \frac{2D(\omega)\omega_1\omega_n}{\omega_1\omega_n}, \quad \beta = \frac{2D(\omega)}{\omega_1 + \omega_n} \tag{4.15}$$

where ω_1 and ω_n represent the first and the highest natural frequency of the vibration respectively. A constant damping can be achieved over a bounded frequency range (ω_1 to ω_n) by specifying the values of Rayleigh damping parameters α and β. In this study, values of ω_1 and ω_n are taken as 45 and 850 KHz respectively. The frequency variations of Rayleigh, stiffness and mass damping are shown in Figure 4.4.

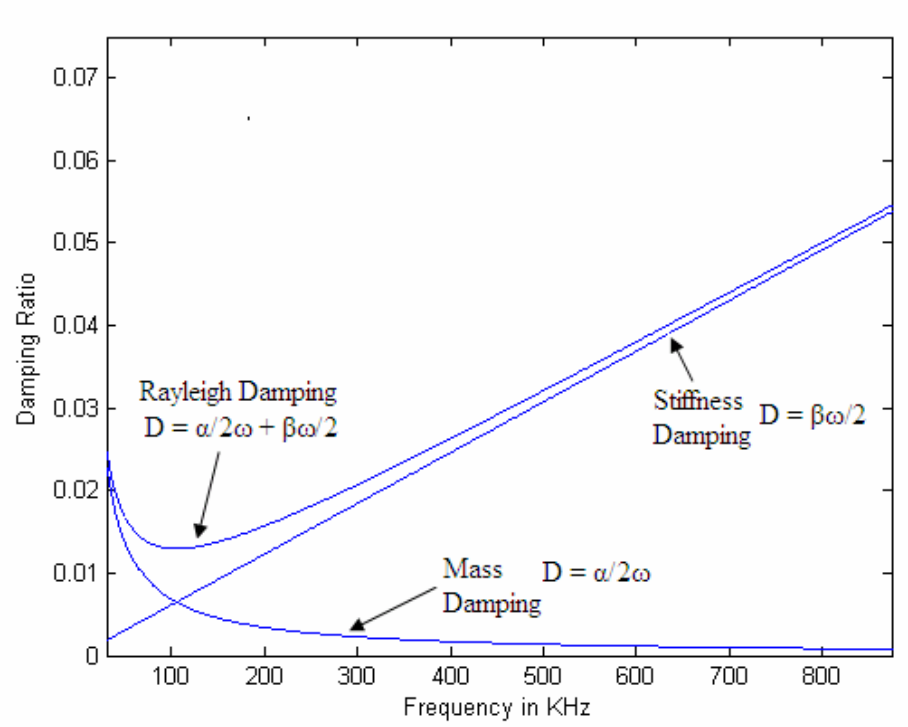


Figure 4.4: Relationship between Rayleigh damping parameters and damping ratio

Theoretical Response: Solution to Lamb Problem

The analytical solution to Lamb's problem for R-wave displacement is given by Bath and Berkhout (1984). The vertical (Equation 4.16) and horizontal (Equation 4.17) displacements are given by

$$u_v(t,d) = \frac{RF_b}{4\pi G\Psi^2 V_r}\sqrt{\frac{2\Psi V_R}{d}}\cos\left(\frac{\pi}{4}-\frac{3}{2}\chi\right)\cos^{\frac{3}{2}}(\chi) \qquad (4.16)$$

$$u_h(t,d) = \frac{RF_b}{4\pi G\Psi^2 V_r}\sqrt{\frac{2\Psi V_R}{d}}\sin\left(\frac{\pi}{4}-\frac{3}{2}\chi\right)\cos^{\frac{3}{2}}(\chi) \qquad (4.17)$$

$$\text{given } \chi = \tan^{-1}\left(\frac{t-\frac{d}{V_R}}{\Psi}\right)$$

where R is a constant that depends on the compressional, shear and Rayleigh wave velocities, V_R is the Rayleigh wave velocity, G is the shear modulus and d is the distance from the source. For calibration of the numerical model, only the solution to the vertical displacement due to Rayleigh wave is used in this study. Figure 4.5 (a) & (b) shows typical vertical and horizontal responses at different distances.

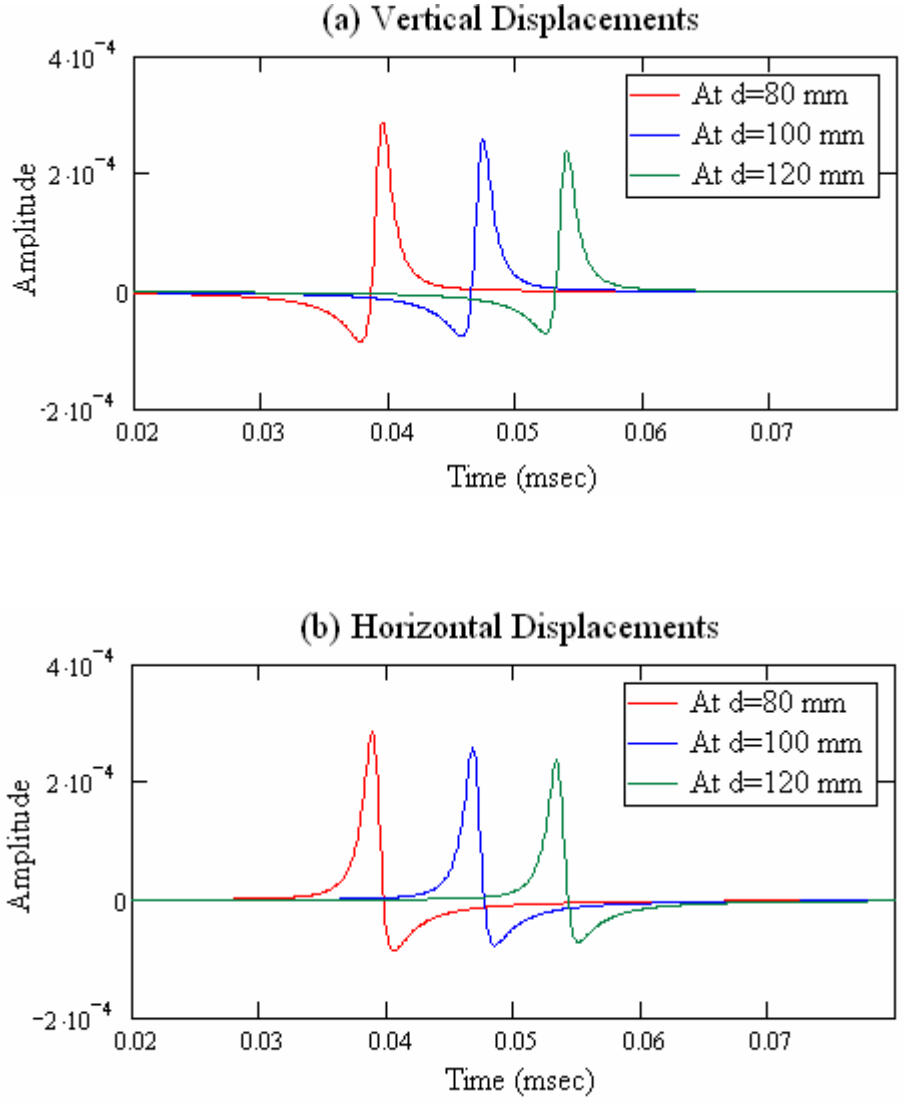


Figure 4.5: Typical vertical (a) and horizontal displacements (b) of Rayleigh wave

Spatial and temporal discretization parameters

Selection of mesh size Δs and time step Δt is based on the consistency, stability criteria described given in section 4.2.2. The mesh size is calculated by first computing the maximum mesh dimension. The maximum dimension depends on the highest propagating frequency, Rayleigh wave velocity and the constant χ (Equation 4.4). The value of the constant χ is adopted as 0.25 for the consistent mass approach used by LS-DYNA. The source has a wide frequency bandwidth from 1 KHz to 1MHz with the highest propagating frequency selected as 750 KHz which gives the maximum mesh dimension as 0.68 mm. An element size of 0.42 mm was chosen for the model which gives a total of 354025 elements. Plain strain shell elements are used mostly to model linear, elastic materials. The shell element is a four node element with six degrees of freedom at each node: translations in the x, y and z directions and rotations about the x, y, and z.

LS-DYNA uses an explicit finite element scheme which is conditionally stable. The time step must be smaller than the length of time required for a signal to traverse the distance between the node points of a single element. The time required for a signal to travel through an element is called the critical time Δt_{crit} and is computed automatically by LS-DYNA based on the bar's wave speed and maximum of the shortest side or the area/ (minimum of the longest side or the longest diagonal). In order to avoid instabilities, the critical time step is scaled down by a scale factor of 0.5 which gives a value of 4.12×10^{-5} ms and the maximum dynamic time (t_{max}) is selected as 0.12 ms.

Normalization and characteristic wavelength

The concept of characteristic wavelength λ_{ch} was introduced inorder to generalize the results (Nasseri, 2006). In other words, λ_{ch} is used as a benchmark for the measurement of distance and time. Figure 4.6 shows the frequency spectrum of the response at the first receiver. Wavelengths corresponding to the frequency of maximum spectral amplitude and frequency at the center of gravity of the spectrum were computed.

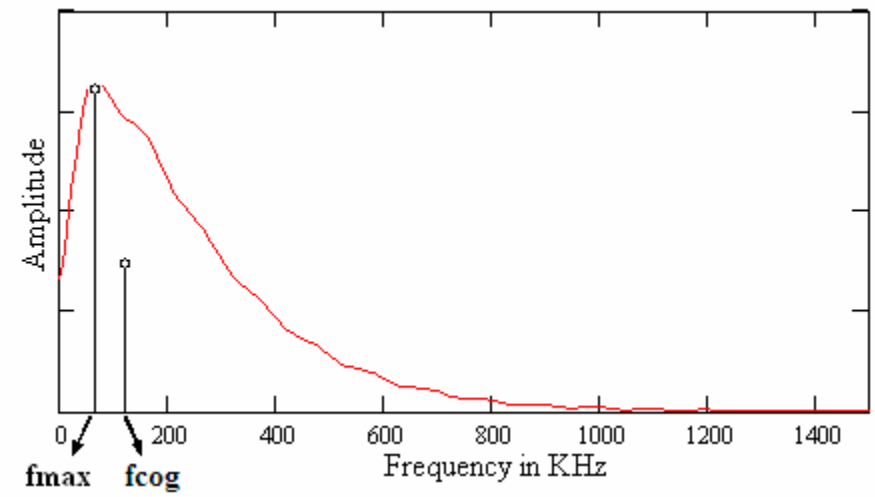


Figure 4.6: Frequency Spectrum of the response at a distance of 0.42mm from the source

Wavelength corresponding to $f_{\max} = 75\,\text{KHz}$ and $V_R = 2550$ m/s is 34 mm $(\lambda_{\max})$

Wavelength corresponding to $f_{cog} = 151$ KHz and $V_R = 2550$ m/s is 15 mm (λ_{cog})

The maximum of the two wavelengths is taken as the characteristic wavelength.

$$\lambda_{ch} = 34\,\text{mm} \tag{4.18}$$

The time taken by the Rayleigh wave to travel a length equal to the characteristic wavelength is defined as the characteristic time.

$$t_{ch} = \frac{\lambda_{ch}}{V_R} = 0.013\,\text{ms} \tag{4.19}$$

Based on the above definitions, the grid spacing ($\Delta s = 0.42$ mm) is $0.012\,\lambda_{ch}$ and the critical time step ($\Delta t = 4.12 \times 10^{-5}$ ms) is $3.17 \times 10^{-3}\, t_{ch}$

4.5 Model Validation

Comparison between the theoretical and numerical results

Verification of the numerical model was performed by comparing the model responses to the theoretical responses in time domain. For calibration only the vertical displacements at different locations along the surface were considered. Graphical representation of the calibration of typical numerical responses with theoretical vertical responses is showed in Figure 4.8(a) (where d indicates the distance from the source). The model responses were found to be in good agreement with the theoretical responses after a certain distance from the source which is referred to as the near field effect which occurs due to body wave interference (Zywicki and Rix, 2005). Since the geometric attenuation of body waves from an active source is larger than Rayleigh waves, the surface waves tend to dominate the wave field in relatively large distances. The commonly used criteria to overcome the near field effects has been proposed by Heisey et al. (1982), for the case, where, the distance between the source and first receiver is equal to the receiver spacing Δs which is given as

$$\Delta s > \frac{\lambda}{3} \tag{4.20}$$

where λ is the wavelength of Rayleigh wave. However, Nasseri et al (2006) found that near field effects are important for $\Delta s < 2\lambda$ .Thus near field effects would be upto approximately one to two λ_{ch} where $\lambda_{ch} = 34mm$. Figure 4.7 shows that after a distance of 34 mm ($1\,\lambda_{ch}$) away from the source, the numerical response is found to agree well

with the theoretical response of Lamb **(1904)** and after a distance of 70 mm ($2\lambda_{ch}$) the numerical responses is identical to the theoretical response. The oscillations at the end of the responses are the effects of the spatial and temporal discretization. Thus it can be concluded that the model has been well calibrated with the theoretical Lamb solution.

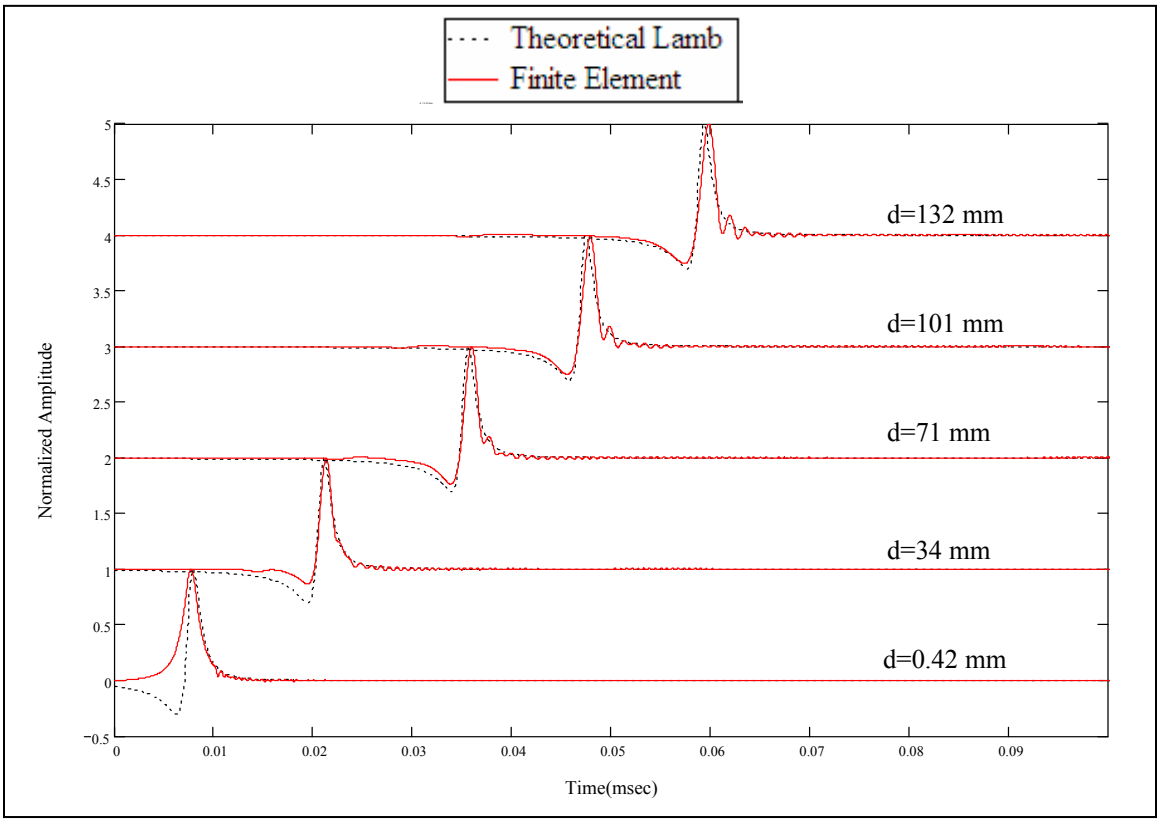


Figure 4.7: Comparison of typical numerical responses with theoretical Lamb's solution; d represents the distance from the source

For the purpose of validation, theoretical horizontal responses of Lamb's solution were utilized. Graphical representation of the validation of typical numerical responses with theoretical horizontal responses is showed in Figure 4.8 (where d indicates the distance from the source).

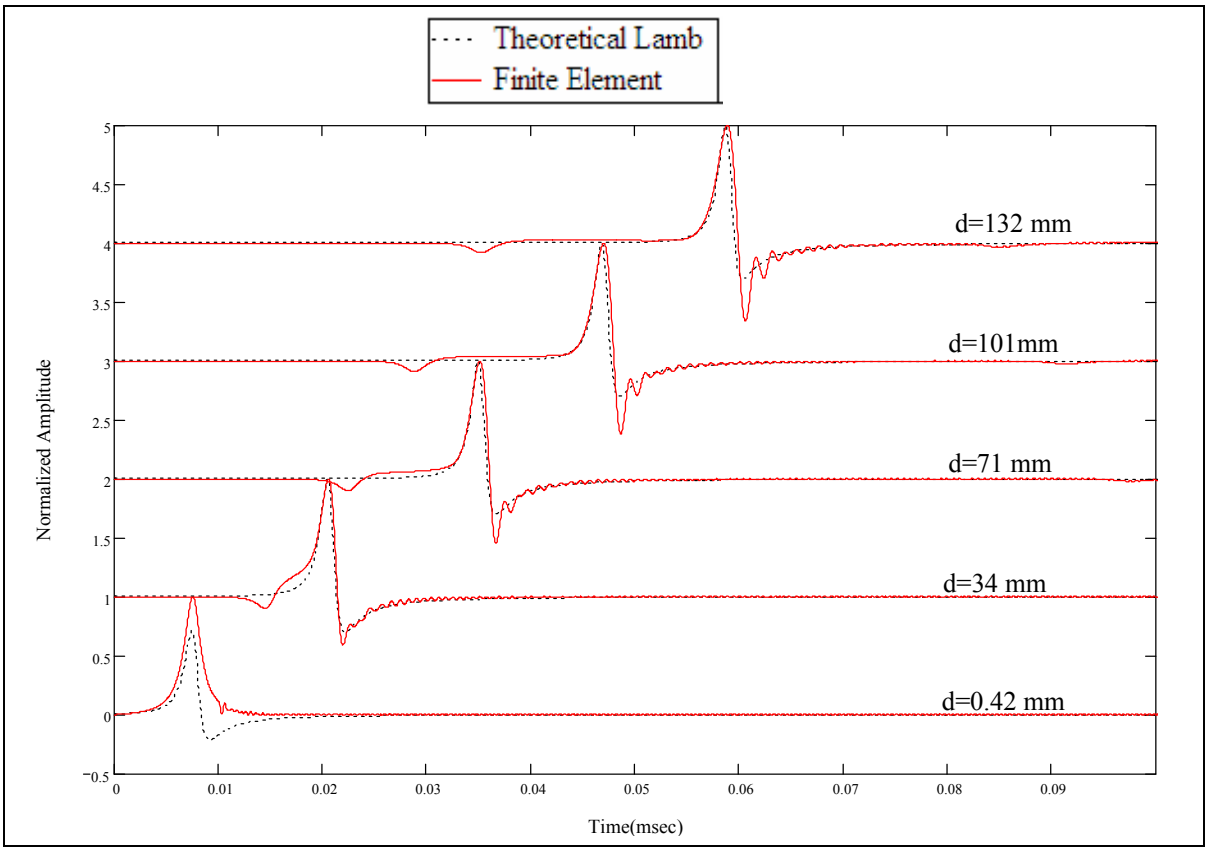


Figure 4.8: Validation of typical numerical responses with theoretical Lamb's solution; d represents the distance from the source

Model validation is essential in order to check if the calibrated model responds well to another type of an independent response. The model responses were found to be in good agreement with the theoretical horizontal responses which establishes the fact that the model responds well to sources other than the source (theoretical vertical responses) used for calibration.

4.6 Surface Responses

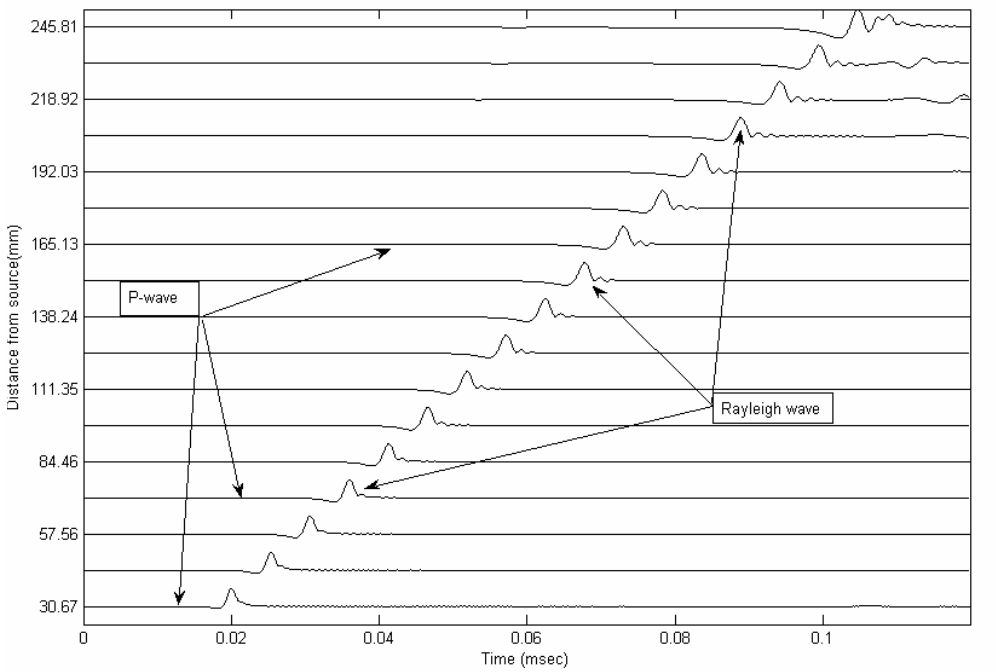
Time Responses

The wiggle plots of typical vertical and horizontal responses recorded along the surface of the model are shown in Figures (4.9 & 4.10). In these plots, the horizontal axis represents time in seconds and the vertical axis represents distance from the source. All

the responses are normalized by dividing it by the maximum of the responses and hence, the maximum value of each response will be equal to 1. Normalization of the traces is important in order to ensure that all the responses have equal importance. The large peaks seen in all the traces indicate the Rayleigh wave arrival, whereas, the small dips (seen more prominently in horizontal responses) indicate the arrival of P-wave. The oscillations seen at the end of the responses are the effects of spatial and temporal discretization. Due to the input source type (vertical impulse), relative amplitude of P-waves is found to be smaller in comparison with R-waves. On the other hand, a horizontal impulse would generate a stronger P-wave. As the Rayleigh wave velocity is close to the shear wave velocity (the ratio of R-wave to S-wave velocity is 0.9), the large amplitude of R-waves overshadows the S-wave arrival.

As the particle motion of p-wave is parallel to the direction of propagation, at surface the p-waves can be traced better in the horizontal responses as shown by both the wiggle and contour plots (Nasseri, 2006). The slope of each event represents the parameter $(\Delta d/\Delta t)$ which gives the apparent velocity of the event. The measured values of apparent velocity for p-wave and Rayleigh wave are 4761 (m/s) and 2532(m/s). These values agree well with the theoretical p-wave and Rayleigh wave velocities related to the model (Table 3.2). Reflection of the incident Rayleigh wave from the model boundaries can be also be seen in these figures; though p-wave reflections from model boundaries can be seen well only in the contour plot of the horizontal responses Figure 4.10(b).

