LEARNING PARADIGMS FOR THE IDENTIFICATION OF ELASTIC PROPERTIES OF COMPOSITES USING ULTRASONIC GUIDED WAVES

By

Karthik Gopalakrishnan

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering—Master of Science

2020

ABSTRACT

LEARNING PARADIGMS FOR THE IDENTIFICATION OF ELASTIC
PROPERTIES OF COMPOSITES USING ULTRASONIC GUIDED WAVES

By

Karthik Gopalakrishnan

Identification of elastic properties of composites is relevant for both nondestructive materials characterization as well as for in-situ condition monitoring to assess and predict any possible material degradation. Learning paradigms have been well explored when it comes to detection and characterization of defects in safety-critical structures, but are relatively unexplored when it comes to structural materials characterization. In this thesis we propose a learning paradigm that includes the potential use of Machine Learning (ML) and Deep Learning (DL) algorithms to solve the inverse problem of material properties identification using ultrasonic guided waves. The propagation of guided waves in a composite laminate is modelled using two different modelling techniques as part of the forward problem. Here, we use the two fundamental modes of guided waves, i.e. the anti-symmetric (A0) and the symmetric modes (S0) as features for the proposed learning models. As part of the inverse problem, different learning models are used to map feature space to target space that consists of the material properties of composites. The performance of the algorithms is evaluated based on different metrics and it is seen that the networks are able to learn the mapping and generalize well to unseen examples even in the presence of noise at various levels. Overall, we are able to develop a complete framework consisting of many interlinking data processing algorithms that can effectively estimate and predict the material properties of any given composite.

ACKNOWLEDGEMENTS

I would like to first acknowledge my advisor, Dr Yiming Deng, for his unwavering support and guidance throughout this journey. The amount of freedom he gave me in pursuing research in general was fantastic, and I have been able to grow my scientific temper greatly working under him. His work ethic and approach to science inspires me and will certainly guide me in the future. I would also like to place my deepest thanks to the members of my committee: Dr Lalita Udpa and Dr Mi Zhang for always being available and guiding me throughout this journey. In particular, Dr Lalita Udpa has been of immense help to me in the last couple of years. Apart from the invaluable technical inputs that I have received from her, she has always gone out of her way looking out for me and making sure I am doing well. I really appreciate and am grateful for the opportunity of working and interacting with her on a daily basis.

I would also to thank all the members of the Nondestructive Evaluation Laboratory (NDEL). Having a large research group at NDEL has been very beneficial and has led to many productive and fun conversations. In particular, I would like to thank Srijan, Subrata, Bharath, Vivian, Guillermo and Vivek for the banter, and the various conversations we have had ranging from football, food and what not. Without them, going to the lab would have become routine and mundane. I would also like thank all the faculty members associated with the NDEL who have in some way or other enriched my experience here.

When I initially moved to Michigan State University, I had the opportunity to work with Dr Sunil Kishore Chakrapani for a year. I would like to express my deepest of gratitude towards him for placing his trust on me when I was just a fresh undergraduate from India. I learnt a great deal in conducting rigorous experimental approach under him, and also learnt a lot on writing good scientific documents. I am inspired by his approach to science and research in general, and I will always carry his teachings and his advices with me. I also want to take this opportunity to thank Dr Shankar Balasubramaniam for always being available and giving me some valuable advice on choosing my career path.

I have had the pleasure of being a teaching instructor for two different courses during my time here. I would like thank Dr Gregory Wierzba and Dr Sunil Kishore Chakrapani for being wonderful supervisors, and always letting me perform my duties without a lot of interference, but at the same time always being available in case of any trouble. I also took many interesting courses during my time here, and I would like to thank all the teachers who thought the courses for their efforts and endeavors.

Also, big thanks to all the staff at the Dept. of Electrical and Computer Engineering. In particular, I am eternally grateful to Meagan, I have troubled her multiple times with multiple different requests and she has always solved it for me in a very timely manner and with a smile on her face. I also want to thank Laurene, for always processing my time sheets on time even if I submit it a day late. Big thanks to Brian, for always being available for any technical help or requests that I have had over the last two years. People like them are the unsung heroes of every department, and a lot of it wouldn’t be possible if not for their timely help and services.

Moving on to the academic collaborators, I would like to first thank Magna International Inc. I was fortunate to work on a very interesting project funded by them, where I learnt the basics of adhesion and adhesion bonding in materials. I also worked on a project funded by the US Department of Transportation, where I worked on evaluating the integrity of oil and gas pipelines. As part of this work, I also had the opportunity to present my work in an R&D seminar held by them. I would like to thank them for providing me with such an amazing opportunity. Finally, I would like to place my biggest thanks to Mr Mahindra Rautela, who has greatly helped me with my work that has eventually become my dissertation. A lot of the work that I have presented in my thesis is in collaboration with him, and I have enjoyed working with him all through. His drive, passion and zeal towards research and technology are something that I am inspired by and I’ll take along with me to the future.

Before moving here, I have interacted with lot of different people that have helped me shape my career. In particular, I would like to thank Dr G K Ananthsuresh, and Dr Suresh Sundaram for taking me under their wing at IISc and NTU respectively, when I was just a curious undergraduate trying to understand what research was all about. I also want to thank Dr Senthil Jayavelu, who helped me with a lot of my work for my undergraduate thesis that really helped me develop a firm interest in the area of Deep Learning. I also want to thank all my teachers at the Amrita University, Bangalore Campus from where I obtained my undergraduate degree.

Moving outside the academic sphere, I have had the pleasure of meeting some wonderful people who have all been a big part of my graduate life. Cricket is a big part of who I am, and I have had the opportunity of playing a good level of cricket at Michigan. I would like to thank the entire cricket family here at Michigan for giving me an opportunity to have something else to look forward to in life. I also want to thank the entire *Desi Boys* group: Sai Chaintanya, Ankit, Hitesh, Ronak, Bharath, Yash, Abhiroop, Siddartth and Abu Bakr for their company through the last two years. A lot of my weekends went by chilling with them over food and drinks, or by watching random movies or playing board games.

Moving to MSU, I was able to meet one of my best friends in my life, Sejal. I am thankful for all the amazing times we have spent together. From watching movies, cooking meals to playing and fighting over board games, I want to thank you for your companionship and for always being there no matter what. I would also like to thank Manali, who along with Sejal accompanied me on multiple trips all across Michigan. I also want to thank all my friends back home in India, who have all indirectly been a part of this wonderful journey.

Finally, but most importantly I am eternally grateful to my parents, Gopalakrishnan and Anuradha. Your support has been unwavering, and I am who I am because of you and your nonstop consistent support. I am very fortunate to have a dad who is himself a revered professor. I want to thank him for always taking out time to review my presentations, reports, papers and always giving me feedback to improve it. I want to thank my mom for always willingly taking in all my rants and being a big emotional support. I would like to thank my sister Keerthana, for always being the sibling whose leg I could always pull. Nothing would have been possible if not for you guys. Thanks for everything!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The idea of structural design has undergone a radical change. Due to the continuous technological advances achieved by the human mankind, lighter and sleeker structures have replaced the conventional bulky and heavy structures. But newer solutions always bring with it a newer set of problems that requires solving. The new age structures have introduced severe constraints on design methodologies that are currently in practice [1], which requires newer techniques to monitor and assess the integrity of structures. *Structural Health Monitoring* (SHM) is one such technology, which can provide vital information on the state of the structure at any given time. Combined with the powerful computational and signal processing tools available, one can not only precisely determine the health of any structure, and classify and characterize defects that affect the structural integrity, but also predict the future performance of the structure over time.

1.1. Overview on Structural Health Monitoring

Structural Health Monitoring is the process of evaluating the health state of a structure and predicting its remaining life. This process often involves the constant observation of a system over time using periodically sampled response measurements from a sensor actuator system, extraction of healthy and damage sensitive features from these measurements and the statistical analysis of the these features to establish the state of the system health, and then further predict the remaining life of the structure.

All structures like bridges, aircrafts, and pipelines have a finite lifetime, and begin to deteriorate when put into service. Due to a combination of environmental, material and other effects, processes such as corrosion, fatigue, erosion, overloads and general wear and tear degrade them until they are no longer useful. In worse cases, they can lead to fatal damage that can endanger people’s lives, their livelihoods and the environment in general. It is therefore essential to examine structures of importance periodically, and determine whether or not remedial action is needed. SHM serves as an early warning system and helps in resolving the health of the structure before they can progress to cause potential damage.