(a)



(b)

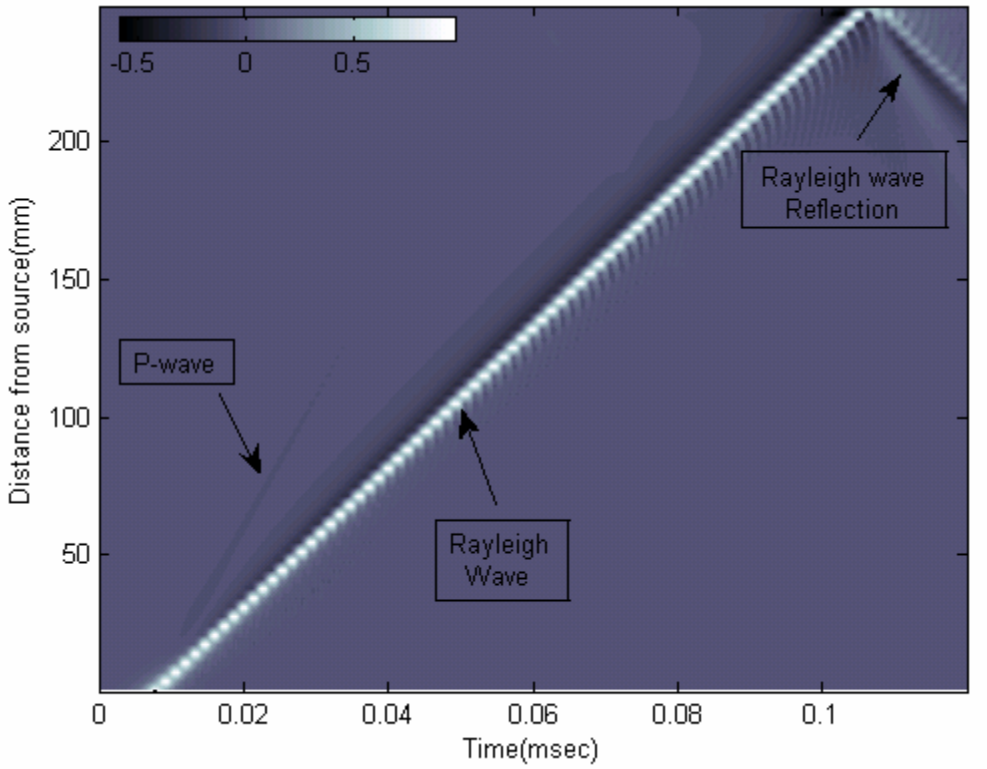
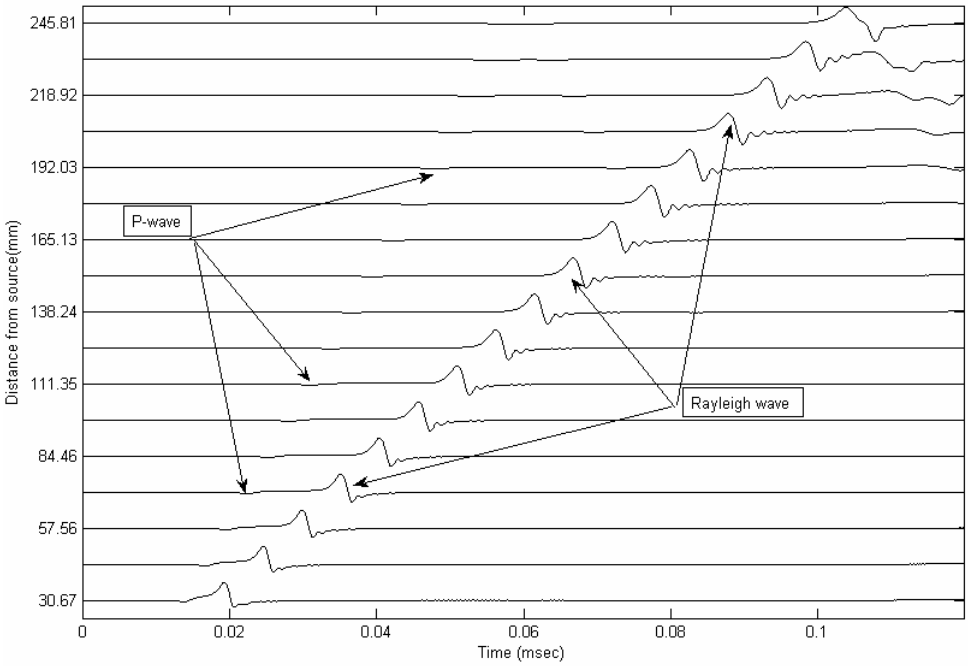


Figure 4.9: Normalized vertical responses along the surface of the model in form of (a) Wiggle Plot (b) Contour Plot (Model_1)

(a)



(b)

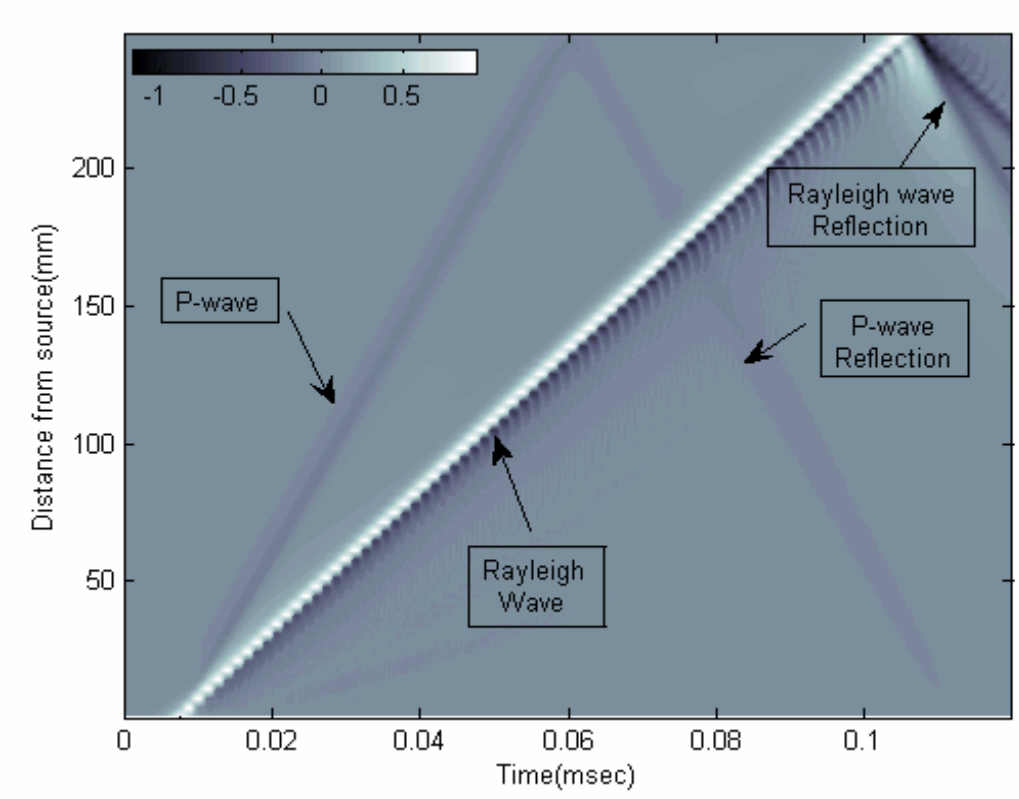


Figure 4.10: Normalized horizontal responses along the surface of the model in form of (a) Wiggle Plot (b) Contour Plot (Model_1)

Frequency Responses

The frequency spectrum of typical normalized vertical responses recorded along the surface is shown in Figure 4.11. The shape of the spectra does not change with distance, though reflections corresponding to incident Rayleigh wave in the time domain can be seen as successive peaks in the frequency domain. Contour plot of the same data is shown in Figure 4.12. The ripples in the contour plot indicate the presence of reflections in the recorded data (Robinson and Durrani, 1986).

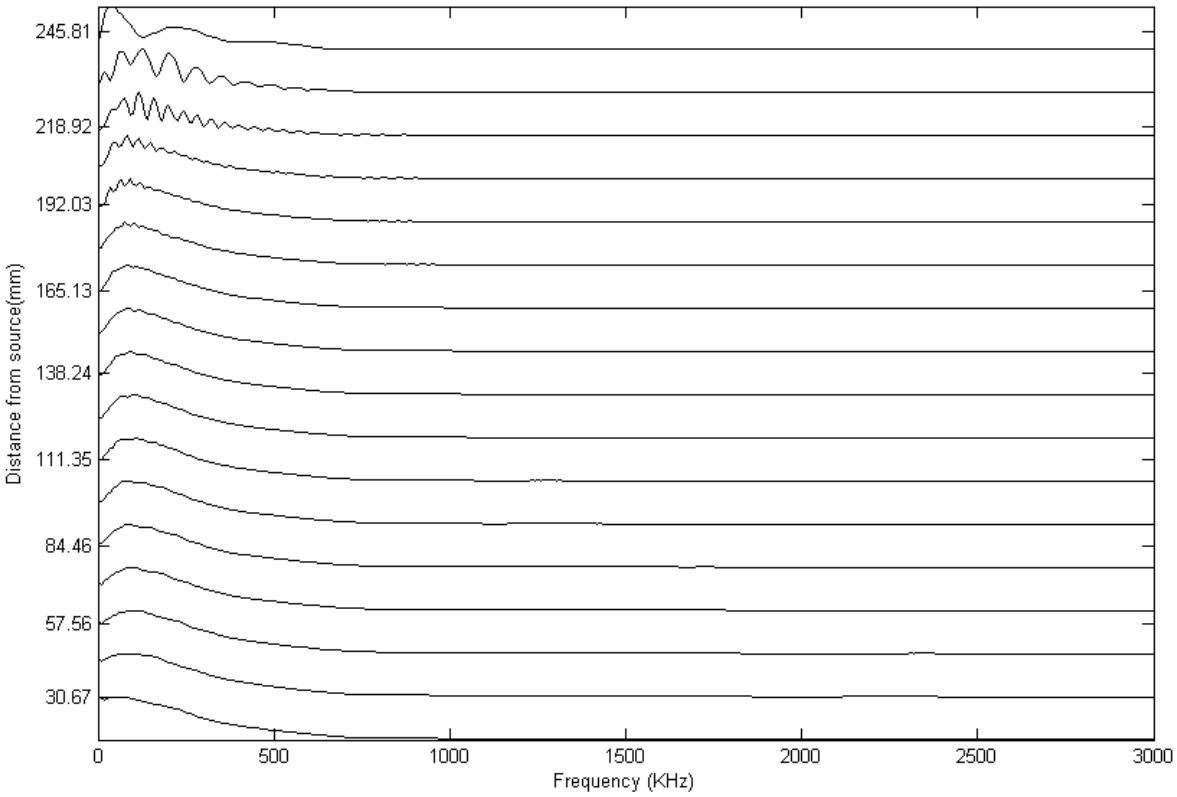


Figure 4.11: Frequency spectrum of the normalized vertical responses from Model_1

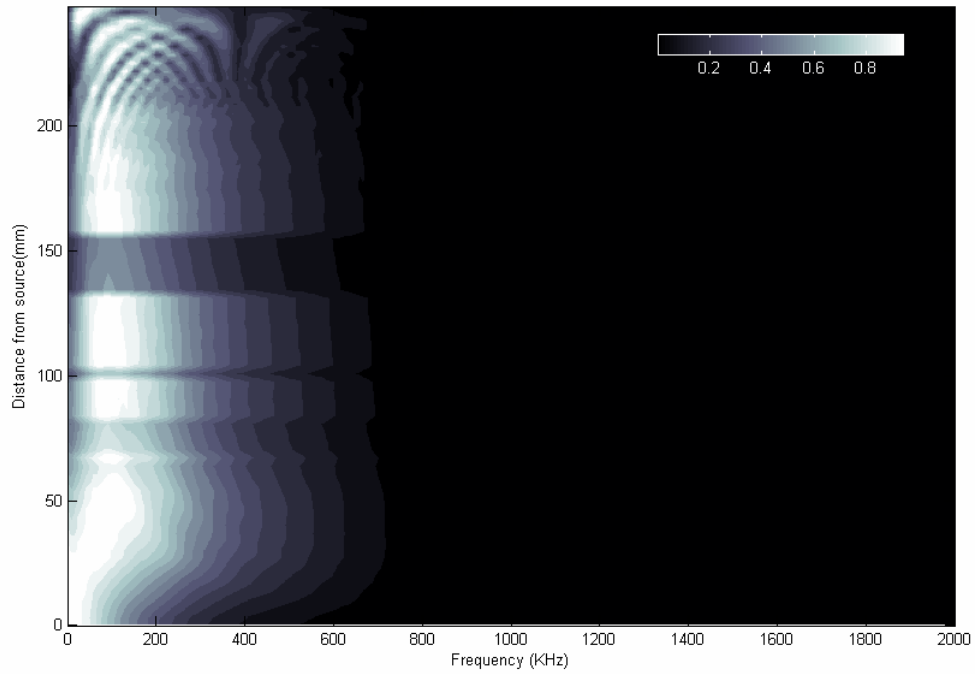


Figure 4.12: Contour plot of the frequency spectrum of the normalized vertical responses from Model_1

4.7 Finite element modeling of voids

The development of a decision support system for classification between a perfect model (without any void) and imperfect model with varying sizes of voids is attempted in this section. Three different sizes of voids were considered; small, medium and large, the dimensions of which are given in Table 4.5

Table 4.5: Dimensions of the different sizes of voids; dimensions in terms of the characteristic wavelength are also indicated

Type of Void	Embedment Depth (h) (mm)	Height (a) (mm)	Width (b) (mm)
Small	6.3 (.185 λ_{ch})	6.3 (.185 λ_{ch})	37.8 (1.1 λ_{ch})
Medium	6.3 (.185 λ_{ch})	12.6 (.37 λ_{ch})	75.6 (2.2 λ_{ch})
Large	6.3 (.185 λ_{ch})	25.2 (.74 λ_{ch})	151.2 (4.4 λ_{ch})

The general geometry of the model with void is shown in Figure 4.13. The voids are always centered at 125 mm from the source, thus the center of the voids is coincides with the center of the model.

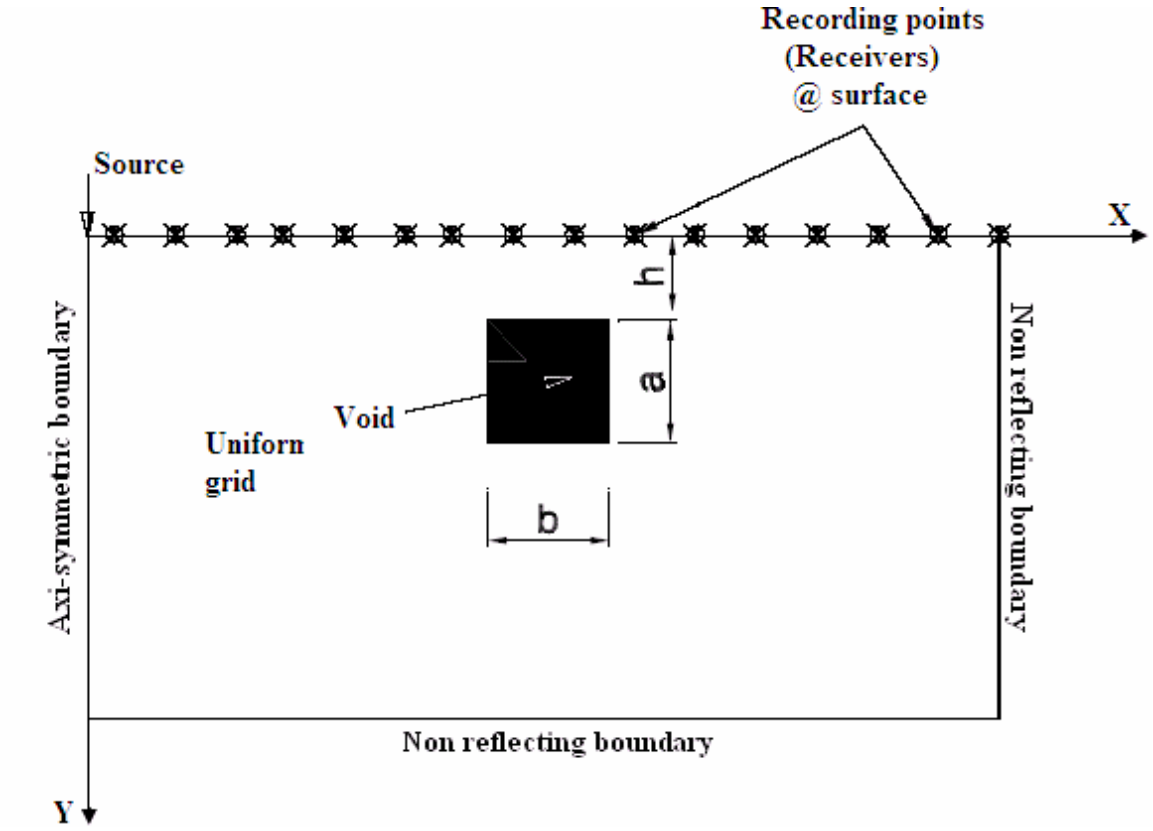


Figure 4.13: General geometry of the model with void

Generation of sample data set

The most significant factor in ensuring a good accuracy of any classification process is the presence of enough variance in the sample data set. This in turn requires generation of sufficient number of samples for each of the considered cases. For each of the considered cases, different samples were created by changing the type of input loading. Four different loading conditions were considered for each of the four different cases which yielded 16 numerical simulations.

In order to ensure sufficient variance in the dataset, the sources were chosen such that there is sufficient variation in the frequency content of the pulses generated by these sources. The sources used are Lamb (1904), Ricker (1945), impact and sinusoidal which are some of the commonly used sources in geophysics. The source functions are expressed as follows:

- ***Lamb***

The same source used for calibration (Equation 4.12) is used as the input source which is expressed as

$$F(t) = \frac{F_b}{\pi\left(t^2 + \Psi^2\right)} \tag{4.21}$$

where F_b alters the force amplitude, ψ controls the width of the pulse (or the frequency content) and t represents time. Graphical Representation of Lamb force is shown in Figure 4.14

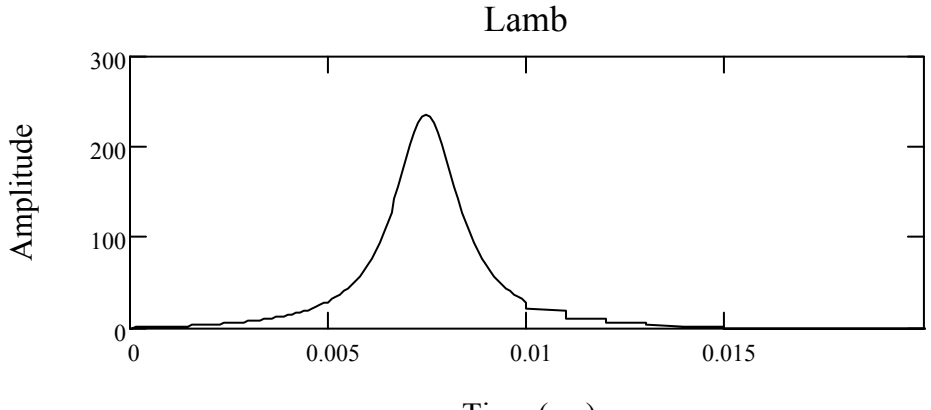


Figure 4.14: Lamb Source

- ***Ricker***

The Mexican hat wavelet often called the Ricker wavelet in Geophysics is usually used to model the seismic data. The Ricker wavelet is the normalized second derivative of a Gaussian function and is defined as

$$r(t) = \frac{F_b}{\sqrt{2\pi}}\left[1 - \frac{t^2}{\psi^2}\right]e^{\frac{-t^2}{2\psi^2}} \tag{4.22}$$

where F_b alters the force amplitude, ψ controls the width of the pulse (or the frequency content) and t represents time. Graphical representation of Ricker function is shown in Figure 4.15.

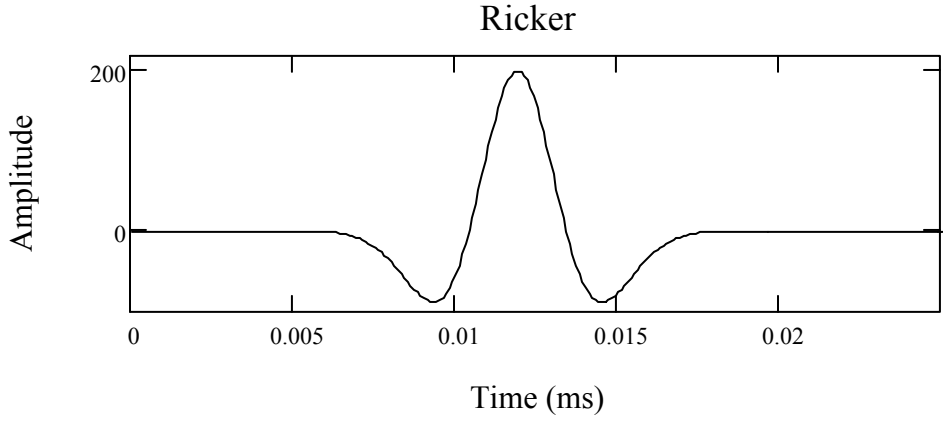


Figure 4.15: Ricker Source

- ***Impact***

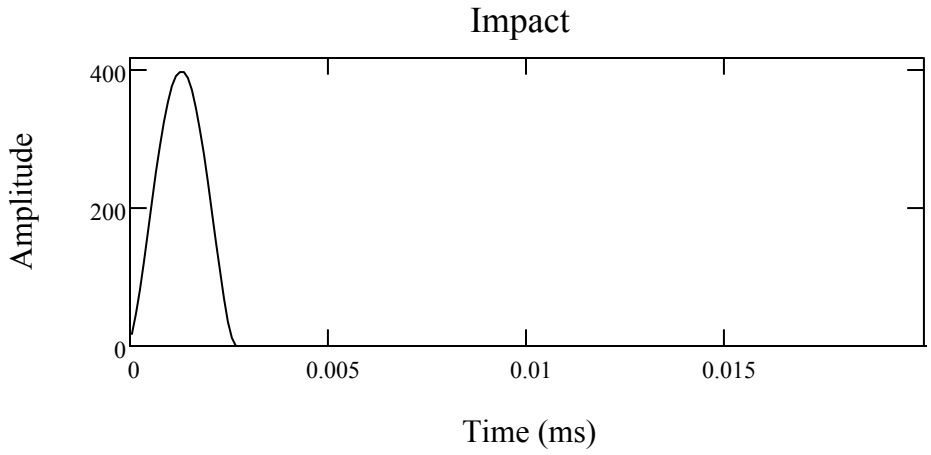
Impact is a commonly used source for generating stress waves in materials for example impact-echo method. The impact source is expressed as:

$$i(t) = F_b \sin(\alpha\pi t) q(t) \tag{4.23}$$

where F_b alters the force amplitude, α controls the width of the pulse (frequency content), t represents time and $q(t)$ is given by

$$q(t) := \begin{cases} \sqrt{\sin(\alpha \cdot \pi t)} & \text{if } \alpha \cdot \pi t < \pi \\ 0 & \text{otherwise} \end{cases} \tag{4.24}$$

Graphical representation of impact function is shown in Figure 4.16



4.16: Impact source

- ***Decaying Sinusoidal***

A decaying sinusoidal source with seven cycles has been used and is expressed as

$$s(t) = F_b \sin(\omega t) e^{(-\omega \xi t)} \tag{4.25}$$

where F_b alters the force amplitude, ω controls the width of the pulse (frequency content), ξ controls the decay..

Graphical Representation of Sinusoidal function is shown in Figure 4.17

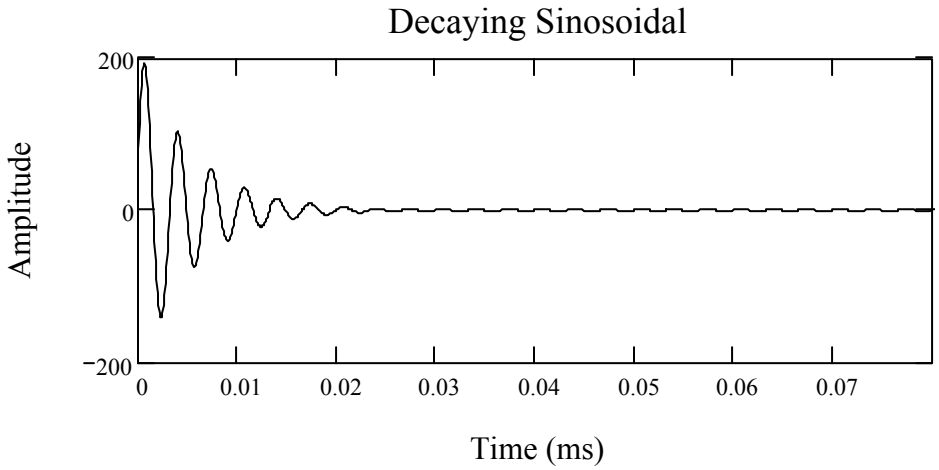


Figure 4.17: Sinusoidal Source

Comparison between the frequency content of the sources is showed in Figure 4.18 .There is a sufficient variation between the frequency content of the sources.

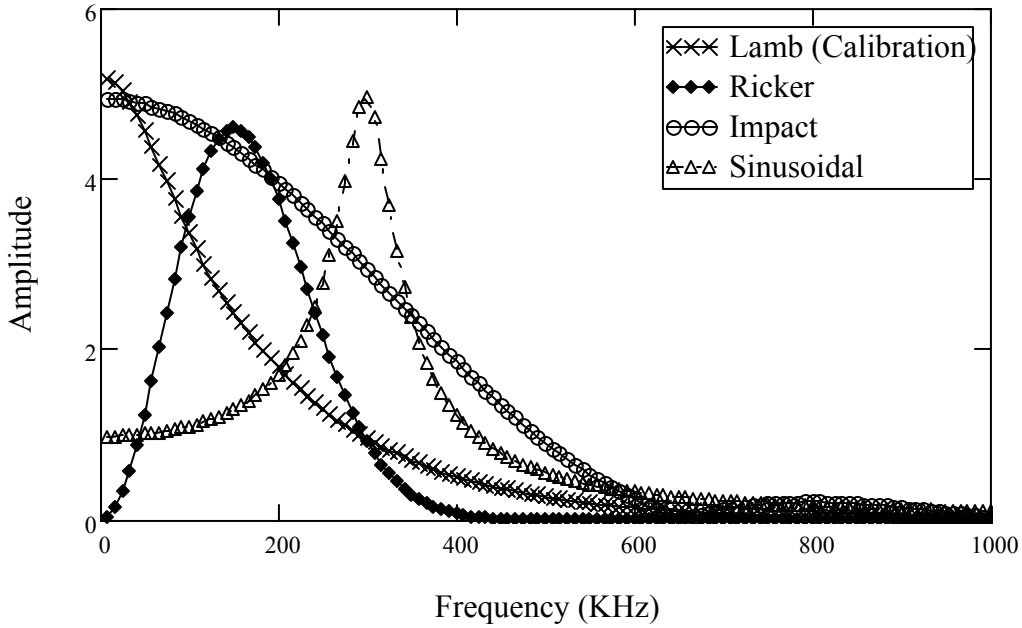


Figure 4.18: Frequency Spectrum of the sources

In each of the simulations, both horizontal and vertical responses were recorded at 75 locations along the surface of the model which gives a total number of 2400 samples (16 simulations * 75 horizontal responses * 75 vertical responses). A block diagram showing the different types of sample data set generation is presented in Figure 4.19

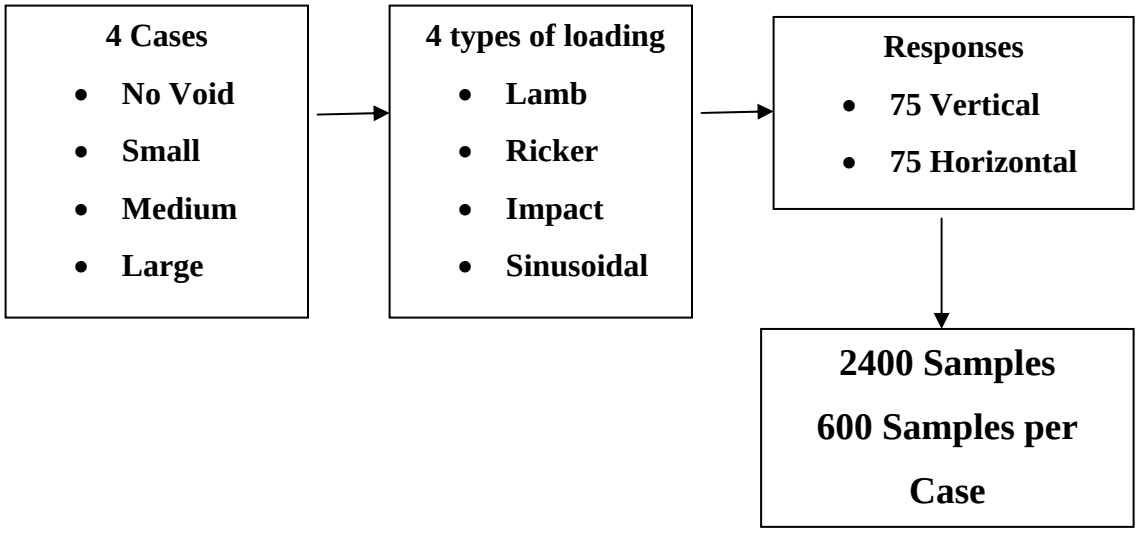


Figure 4.19: Process of data set generation

4.8 Surface responses with the voids

Figures (4.20 to 4.22) show the contour plots of the normalized vertical surface responses of the model when subjected to the Lamb source in the presence of voids of different sizes. Figures (4.23 to 4.25) show the contour plots of the frequency spectra responding to Figures (4.20 to 4.22) respectively. The general dimensions of the voids and the dimensions in terms of the characteristic wavelength are provided in Table 4.5. In the contour plots, the dashed lines represent the void boundaries projected onto the surface. The first dashed line represents the near boundary (NB) of the void, whereas, the second represents the far boundary (FB) of the void. The following trends were observed in all the three cases of different void sizes. The incident Rayleigh wave R1 undergoes reflection and transmission when it interacts with the near boundary of the void. The reflected part is represented as R2 while, the transmitted part as R3. The reflected wave R2 gets reflected from the boundary of the model represented as R4. The transmitted Rayleigh wave R3 is reflected as R5 and transmitted as R6 when it encounters the far

boundary of the void. The transmitted wave R6 gets reflected as R7 when it encounters the model boundary. The dimension of the voids plays an important role in determining the occurrence of the various events which can be seen in the contour plots. Numerical investigations with other input functions confirm these observed trends. The results of the models with the other input functions are presented in the Appendix. Nasseri (2006) found that the distance between the ripples seen in the contour plots of the frequency spectra is a function of the time delay between the main signal and its reflection, i.e. larger the delay, the closer will be the ripples to each other. This trend can be observed in Figures (4.23 to 4.25). The delay between the main signal and its reflection is larger for Model_2 than in Model_3 and Model_4 as seen in Figures (4.20 to 4.22). As a result of the differences in the time delays, ripples in Figure 4.23 (small void) are much closer to each other than in Figures 4.24 and 4.25.