Structural design has undergone many changes across years, where stringent restrictions are placed on design parameters to produce the most efficient structures having superior structural integrity. Such structures are generally geometrically optimized to guarantee their resistance to sustain the high design loads. However, they are more susceptible to small damages such as horizontal, vertical or inclined cracks, corrosion in metallic structures, and delamination, fibre breakages in the case of composites. These defects severely affect the structural health, and therefore need consistent monitoring.

SHM can be viewed as a generalized tool box that has the objective of providing necessary techniques for the constant/periodic monitoring of structures. These techniques are specifically designed for the various materials used in critical structures like buildings (concrete), bridges (metal) and aircrafts (composites/metals) etc. SHM therefore has innumerable applications in varied disciplines including aerospace, mechanical and civil engineering [2] [3]. SHM potentially offers increased safety, since faults cannot grow to a dangerous level, and avoids the vagaries of human behaviour. The benefits of SHM can be mainly categorized as:

- To use structures to their optimal best, a minimized downtime, and avoiding fatal failures.
- To give designers an improvement on their products.
- Minimization of human involvements, and therefore possible human errors. This directly improves safety and reliability.

1.1.1. Basic Elements of SHM Systems

A typical system contains both hardware and software elements. The hardware elements basically are the sensor actuator setup and its associated instrumentation, while the software components can vary but generally compromises of damage modelling and damage characterization algorithms. Sensors may be active or passive. Passive sensor like strain gauges only sense (receive) while active sensors transmit and receive. Commonly used sensors include *Poly Vinyl Di-Fluoride* (PVDF) sensors, Piezoceramic sensors made from *Lead Zirconate Titanate*, commonly known as PZT sensors or fibre optic sensors. The responses obtained from the sensors vary sensor to sensor, but typically, they are all time histories of a certain variable. It is important to note that SHM is a time dependent process, and any sensor should be able to monitor a parameter/variable over time. The most common response often received from a PZT or a PVDF sensor, is the voltage history. These responses are post processed and manipulated to extract healthy and damage features to effectively characterize the health state of the structure.

SHM compromises of two main components, the *Diagnosis* and the *Prognosis. Diagnosis* aims to, at every instant of time give a procedure to determine the health of the structure while *Prognosis* involves computation of the severity of the defects detected during *diagnosis* in terms of fracture mechanics parameters, and derive the structures residual life. *Diagnosis* normally gives information about the onset of damages like cracks, its location and geometric parameters. Considering only the diagnosis component, SHM can be described as a new and improved way of performing *Nondestructive Evaluation* (NDE). By combining integration of sensors, smart materials, computational modelling, data transmission and processing ability, it extends the traditional NDE approach to reconsider the structure design and its lifetime management. An alternate view of SHM is that of a discipline combining the following four subjects [3] (see Figure 1.1). In general, an SHM system encompasses the following components:

- Test structure/ simulation Model
- Sensors
- Data acquisition systems
- Signal processing algorithms
- Damage modelling and classification algorithms
- Data transfer, handling, management and storage mechanism

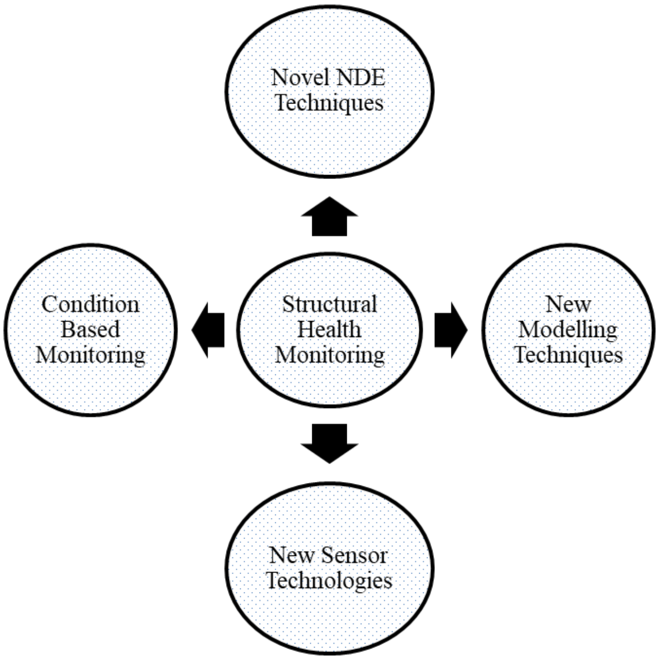


Figure 1.1 Elements of Structural Health Monitoring

Modelling is a critical part of any SHM framework. The measured response isn't useful if not for powerful and robust post processing algorithms that converts raw data to meaningful information. Common modelling techniques employ the *Finite Element Method* (FEM), which is very adept at modelling complex geometries. FEM requires the use of very fine meshes to detect and characterize very small defects, which results in very high computation time and cost. An example of using FEM for SHM can be found in [4].

Therefore, for characterizing smaller defects more efficiently, a suitable mathematical model should be based on the physics of wave propagation, and one of the most well exploited models is the *Spectral Finite Element Method* (SFEM) [5].

Modelling has mainly two components, namely the flaw modelling, and damage detection algorithms. In metallic structures, some of the most common types of defects are caused by pitting corrosion. Modelling pitting corrosion is relatively easy, and has been extensively studied in [6][7][8]. Another common type of defect in metallic structures is the horizontal/vertical cracks which are sometimes through thickness. Damage modelling in composites is whereas a challenging task due to many different types of failure modes. Some of the commonly occurring modes are delamination, fibre breakage, matrix cracks and debonds. In addition, composites are prone to moisture absorption due to its highly porous nature because of unavoidable errors during manufacturing. Hence, one needs to be able to come up with simplified mathematical models to describe the various types of flaws that commonly occur in critical structures. The second aspect of modelling is in devising robust damage characterization algorithms. These algorithms should be able to easily detect defects, distinguish healthy to faulty samples, and be able to clearly extract the damage features from a SHM response. The features should directly or indirectly give information about the state of the health of the structure. There is always a lot of noise present when conducting field experiments and these algorithms should be able to work well even in the presence of noise.

1.1.2. Levels of Structural Health Monitoring

SHM can be thought as a system identification problem. Through diagnosis, one can get information about any defect or anomaly present and their characteristics, and the prognosis uses the information obtained from diagnosis and determines the residual life of the structure. We can broadly divide SHM to five levels [2].

- **Level 1**: Detecting the damage i.e. being able to distinguish healthy and faulty responses.
- **Level 2**: Defect location and geometry
- **Level 3**: Severity of the damage
- **Level 4**: Damage control i.e. possibility of controlling or delaying the growth of damage.
- **Level 5**: Determining the residual life.

The first four levels effectively constitute the diagnosis component of SHM, while Level 5 is effectively prognosis. Level 1 SHM is relatively easy to achieve, and it can be achieved by using passive sensors to monitor parameters such as strain energy, fundamental natural frequency, phase information, stiffness reduction over time. The most common method though is using natural frequencies. As damage reduces stiffness, it induces changes in the natural frequencies. Comparing to the baseline fundamental frequency should confirm the presence of damage, and act as a reliable feature to isolate faulty samples.

Level 2 SHM is relatively harder the level 1, as from the known input and the measured SHM response, it is necessary to determine the location and orientation of the flaw. A simple way is capturing the response at some known location. This response will contain the reflected energy packet coming from the flaw, knowing the speed of the wave in the medium, and the time of arrival of the reflected pulse, we can effectively locate the flaw. This is easier said than done, as the reflection from the flaw will mostly be very small in amplitude, and can easily be buried as noise or as part of the reflected wave packets. This makes the designing of the detection algorithms challenging. An ideal algorithm would be one which works accurately without a baseline response.

Once the damage characteristics are evaluated in Level 2, it is important to determine the severity of the damage. For example, if the damage is a crack, then we need to estimate the *Stress Intensity Factor* (SIF) or *Strain Energy Release rate* (SERR). If these parameters reach a pre-defined threshold value, these cracks will grow. Now, if the defect is found to be severe, Level 4 SHM deals with the immediate measures to be taken to arrest the growth of cracks. Level 5 SHM is closely related with level 4 SHM, wherein the estimation of fracture parameters is used to perform fatigue life analysis to determine the residual life. The analysis is mainly statistical in nature and generally encompasses novel data processing techniques.