Therefore, the surface responses reflect the differences between the various classes in both time and frequency domains, for example the pattern of the responses obtained from a small void was distinct from those caused by a medium sized void. These differences were useful in the detection and classification process of the void sizes.

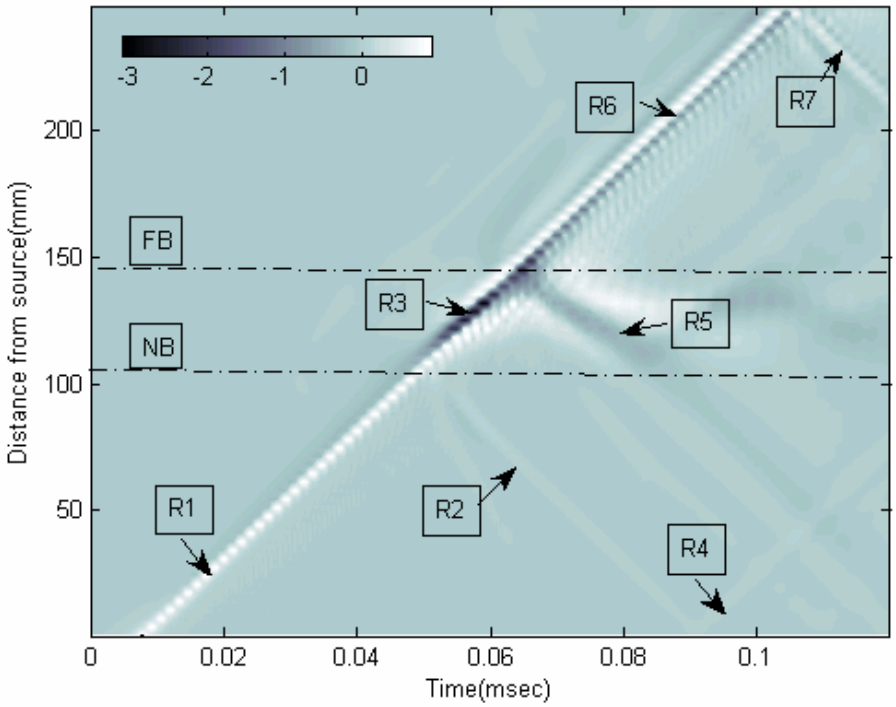


Figure 4.20: Contour plot of the normalized vertical responses along the surface of the model in the presence of a small void (Model_2)

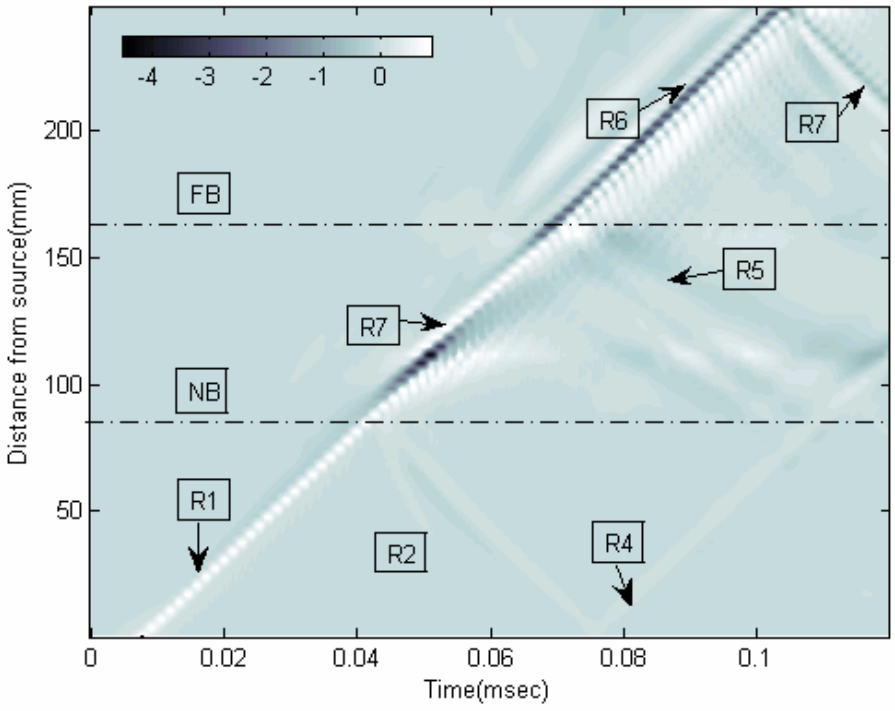


Figure 4.21: Contour plot of the normalized vertical responses along the surface of the model in the presence of a medium void (Model_3)

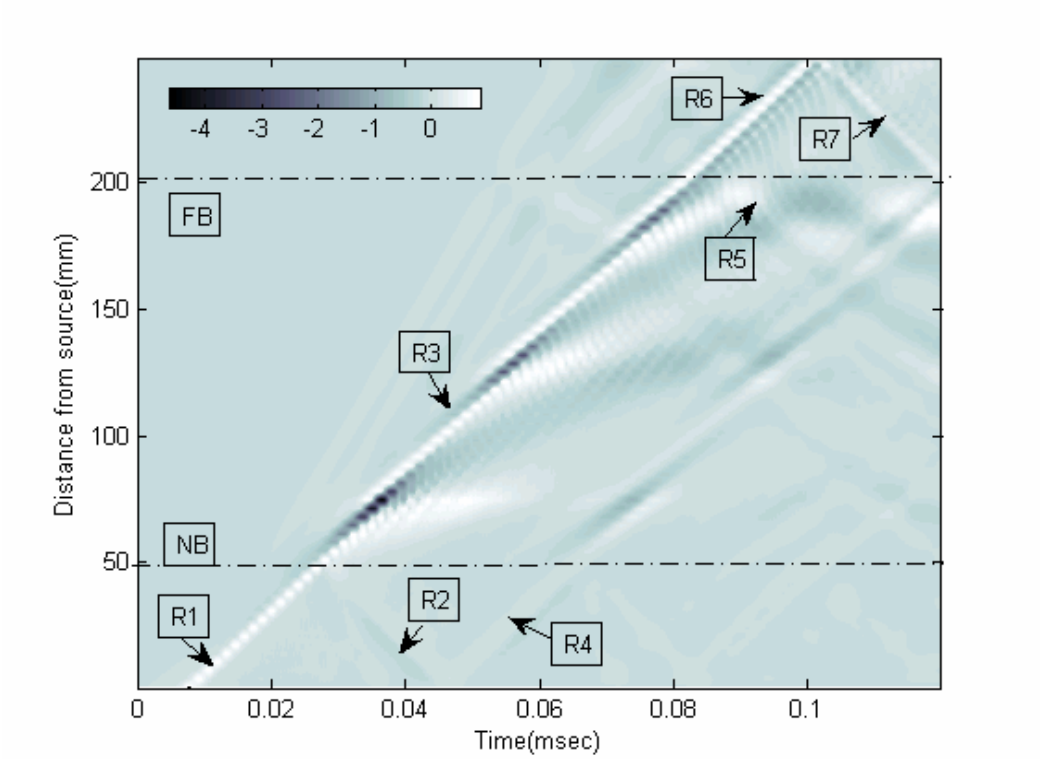


Figure 4.22: Contour plot of the normalized vertical responses along the surface of the model in the presence of a large void (Model_4)

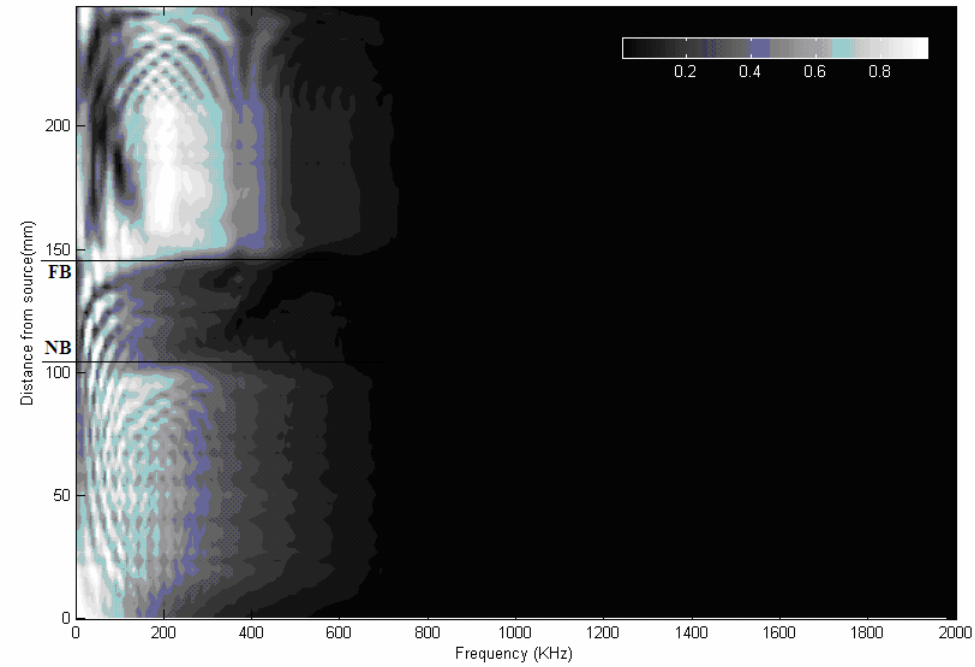


Figure 4.23: Contour plot of the frequency spectra of normalized vertical responses along the surface of the model in the presence of a small void (Model_2)

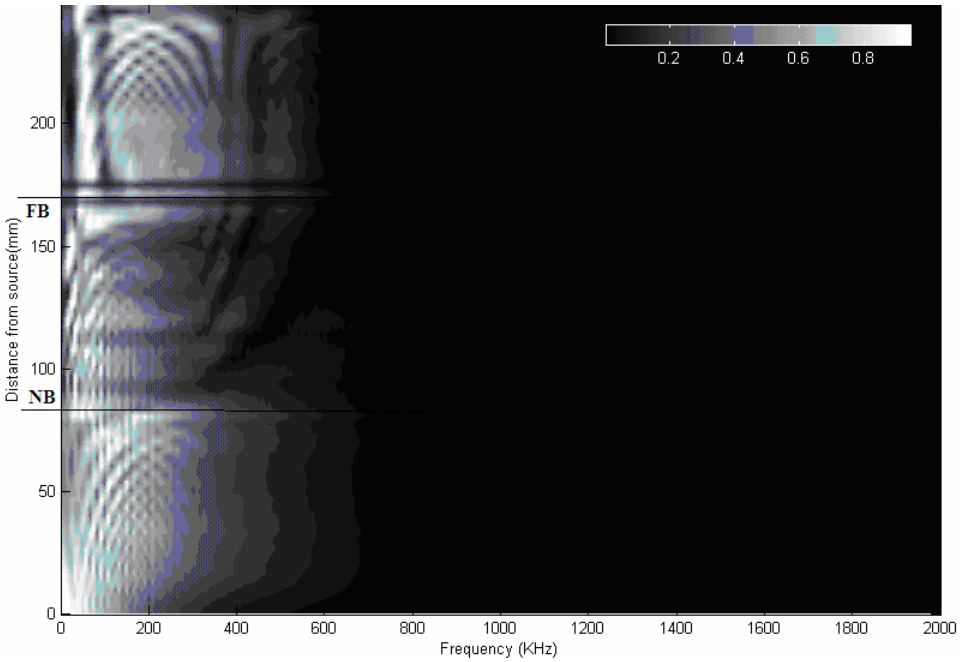


Figure 4.24: Contour plot of the frequency spectra of normalized vertical responses along the surface of the model in the presence of a medium void (Model_3)

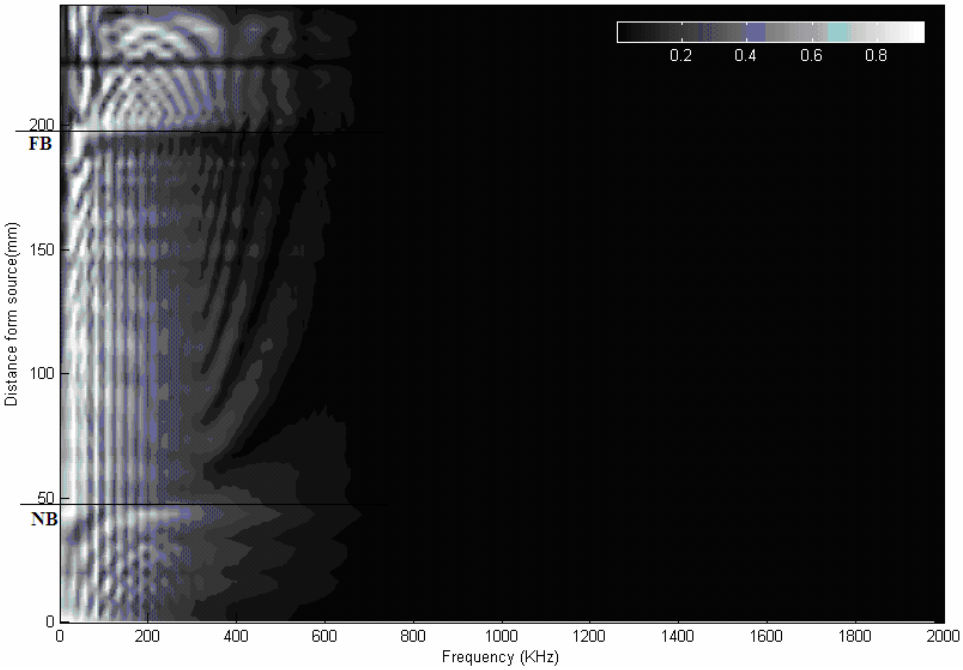


Figure 4.25: Contour plot of the frequency spectra of normalized vertical responses along the surface of the model in the presence of a large void (Model_4)

4.9 Summary

This chapter presented the set-up, calibration and verification of an explicit scheme finite element numerical model developed with LS-DYNA to simulate the propagation of Rayleigh waves in a semi-infinite medium. The set-up of the model includes procedures for choosing the model size, material properties, temporal and spatial discretization parameters, damping parameters and boundary conditions. Calibration of the numerical model was done using Lamb's solution for vertical displacements to assure the quality of the obtained data. Validation of the numerical model was performed by comparing the horizontal responses of numerical model to the theoretical horizontal responses of Lamb's solution. The final result of this calibration is a finite element model that accurately simulates the propagation of surface wave in a homogeneous semi-infinite medium.

The concept of characteristic wavelength was computed to normalize the numerical results. A total number of 600 samples were generated for each of the four considered cases resulting in a total of 2400 samples altogether. The effect on the surface responses due to the presence of voids of different sizes has also been presented. The interaction of the surface responses with the voids depends on the dimensions of the voids. Therefore the surface responses are capable of reflecting the differences between the void classes.

Chapter 5

Data Analysis by Classification Techniques

5.1 Introduction

In this chapter, results of applying various classification tools for the purpose of detection and characterization of defects are presented. The classification system for defect diagnosis is composed of three modules: a data generator, feature extractor, and a classifier. The sample data set consists of surface responses obtained from numerical models developed using the finite element method presented in Chapter 4. The purpose of the feature extractor is to generate a good set of parameters that are capable of reflecting the true nature of the underlying dataset consisting of a large number of data values categorized as signals. The classifier helps to make a decision on the class that any sample may belong to. A set of 37 features was extracted from all the samples in various domains by applying suitable transformations. Both supervised and unsupervised classifiers were employed in this study. The classification results obtained from the various classifiers are reported in terms of the confusion matrix. The obtained classification results are quite encouraging showing the suitability of the proposed techniques for the development of a decision support system for automatic characterization of defects in the field of non-destructive testing. Even though the results were obtained for a specific material (concrete), all the conclusions presented here are applicable for many other cases as soon as the response is normalized.

5.2 Feature Extractions and Preprocessing

5.2.1 Feature Extractions

A major step in the design of any signal classification system is the selection of a "good" set of features that are capable of separating the signals in the feature space. A classification algorithm will always give some kind of result, but a poor feature representation will lead to a result that does not reflect the true nature of the underlying data. The data should be simplified without loosing information. Usually, it is desirable to reduce the dimensionality of the input data to improve performance and increase the computational efficiency, and in many cases this can be achieved by applying suitable additional transformations. The choice of features needs to fulfill a basic criterion; the features should preserve all and only the important information that is contained in the data. This requirement has strong implications on what transforms should be used in the feature extraction process. Finding the best features is a very difficult task, and it often can only be accomplished through a trial-and-error process.

Feature extraction in the most general sense, consists of applying a mapping of the multidimensional space into a space of fewer dimensions. Feature extraction methods can be classified as linear or nonlinear methods. Linear methods attempt to find a globally flat subspace, while, the nonlinear methods attempt to find a locally flat subspace. Linear methods are simpler and better understood, while, nonlinear methods are more general and difficult to analyze. In the proposed work, only the linear mode of feature extraction was performed. Table 5.1 gives a brief outline concerning the domains from which the features were extracted.

Table 5.1: Domain of feature vectors

Numbers of Features	Feature Type
6	Time Traces
3	Derivatives from time traces
9	Fourier Transformation
6	Log Discrete Fourier Transforms
9	Wavelet Coefficients
4	Cepstrum Coefficients

The following sections discuss the features extracted in different domains. Programs for extracting features were developed using Matlab ®.

Time Traces

The following features were extracted from the responses in time domain

- Amplitude (+/-)

  Maximum and minimum values of time domain amplitude

- Mean value

  Probability distribution functions can be characterized by their moments. The well known moment mean is the first order moment

$$\bar{x} = Average = \frac{\sum x_i}{n} \tag{5.1}$$

  where, x_i represents the data and n is the number of data points

- Standard deviation or second order moment

  Standard deviation measures the 'spread' of the distribution

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}} \tag{5.2}$$

- Skewness or third order moment

  Skewness measures the degree of asymmetry of a distribution

$$Skewness = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^3}{(n-1)\sigma^3} \tag{5.3}$$

- Kurtosis or fourth order moment measures the degree of peakedness or flatness of a distribution compared with normal distribution

$$Kurtosis = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^4}{(n-1)\sigma^4} - 3 \tag{5.4}$$

Derivatives

- First derivative of displacement responses with respect to time gives velocity
- Second derivative of displacement with respect to time gives acceleration
- Third derivative of displacement with respect to time gives jerk

Fourier Transformation

The following features were extracted from the frequency spectra of the responses by applying the fast Fourier transformation.

- Maximum spectral amplitude (A_{max})
- Frequency corresponding to A_{max}
- Area under the spectrum
- First order moment with respect to area
- Second order moment with respect to area
- Mean value
- Standard deviation

- Skewness
- Kurtosis

A Matlab® coding of (Tallavo, 2006) was adopted for performing Fast Fourier Transformation.

Log Discrete Fourier Transformation (log DFT)

Log DFT is obtained by taking the logarithm value of the Fourier amplitude. The following features were extracted by applying log DFT.

- Maximum value
- Minimum value
- Mean value
- Standard deviation
- Skewness
- Kurtosis

Wavelet Coefficients

The wavelet transformation generates the approximate coefficient vector, CA and detail coefficient vector, CD obtained by a single level wavelet decomposition of the response vector using the Daubechies window of order 2.

Features extracted from both the coefficient vectors (CA and CD) are given below

Approximate Coefficients

- Maximum value
- Minimum value
- Mean value
- Standard deviation

- Skewness
- Kurtosis

Detail Coefficients

- Mean value
- Standard deviation

Cepstrum Analysis

The following features were extracted from the complex cepstrum of the responses

- Mean value
- Standard deviation
- Skewness
- Kurtosis

A total of 37 features were extracted from all the samples corresponding to each of the considered classes. The size of feature space is 2400×37, where, the number of rows corresponds to the total number of samples for all the classes (i.e., six hundred samples per class) and the number of columns responds to the number of features (37 features per sample).

5.2.2 Feature Preprocessing

Scaling and Normalization

Scaling of the extracted features is essential especially, when, the data spans different ranges. After scaling, normalization of the data set is carried out so that all the inputs and the target outputs have the same means of zero and standard deviation of 1. The purpose of normalization is to ensure that all features receive about equal weight and are not dominated by the larger-valued measurements.

Principal Component Analysis (PCA)

One of the main problems concerning neural networks applications in the field of classification/pattern recognition is the choice of a successful range of features providing the independent or discriminate information needed by the network for the correct classification. Often, a large set of potentially useful features is collected, and by feature reduction, the most suitable features are selected. In fact, not all the extracted features are correlated to the identification of the pattern; these features introduce uncertainty, thereby, reducing the capacity of recognition.

In this study, principal component analysis was employed for feature reduction for overcoming the problem of feature redundancy. PCA is a technique for simplifying the input dataset (in this case the extracted features) so that the data set is uncorrelated. This analysis based on linear transformation transforms, the input data thereby, minimizing the correlation between the input data (Jolliffe, 2002). PCA transforms the data to a new coordinate system such that, the greatest variance by the projection of the data lies on the first coordinate called the first principle coordinate and each successive orthogonal components account for as much as remaining variability in the dataset. Thus, it tries to encapsulate the variance in the dataset in terms of principal components. In addition, PCA reduces the dimensionality of the dataset, thereby, retaining only those components which contribute more than a specified fraction of the total variation in the dataset. PCA has the merit of being the optimal linear transformation for keeping the largest variance in the dataset without much loss of the information in the dataset. Other advantage of PCA is that it does not have a fixed set of basic vectors as their basic vector depends on the dataset unlike the other linear transformation techniques.

5.3 Assessment of classification results: Confusion Matrix

As the problem considered here concerns a classification problem, a good assessment of the success of the network can be obtained by looking at the confusion matrix. The confusion matrix (Kohavi and Provost, 1998) contains information about actual and predicted classifications done by a classification system. A representation of the confusion matrix is shown in Table 5.2.

Table 5.2: Confusion matrix

		Predicted	**Class**
		Class 1	Class 2
Actual	Class 1	a	b
Class	Class 2	c	d

Each column of the matrix represents the instances in a predicted class, while, each row represents the instances in an actual class. Classification accuracy can be reported by the following standard terms:

Overall accuracy is based on the proportion of the total number of predictions that were correct and is determined by the expression

$$AC = \frac{a+d}{a+b+c+d} \tag{5.5}$$

True rate for class 1 is based on the proportional of number of correct predictions for class 1 and is given by

$$TR_1 = \frac{a}{a+b} \tag{5.6}$$

False rate for class 1 is based on the proportional of number of class 2 cases incorrectly classified as class 1 and is given by

$$FR_1 = \frac{c}{c+d} \tag{5.7}$$

True rate for class2 is based on the proportional of number of correct predictions for class2 and is given by

$$TR_2 = \frac{d}{c+d} \tag{5.8}$$

False rate for class 2 is based on the proportional of number of class 1 cases incorrectly classified as class 2 and is given by

$$FR_2 = \frac{b}{a+b} \tag{5.9}$$

5.4 Classification/Clustering techniques for detection and classification of voids

The physical phenomena arising due to the existence of voids of different sizes were presented in Chapter 4. It was observed that the surface responses were dependent on the dimensions of the voids. Therefore the surface responses were capable of reflecting the differences between the void classes. These differences were useful in the detection and classification process of the voids. Features were extracted from the surface responses (vertical and horizontal displacements) of the medium through the feature extraction procedure described in section 5.2.1. The feature space consisting of a total of 37 features extracted from 2400 samples was subjected to feature preprocessing by applying scaling and normalization.

Once the features were extracted, scaled and normalized, three types of classifications were performed for the detection and characterization of the defect classes. The accuracy of the classifier was defined by the percentage of samples from each class correctly identified. Table 5.3 shows the three types of classification

Table 5.3: Types of classifications

Classification 1	Classification between perfect medium without any void and imperfect medium with any void
Classification 2	Classification between perfect medium and medium with a small void considered as one class and medium with a medium sized void and a large void considered as the second class
Classification 3	Classification between the different sizes of the void as well as the no void case

The objective of all the three classification types is to primarily determine the stability of the medium by checking if the medium is prefect or imperfect with voids inside it. The next step is to fine tune the classification process, wherein, the perfect medium and medium with a small void are considered as belonging to one class and the medium with medium sized and large sized void are considered to be the second class .The third step is to determine the extent of damage due to the presence of voids in the medium which is done by determining the size of the void present in the medium. The third type of classification is more complicated than the other two types of classifications since not only the presence of a void is detected but also the size of the void is determined.

In general, the term classification deals with assigning samples to different classes based on the fact that similar samples would be classified into the same class and dissimilar objects into different classes. This classification can be achieved in a supervised and unsupervised manner. If the classes are predefined and this information is used during the training process, the process is called supervised classification or simply classification. On the other hand, if the classes are not predefined, the process is known as clustering, wherein, the classes are themselves called clusters.

In this study, both supervised classification and unsupervised/clustering tools were employed for the defect characterization. In supervised learning scheme, the system is provided with training examples and is trained to recognize them. Two main phases can be recognized in any diagnostic task using supervised learning: the training phase and the defect characterization phase. The training phase consists of presenting inputs from training set along with corresponding desired target outputs (here, classes) and is used to train the network until it can associate the input vectors with specific target outputs. The testing phase consists of presenting new inputs from the test set to the trained model, to see how well the network has learnt and how well the network has performed. The training set is composed of 75% of the total data set (1800 samples) and the testing set consisting of the remaining 25% (600 samples). The samples of the training and testing set are randomly selected from the data set. Table 5.4 gives the desired target outputs for the three types of classification

Table 5.4: Target outputs for the three types of classification

Classification Type 1:

Class	**Target Output**
Perfect Medium (No void)	1
Imperfect Medium (Presence of Void)	2

Classification Type 2:

Class	**Target Output**
No void and Small void	1
Medium and Large void	2

Classification Type 3:

Class	Target Output
No Void	1
Small Void	2
Medium Void	3
Large Void	4

Since the primary objective of this study is defect characterization which in turn requires partition of the space that contains vector representations (features) of the classes into regions. Each region ideally must contain samples from a single and unique class only. This characterization can be achieved by either artificial neural networks or conventional statistical methods. Based on nature of the data set, the type of classification method can be chosen. Artificial neural networks are employed when the class regions cannot be separated by a hyperplane and the vice versa for statistical methods. In order to investigate the nature of the feature set, both the methods were utilized.