It is to be noted, the knowledge of material properties of the structure under test is critical to perform or achieve any level of SHM. For example, in the most basic and probably easiest level i.e Level 1 involves detecting possible defects. This is normally done by monitoring the material properties of the structure periodically over time. This requires an user to know the material properties of the structure before it's in service, and also periodically over time. The following levels of SHM are all based on the results of the detection process in Level 1. Therefore, the material characterization of the structure is the fundamental and most important task that needs to be carried out in any SHM framework or technique.

1.2. Forward and Inverse Problems in NDE/SHM

Forward problems typically use known models of the system of interest along with a known input to establish the characteristics of the output response. Inverse problems in most cases meanwhile, use an observed output, along with a known input to estimate the properties of the model. Figure 1.2 below shows the block diagrams of a basic forward and inverse solver.

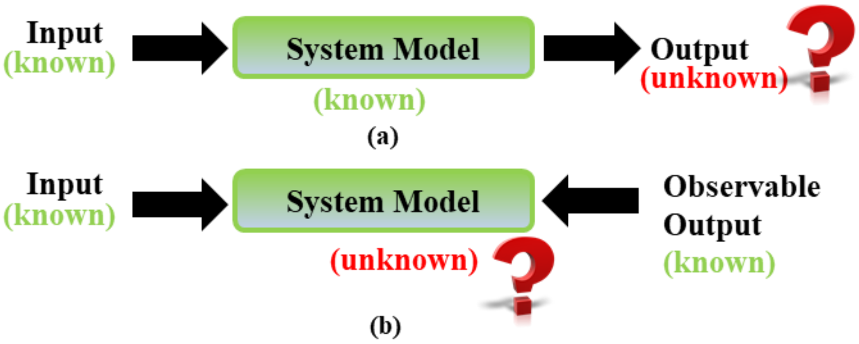


Figure 1.2 (a) Forward Process (b) Inverse Process

In NDE, and more generally in SHM, physics based mathematical models are utilized to describe systems of interest. Some of the common methods used are the FEM and SFEM. Waves excited by different actuating mechanism act as the inputs, and as described before, different type of responses like voltage histories, displacement histories, strain rates are the outputs of a typical NDE system. Forward problems aren't easily solvable in this field, because it is very hard to know the exact physical properties of the system or model; hence one has to reverse engineer using the output response or measurement to gain information about the system in general. In SHM particularly, known inputs and observed outputs needs to be used to establish the physical and material state of the critical structure in question. This essentially is the inverse process, and is a fundamental part of any SHM/NDE framework.

A problem should be well posed in order to be solved. In general, one can describe three conditions for an inverse problem to be well posed [9]. The systems used in SHM typically have distinctive physical and material properties that can be mathematically well defined, and this is necessary to ensure that the problem in hand has a realistic solution. As is the case in solving any engineering related challenges, the solution needs to be unique. For example in Figure 1.3, when *Ultrasonic Guided Waves* (UGW) is used to study pitting corrosion in long pipelines, it is clearly visible that the responses for a healthy and faulty pipe with a very small defect aren't very different. The differences are minimal, and a defect response can easily be misclassified as a healthy one. This means that in order to obtain a unique accurate representation of the system, one needs to collect more data where more informative features can be extracted. Also, the solution must be stable for small deviations in the measured data, or in the presence of noise. In other words, the solution must be robust and should incorporate factors that affect systems in practical situations. But by incorporating more information or measurements, one risks the stability of the system as more data brings in more stochastic variations.

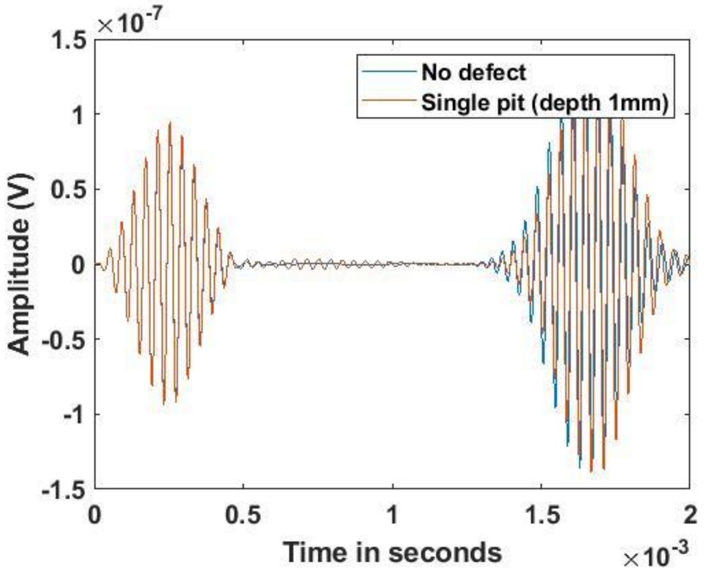


Figure 1.3 Ultrasonic Guided Wave responses for healthy pipeline vs a faulty pipeline with a single 1mm deep corrosion pit

Damage detection and characterization is a complex procedure, and it is impossible to solve these kinds of inverse problems independent of the forward process. Typically, materials characterization forms the crux of almost all damage characterization techniques. Knowledge of the material properties of the structure before being put into service and during its service period is very critical for condition monitoring. Therefore, such problems generally involve both forward predictors and inverse detectors. SHM techniques normally combines the forward solver and the inverse solver to establish robust models, where both forward predictor and inverse detector models maximize their sensitivity to different features of the signals, and at the same time minimize their sensitivity to cofounding factors caused due to the variations in the mechanical, physical and material properties of the system. It can be effectively described as a symbiotic relationship, where information from the forward process is used to improve the models established in the inverse process, and these improved models are then used in the forward process to obtain better measurements. Figure 1.4 shows the inter relationship between the forward and inverse process, and how it can be used to establish efficient models.

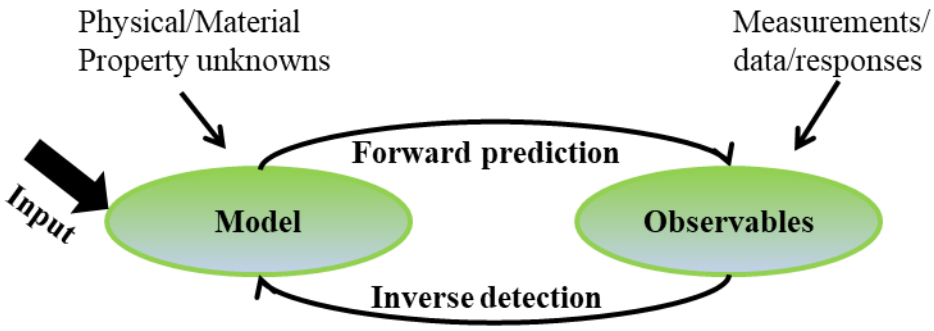


Figure 1.4 Interdependence of the forward and inverse process

1.3. SHM for Composites

Composites are materials made by combining two or more natural or artificial elements with different physical and chemical properties that results in a mechanically superior material compared to its constituent material. Generally, fibre reinforced polymer composites consist of a polymer matrix reinforced with a man-made or natural fibre. Compared with traditional metallic materials, the main advantages of composites are: a) low density and high specific strength and stiffness b) good vibration damping ability, long fatigue life and high wear, creep, corrosion and temperature resistances; b) strong tailoring ability in both microstructures and properties that make them design efficient to satisfy different application needs; c) since detail accessories can be combined into a single cured assembly, the number of required fasteners and the amount of assembly labour can be significantly reduced [12][13]. The most common types of damages in composites are fibre breakage, matrix cracking, fibre-matrix debonding and delamination between plies, most of them which occur through the thickness, and are barely visible. They can severely degrade the performance of a structure and can cause fatal damages at the worst. Due to the above advantages, composites are used in many critical structures across different industries like aviation, automobile etc. and therefore require constant periodic monitoring to guarantee optimum performance. Figure 1.5 below shows a typical setup to perform active SHM of composites.