Unsupervised learning scheme (Clustering) consists of allowing a system to cluster samples together based on similarities it perceives in the feature space. By contrast with supervised learning, there are no explicit target outputs in unsupervised learning. The entire data set is presented to the system; the system discovers collective properties and organizes the data into clusters or classes.

The results obtained from the classification/clustering techniques used in this study are described in the following sections. A detail description of these techniques is presented in Chapter 3 on classification and clustering. All the techniques used in this study were developed using Matlab ®.

Sensitivity Analysis

One of the main limitations of soft computing methods occurs when the number of samples in the training set is limited (as in the present work) and overfitting of the network occurs.

Francesca Cau et al. (2006) established the fact that overfitting tends to fit samples in a more complex fashion than required. Overfitting occurs when the network has many degrees of freedom and this can be overcome by limiting the size of the network by reducing the number of hidden layers and the number of neurons in the hidden layer. However, in most of the applications reported in literature, the size and the number of neurons in the hidden layers have to be heuristically determined (Oscar et al., 2006) which in turn calls for a sensitivity analysis to be performed in order to choose the most adequate network. The sensitivity analysis consists of training the network with heuristically selected number of hidden layers and the number of neurons in the hidden layers. The performance of the network is assessed in terms of the overall prediction accuracy obtained through the confusion matrix. If the performance is satisfactory, the procedure ends, otherwise, a new network having more hidden neurons or more number of hidden layers is trained and the process of training the network goes on until the network reaches the desired performance. Overfitting can also occur from overtraining the network, wherein, the network has not only learnt the basic mapping associated with input and output data, but also the noise present in the training set. As a result, the network memorizes only the training set and therefore, the network performs well on the training set. On the other hand, the network loses its ability to generalize to new data (testing set). This can be avoided by evaluating the mean square error between the

predicted and actual results during the testing phase which gives an indication regarding the overtraining. As the error begins to increase, the training process is terminated.

5.5 Supervised Classification

5.5.1 Soft Computing Methods

5.5.1.1 General

Two types of soft computing methods were employed for supervised classification: Backpropagation Neural Network (BPNN) and Adaptive Neuro-Fuzzy Inference System (ANFIS). Sensitivity analysis for both the methods was performed for all the three types of classifications. Detailed description of these methods is provided in Chapter 3.

5.5.1.2 Backpropagation Neural Network (BPNN)

BPNN comprises of an input layer, output layer and a number of hidden layers. The first layer has weights from the input. Propagation of data takes place from input layer to the output layer. There is no connectivity between neurons in a layer. This type of neural network is trained using a process of supervised learning in which the network is presented with a series of matched input and output patterns and the connection strengths or weights of the connections automatically adjusted to decrease the difference between the actual and desired outputs. Detailed description of the steps involved in BPNN is presented in Chapter 3. An example of the training of BPNN is shown in Figure 5.1

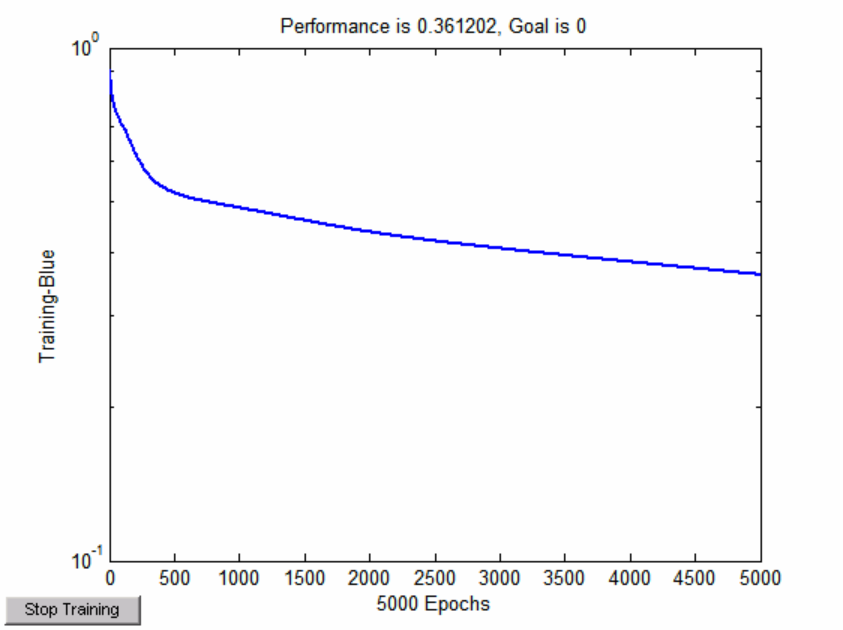


Figure 5.1: Training of BPNN

As Figure 5.1 shows, the performance is the RMSE calculated at 5000 epoch and goal is the minimum threshold value. After the training phase, the testing data set is presented to the trained model, to see how well the network has learnt and how well the network has performed.

Sensitivity Analysis

Sensitivity analysis was performed for all the three types of classifications in order to select the optimal parameters that give the maximum overall classification accuracy obtained through the confusion matrix. The parameters include number of epochs, number of hidden layers, and number of neurons in the hidden layers. The input and output layers consists of one neuron. All the results reported were obtained by testing the trained network with the testing data.

❖ ***Classification 1 (Void and no void)***

One Hidden Layer

5 neurons

Number of Epochs	Classification Accuracy (%)
1000	79
2500	80
5000	83
10000	84

10 neurons

Number of Epochs	Classification Accuracy (%)
1000	80
2500	81
5000	84
10000	88

20 neurons

Number of Epochs	Classification Accuracy (%)
1000	80
2500	82
5000	84
10000	85

Two Hidden Layers

10 and 5 Neurons in two hidden layers, respectively

Number of Epochs	Classification Accuracy (%)
1000	81
2500	84
5000	84
10000	82

10 and 20 Neurons in two hidden layers, respectively

Number of Epochs	Classification Accuracy (%)
1000	78
2500	83
5000	87
10000	90
50000	**92**

❖ ***Classification 2 (No & small void and medium void & large void)***

One Hidden Layer

5 neurons in the hidden layer

Number of Epochs	Classification Accuracy (%)
1000	69
2500	70
5000	72
10000	68

10 neurons in the hidden layer

Number of Epochs	Classification Accuracy (%)
1000	67
2500	67
5000	66
10000	68

20 neurons in the hidden layer

Number of Epochs	Classification Accuracy (%)
1000	68
2500	68
5000	71
10000	73

Two Hidden Layers

10 and 5 neurons in two hidden layers, respectively

Number of Epochs	Classification Accuracy (%)
1000	66
2500	70
5000	72
10000	66

10 and 20 neurons in two hidden layers, respectively

Number of Epochs	Classification Accuracy (%)
1000	64
2500	66
5000	70
10000	**78**
50000	74

❖ ***Classification 3 (No, small, medium and large void)***

One Hidden Layer

5 neurons in the hidden layer

Number of Epochs	Classification Accuracy (%)
1000	37
2500	38
5000	40
10000	44

10 neurons in the hidden layer

Number of Epochs	Classification Accuracy (%)
1000	36
2500	39
5000	43
10000	47

20 neurons in the hidden layer

Number of Epochs	Classification Accuracy (%)
1000	37
2500	38
5000	43
10000	47

Two Hidden Layers

10 and 5 neurons in two hidden layers, respectively

Number of Epochs	Classification Accuracy (%)
1000	33
2500	40
5000	43
10000	45

10 and 20 neurons in two hidden layers, respectively

Number of Epochs	Classification Accuracy (%)
1000	34
2500	43
5000	45
10000	44

Three Hidden Layers

20, 10 and 5 Neurons in three hidden layers, respectively

Number of Epochs	Classification Accuracy (%)
1000	35
2500	42
5000	51
10000	**56**

Optimal Selection

The number of hidden layers was decided by experimenting with the number of layers ranging from 1 to 3. The number of neurons for the hidden layers was selected by experimenting with different number of neurons ranging from 5 to 20. The number of epochs was varied from 1000 to 50000. Final setting of the network in classification 1 trained for 50000 epochs was two hidden layers with 10 and 20 neurons. The confusion matrix with this optimal selection of parameters for classification1 is shown in Table 5.5

Table 5.5: Confusion matrix for optimal selection of parameters for classification 1

Class	**No Void**	**Void**
No void	**120**	26 S
Void	24 U	**430**

The overall classification accuracy is 92% wherein, the accuracy in predicting the absence of a void is 82% and whereas 18% of the samples are incorrectly predicted as presence of voids. On the other hand, the accuracy in predicting the presence of a void is 95% whereas 5% of the samples are incorrectly predicted as absence of void. Out of the false predictions, 18% of the samples incorrectly predicted as the presence of void is on the safe side (indicated by "S"), from the structural point of view, whereas, 5% of samples incorrectly predicted as sound samples are unsafe (indicated by "U") while, they actually have voids in them. The classification of the incorrectly predicted samples into

safe and unsafe classification is essential. The safe misclassification probably results in a high cost but it is structural safe. The safe and unsafe misclassifications are indicated by "S" and "U" in all the classifications performed by all the classifiers.

The final setting of the network in classification 2 was two hidden layers with 10 and 20 neurons respectively trained for 10000 epochs. The confusion matrix with this optimal selection of parameters for classification 2 is shown in Table 5.6

Table 5.6: Confusion matrix for optimal selection of parameters for classification 2

Class	No void & Small void	Med & Large void
No void & Small void	**231**	63 S
Med & Large void	71 U	**235**

The overall classification accuracy is 78%. The accuracy in correctly predicting the class 1 (no void and small void) is 79% whereas 21% of the samples are incorrectly predicted as belonging to class 2 (medium and large void). The safe "S" and unsafe "U" false predictions are indicated in the confusion matrix.

The final setting of the network in classification 3 was three hidden layers with 5, 10 and 20 neurons respectively trained for 10000 epochs. The confusion matrix with this optimal selection of parameters for classification 3 is shown in Table 5.7

Table 5.7: Confusion matrix for optimal selection of parameters for classification 3

Class	No Void	Small Void	Medium Void	Large Void
No void	**94**	34 S	17 S	1 S
Small Void	15 U	**59**	66 S	8 S
Medium Void	7 U	42 U	**104**	23 S
Large Void	1 U	3 U	46 U	**80**

The overall classification accuracy is 56%. The percentage of correctly classified samples is 64% (no void class), 40% (small void), 59% (medium void) and 62% (large void). The misclassification between the different classes and the safe "S" and unsafe "U" false predictions are indicated in the confusion matrix.

5.5.1.3 Adaptive Neuro-Fuzzy Inference System (ANFIS)

ANFIS combines the desirable qualities of both fuzzy logic and neural networks. Detailed description of step by step procedure of ANFIS is given in chapter 3. One of the drawbacks of ANFIS lies in the dimensionality of the feature data set not being able to exceed a value of 3. For this purpose, principal component analysis was employed to reduce the dimensionality of the feature set from 37 to 3 without much loss of the information in the data set.

Sensitivity Analysis

Sensitivity analysis was performed for all the three types of classifications inorder to select the optimal parameters which give the maximum overall classification accuracy obtained through the confusion matrix. The parameters include number of epochs, number of membership functions per input. All the results reported were obtained by testing the trained network with the testing data.

- ***Classification1 (Void and no void)***

Two Membership Functions

Epochs	Accuracy
5	47
10	47
20	47
50	**47**

Three Membership Functions

Epochs	Accuracy
5	43
10	44
20	44
50	43

- ***Classification 2 (No & small void and medium & large void)***

Two Membership Functions

Epochs	Accuracy
5	44
10	44
20	44
50	**44**

Three Membership Functions

Epochs	Accuracy
5	43
10	43
20	43
50	43

- ***Classification 3 (No, small, medium and large void)***

Two Membership Functions

Epochs	Accuracy
5	26
10	26
20	26
50	**26**

Three Membership Functions

Epochs	Accuracy
5	22
10	22
20	22
50	22

Optimal Selection

The number of membership functions was varied from 2 to 3. The number of epochs was varied from 5 to 50. The misclassification between the different classes, the safe and unsafe false predictions is indicated in the confusion matrix. Final setting of the parameters for all the classifications was: two membership functions per input trained for 50 epochs. The results obtained by the model with the testing data for the different classifications presented in the form of confusion matrices are shown in the following tables.

Table 5.8: Confusion matrix for optimal selection of parameters for classification 1

Class	**No Void**	**Void**
No void	**70**	76 S
Void	247 U	**207**

The overall classification accuracy in correctly predicting the classes is 48%.

Table 5.9: Confusion matrix for optimal selection of parameters for classification 2

Class	**No void & Small void**	**Med & Large void**
No void & Small void	**165**	129 S
Med & Large void	207 U	**99**

The overall classification accuracy in correctly predicting the classes is 44%

Table 5.10: Confusion matrix for optimal selection of parameters for classification 3

Class	**No Void**	**Small Void**	**Medium Void**	**Large Void**
No void	**37**	34 S	38 S	37S
Small Void	50 U	**42**	31 S	25 S
Medium Void	68 U	46 U	**32**	30 S
Large Void	55 U	41 U	18 U	**16**

The overall classification accuracy in correctly predicting the classes is 26%

5.5.2 Supervised Classification: Statistical Techniques

5.5.2.1 General

Two types of statistical methods were employed for supervised classification: k-nearest neighbor classification (k-NN) and classification based on linear discriminate analysis (LDA). Detailed description of these methods is given in chapter 3.

5.5.2.2 k-Nearest Neighbor Classifier (k-NN)

The k-nearest neighbor classifier is based on finding the number of k-nearest neighboring samples around an unknown sample and the class of the sample is determined by the class that has the highest percentage of neighbors. The results obtained from the model with the testing data for the different classifications presented in the form of confusion matrices are shown in the following tables. The misclassification between the different classes and the safe "S" and unsafe "U" false predictions are indicated in the confusion matrix.

- ***Classification 1 (Void and no void)***

Table 5.11: Confusion matrix for classification 1

Class	No Void	Void
No void	**127**	19 S
Void	17 U	**437**

A very high classification accuracy of 94% was obtained in detecting the presence of voids in the medium.

❖ ***Classification 2 (No & small void and medium & large void)***

Table 5.12: Confusion matrix for classification 2

Class	No void & Small void	Med & Large void
No void & Small void	**258**	36 S
Med & Large void	35 U	**271**

An overall classification accuracy of 88% was obtained in classifying between the two classes (no void and small void) and (medium and large void).

❖ ***Classification 3 (No, small, medium and large void)***

Table 5.13: Confusion matrix for classification 3

Class	No Void	Small Void	Medium Void	Large Void
No void	**127**	6 S	13 S	0 S
Small Void	8 U	**117**	13 S	10 S
Medium Void	8 U	20 U	**127**	21 S
Large Void	1 U	6 U	10 U	**113**

An overall classification accuracy of 81% was obtained in determining the size of the void in the medium.

5.5.2.3 Linear Discriminate Classifier

In LDA, the original data is transformed into a new feature space in which class separability can be carried out more effectively. LDA maximizes the ratio of between-class variance to within-class variance. The main purpose is to maximize this ratio so that adequate class separability is achieved. After transformation, Euclidean distance is used to classify data points. In the testing phase, test vectors are transformed and the Euclidean distance of the test vectors from the class means is calculated. The test vector is classified as belonging to the class that has the shortest distance. The results obtained from the

model with the testing data for the different classifications presented in the form of confusion matrices are shown in the following tables. The misclassification between the different classes and the safe "S" and unsafe "U" false predictions are indicated in the confusion matrix.

- ***Classification 1 (Void and no void)***

Table 5.14: Confusion matrix for classification 1

Class	No Void	Void
No void	**115**	31 S
Void	74 U	**380**

An overall classification accuracy of 83% was obtained in detecting the presence of voids in the medium.

- ***Classification 2 (No & small void and medium & large void)***

Table 5.15: Confusion matrix for classification 2

Class	No void & Small void	Med & Large void
No void & Small void	**219**	75 S
Med & Large void	78 U	**228**

An overall accuracy of 75 % was obtained in predicting the considered classes

- ***Classification 3 (No, small, medium and large void)***

Table 5.16: Confusion matrix for classification 3

Class	No Void	Small Void	Medium Void	Large Void
No void	**97**	37 S	5 S	7 S
Small Void	25 U	**73**	24 S	26 S
Medium Void	16 U	53 U	**64**	43 S
Large Void	4 U	23 U	25 U	**78**

An overall classification accuracy of 52% was obtained in determining the size of the void in the medium.

5.6 Unsupervised Classification or Clustering

5.6.1 Fuzzy C-means Clustering

Fuzzy c-means clustering is one of the most commonly used clustering techniques. In real world applications, there is often no sharp or crisp boundary between classes that makes fuzzy clustering appropriate for such classifications. Fuzzy C-means clustering is accomplished by grouping of patterns based on similarities between the individual patterns. The results obtained for the different classifications presented in the form of confusion matrices are shown in the following tables. The misclassification between the different classes and the safe "S" and unsafe "U" false predictions are indicated in the confusion matrix.

- ***Classification 1 (Void and no void)***

Table 5.17: Confusion matrix for classification 1

Class	**No Void**	**Void**
No void	**202**	398 S
Void	612 U	**1188**

An overall classification accuracy of 58% was obtained in detecting the presence of voids in the medium.

- ***Classification 2 (No & small void and medium & large void)***

Table 5.18: Confusion matrix for classification 2

Class	**No void & Small void**	**Med & Large void**
No void & Small void	**437**	763 S
Med & Large void	377 U	**823**

An overall accuracy of 53% was obtained in predicting the considered classes

- ***Classification 3 (No, small, medium and large void)***

Table 5.19: Confusion matrix for classification 3

Class	No Void	Small Void	Medium Void	Large Void
No void	**175**	175 S	177 S	73 S
Small Void	184 U	**173**	154 S	89 S
Medium Void	225 U	138 U	**147**	90 S
Large Void	296 U	89 U	125 U	**90**

An overall classification accuracy of 24% was obtained in determining the size of the void in the medium.

5.7 Comparison of results from various classification techniques

The assessment of the classifiers' performance used for detection and characterization of defects has proved to be valuable in comparing the various techniques. The performance of all the classifiers were reported in terms of their overall classification accuracy obtained through the confusion matrix (Table 5.20)

Table 5.20: Overall classification accuracy given by different classifiers

- Classification between presence and absence of voids (Classification1)

Supervised : Soft Computing	Accuracy (%)
Feedforward Backpropagation Neural Network	92
Adaptive Network based Fuzzy inference system	47

Supervised: Statistical	Accuracy (%)
Classify: Linear Discriminate Analysis	83
k-Nearest Neighbor	***94***

Unsupervised	Accuracy (%)
Fuzzy C-means Clustering	53

❖ Classification between (no void, small void) and (medium, large void) (Classification2)

Supervised : Soft Computing	**Accuracy (%)**
Feedforward Backpropagation Neural Network	78
Adaptive Network based Fuzzy inference system	44

Supervised: Statistical	**Accuracy (%)**
Classify: Linear Discriminate Analysis	75
k-Nearest Neighbor	***88***

Unsupervised	**Accuracy (%)**
Fuzzy C-means Clustering	53

❖ Classification between different sizes of the void (Classification3)

Supervised : Soft Computing	**Accuracy (%)**
Feedforward Backpropagation Neural Network	56
Adaptive Network based Fuzzy inference system	26

Supervised: Statistical	**Accuracy (%)**
Classify: Linear Discriminate Analysis	52
k-Nearest Neighbor	***81***

Unsupervised	**Accuracy (%)**
Fuzzy C-means Clustering	24

Therefore the performance of k-nearest neighbor for all the three classifications has proved to be superior to the other classifiers. An overall accuracy of 94% was obtained in detecting the presence of a void in a medium and an accuracy of 81% was obtained in determining the size of the void in the medium. In all the classifications, the incorrectly predicted samples are classified into safe and unsafe misclassifications (indicated by “S” and ”U” in the confusion matrices) because though the safe misclassification probably might result in a high operational cost but it is safe from structural point of view.

The performance of feedforward backpropagation neural networks (BPNN) and linear discriminate classifier were found to be satisfactory. However the classification accuracies reported by adaptive neuro-fuzzy inference system and fuzzy c-mean clustering was very low, a possible result because of severe reduction in dimensionality from 37 to 3 in case of ANFIS and the closely clustered nature of the data set in case of FCM. Thus the assessment of the various classifiers' performance in this application is useful in comparing the various techniques and establishes the advantages of using simplified classification techniques like k-NN for defect characterization.

Further to test the severity of the best classification technique namely, the k-NN classifier, the classification was performed by training the classifier with 50% of the dataset (1200 samples) and testing with the rest 50% as the testing data (1200 samples). The results obtained for the different classifications presented in the form of confusion matrices are shown in the following tables.

- ***Classification 1***

Table 5.21: Confusion matrix for classification 1

Class	**No Void**	**Void**
No void	**244**	58 S
Void	47 U	**851**

An overall classification accuracy of 91% was obtained in detecting the presence of voids in the medium.

❖ ***Classification 2***

Table 5.22: Confusion matrix for classification 2

Class	No void & Small void	Med & Large void
No void & Small void	**483**	88 S
Med & Large void	109 U	**520**

An overall accuracy of 84 % was obtained in predicting the considered classes

❖ ***Classification 3***

Table 5.23: Confusion matrix for classification 3

Class	No Void	Small Void	Medium Void	Large Void
No void	**244**	23 S	29 S	6 S
Small Void	21 U	**195**	32 S	21 S
Medium Void	20 U	62 U	**219**	49 S
Large Void	6 U	21 U	29 U	**223**

An overall classification accuracy of 73% was obtained in determining the size of the void in the medium. Comparison between the results obtained from 75% & 25% of the dataset used as training and testing and 50% & 50% is shown in Table 5.24

Table 5.24: Comparison of the classification accuracies between 75% & 25% and 50% & 50% of the dataset used as training and testing

Type	75 % training & 25% testing Classification accuracy (%)	50% training & 50% testing Classification accuracy (%)
Classification 1	94	91
Classification 2	88	84
Classification 3	81	73

It can be seen that the results were encouraging as there is not much difference in the classification accuracies between the two methods.

5.8 Critical Features

Based on the classification results by different techniques, it can be concluded that k-Nearest neighbor classifier gives the highest classification accuracy for all the classification types. Selection of critical features is essential in order to identify the critical feature set which gives the maximum classification accuracy.

Table 5.20 gives the overall classification accuracy for different category of features vectors using k-nearest neighbor Classifier

Table 5.25: Classification accuracy for individual categories of feature vectors

Accuracy (%)	Time Traces	Derivatives	Fourier Transforms	Log DFT	Wavelet Coeff	Cepstrum Coeff
Classification1	92	69	93	83	90	81
Classification2	81	54	86	73	83	74
Classification3	71	32	76	64	69	59

From the above table, features extracted from time traces, Fourier transformation and Wavelet transformation gives better results when compared with the other categories. The classification accuracy obtained by combining features from these three domains is presented in Table 5.26.

Features extracted from the time traces and from the frequency spectra (by applying Fourier transformation) gives the maximum classification accuracy for all the three classification types; however there is only a marginal increase in the classification accuracy.

Table 5.26: Classification accuracy by combining features extracted from time traces, Fourier transformation and Wavelet transformation

Accuracy (%)	(Time, Fourier, Wavelet)	(Time and Fourier)	(Time and Wavelet)	(Fourier and Wavelet)
Classification 1	95	**95**	90	94
Classification 2	88	**90**	84	85
Classification 3	80	**82**	71	77

5.9 Summary

A classification and Clustering method for non-destructive testing for void characterization in half space media using ultrasonics has been implemented.

The sample database for training and validation has been obtained from finite element simulations. Significant features extracted from the dataset in different domain by applying suitable transformations were subject to feature preprocessing. Both supervised and unsupervised classification techniques were applied for void detection and characterization.

The performance of each classifier for detection and classification of voids is given in Table 5.20. The assessment of the various classifiers' performance in this application has proved to be valuable in comparing the various techniques. From the classification results, it was discovered that the performance of k-nearest neighbor classifier proved superior when compared with the other techniques. An overall accuracy of 94% was obtained in detecting the presence of a void in a medium and an accuracy of 81% was obtained in determining the size of the void in the medium. Further the features extracted from the time traces and from the frequency spectra (by applying Fourier transformation) were found to give the maximum classification accuracy. The severity of the best

classification method, namely the k-NN classifier was tested by training the classifier with 50% of the dataset and testing with the rest 50%. The results obtained were encouraging as there was not much difference in the classification accuracies when compared with 75% and 25% of the dataset used for training and testing. These results therefore help in assessing the applicability of simplified classification methods such as k-NN in defect characterization.

Chapter 6

Summary and Conclusions

A diagnostic system based on supervised and unsupervised learning paradigm was developed for the identification and classification of voids in a half-space medium. The first step involved the construction of numerical models using LS-DYNA, a commercial finite element code to simulate the propagation of Rayleigh waves in a homogeneous elastic half-space. These 2-D axi-symmetric finite element models simulate the MASW (Multiple Analysis of Surface Waves) test method that facilitates the investigation of the behaviour of Rayleigh wave in the presence and absence of voids in the medium. To establish the validity of finite element models used in this study, they were first calibrated with a known theoretical solution. This was done to ensure that the finite element results provided a certain degree of confidence in their accuracy, because exact analytical solutions were not available for complex problems like the one taken up under the present study. Once the model was calibrated and validated, a detailed parametric study can be carried out in less time. The investigation into the numerical modeling of surface waves examined the model element size, time step, model size, boundary conditions, damping parameters, source configuration and calibration to Lamb's theoretical solution by comparing the numerical responses with the theoretical responses.

Once the model was calibrated, voids of different sizes were introduced into the model to simulate anomalies. The most significant factor in ensuring a good accuracy of any classification process is the presence of enough variance in the sample data set. This was achieved by generating sufficient numbers of samples for each case by changing the type of input loading. Inorder to ensure sufficient variance in the dataset, the sources were

chosen such that there is sufficient variation in the frequency content of the pulses generated by these sources since the responses obtained from the medium is dependent on the frequency content of the source. A set of 600 samples were generated for each of the four considered cases which gave a total of 2400 samples. The effect on the surface responses due to the presence of voids of different sizes revealed that the pattern of the surface responses obtained was dependent on the void dimensions. These differences were useful in the involved extraction of features from identification and classification of the voids.

The second step consists of selecting a good set of parameters from the responses that are capable of reflecting the true nature of the underlying dataset. This was achieved through the feature extraction process which in the most general sense is applying a mapping of the multidimensional space into a space of fewer dimensions by applying suitable transformations. A set of 37 features were extracted from all the samples in various domains.

The third step involved application of applying classifications methods based on supervised and unsupervised learning on the feature dataset which helps in making a decision on the class any sample belongs to. The objective of the three types of classifications that were performed was to firstly identify the presence of a void in a medium and then to determine the class or the size of the void in the medium. In classifiers based on supervised learning, this was achieved by training the system with training examples by which the system learns and is capable of predicting with the testing dataset. Supervised classifiers based on soft computing methods such as backpropagation

neural network, adaptive neuro-fuzzy inference system and statistical methods such as k-nearest neighbor and linear discriminate analysis were employed.