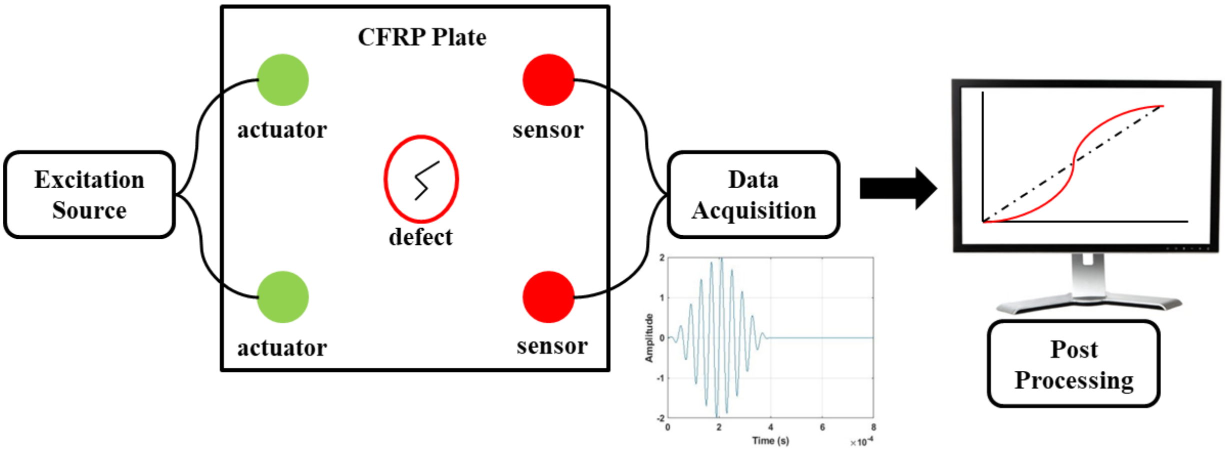


Figure 1.5 Active SHM Setup of a CFRP Composite Plate

1.4. Machine Learning in SHM

Machine Learning (ML) is considered to be the natural evolution of statistical learning. It is one of the major dominant subset of *Artificial Intelligence* (AI). To summarize in a single sentence, ML is the science of developing statistical models that learn with time, and can perform a function/task of interest without necessary human intervention. Most of the current SHM frameworks aim to perform fast in-situ testing. NDE progressively has required a higher level of automatization to handle the huge amount of data generated. This in term means quick processing of the signals obtained after an experiment. Conventional data analysis techniques require expertise and involve manual labour, but with the advent of better computational facilities like GPUs, powerful machine learning based classification and characterization algorithms can be developed. The complexity involved in defect and materials characterization algorithms have been described before. This is mainly due to the fact that, there doesn’t exist a simple functional mapping of the input feature space to the measured output response space. Neural Networks are in other words known as universal function approximators [10[[11][26][27]. Quite simply, they help in mapping the input feature space to the output response space. In general, SHM is an inverse problem that is data intensive and which does not have a unique solution. Hence, ML plays a very important role in mapping output signals to input features to discover important details of the structure. Overall, the benefits of Machine Learning in SHM can be described as:

- Automatization of the SHM framework i.e. performs various levels of SHM without any human intervention.
- Improves the robustness of the algorithms. These algorithms are powerful and work very well even in presence of noise.
- Reliability. Once trained well, the networks can sustain and perform the described process multiple times with the same accuracy.
- Mapping input to output space, thereby gaining vital information about the system parameters
- Efficient and fast post processing of the observed outputs.

1.5. Material Property Identification of Composites

In any Structural Health Monitoring framework, the knowledge of the material properties of the structure under inspection is fundamental. The knowledge of material properties is required before the structure is put into service. Because of the inherent heterogeneity of composites in general, coupled with the large uncertainties expected in the manufacturing process, it becomes extremely difficult to predict the material properties of the final part from those of the individual constituents. Once in service, it is also essential to continuously compute the material properties of the structure periodically to capture possible material degradation. Not only that, by monitoring the material properties over time, potential defects can be identified in the structure by detecting the changes in the material properties over time. Material degradation can cause a structure to fail if the strength falls below a certain threshold. Having efficient material property identifications schemes can potentially help fine tune the manufacturing process to reduce the amount of error that occurs normally. Therefore identification of material properties is not only required for materials characterization but also for in-situ monitoring. The ideal requirements for any material property identification scheme are that the method is non-destructive in nature, and is able to be deployed for real time online in-situ monitoring. The algorithm or framework should also be able to generalize well on any composite.