Unsupervised learning scheme consists of allowing a system to cluster samples together based on similarities it perceives in the feature space. By contrast with supervised learning, there is no training and testing in unsupervised learning. The entire data set is presented to the system; the system discovers collective properties and organizes the data into clusters or classes. Fuzzy c-mean clustering based on unsupervised learning was employed.

From the classification results, it was discovered that the performance of k-nearest neighbor classifier proved superior when compared with the other techniques. An overall accuracy of 94% was obtained in identification and an accuracy of 81% was obtained in determining the size of the void in the medium. The assessment of the various classifiers' performance has proved to be valuable in comparing the different techniques and establishing the applicability of simplified classification methods such as k-NN in defect characterization.

The obtained classification accuracy for the detection and classification of voids are very encouraging, showing the suitability of the proposed approach to the development of a decision support system for non-destructive testing of materials for defect characterization.

Recommendation for future work includes validating the obtained results with actual measures. The applicability of these results can be extended in more complex geometries (ex. more boundaries, different materials and defects with a wider range of sizes and shapes) provided enough number of samples is taken to train the classifiers. More work is

necessary in transferring the above work by applying these classification tools on experimental or field data. This will require additional signal processing techniques including more analyses into the wavelet transformation of the signals inorder to obtain significant features that lead to more robust classification processes.

References

Achenbach, (1973), Wave Porpagation in elastic solids., North Holland Publishing Co./ American Elsevier, Amstrdam/New york.

Al-Hunaidi, M. O.,(1993) "Insights on the sasw nondestructive testing method," Canadian journal of civil engineering, vol. 20, pp. 940—950.

Alleyne D, Cawley P., (1998) "A two-dimensional Fourier transform method for measurement of propagating multimode signals" J Acoust Soc Am; (in press).

Alterman and Burridge, R. Z., (1981) "The elastic radiation from an expanding spherical cavity," Geophysics. J.R. ast., vol. 30,pp.,1743-1761.

Amit Konar., (2000) Artificial Intelligence and Soft Computing: behavioral and cognitive modeling of the human brain. CRC Press, Inc.

Amitava Roy., Bharat, P., and Swapan Kumar De., (1995),"Materical classification through neural networks," Ultrasonics Vol 33 No 3.

ANSYS user's manual for revision 5.0 (1992) Houston, TX: Swanson Analysis Systems.

Asghar Bhatti, M., Fundamental Finite Element Analysis and Applications *with Mathematica and MATLAB* Computations,John Wiley &Sons, Inc.

Belesky, R.M., and Hardy, H.R., (1986)," Seismic and microseismic methods for cavity detedtion and stability monitoring of near surface voids," Proceedings of the 27th U.S Symposium on Rock Mechanice, Rock Mechanics: Key to Energy Production, University of Alabama: 248-258.

Berthholf, L.D., (1967) "Journal of Applied Mechanics., vol. 30,pp. 725.

Boore, D.M., (1972) "Finite difference methods for seismic wave propagation in heterogeneous materials," Methods in computational Physics, vol. 11, pp. 1.

Brian M. Lempriere., (1968) "Ultrasound and elastic waves, frequently asked questions," Academic press, Elsevier science.

Bullen, K.E., (1963),"An introduction to the theory of seismology," Cambridge Univ. Press.

Burch, S. F., Bealing, N. K., (1986)"A physical approach to the automated ultrasonic characterization of buried weld defects in ferritic steel," NDT International. Vol 19. No 3.

Carino, N.J., (1984), "Laboratory Study of Flaw Detection in Concrete by the Pulse-echo Method, in In-Situ/Nondestructive Testing of Concrete," V.M. Malhotra, Editor, ACI SP-82, American Concrete Institute, pp 557-579.

Carino, N. J., Sansalone, M. and Hsu, N. N., (1986), "Flaw detection in concrete by frequency spectrum analysis of impact echo waveforms," International Advances in Nondestructive Testing, 12th Edition, New York, pp 1-20.

Carino, N. J., and Sansalone, M., (1988), "Impact-echo: a new method for inspecting construction materials," Proceedings of Nondestructive Testing and Evaluation of Materials for Construction, University of Illinois.

Carino, Nicholas, J., (1986)" Point source-point receiver pulse-echo technique for flaw detection in concrete", Journal of The American Concrete Institute, v 83, n 2, Mar-Apr, p 199-208.

Chan, T. Y., (1995)"Wavelet Basics," Kluwer Academic Publishers, Boston.

Chaoqiang and Stephen., (2006) "Evaluation of MASW techniques to image steeply dipping cavities in laterally inhomogeneous terrain," Journal of Applied Geophysics 59 106–116.

Chih-Ping Lin, Cheng-Chou Chang, Tzong-Sheng Chang., (2004)" The use of MASW method in the assessment of soil liquefaction potential," Soil Dynamics and Earthquake Engineering 24 689–698.

Chiou, C. P., Schmerr, L.W., (1991)" Quasi pulse-echo ultrasonic technique for flaw classification," Ultrasonics; 29:471-81.

Cohen, L., "Time-Frequency Distributions-A Review", Proc. IEEE, Vol. 77, No 7, pp. 941-981.

Curro, J. R., (1983)," Cavity detection and delineation research," Report2, Seismic Methodology, Medford Cave Site, Florida, U.S Army Waterway Experiment Station Technical Report GL-83-1, U.S Army Waterways Experiment Station, Vicksburg, Mississipi.

Damarla, T., Ghosal, S., and Karpur, P., (1991)" Application of neural networks for classification of ultrasonic signals," Intelligent Engineering Systems Through Artificial Neural Networks (Eds Dagli, C.H., Kumara, S.R.T. and Shin, Y.C.) ASME Press, New York 377-382.

Drai R., Khelil, M., Benchaala, A., (2000)" Elaboration of some signal processing algorithms in ultrasonic techniques: Application of material NDT," Ultrasonic 2000; 38(1-8): 503-7.

Drai Redouane, Mohamed Khelil, Amar Benchaala, (2002) "Time frequency and wavelet transform applied to selected problems in ultrasonic NDE," NDT&E International 35(2002) 567-572.

Dravinski, M., (1983),"Ground motion amplification due to elastic inclusions in a half-space", Earthquake Engineering and Structural Dynamics, 11:313-335.

Drossaert, F.H., (2001), "Determination of geotechnical parameters of a railroad embankment byseismic methods without possession management," Drs dissertation, University of Utrecht.

Dudgeon Dan, E., and Mersereau Russel, M., (1984) "Multidimensional digital signal processing," Prentice Hall Inc., Englewood Cliffs, NJ.

Flandrin, P., (1988)"Nondestructive evaluation in the time-frequency domain by means the Wigner-Ville Distribution", Signal Processing & Pattern Recognition in Nondestructive Evaluation of Materials, Springer-Verlag.

Francesca Cau., Alessandra Fanni., Augusto Monitisci., Pietro Testoni., Mariangela Usai., (2005)"Artificial Neural Networks for Non-Destructive Evaluation with Ultrasonic Waves in not Accessible Pipes," Conference Record - IAS Annual Meeting (IEEE Industry Applications Society), v 1, Conference Record of the 2005 IEEE Industry Applications Conference, 40th IAS Annual Meeting, 2005, p 685-692.

Francesca Cau., Alessandra Fanni., Augusto Monitisci., Pietro Testoni., Mariangela Usai., (2006) "A signal processing tools for non-destructive testing of inaccessible pipes," Engineering Applications of Artificial Intelligence 19 753-760.

Fukunaga, K., (1990),"Introduction to Statistical Pattern Recognition," Academic Press, San Diego, California.

Gang Tian., Don W. Steeples., Jianghai Xia., Richard D. Miller., Kyle T. Spikes., Matthew D. Ralston., (2003)" Multichannel analysis of surface wave method with the autojuggie," Soil Dynamics and Earthquake Engineering 23 243–247.

Graff, K. F., (1991), Wave motion in elastic solids, Dover, New York.

Hallquist, J.O., (1991-1998) LS-DYNA *Theoretical Manual*. Livermore Software Technology Corporation.

Hiltunen, D. R., and Woods, R. D., (1989)"Influence of source and receiver geometry on the testing of pavements by the surface waves method," in Nondestructive testing of pavements and backcalculation of moduli, A.J. BushIII and G.Y. Baladi, Eds., ASTM STP 1026, pp.138—153. American society for testing and materials, Philadelphia.

Hiltunen, D. R., and Woods, R. D., (1990) "Variables affecting the testing of pavements by the surface wave method," Transportation research record, , no. 1260, pp. 45—52.

Hirao, M., and Fukuoka, H., (1982), "Scattering of Rayleigh surface waves by edge cracks: numerical simulation and experiment," The Journal of the Acoustical Society of America. vol. 72:2 pp.602-606.

Huch Doyle., Seismology, John Wiley and Sons, New York, 1995.

Imran, I., Nazarian, S., and Piccornell, M., (1995), "Crack detection using time-domain wave propagation technique", Journal of Geotechnical Engineering, vol 121:2 pp.198-207.

Jang, J., (1993) ANFIS: Adaptive-Network- Based Fuzzy Inference System. IEEE Trans. On Systems, Man and Cybernetics. 23(3):665-685.

Jolliffe, I.T., (2002), Principle component analysis, Springer, Second edition.

Kamen Edward W., (2000) "Fundamentals of signals and systems using the web and matlab," Second edition, Prentice Hall, Upper Saddle River, New York.

Karray, F., and De Silva., (2004), "Soft Computing and Intelligent Systems Design" Addison Wesley, Pearson Education Ltd.

Kramer Steven, L., (1996)"Geotechnical earthquake engineering," Prentice Hall, Upper Saddle River, New Jersey.

Kurt J. Marfurt., (1998) "Accuracy of finite-difference and finite-element modeling of the scalar and elastic wave equations", Geophysics, Vol. 49, NO 5 (May 1984); Pg 533-549.

Lai, C. G., (1998), Simultaneous inversion of Rayleigh phase velocity and attenuation for near-surface site characterization, PhD Diss., Georgia Inst. Of Techn., Atlanta (Georgia, USA) , 370 pp.

Lamb, H., (1904) "On the propagation of tremors over the surface of an elastic solid," in Philosophical transactions of the Royal society of London, A203, 1-42.

Legendre, S., Goyette, J., and Massicotte, D., (2001) "Ultrasonic NDE of Composite material structures using Wavelet Coefficients"; NDT&E International, 34, pp. 31-37.

Love, A.E.H., (1911), "Some Problems of Geodynamics," Cambridge, U.K.: Cambridge University Press, pp. 160-165.

Liu, M., and Gorman, D.G., (1995), "Formulations of Rayleigh damping and its extensions." *Comput. Struct.,* 57(2), 277-285.

Lin, Y., and Sansalone, M., Jul.-Aug. (1992), "Detecting flaws in concrete beams and columns using the impact-echo method," ACI Materials Journal, v89 n4, pp396-405.

Liu, S.W., Jin H.Huang., Sung, J.C., Lee, C. C., (2002) "Detection of cracks using neural networks and computational mechanics",Computer methods in applied mechanics and engineering 191 2831-2845

Lysmer, J., (1965), "Vertical motion of rigid footings", Contract Report No. 3-115, conducted for WES, Vicksburg, Mississippi, 137 pp

Malcolm Slaney., Auditory toolbox, version 2. Technical Report 1998-010, Interval Research Corporation.

Mallat, S.G., (1989)," A theory for multiresolution signal decomposition: the wavelet Representation," IEEE Transactions on pattern analysis and machine intelligence, 11, No 7, 674-693.

Margrave, F. E., Rigas, K., Bradley, D.A., and Barrowcliffe, P., (1991) "The use of neural networks in ultrasonic flaw detection," Measurement 25 143-154.

Masnata, A., and Sunseri, M., (1995) "Neural network classification of flaws detected by ultrasonic means," NDT&E Int. 29 (2) 87–93.

McCarthy, J., Minsky, M.L., Rochester, N., and Shannon, C. E., (1955), "A proposal for the Darthmouth summer research project on artificial intelligence".

Miller, G. F., and Pursey, H., (1955)"On the partition of energy between elastic waves in a semiinfinite solid," in Proceedings of Royal society of London, vol. 233 of A, pp. 55-59.

Nasseri-Moghaddam, A., Cascante, G., Phillips, C., and Hutchinson, D. J., (2004) " Detection of underground anomalies by studying their damping effects on Rayleigh waves," In: 54th Canadian geotechnical conference, Quebec City, Quebec, October.

Nasseri-Moghaddam, Ali., (2006) "Study of the effect of lateral inhomogeneities on the propagation of Rayleigh waves in an elastic medium," PhD thesis, Civil Engineering, University of Waterloo.

Nasseri-Moghaddam., Cascante, G., Phillips, C., and Hutchinson, D.J., (2006)" Effect of underground cavities on Rayleigh waves-Field and numerical experiments," Soil Dynamic and Earthquake Engg 27, 300-313.

Nisee., National information service for earthquake engineering (1998), University of California, Berkeley.

Nazarian, S., (1984), "In situ determination of soil deposits and pavement systems by spectral analysis of surface waves method," PhD thesis, Univ. of Texas at Austin.

Oppenheim, A.V., and Wilsky, A. S., (1983) "Signals and Systems," Englewood Cliffs, NJ: Prentice Hall.

Obaidat, M. S., Suhail, N.A., and Sadoun, B., (2001)"An intelligent simulation methodology to characterize defects in materials," Information Sciences 137, pp 33-41.

Oscar Martin., Manuel Lopez., and Fernado Martin., (2006) "Artificial neural networks for quality control by ultrasonic testing in resistance spot welding," Journal of Materials Processing Technology.

Pain, H. J., "The physics of vibrations and waves," Fourth edition, Wiley Publishers.

Papoulis, A., July (1989) "The Fourier integral and its applications," McGraw-Hill, New York 1962.

Popovics, J. S., and Rose, J. L., November, (1994), "A survey of developments in ultrasonic NDE of concrete", IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, v41, n1, pp140-143.

Park, C.B., Miller, R. D., and Xia, J.,(1999) "Multichannel analysis of surface waves", Geophysics; 64(3):800–8.

Pedrycz, W., (1997), "Fuzzy clustering with partial supervision," IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, 27(5), pp. 787-795.

Phillips, C., Cascante, G., and Hutchinson, D. J., (2000)"Seismic surface waves to detect underground voids," In: Proceedings of the symposium on the application of geophysics to engineering and environmental problems, Arlington, Virginia. p. 29–37.

Phillips, C., Cascante, G., and Hutchinson, J., (2002) "The innovative use of surface waves for void detection and material characterization," In: Proceedings of the symposium on the application of geophysics to engineering and environmental problems, Las Vegas, Nevada.

Phillips, C., Nasseri-Moghaddam, A., Moore, T., Cascante, G., and Hutchinson, D. J., (2003) "A simple automated method of sasw analysis using multiple receivers," in Proceedings of the Symposium on the Application of Geophysics to Engineering and Environmental Problems, San Antonio, Texas, pp. 1582—1600, SEG.

Pitts, W., and McCulloch, W.S. (1947) "How we know universals: the perception of auditory and visual forms," Bulletin Math, Biophys, 9:27-47.

Qian Shie., (2002), Time-frequency and wavelet transforms, Prentice-Hall Inc.

Wolfgang Hardle and Heiko Lehmann., (2004), "Neural networks", (http://www.quantlet.com/mdstat/scripts/xlg/html/xlghtmlnode45.html).

Rayleigh, L., (1885), "On waves propagated along the plane surface of an elastic solid," Proceedings of the London Mathematical Society, v17, pp 4-11.

Rechtien, R. D., Stewart, D. M., (1975)"A seismic investigation over a near surface Cavern," Geoexploration, 13:235–46.

Richard, F.E., Jr., Hall, J.R., Jr., and Woods, R.D., (1970), Vibrations of Soil and Foundations, Prentice-Hall, Inc., Englewood Cliffs, NJ, 414 pages.

Rhim. J, Lee.S.W, (1995),"A neural network approach for damage detection and identification of structures," Comput. Mech. 16 437–443.

Raju Damarla, T., Karpur, P., and Bhagat, P.K., (1992)"A self-learning neural net for ultrasonic signal analysis," Ultrasonics 30 (5) 317– 324.

Robinson, E.A., and Durrani, T.S., (1986), Geophysical signal Processing, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Roesset, J. M., Chang, D.W., Stokoe, K. H., and Aouad, M., (1989), "Modulus and Thickness of the Pavement Surface Layer from SASW Tests," Transportation Research Record, No. 1260, pp. 52-63.

Rose, J. L., (1977) "A 23 sorting study in ultrasonics and pattern recognition," *Mat Eval;* 35:87-96.

Rose., and Joseph, L., (1999) "Ultrasonic waves in solid media," Cambridge, U.K: Cambridge University Press.

Sanchez-Salinero, I., Roesset, J.M., Shao, K-Y., Stokoe, K.H., Ii, and Rix, G.J., (1987), "Analytical Evaluation Of Variables Affecting Surface Wave Testing Of Pavements," Transportation Research Record, No.1136, pp.86-95.

Santamarina J. Carlos., (2001) "Introduction to discrete signals and inverse problems in civil engineering," ASCE Press, Reston, Virginia.

Santos, J. B., and Perdigao, F., (2001) "Automatic defects classification-a contribution," NDT & E International 34, pp 313-318.

Shankar, R., Mucciardi, A.N., Whalen, M.F., and Johnson, M.D., (1978) *Proc. ARPAIAFML Review of Progress in Quantitative NDE* (Ed Thompson, D.O.) Technical Report AFML-TR-78-55, Wright-Patterson Air Force Base, Ohio 50-72.

Shensa, M.J., Oct (1992) “The Discrete Wavelet Transforms: Wedding the À’ Trous and Mallat Algorithms”, IEEE Trans. Signal Processing, Vol. 40, No.10, pp. 2464-2482.

Sheu, J.C., Rix, G.J., and Stokoe, K.H., II., (1988), Rapid Determination of Modulus and Thickness of Pavement Surface. Layer”, presented at the Transportation Research Board annual meeting.

Socco, L.V., and Strobbia, C., (2004) “Surface-wave method for near surface characterization a tutorial,” Near surface geophysics, vol. 2, no. 4, pp. 165-185.

Song, S.J., Schmerr, L.W., (1992) “Ultrasonic flow classification in weldments using probabilistic neural networks,” J. Nondestr. Eval. 11 (2) (1992) 69–77.

Steward A. Rounds., (2002), “Development of a neural network model for dissolved oxygen in the Tualatin river, Oregon,” *in* Proceedings of the Second Federal Interagency Hydrologic Modeling Conference, Las Vegas, Nevada.

Suaris, W., and Fernando, V., (1987). “Ultrasonic pulse attenuation as a measure of damage growth during cyclic loading of concrete.” ACI Mater. J., 84(3), 185–193

Tallavo Fernado., (2006), personal communcitation of Matlab® coding for implementing Fast Fourier Transformation, Ph.D Scholar, University of Waterloo.

Upda, L., and Udpa, S.S., (1990)”Eddy current defect characterization using neural networks”, Mat Eval 342-347.

Valliappan, H.S., and Murti, V., (1984). “Finite element constraints in the analysis of wave propagation problems.” UNICIV Rep. No. R-218, School of Civil Engineering, Univ. of New South Wales, New South Wales, Australia.

Walker James, S., (1996) “Fast Fourier transforms,” CRC-Press, 2nd edition.

Watkins, J.S., Godson, R.H., and Watson, K.,(1967)” Seismic detection of near surface cavities,” (Geological Survey Professional Paper 599-A). National Aeronautics and Space Administration. Washington, DC: US Government printing office.

Werbos, P., (1974),”Beyond regression: new tools for prediction and analysis in the behavioral sciences,” PhD dissertation, Harvard University.

White-W., Valliappan, S., and Lee, I.K., (1977), ‘Unified Boundary for Finite Dynamic Models’, Journal of Eng Mech. Div., ASCE, vol.103, no.EM5, pp 949-964.

Woods, R.D., (1968) “Screening of surface waves in soils,” Journal of the soil mechanics and foundations division, vol. 94, no. SM4, pp. 951-979.

Xia, J., Miller, R.D., and Park, C.B., (1999),"Estimation of near-surface shear-wave velocity by inversion of Rayleigh waves," Geophysics 64(3): 691–700.

Zerwer, A., Cascante, G., and Hutchinson, J., (2002) "Parameter Estimation in Finite Element Simulations of Rayleigh Waves," Journal of Geotechnical and Geoenvironmental Engineering, Vol 128, No. 3.

Appendix A

Developed Mathcad® Sheets

This appendix contains the developed Mathcad work sheets that were used for the calculations of the following

Mathgram Chapter 4-1: shows the Lamb solution and calculates the corresponding displacements

Mathgram Chapter 4-2: shows the details of the numerical model and the calibration procedure

Mathgram Chapter 4-3: shows the different sources used for the sample data generation

Appendix A: Mathcad Files

Mathgram Chapter 4-1: Lamb Solution

References: Bath and Berkout (1984)

Material Properties:

Poisson ratio =0.25 Density (kg/m^3) = 2340 Elastic Modulus (MPa) = 45000

Shear Modulus G1 (MPa) = 18000 P-Wave Velocity (m/s) = 4800

Shear Wave Velocity (m/s)= 2770 Rayleigh Wave Velocity Vr (m/s) = 2550

Time domain parameters:

Number of points: $N0 := 1200$

$i := 1..N0$ $dt := 10^{-7}$ s $t_i := i \cdot dt$

Frequency domain parameters:

Number of points in frequency: $Nf := \frac{N0}{2}$

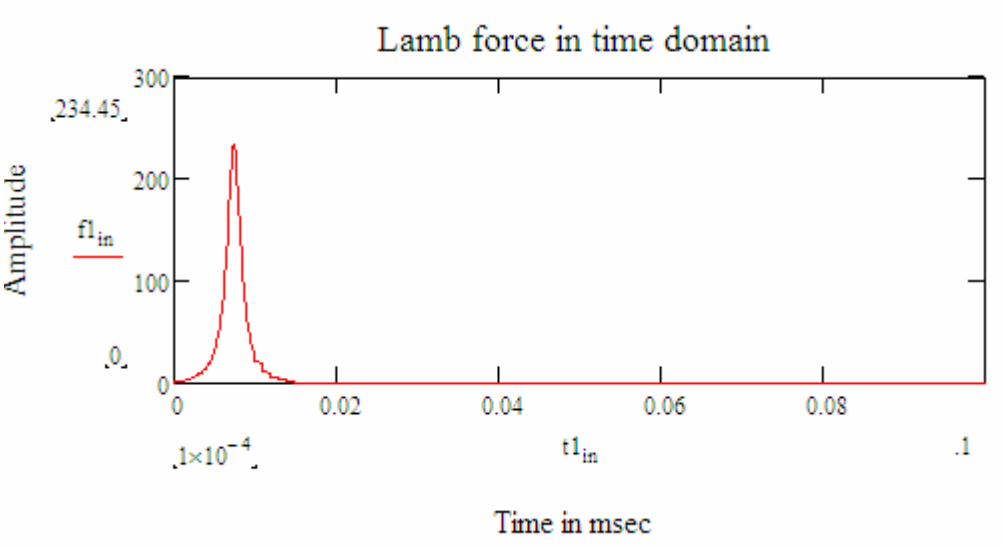
$j := 1..Nf$ $\Delta f := \frac{1}{N0 \cdot dt \cdot 1000}$ Hz $freq_j := \Delta f \cdot j$

Input source:

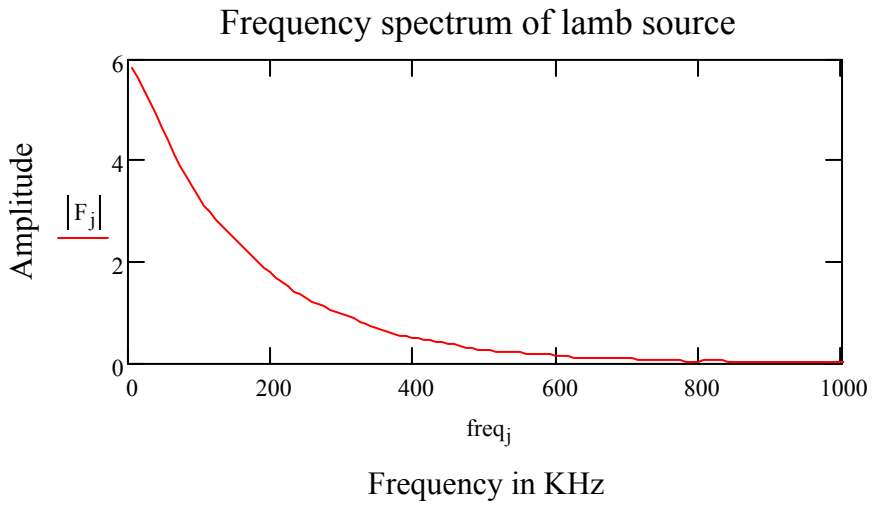
Source freq content or width of the pulse: $\psi := 10^{-6}$ s Source Amplitude: $Fb := .75$

Time shift: $delay := 75 \cdot dt$ $t1_i := t_i \cdot 1000$ ms

Source function: $f(t) := \frac{Fb \cdot \psi}{\pi\left[(t) - delay\right]^2 + \psi^2\right]}$ $fo_i := f(t_i)$ $f1 := \frac{fo}{1000}$



$F := CFFT(f1)$



Theoretical solution for Rayleigh wave displacement:

Spacing between the receivers

$z := 1 .. 595 \qquad \Delta d := \frac{250}{595} \qquad d_z := \Delta d \cdot z$

Constant: $R := 1 \qquad \psi 1 := \psi \cdot 100(\qquad G1 := 1800(\qquad Vr := 255($

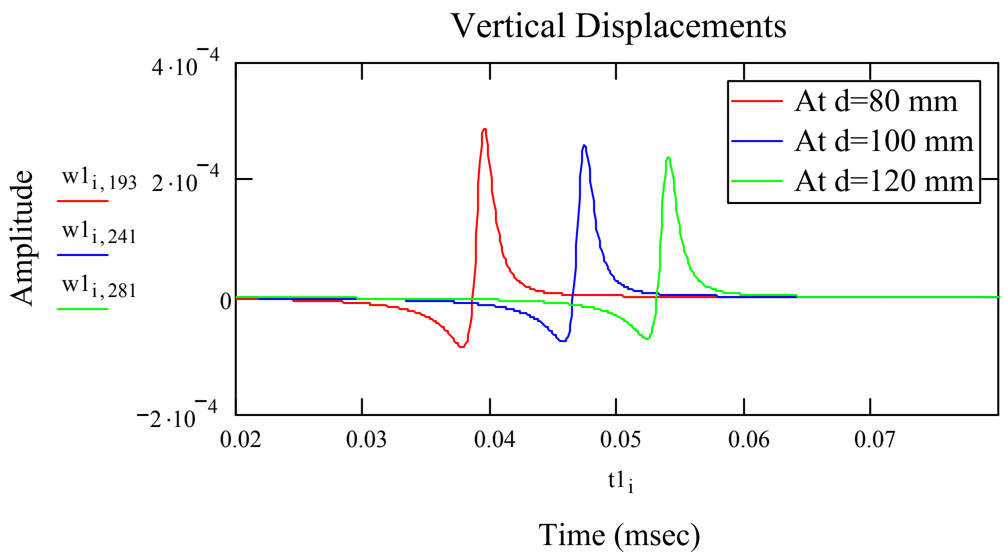
Vertical Surface Displacements far from the source:

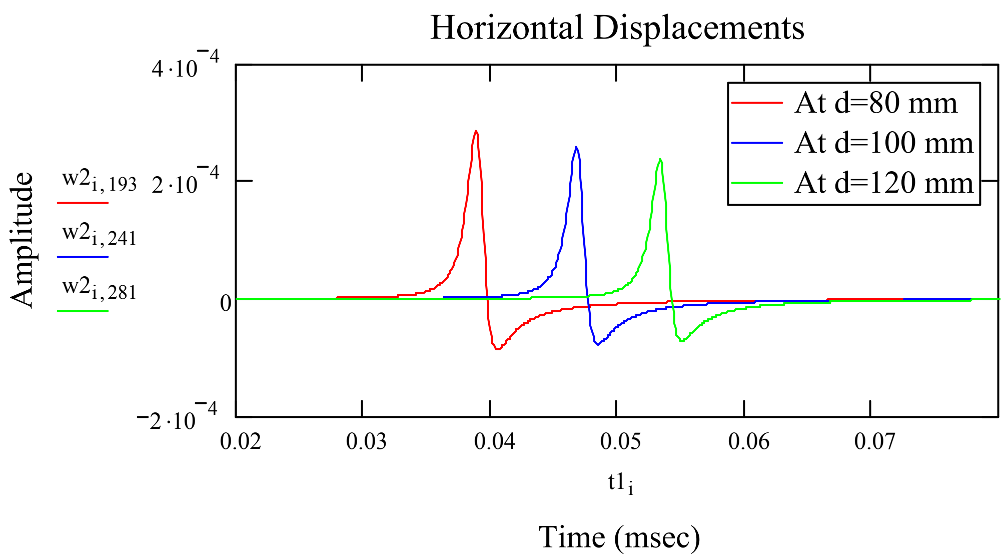
$$v(t1, d) := \operatorname{atan}\left(\frac{t1 - delay \cdot 1000 - \frac{d}{Vr}}{\psi 1}\right)$$

$$wv(t1, d) := \frac{R \cdot Fb}{4 \cdot \pi \cdot G1 \cdot \psi 1^2 \cdot Vr} \cdot \sqrt{\frac{2 \cdot \psi 1 \cdot Vr}{d}} \cdot \cos\left(\frac{\pi}{4} - \frac{3}{2} \cdot v(t1, d)\right) \cdot \cos(v(t1, d))^{\frac{3}{2}} \qquad w1_{i,z} := wv\left(t1_i, d_z\right)$$

Horizontal Surface Displacements far from the source:

$$wh(t1, d) := \frac{R \cdot Fb}{4 \cdot \pi \cdot G1 \cdot \psi 1^2 \cdot Vr} \cdot \sqrt{\frac{2 \cdot \psi 1 \cdot Vr}{d}} \cdot \sin\left(\frac{\pi}{4} - \frac{3}{2} \cdot v(t1, d)\right) \cdot \cos(v(t1, d))^{\frac{3}{2}} \qquad w2_{i,z} := wh\left(t1_i, d_z\right)$$





%%

Mathgram Chapter 4-2: Calibration of the numerical model

References: (Valliappan and Murti, 1984) & (Cascante et al.,2002)

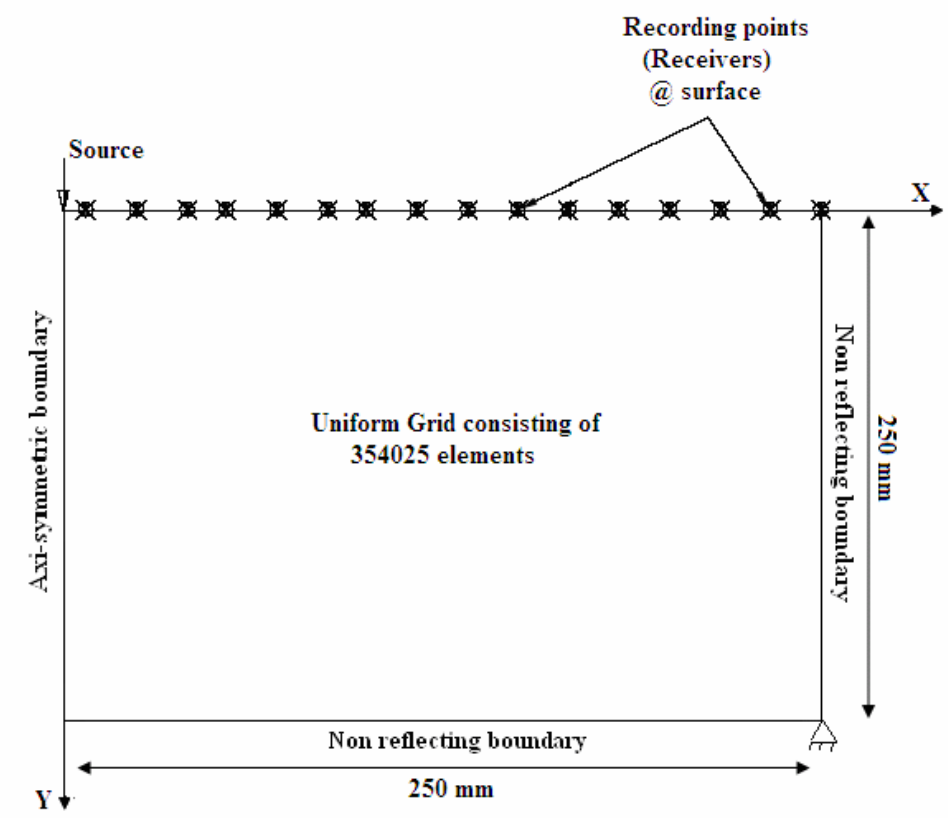
Properties of the numerical model

Model Dimensions = 250*250 Termination time of the simulation t_{max}=0.12 ms

Total Number of elements = 354025 Element dimension = 0.4202 mm

Critical time chosen by LS-DYNA based on the model's wave speed and maximum of the shortest side or the area/ (minimum of the longest side or longest diagonal) is scaled down by a scale factor of 0.5 which gives a value of $4.12*10^{-5}$

Number of receivers =75 Distance between the receivers = 3.36 mm



Material Properties:

Shear Modulus G1 (MPa) = 18000 Poisson ratio =0.25 Density (kg/m^3) = 2340

Rayleigh Wave Velocity Vr (m/s) = 2550 P-Wave Velocity (m/s) = 4800

Shear Wave Velocity (m/s) = 2770

Boundary Conditions:

In this study, the bottom and right boundaries of the model are free and non-reflecting boundary conditions were applied to them inorder to represent an infinite half space and to minimize the effect of reflections. The left boundary is fixed in the horizontal direction, thereby, representing the axes of symmetry. The free body motion of the model in X and Y direction is prevented by applying translational constraints in both the X and Y directions on the lower right corner of the model.

Damping Parameters:

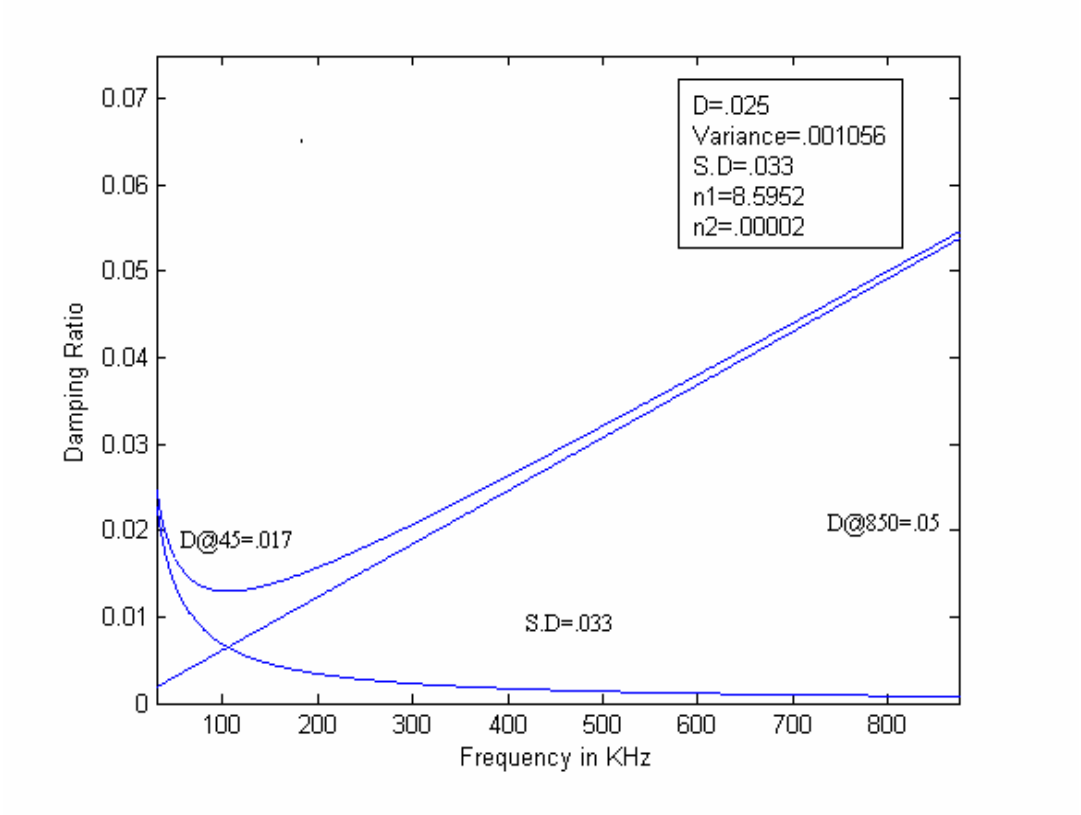
The following equations were solved to obtain the damping parameters:

$$D := \frac{\left[\frac{\eta 1}{2}\ln\left(\frac{\omega 1}{\omega n}\right) + \frac{\eta 2}{4}\cdot\left(\omega n^2 - \omega 1^2\right)\right]}{\omega n - \omega 1} \qquad \sigma := \frac{(\omega n - \omega 1)\left[\frac{\eta 1^2}{4}\left(\frac{1}{\omega 1} - \frac{1}{\omega n}\right) + \frac{\eta 2^2}{12}\left(\omega n^3 - \omega 1^3\right) + \frac{\eta 1 \cdot \eta 2}{2}(\omega n - \omega 1)\right] - D^2}{(\omega n - \omega 1)^2}$$

Rayleigh Damping D = 2.5% Variance σ =.00105

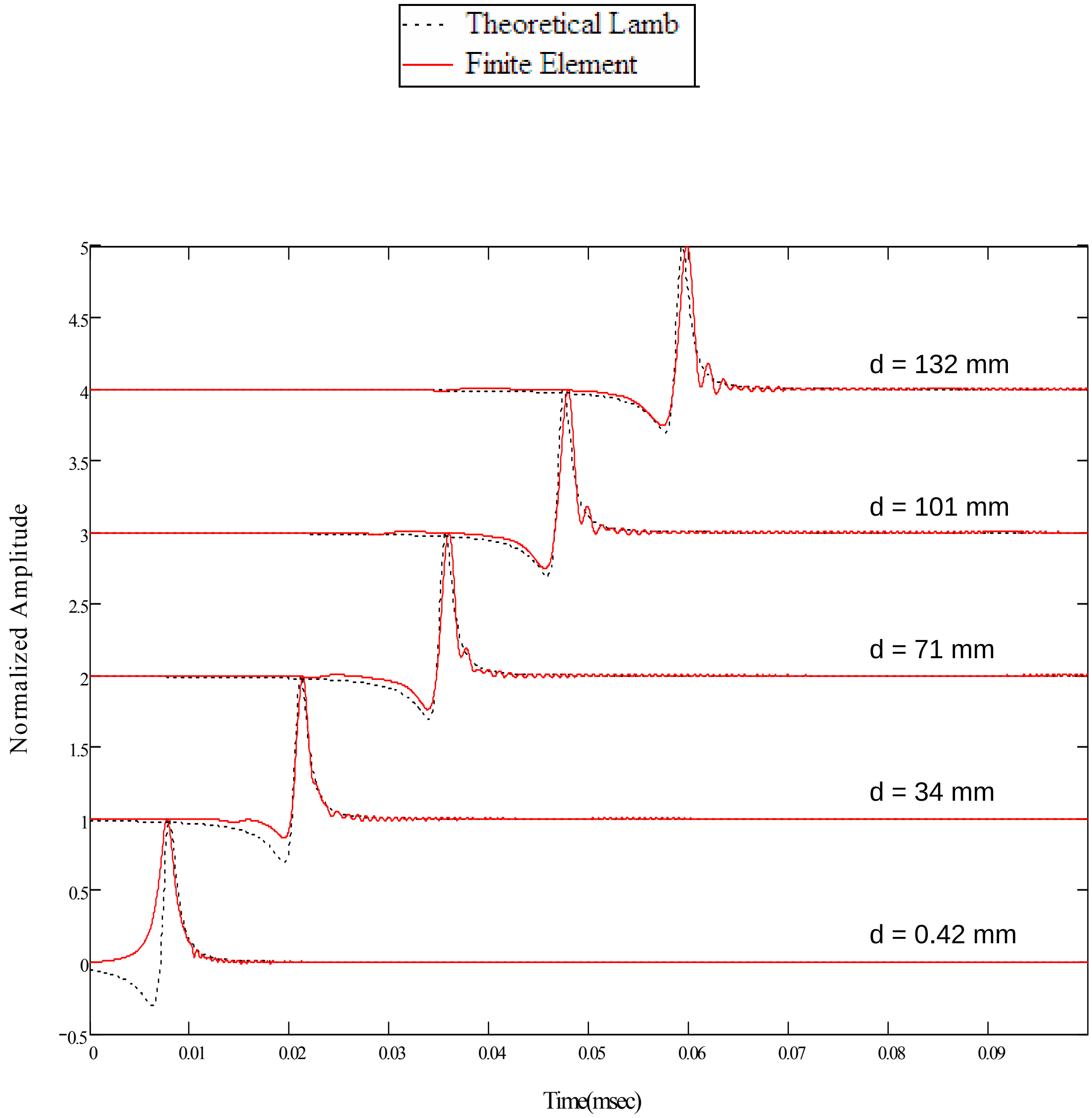
$\omega 1$ = first natural frequency which is taken as 45 KHz

ωn = highest natural frequency which is taken as 850 KHz



Calibration of the numerical model: Comparison of the numerical responses with the theoretical responses obtained from Lamb solution (Mathgram Chapter 4-1)

Camparing the numerical responses with the theoretical responses at typical distances (d) from the source



where d indicates the distance from the source

%%%

Mathgram Chapter 4-3: Sources used

References: Bath and Berkout (1984), Ricker (1945)

Source 1: Lamb

Time domain parameters:

Number of points: $N0 := 120($

$i := 1..N0$ $\quad dt := 10^{-7}$ s $\quad t_i := i \cdot dt$

Frequency domain parameters:

Number of points in frequency: $Nf := \frac{N0}{2}$

$j := 1..Nf$ $\quad \Delta f := \frac{1}{N0 \cdot dt \cdot 1000}$ Hz $\quad freq_j := \Delta f \cdot j$

Input source:

Source freq content or width of the pulse: $\psi := 10^{-6}$ s

Source Amplitude: $Fb := .75$

Time shift: $delay := 75 \cdot dt$ $\quad t1_i := t_i \cdot 1000$ ms

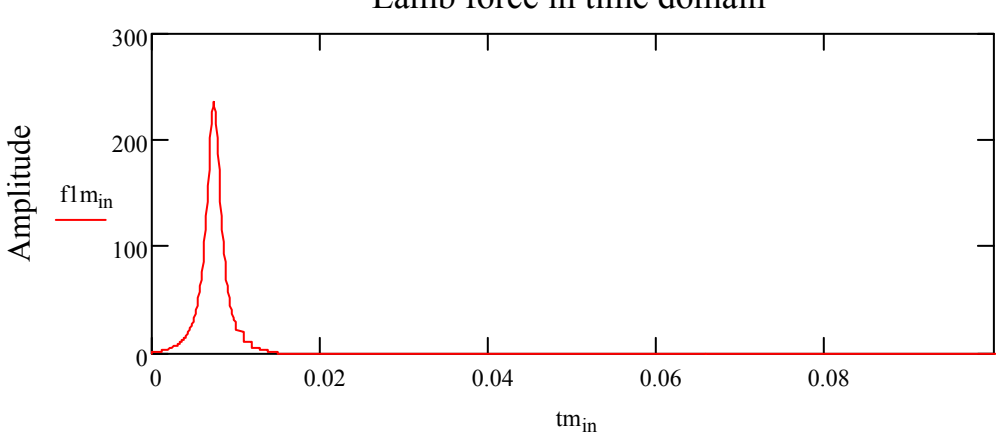
Source function: $f(t) := \frac{Fb \cdot \psi}{\pi\left[(t) - delay\right]^2 + \psi^2\right]}$ $\quad fo_i := f\left(t_i\right)$ $\quad f1 := \frac{fo}{1000}$

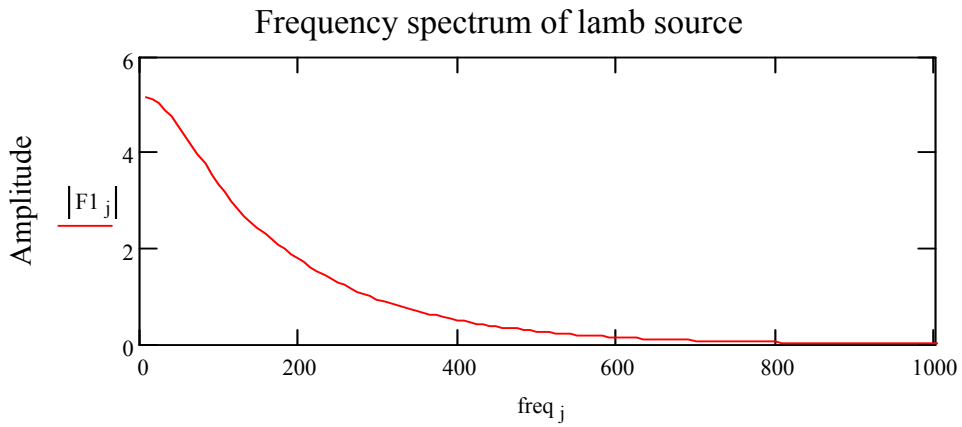
The source was modified so that the function starts at zero. This was done to remove oscillations in the numerical responses.

$Load := READPRN("G:\Extra files1\\load.cvr")$

$rows(Load) = 1.2 \times 10^3$ $\quad cols(Load) = 2$ $\quad n := rows(Load)$ $\quad in := 0..n-1$

$f1m_{in} := \left(Load^{\langle 1 \rangle}\right)_{in}$ $\quad tm_{in} := \left(Load^{\langle 0 \rangle}\right)_{in}$ $\quad F1 := CFFT(f1m)$





Source 2: Ricker

Time domain parameters:

N0 := 120(i := 1.. N(

dt := 10^{-7} s $t_i := i \cdot dt$

Frequency domain parameters:

$Nf := \frac{N0}{2}$ j := 1.. Nf

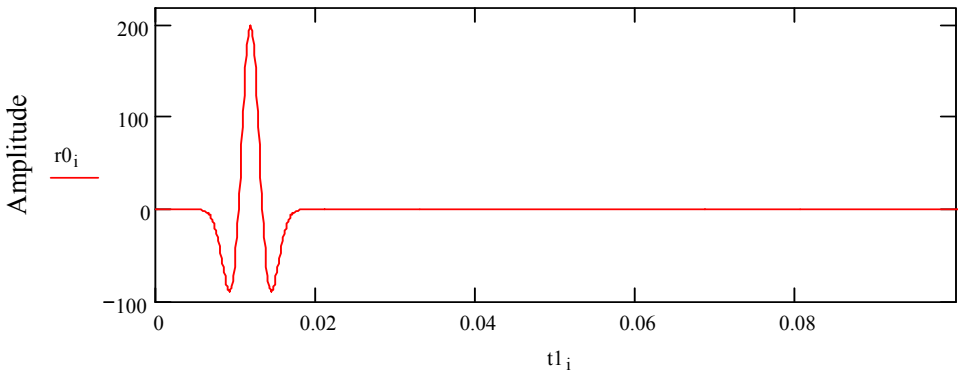
$\Delta f := \frac{1}{N0 \cdot dt \cdot 1000}$ $freq_j := \Delta f \cdot j$

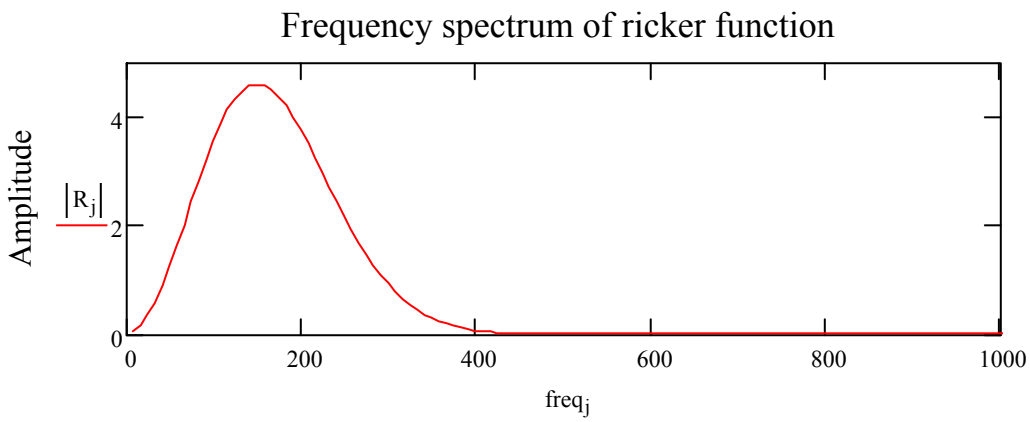
Input source:

Source freq content or width of the pulse: ψ1 := .001[illegible] ms Source Amplitude: Fb1 := 50(

Time shift: dt := dt·100(msec $t1_i := t_i \cdot 100($ ms delay1 := 120·dt

Source function: $rw(t1) := \frac{Fb1}{\sqrt{2\pi}}\left[1 - \frac{(t1 - delay1)^2}{\psi 1^2}\right] \cdot e^{\frac{-(t1-delay1)^2}{2\psi 1^2}}$ $r0_i := rw(t1_i)$ R := CFFT(r0)





Frequency in KHz

Source 3: Impact Source

Time domain parameters:

$N0 := 120($ $\quad i := 1..N0$

$dt := 10^{-7}$ s $\quad t_i := i \cdot dt$

Frequency domain parameters:

$Nf := \frac{N0}{2}$ $\quad j := 1..Nf$

$\Delta f := \frac{1}{N0 \cdot dt \cdot 1000}$ $\quad freq_j := \Delta f \cdot j$

Input source:

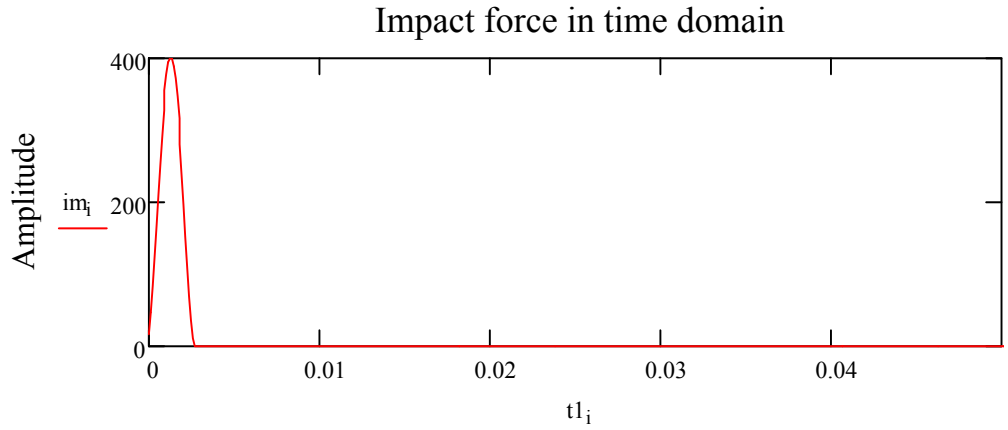
Source freq content or width of the pulse: $\alpha := 375$ Source Amplitude: $Fb2 := 40($

Time shift: $dt := dt \cdot 100($ msec $\quad t1_i := t_i \cdot 100($ ms

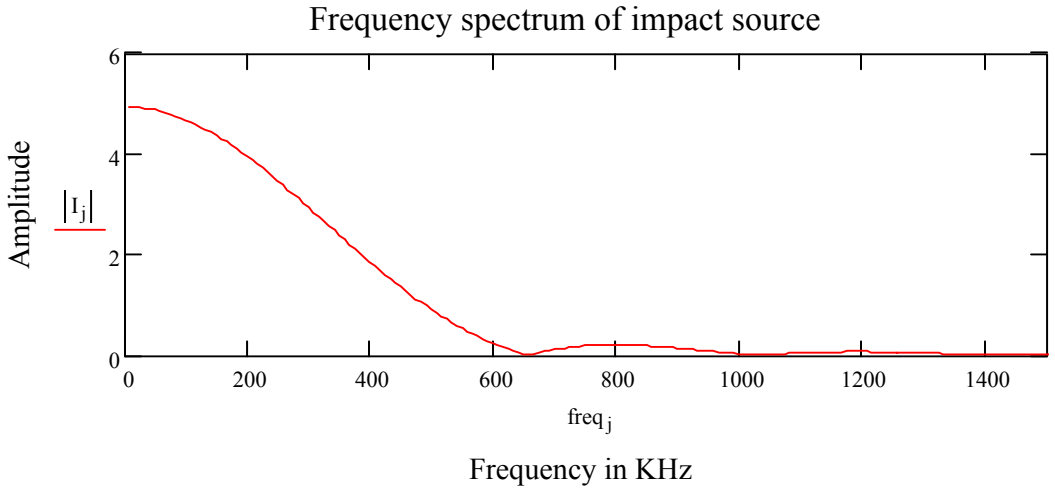
Source function:

$$q1(t1) := \begin{cases} \sqrt{\sin(\alpha \cdot \pi t1)} & \text{if } \alpha \cdot \pi t1 < \pi \\ 0 & \text{otherwise} \end{cases}$$

$q2(t1) := Fb2 \sin(\alpha \pi t1) \cdot q1(t1)$ $\quad im_i := q2(t1_i)$ $\quad I := CFFT(im)$



Time in msec



Source 4: Decaying Sinusoidal Source

Time domain parameters:

$N0 := 1200$ $\quad i := 1..N0$

$dt := 10^{-7}$ s $\quad t_i := i \cdot dt$

Frequency domain parameters:

$Nf := \frac{N0}{2}$ $\quad j := 1..Nf$

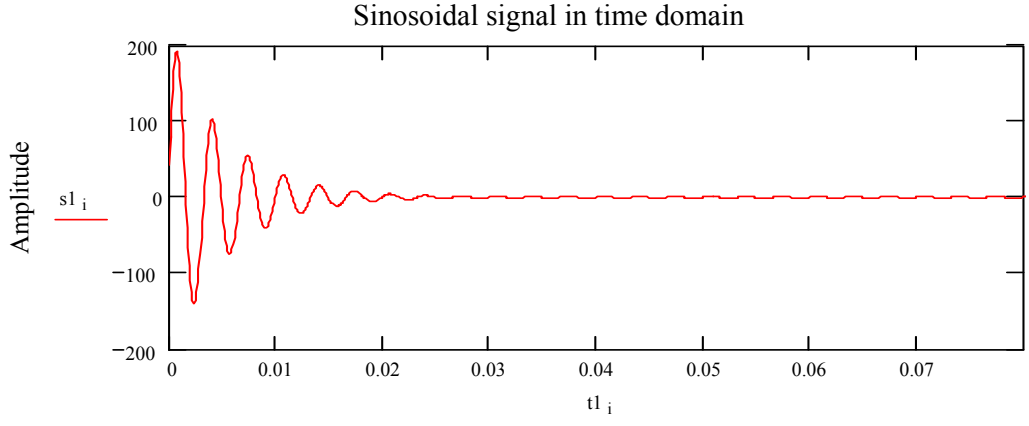
$\Delta f := \frac{1}{N0 \cdot dt \cdot 1000}$ $\quad freq_j := \Delta f \cdot j$

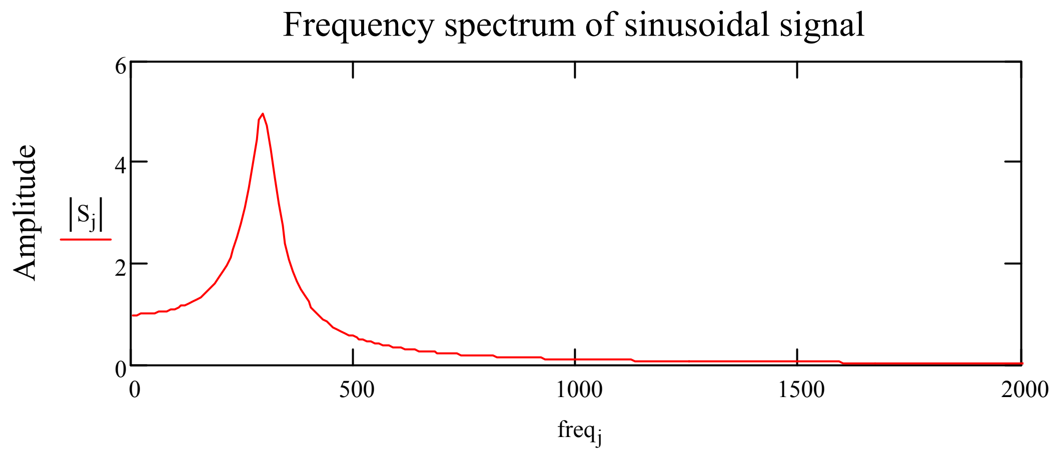
Input source:

Source freq content or width of the pulse: $\xi := 0.1$ Source Amplitude: $Fb3 := 225$

Time shift: $dt := dt \cdot 1000$ msec $\quad t1_i := t_i \cdot 1000$ ms $\quad f_0 := 300$ $\quad \omega := 2 \cdot \pi \cdot f_0$

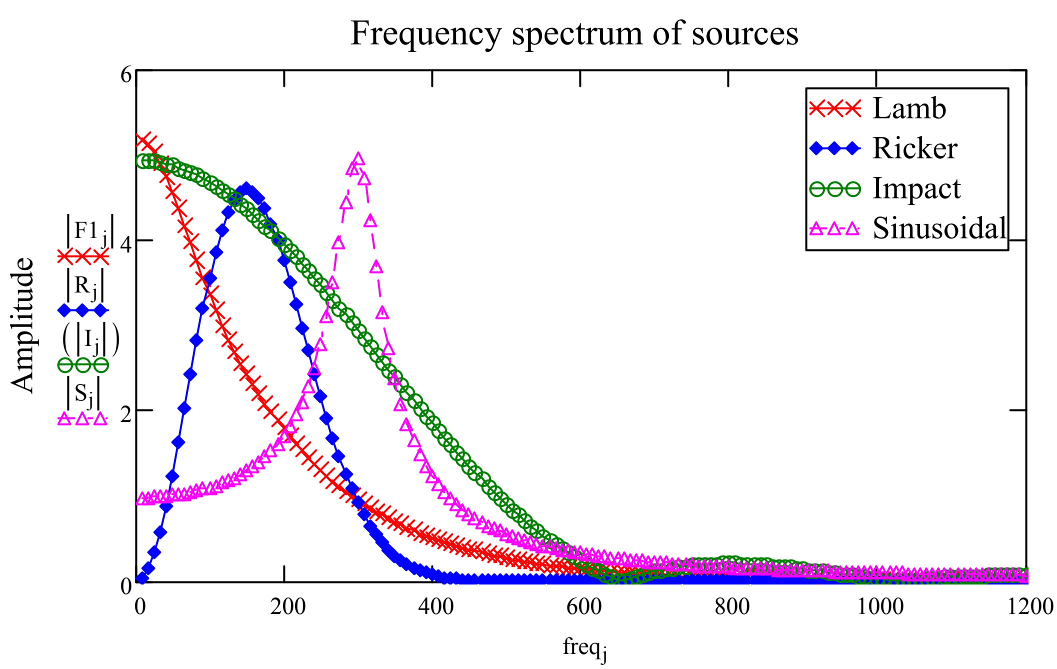
Source function: $s(t1) := Fb3 \cdot \sin(\omega \cdot t1) \cdot \exp(-\omega \cdot \xi \cdot t1)$ $\quad s1_i := s(t1_i)$ $\quad S := CFFT(s1)$





Frequency in KHz

Comparison of the frequency spectrum of the different sources:



Frequency in KHz

Source type	Fcog in KHz	Fmax in KHz	F (5%)	F (95%)	Average
Lamb	169.61	8.333	14	251	31.923
Morlet	172	170	79	260	4.539
Ricker	169.5	150	80	250	0.000334
Impact	370	8.333	26	536	4.631
Sinusoidal	487.17	300	84	425	9.27

%%%

Appendix B

Developed Matlab® Sheets

This appendix contains the developed Matlab work sheets that were used for the calculations of the following

Matlab Chapter 4-1: Contour Plots

Matlab Chapter 5-1: Process of feature extraction

Matlab Chapter 5-2: Scaling of features

Matlab Chapter 5-3: Computation of the Confusion matrix

Matlab Chapter 5-4: Development of Backpropagation neural network

Matlab Chapter 5-5: Development of Adaptive neuro-fuzzy inference system

Matlab Chapter 5-6: Development of k-Nearest Neighbor Classfier

Matlab Chapter 5-7: Development of Linear Discriminate Classifier

Matlab Chapter 5-8: Development of Fuzzy C-means Clustering

%Matlab Chapter 4-1: Contour Plots

```
%%%Program for plotting contour plots%%%%
clear;
clc;

%load the data
load xs_lambB.txt

%load the data
load lambB_2916.txt %
%load the time data
time=lambB_2916(:,1);

%get the distances between the traces
disp=0.4202; %distance between the source and the first trace
for i=2:75;
disp(i)=disp(i-1)+3.3616;
end

%normalize all the traces to the max value of the corresponding traces so the max
%value in each trace is equal to 1
[m,n]=size(xs_lambB);

for i=1:n;
      dataN(:,i)=xs_lambB(:,i)/max(xs_lambB(:,i));
end;

contourf(time,disp,dataN',64)
hold on;
shading flat;
hold off;
xlabel('Time(msec)')
ylabel('Distance from source(mm)')
colorbar
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

%Matlab Chapter 5-1: Feature Extraction

```
%%%%%%%%%%%%%%Program for extracting features%%%%%%%%%%%%
clear;
clc;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
format long;
%load the data from which the features have to be extracted
load x_nolambB.mat
data=[x_nolambB];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Features from Time traces
%Features from time domain: [max min mean stddev skewness kurtosis]
%5features

for i=1:size(data,2);
x(:,i)=data(:,i);
v(:,i)=var(x(:,i));
mxx(:,i)=max(x(:,i)); %max
mix(:,i)=min(x(:,i)); %min
meanx(:,i)=mean(x(:,i)); %mean
stx(:,i)=std(x(:,i)); %stddev
skx(:,i)=skewness(x(:,i)); %skewness
kux(:,i)=kurtosis(x(:,i)); %kurtosis
end;

timef = [mxx' mix' meanx' stx' skx' kux'];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Features obtained from the derivatives of the responses
%3 features
for i=1:size(data,2);
q(:,i)=sum(abs(diff(data(:,i))));
q1(:,i)=sum(abs(diff(diff(data(:,i)))));
q2(:,i)=sum(abs(diff(diff(diff(data(:,i))))));
end

der=[q' q1' q2'];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Features from wavelet transformation
%Wavelet Analylis:
%[ca1mean1 ca1std1 cd1mean1 cd1std1 ca1CDVD3 ca1skew1 ca1kurt1]
%7 features
```

```
for i=1:size(data,2);
s(:,i)=data(:,i);
s1(:,i)=[0;s(:,i)];
%s1(:,i)=[s(:,i)];

%Wavelet Transformation computes the approximate coefficients
%vector, CA and detail coefficients vector, CD obtained by a
%single level wavelet decomposition of the response vector
%using the Daubechies window of order 2.

[ca1(:,i),cd1(:,i)] = dwt(s1(:,i),'db2');
ss(:,i) = idwt(ca1(:,i),cd1(:,i),'db2');
err(:,i) = norm(s1(:,i)-ss(:,i));
[Lo_R,Hi_R] = wfilters('db2','r');
ss(:,i) = idwt(ca1(:,i),cd1(:,i),Lo_R,Hi_R);
ca1max(:,i)=max(ca1(:,i)); %ca max
ca1min(:,i)=min(ca1(:,i)); %ca min
ca1mean(:,i)= mean(ca1(:,i));%ca mean
ca1std(:,i) = std(ca1(:,i)); %ca std
ca1CV(:,i)=ca1std(:,i)/ca1mean(:,i); %ca coeff of var
ca1skew(:,i)= skewness(ca1(:,i)); %ca skewness
ca1kurt(:,i)= kurtosis(ca1(:,i)); %ca kurtosis
cd1mean(:,i)=mean(cd1(:,i)); %cd mean
cd1std(:,i)= std(cd1(:,i)); %cd std
end;

wc=[ca1max' ca1min' ca1mean' ca1std' ca1CV' ca1skew' ca1kurt' cd1mean' cd1std'];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Features by taking logarithmric value of the Fourier amplitude
%logDFT Analylis: [max min mean stddev skewness kurtosis]
%6 features

for i=1:size(data,2);
d(:,i)=data(:,i);
Y(:,i) = fft(d(:,i));
Pyy=log(Y);
lmxp=real(max(Pyy)); %max
lmip=real(min(Pyy)); %min
lmpow=real(mean(Pyy)); %mean
lstpow=real(std(Pyy)); %stddev
lskpow=real(skewness(Pyy)); %skewness
lkupow=real(kurtosis(Pyy)); %kurtosis
end;

logdft = [lmxp' lmip' lmpow' lstpow' lskpow' lkupow'];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Features from Cepstral analysis
% MFCC Analylis: [mean stddev skewness kurtosis]
%4 features

for i=1:size(data,2);
   x(:,i)=data(:,i);
ccepsco(:,i)=cceps(x(:,i));
mcep(:,i)=mean(ccepsco(:,i));%mean
stcep(:,i)=std(ccepsco(:,i));%std
skcep(:,i)=skewness(ccepsco(:,i));%skewness
kucep(:,i)=kurtosis(ccepsco(:,i));%kurtosis
end;

mfcc = [mcep' stcep' skcep' kucep'];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Combining all the features extracted

features=[timef,wc,logdft,mfcc,der];
save features features

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

%Matlab Chapter 5-2: Scaling

```
%%%%%%%%%%%%%%Program for extracting features%%%%%%%%%%%%
clear;
clc;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%load the data
load feaB.mat;
checkmax = max(feaB);
checkmin = min(feaB);
check = [checkmin;checkmax];
size_matrix = size(feaB);
total_samples = size_matrix(:,1);
total_parameters = size_matrix(:,2);
for k=1:total_parameters
   minparam = check(1,k);
   maxparam = check(2,k);
   diff_param = maxparam - minparam;
   if diff_param > 0
      for l=1:total_samples
         replace_val=(feaB(l,k)-minparam)/diff_param;
         feaB_scaled(l,k) = replace_val;
      end
   end
end
save feaB_scaled feaB_scaled

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

%Matlab Chapter 5-3:Confusion Matrix

```
%%%%%%%%%%%%%%%Program for Confusion Matrix%%%%%%%%%%%%%
% CONFMAT Generates a confusion matrix%
% c = confmat(x,y)%
% Author Adrian Chan%
% This function generates a confusion matrix.%
% Inputs
%    x: vector of what the signal should have been
%    y: vector of what the signal was classified as%
% Outputs
%    c: confusion matrix (rows are inputs, colums are outputs)%
% Modifications
% 00/02/01 AC First created.
% 01/01/18 AC c(i,j) = length(find(z == j))
%                                  changed to c(i-minx+1,j-minx+1) = length(find(z == j))
%            This allows any minx.

function c = confmat(x,y)

minx = min(x);
maxx = max(x);

c = zeros(maxx-minx);
for i = minx:maxx
  index = find(x == i);
  for j = minx:maxx
    z = y(index);
    c(i-minx+1,j-minx+1) = length(find(z == j));
  end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

%Matlab Chapter 5-4: Backpropagation Neural Network (BPNN)

```
%Also includes normalization of the dataset

%%%%%%%%%%%%%%Program for BPNN%%%%%%%%%%%%%%%%%%%
clear;
clc;

%generation of training and testing data
training_data = []; %75 training
testing_data = []; %25 testing

%Load the scaled feature data set
load featuresB_scaled.mat;
features=featuresB_scaled;

%Load the target output file for the different classes
load out_basicB.mat;
output=out_basicB;

%Selecting the training and testing data set randomly
num_of_training = round(.75*length(features(:,1)));
num_of_testing = length(features(:,1)) - num_of_training;
sam_number = [1:length(features(:,1))]';
alldata_scaled = [sam_number features output];

rand('seed',7171751);

for i=1:num_of_training
   cur_index = randint(1,1,[1 length(features(:,1))]);
   training_data = [training_data;alldata_scaled(cur_index,:)];
end

for i=1:num_of_testing
   cur_index = randint(1,1,[1 length(features(:,1))]);
   testing_data = [testing_data;alldata_scaled(cur_index,:)];
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Normalizing the training data set
x=training_data(:,2:38)';
p=x;
perexplained=0.01;
      [pn,meanp,stdp] = prestd(p);
x=pn';
ninput=size(x(1,:));
```

```
y=training_data(:,39);
nsample=length(y);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Normalizing the testing data set
xtest=testing_data(:,2:38)';
ptest=xtest;
[pntest,meanp,stdp] = prestd(ptest);
xtest=pntest';
ytest=testing_data(:,17);
nsample1=length(ytest);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Input parameters for the BPNN
no=input('Number of neurons in the hidden layer = ');
%input the number of epochs
e=input('Number of epochs = ');
%input the termination error tolerance
goal=input('Enter the maximum tolerable error = ');
P=x;
T=y;
a=xtest;
s=ytest;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Preprocesfeatures of the data
%Some transfer functions need that the inputs and targets are scaled so that
%they fall within a specified range. In order to meet this requirement we
%need to pre-process the data
%For PrepocesfeaturesBg Premnmx is used so that the maximum is +1 and minimum is
%-1
[pn,minp,maxp,tn,mint,maxt]=premnmx(P',T');
[an,mina,maxa,sn,mins,maxs]=premnmx(a',s');

%Feed forward network
%Creating a feed forward neural net with one hidden layer,one hidden layer
%with tangent sigmoid as transfer function in hidden layer and in the output
%layer, and with gradient descent with momentum backpropagation training function
%the syntax is
%net = newff(PR,[S1 S2...SNl],{TF1 TF2...TFNl},BTF,BLF,PF)
%here features set of the data is large the minmax command is used to get the max and
%the min values of the input data
net=newff(minmax(pn),[no 10 1],{'tansig','tansig','tansig','tansig'},'traingdm');
%input the number of epochs for which the network has to be trained
net.trainParam.epochs=e;
%input the maximum tolerable error
```

```
net.trainParam.goal=goal;
%the network is trained with the training data
net=init(net);
net=train(net,pn,tn);

%Testing
%After training the network we simulate our testing data to see how well
%our trained network predicts the output
w=sim(net,an);
%to convert the predicted output to the original scale the data has to be
%postprocessed ufeaturesBg the postmnmx command
predy=postmnmx(w',mins,maxs);
nsample1=length(ytest);

%to see the predicted nt and the actual outputs for the testing data
for i=1:nsample1;
if predy(i) < 1.5
predy1(i)=1;
else
predy1(i)=2;
end
end

d=[predy1'-ytest].^2;
%mean square error
mse=mean(d)
%finding the root mean square error
rmse=sqrt(mse)

%Creating the confusion matrix
cm=confmat(s,predy1')/nsample1
cm1=confmat(s,predy1')

%Accuracy of prediction by the network
Accuracy=(cm(1,1)+cm(2,2))*100

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

%Matlab Chapter 5-5: Adaptive Neuro-fuzzy Inference System (ANFIS)

```
%Also includes feature reduction (Principle Component Analysis PCA)

%%%%%%%%%%%%%%%%%Program for ANFIS%%%%%%%%%%%%%%%%%
clear;
clc;
%generation of training and testing data
training_data = []; %75 training
testing_data = []; %25 testing

%Load the scaled feature data set
load featuresB_scaled.mat;
features=featuresB_scaled;

%Load the target output file for the different classes
load out_basicB.mat;
output=out_basicB;

%Selecting the training and testing data set randomly
num_of_training = round(.75*length(features(:,1)));
num_of_testing = length(features(:,1)) - num_of_training;
sam_number = [1:length(features(:,1))]';
alldata_scaled = [sam_number features output];

rand('seed',7171751);

for i=1:num_of_training
   cur_index = randint(1,1,[1 length(features(:,1))]);
   training_data = [training_data;alldata_scaled(cur_index,:)];
end

for i=1:num_of_testing
   cur_index = randint(1,1,[1 length(features(:,1))]);
   testing_data = [testing_data;alldata_scaled(cur_index,:)];
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Normalizing and applying PCA on the training data set
x=training_data(:,2:38)';
p=x;

%Principle Component Analysis
% PREPCA preprocesses the network input training set by applying a principal
%component analysis.
```

```
% This analysis transforms the input data so that the elements of the input vectors will be
%uncorrelated. In addition, the size of the input vectors may be reduced by retaining only
those components which contribute more than a specified fraction (min_frac) of the total
variation in the data set.

% PREPCA(p,min_frac) takes these inputs:
% P- RxQ matrix of centered input (column) vectors.
% min_frac - Minimum fraction variance component to keep and returns:
% Ptrans   - Transformed data set.
% TransMat - Transformation matrix.

perexplained=0.01;%min_frac
      [pn,meanp,stdp] = prestd(p);
      [ptrans,transMat] = prepca(pn,perexplained);
      while (length(ptrans(:,1))) > 3;
         perexplained=perexplained+0.01;
[ptrans,transMat] = prepca(pn,perexplained);
      end;

x=ptrans';
ninput=size(x(1,:)); %number of input vectors (after PCA) for prediction
y=training_data(:,39);
nsample=length(y);

%Input the number of epochs for which the network is trained
epoch_n = 50;

% genfis1 generates a single-output Sugeno-type fuzzy inference system (FIS) using a
%grid  partition on the data (no clustering). By default, GENFIS1 uses two 'gbellmf' type
%membership functions for each input. Each rule generated by GENFIS1 has one output
% membership function, which is of type 'linear' by default.

% anfis uses a hybrid learning algorithm to identify the membership function parameters
%of single-output, Sugeno type fuzzy inference systems (FIS). A combination of least-
%squares and backpropagation gradient descent methods are used for training.

in_fis  = genfis1([x y],2,'gbellmf');
out_fis = anfis([x y],in_fis,epoch_n);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Normalizing and applying PCA on the testing data set
xtest=testing_data(:,2:38)';
ptest=xtest;

perexplained=0.01;
      [pntest,meanp,stdp] = prestd(ptest);
```

```
      [ptranstest,transMattest] = prepca(pntest,perexplained);
      while (length(ptranstest(:,1))) > 3;
         perexplained=perexplained+0.005;
[ptranstest,transMattest] = prepca(pntest,perexplained);
      end;

xtest=ptranstest';
ytest=testing_data(:,39);
nsample1=length(ytest);

%After training the network we simulate our testing data to see how well the trained
%network predicts the output
predy= evalfis(xtest,out_fis);

%to see the predicted nt and the actual outputs for the testing data
for i=1:nsample1;
if predy(i) < 1.5
predy1(i)=1;
else
predy1(i)=2;
end
end

%confusion matrix
cm=confmat(ytest,predy1')/nsample1
cm1=confmat(ytest,predy1')

%Accuracy of prediction by the network
Accuracy=(cm(1,1)+cm(2,2))*100

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

%Matlab Chapter 5-6: k-Nearest Neighbor Classifier

```
%Also includes normalization of the dataset

%%%%%%%%%%%%%%%%Program for k-NN%%%%%%%%%%%%%%%%%%%%
clear;
clc;
%generation of training and testing data
training_data = []; %75 training
testing_data = []; %25 testing

%Load the scaled feature data set
load featuresB_scaled.mat;
features=featuresB_scaled;

%Load the target output file for the different classes
load out_basicB.mat;
output=out_basicB;

%Selecting the training and testing data set randomly
num_of_training = round(.75*length(features(:,1)));
num_of_testing = length(features(:,1)) - num_of_training;
sam_number = [1:length(features(:,1))]';
alldata_scaled = [sam_number features output];

rand('seed',7171751);

for i=1:num_of_training
   cur_index = randint(1,1,[1 length(features(:,1))]);
   training_data = [training_data;alldata_scaled(cur_index,:)];
end

for i=1:num_of_testing
   cur_index = randint(1,1,[1 length(features(:,1))]);
   testing_data = [testing_data;alldata_scaled(cur_index,:)];
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Normalizing the trining data set
x=training_data(:,2:38)';
p=x;
perexplained=0.01;
      [pn,meanp,stdp] = prestd(p);
x=pn';
ninput=size(x(1,:));
y=training_data(:,39);
```

```
nsample=length(y);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Normalizing the testing data set
xtest=testing_data(:,2:38)';
ptest=xtest;
[pntest,meanp,stdp] = prestd(ptest);
xtest=pntest';
ytest=testing_data(:,17);
nsample1=length(ytest);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Class = knnclassify(Sample, Training, Group) classifies the rows of the data matrix
%Sample into groups, based on the grouping of the rows of Training. Sample and
%Training must be matrices with the same number of columns. Group is a vector whose
%distinct values define the grouping of the rows in Training.

[predy]=knnclassify(xtest,x,y);

%creating confusion matrix
cm=confmat(ytest,predy)/nsample1
cm1=confmat(ytest,predy)

%accuracy of prediction
Accuracy=(cm(1,1)+cm(2,2)+cm(3,3)+cm(4,4))*100

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

%Matlab Chapter 5-7: Linear Discriminate Classifier

```
%Also includes normalization of the dataset

%%%%%%%%%%%%%%%%Program for LDA%%%%%%%%%%%%%%%%%%%
clear;
clc;
%generation of training and testing data
training_data = []; %75 training
testing_data = []; %25 testing

%Load the scaled feature data set
load featuresB_scaled.mat;
features=featuresB_scaled;

%Load the target output file for the different classes
load out_basicB.mat;
output=out_basicB;

%Selecting the training and testing data set randomly
num_of_training = round(.75*length(features(:,1)));
num_of_testing = length(features(:,1)) - num_of_training;
sam_number = [1:length(features(:,1))]';
alldata_scaled = [sam_number features output];

rand('seed',7171751);

for i=1:num_of_training
   cur_index = randint(1,1,[1 length(features(:,1))]);
   training_data = [training_data;alldata_scaled(cur_index,:)];
end

for i=1:num_of_testing
   cur_index = randint(1,1,[1 length(features(:,1))]);
   testing_data = [testing_data;alldata_scaled(cur_index,:)];
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Normalizing the training data set
x=training_data(:,2:38)';
p=x;
perexplained=0.01;
      [pn,meanp,stdp] = prestd(p);
x=pn';
ninput=size(x(1,:));
y=training_data(:,39);
```

```
nsample=length(y);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Normalizing the testing data set
xtest=testing_data(:,2:38)';
ptest=xtest;
[pntest,meanp,stdp] = prestd(ptest);
xtest=pntest';
ytest=testing_data(:,17);
nsample1=length(ytest);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [CLASS,ERR] = CLASSIFY(SAMPLE,TRAINING,GROUP) classifies each row of
%the data in SAMPLE into one of the groups in TRAINING.  SAMPLE and TRAINING
%must be  matrices with the same number of columns.  GROUP is a grouping variable
%for TRAINING or the target output value. Err is the misclassification rate

[predy,erry]=classify(xtest,x,y);

%creating the confusion matrix
cm=confmat(ytest,predy)/nsample1
cm1=confmat(ytest,predy)

%accuracy of prediction and classification
Accuracy=(cm(1,1)+cm(2,2)+cm(3,3)+cm(4,4))*100

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

%Matlab Chapter 5-8: Fuzzy c-mean clustering (FCM)

```
%%%%%%%%%%%%%%%Program for FCM%%%%%%%%%%%%%%%%%%%%
clear;
clc;

%Load the data
load featuresB.mat
data=featuresB;
rand('seed',717171);

% [CENTER, U, OBJ_FCN] = FCM(DATA, N_CLUSTER) finds N_CLUSTER number
%of clusters in the data set DATA. DATA is size M-by-N, where M is the number of
% data points and N is the number of coordinates for each data point. The coordinates for
%each cluster center are returned in the rows of the matrix CENTER. The membership
%function matrix U contains the grade of membership of each DATA point in each
%cluster. The values 0 and 1 indicate no membership and full membership respectively.
%Grades between 0 and 1 indicate that the data point has partial membership in a cluster.
%At each iteration, an objective function is minimized to find the best location for the
%clusters and its values are returned in OBJ_FCN

[center,U,obj_fcn] = fcm(data,2);

maxU = max(U);
% Find the data points with highest grade of membership in cluster 1
index1 = find(U(1,:) == maxU);

% Find the data points with highest grade of membership in cluster 2
index2 = find(U(2,:) == maxU);

 c1(1:length(index1))=1;
 c2(1:length(index2))=2;
 ind1=[index1' c1'];
 ind2=[index2' c2'];

 tot=[ind1;ind2];
 q=sortrows(tot);

 %predicted outputs by FCM
 predy=q(:,2);

%load target output
load out_basicB.mat;
s=out_basicB;
nsample=length(data);
```

```
%creating conusion matrix
cm=confmat(s,predy)
cm1=confmat(s,predy)/nsample

%accuracy of prediction
accuracy=(cm1(1,1)+cm1(2,2))*100

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Appendix C

Developed LS-DYNA Input file

```
*KEYWORD
*TITLE
LS-DYNA keyword deck by LS-PRE
*CONTROL_ACCURACY
$#     osu       inn    pidosu
       1         1
*CONTROL_DYNAMIC_RELAXATION
$# nrcyck     drtol    drfctr    drterm    tssfdr    irelal     edttl    idrflg
     250  0.001000  0.995000     0.000     0.000         0  0.040000
*CONTROL_TERMINATION
$# endtim    endcyc     dtmin    endeng    endmas
 0.120000
*CONTROL_TIMESTEP
$# dtinit    tssfac      isdo    tslimt     dt2ms      lctm     erode     ms1st
   0.000  0.500000         2
$#  dt2msf   dt2mslc
   0.000         0
*DATABASE_NODOUT
$      PID     SECID       MID     EOSID      HGID      GRAV    ADPOPT      TMID
$#      dt    binary
 1.0000E-5         1
*DATABASE_BINARY_D3PLOT
$#      dt      lcdt      beam     npltc
 5.0000E-4
$#   ioopt
       0
*DATABASE_EXTENT_BINARY
$#   neiph     neips    maxint    strflg    sigflg    epsflg    rltflg    engflg
       0         0         3         1         1         1         1         1
$#  cmpflg    ieverp    beamip     dcomp      shge     stssz    n3thdt   ialemat
       0         1         0         1         1         1         2         1
$# nintsld
       1
*DATABASE_HISTORY_NODE_SET
$#     id1       id2       id3       id4       id5       id6       id7       id8
       5
*BOUNDARY_NON_REFLECTING_2D
$#    nsid
       2
       3
*BOUNDARY_SPC_NODE
$#     nid       cid      dofx      dofy      dofz     dofrx     dofry     dofrz
```

```
   355216       0       1       1
*BOUNDARY_SPC_SET
       1       0       1
*SET_NODE_LIST_TITLE
LB
$     SID      DA1      DA2      DA3      DA4
       1
$    NID1     NID2     NID3     NID4     NID5     NID6     NID7     NID8
*LOAD_NODE_POINT
$#     nid      dof     lcid       sf      cid       m1       m2       m3
       1        2        1  1.000000
*PART
$# title
MISS1
$     PID    SECID      MID    EOSID     HGID     GRAV   ADPOPT     TMID
$#    pid    secid      mid    eosid     hgid     grav   adpopt     tmid
       1        1        1
*SECTION_SHELL_TITLE
section
$#   secid    elform      shrf       nip     propt   qr/irid     icomp     setyp
       1       15  0.850000         4         1     0.000         0         1
$#      t1       t2       t3       t4     nloc     marea
    0.000     0.000     0.000     0.000        0     0.000
*MAT_ELASTIC_TITLE
material
$#     mid        ro         e        pr        da        db  not used
       1  0.002340 45000.000  0.250000
*DEFINE_CURVE_TITLE
LCur_1
$    LCID      SIDR       SFA       SFO      OFFA      OFFO    DATTYP
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp
       1         0  1.000000  1.000000
$                A1                O1
$#               a1                o1
*SET_NODE_LIST_TITLE
RB
$     SID      DA1      DA2      DA3      DA4
$#     sid      da1      da2      da3      da4
       2
$    NID1     NID2     NID3     NID4     NID5     NID6     NID7     NID8
$#    nid1     nid2     nid3     nid4     nid5     nid6     nid7     nid8
*SET_NODE_LIST_TITLE
BB
$     SID      DA1      DA2      DA3      DA4
$#     sid      da1      da2      da3      da4
       3
```

```
$    NID1     NID2     NID3     NID4     NID5     NID6     NID7     NID8
$#   nid1     nid2     nid3     nid4     nid5     nid6     nid7     nid8
*SET_NODE_LIST_TITLE
TB
$     SID      DA1      DA2      DA3      DA4
$#     sid      da1      da2      da3      da4
       4
$    NID1     NID2     NID3     NID4     NID5     NID6     NID7     NID8
$#   nid1     nid2     nid3     nid4     nid5     nid6     nid7     nid8
*SET_NODE_LIST_TITLE
STB
$     SID      DA1      DA2      DA3      DA4
$#     sid      da1      da2      da3      da4
       5
$    NID1     NID2     NID3     NID4     NID5     NID6     NID7     NID8
$#   nid1     nid2     nid3     nid4     nid5     nid6     nid7     nid8
       2       10       18       26       34       42       50       58
      66       74       82       90       98      106      114      122
     130      138      146      154      162      170      178      186
     194      202      210      218      226      234      242      250
     258      266      274      282      290      298      306      314
     322      330      338      346      354      362      370      378
     386      394      402      410      418      426      434      442
     450      458      466      474      482      490      498      506
     514      522      530      538      546      554      562      570
     578      586      594
*DAMPING_GLOBAL
$#   lcid   valdmp      stx      sty      stz      srx      sry      srz
       0  8.595253
*DAMPING_PART_STIFFNESS
$#     pid     coef
       1 1.9558E-5
*ELEMENT_SHELL
$   EID    PID   NID1   NID2   NID3   NID4
$#   eid    pid     n1     n2     n3     n4
*NODE
$   NID             X             Y             Z      TC      RC
$#   nid             x             y             z      tc      rc
*END
```

Appendix D: Contour Plots of different sources

Ricker Source

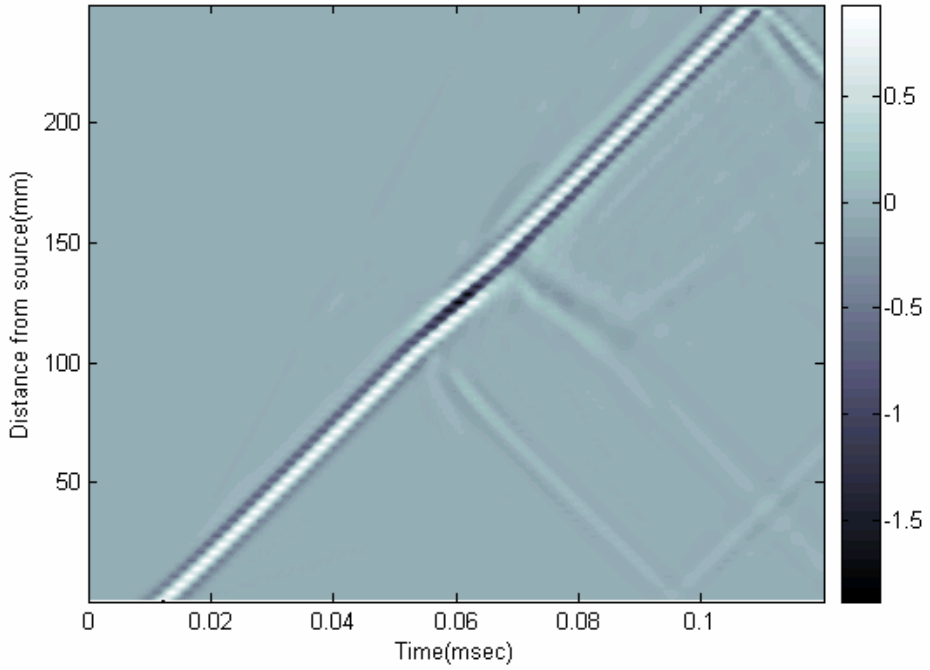


Figure A.1: Contour plot of the normalized vertical responses along the surface of the model in the presence of a small void (Ricker)

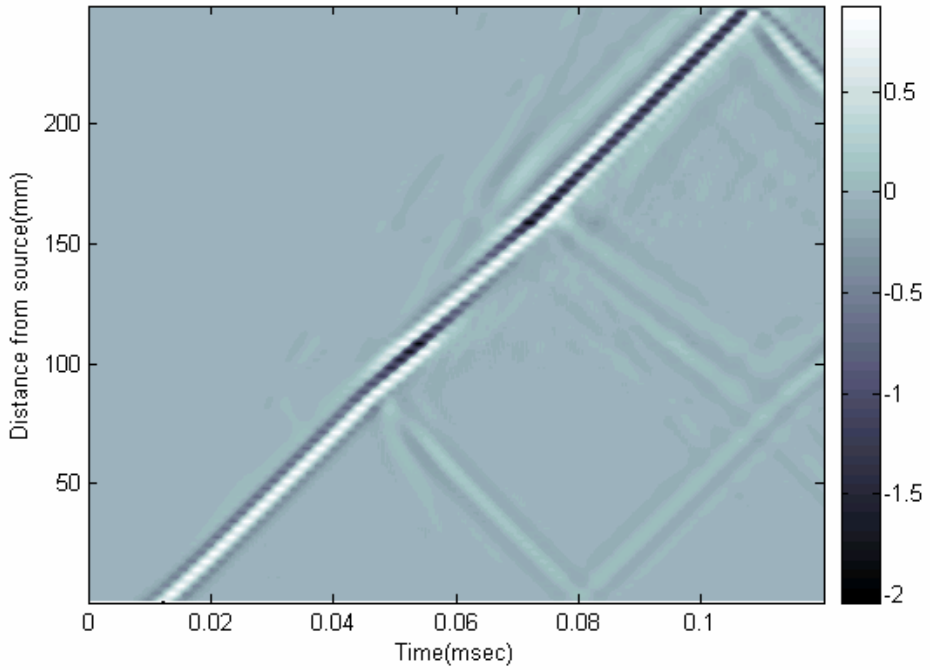


Figure A.2: Contour plot of the normalized vertical responses along the surface of the model in the presence of a medium void (Ricker)

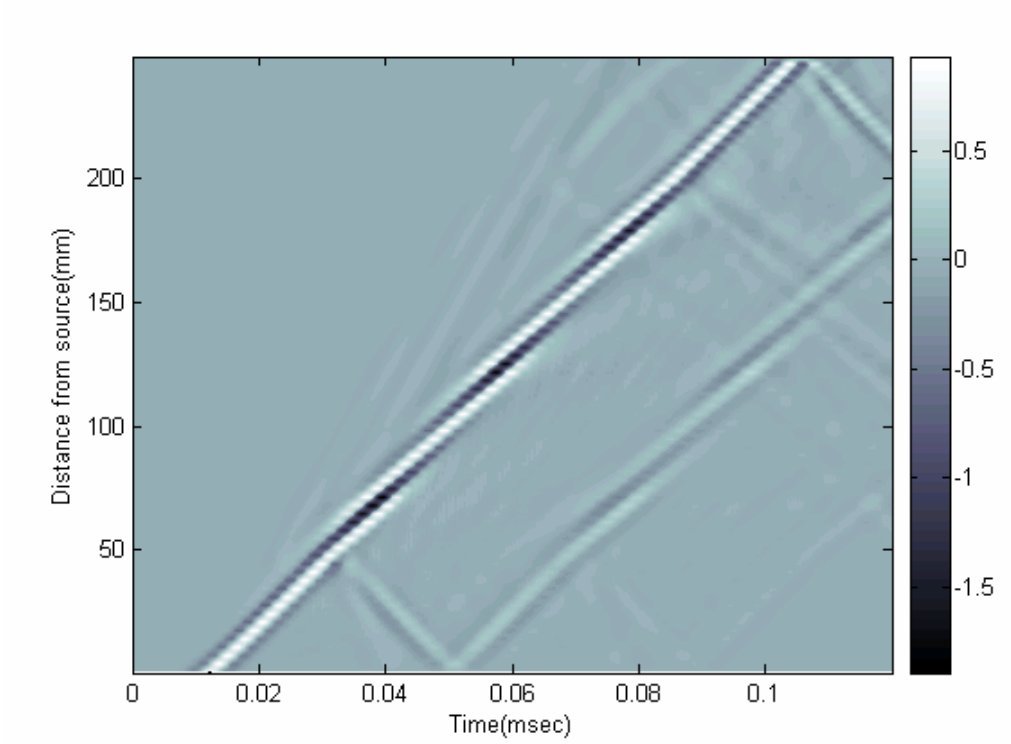


Figure A.3: Contour plot of the normalized vertical responses along the surface of the model in the presence of a large void (Ricker)

Impact Source

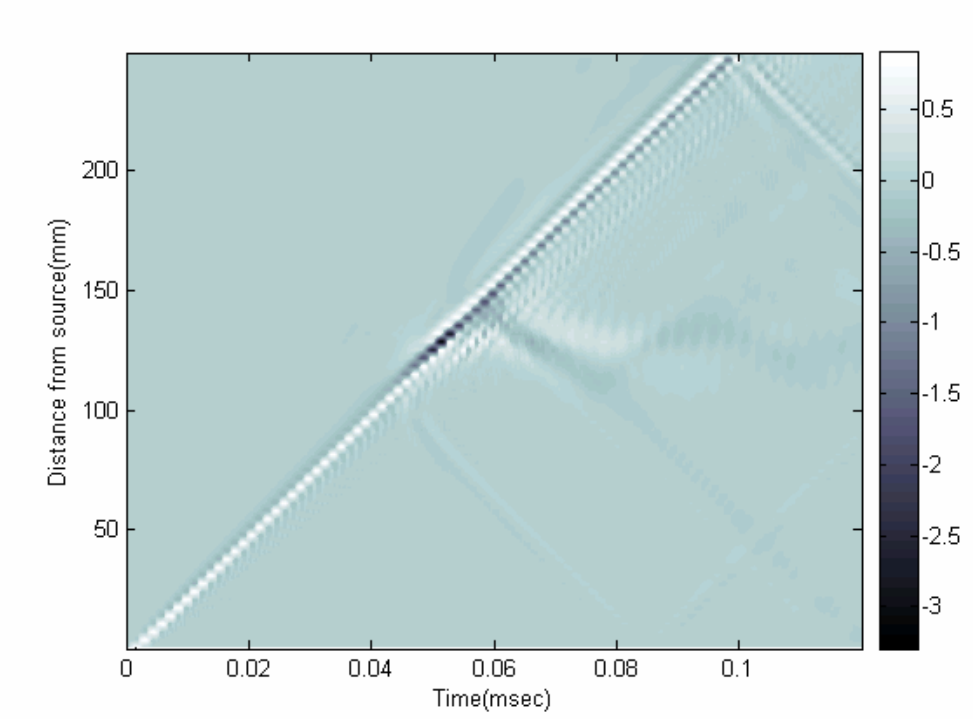


Figure A.4: Contour plot of the normalized vertical responses along the surface of the model in the presence of a small void (Impact)

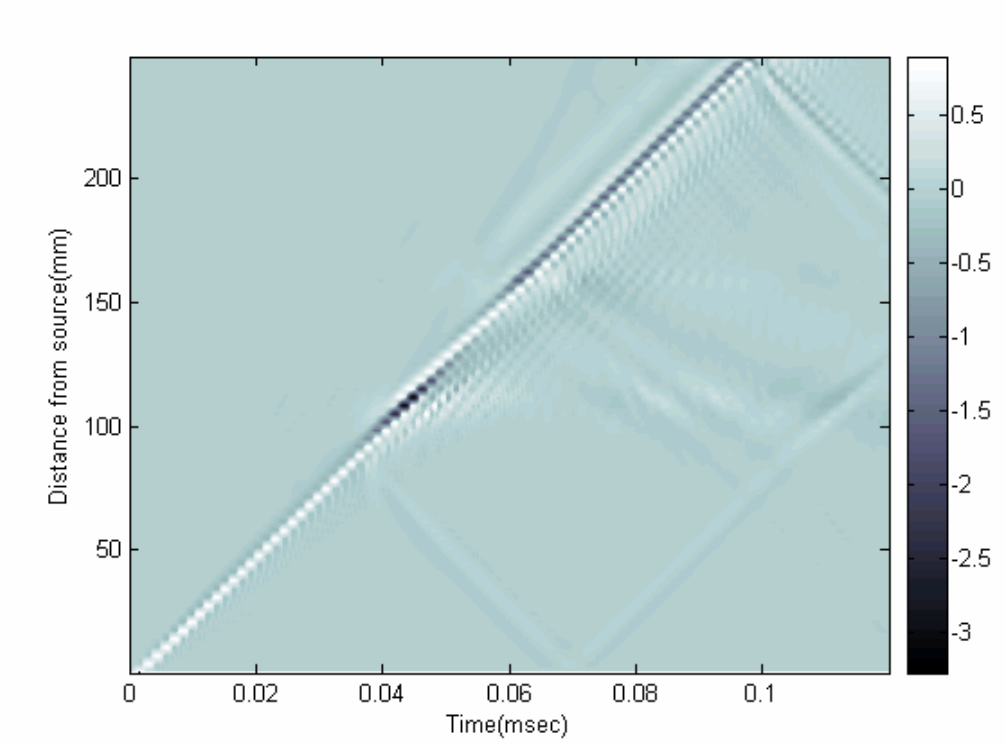


Figure A.5: Contour plot of the normalized vertical responses along the surface of the model in the presence of a medium void (Impact)

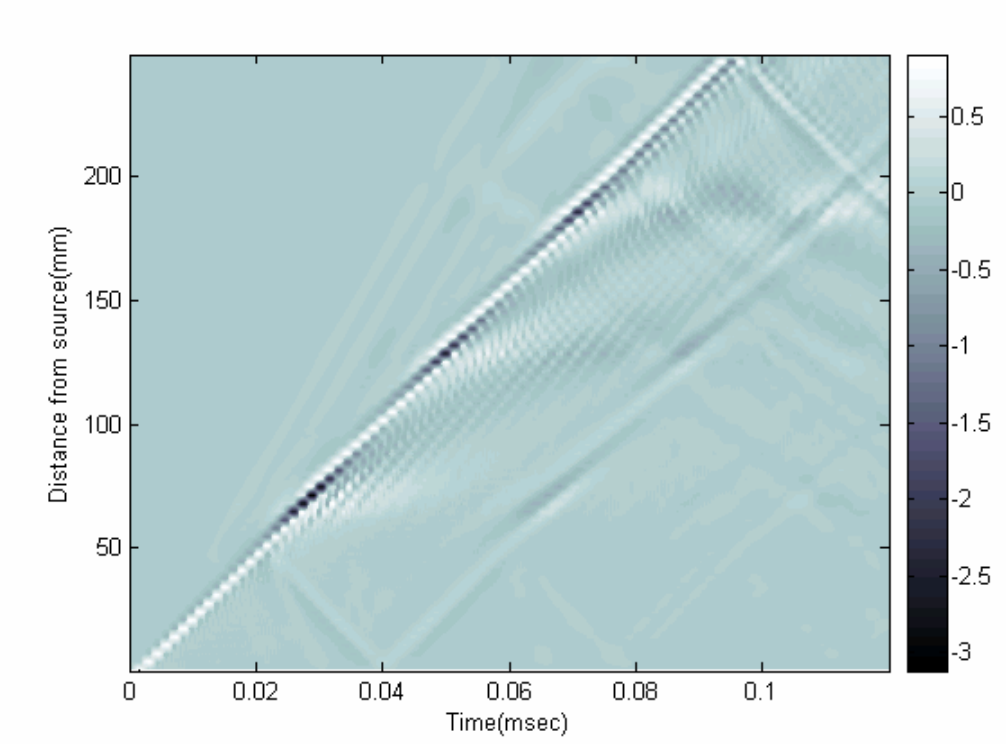


Figure A.6: Contour plot of the normalized vertical responses along the surface of the model in the presence of a large void (Impact)

Decaying Sinusoidal Source

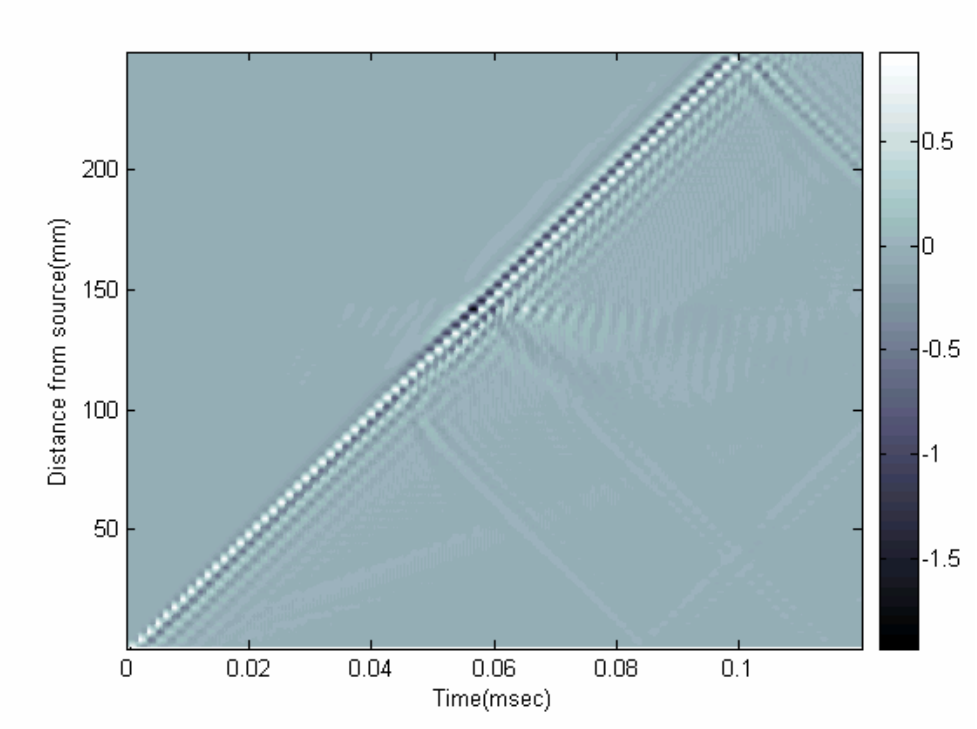


Figure A.7: Contour plot of the normalized vertical responses along the surface of the model in the presence of a small void (Sinusoidal)

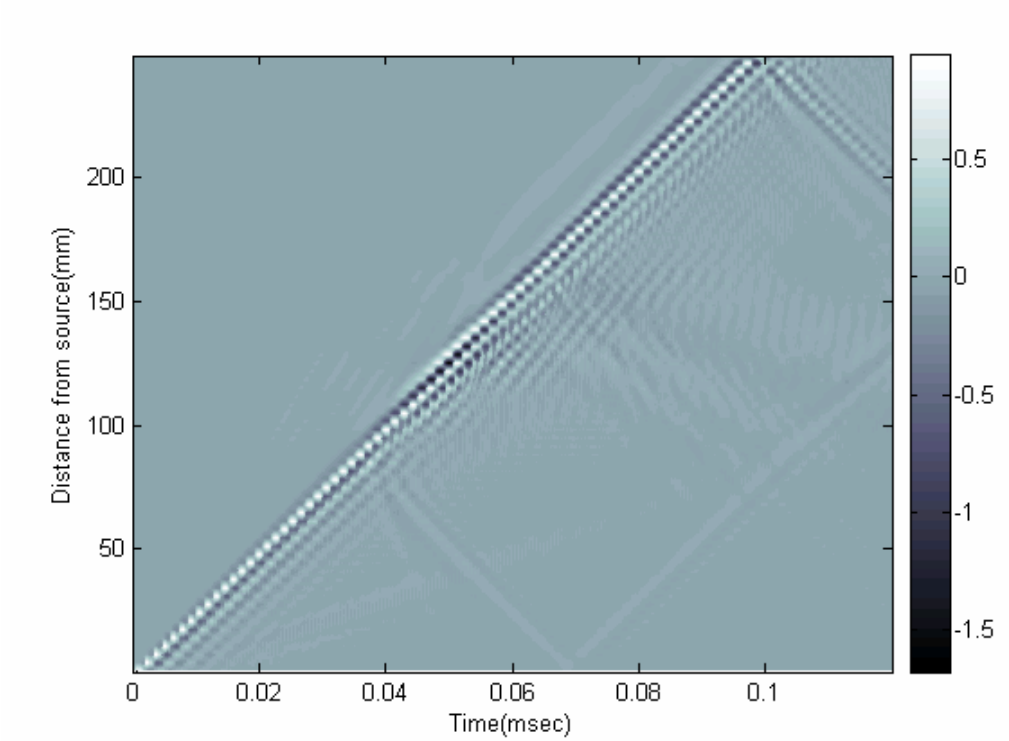


Figure A.8: Contour plot of the normalized vertical responses along the surface of the model in the presence of a medium void (Sinusoidal)

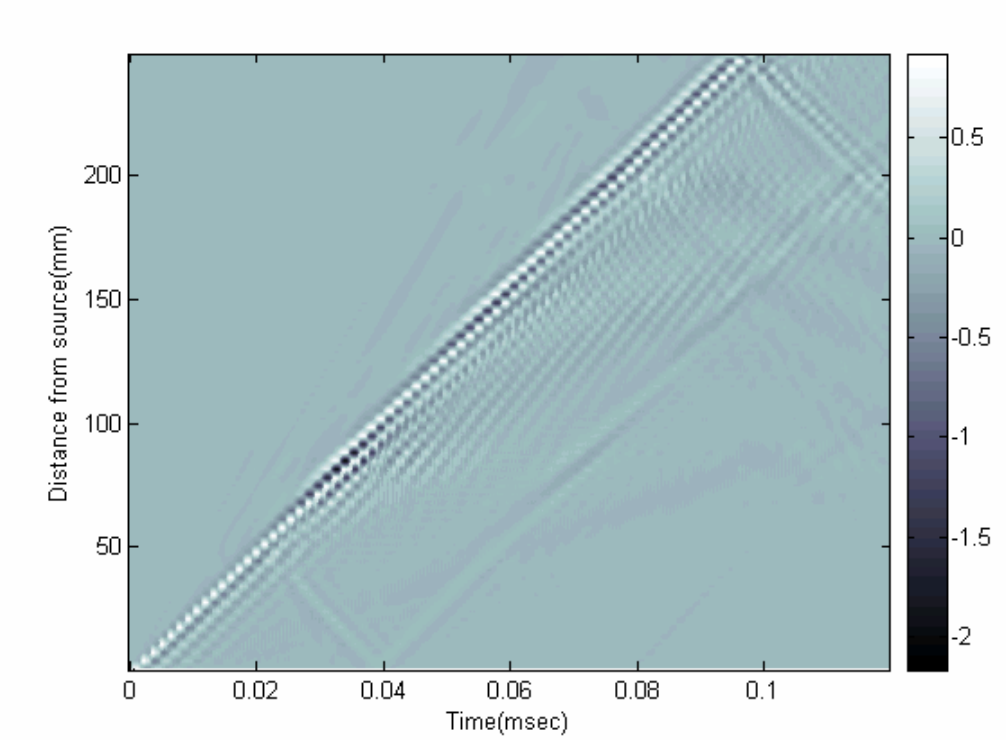


Figure A.9: Contour plot of the normalized vertical responses along the surface of the model in the presence of a large void (Sinusoidal)