Any framework involving guided waves requires a frequency selection procedure. Guided waves are multi-modal waves, and multiple modes often occur at higher frequencies relatively. Therefore, dispersion curves are normally used to study the dispersion phenomenon of guided waves in any material. Based on these curves, an optimal frequency is selected for the problem in hand. The dispersion curves which normally is phase/group velocity plotted as a function of frequency is dependent on the material properties of a composite. Hence uncertainties or differences in the material properties of the composite can lead to dispersion curves that are not accurate, and therefore can result is selecting a frequency that is probably not best suited for the problem in hand. Hence due to the above mentioned reason, it is important to have an automated material property identification scheme that is real time and can be potentially used for in-situ monitoring.

1.6. Literature Review

Researchers have used both vibration-based and ultrasonic guided wave (UGW)-based techniques to estimate material properties of a composite [14]. The vibration-based technique is global in nature and is sensitive to boundary conditions which are not suitable for in-situ condition monitoring [15]. The ultrasonic range of the mechanical guided waves is highly sensitive to composite laminate properties which makes them useful for Non-Destructive Evaluations and Structural Health Monitoring applications [16][17][18]. Traditionally, considerable research has been done on using Deep Learning for damage detection, and damage characterization. Rautela et.al. proposed different deep learning frameworks for damage detection in 1D composite waveguides [48]. Here, they have modelled cracks of various sizes using the Spectral Finite Element formulation, and have used the responses obtained from this model to train on different architectures. But the field of materials characterization using Deep Learning is relatively less explored. Rather, the inverse problem of property identification is investigated using different inversion schemes that in some way or other uses global optimization algorithms. Krishnan Balasubramaniam [7] has explored the feasibility of estimating the stiffness constants in a single layer unidirectional composite laminate, and the ply-up sequence of layered laminates. It is solved as a multi-parameter optimisation function, and genetic algorithm was used to find the optimal solution. Bernard Hosten et.al. [8] have used the phase velocities of lamb wave modes along with a Newton-Raphson scheme to estimate the elastic properties of composites by minimizing the Thomson/Haskell matrix.. In this work, they have used air coupled transducers to generate lamb waves that are sensitive to material properties. J Vishnuvardhan et. al. [9] have utilized a genetic algorithm to measure the elastic properties of an orthotropic plate using ultrasonic velocity data. They have used slowness curves to verify the quality of the reconstruction. Ranting Cui et. al. [2] have investigated a property inversion scheme based on matching phase velocity dispersion curves of relevant guided modes (A0, S0, SH0) using Simulated Annealing optimization algorithm along with a metropolis criteria. In this process, a Semi Analytical Finite Element method is formulated to solve the forward problem. This inversion scheme is used to identify the elastic properties of a unidirectional, quasi-isotropic and anisotropic composite laminates.

All the above mentioned inversion schemes are limited in terms of large scale automation, generalization ability, computational time, in-situ predictions and robustness towards the noise. Also there is a research gap when it comes to using Deep Learning

Algorithms for material property identification. In this thesis, we aim to address this research gap by developing learning models within an overall framework that can identify and estimate material properties of composites.

1.7. Thesis Objectives

The scope of this thesis is to establish a comprehensive framework with different learning models and interlinking data analysis algorithms that can be utilized to characterize composite laminates. The propagation behaviour of the UGW in the structure is utilized to reverse engineering and estimate the material properties. Powerful Deep Learning algorithms like *Convolutional Neural Networks* (CNN) and *Long Short Term Memory Networks* (LSTM), along with multi-layer *Dense Neural Networks* (DNN) are utilized for this purpose. The performance of these models is to be evaluated on data both in presence and absence of noise of various levels to verify its robustness. We use slightly different methodologies to collect data and train the deep networks (1DCNN and LSTMs) compared to that of the DNNs. The approach involving the deep networks will be classified as *"DL based"* approach, while the approach involving DNNs will be classified as *"ML based"* approach hereon in this thesis. Indeed, Deep Learning or DL is a subset of Machine Learning (ML), and the demarcation is made to easily differentiate the two different approaches. Once the framework is well established for complex structures like composites, it can be potentially extended to other materials used in critical structures like metals, concrete etc. To summarize, the main objectives of this thesis can be defined based on the two approaches used i.e. DL based approach and ML based approach. The main objectives in the DL based approach are as follows:

- **Develop (Forward Process)**: Develop Numerical Models (FEM and SFEM) to simulate the two fundamental modes of a UGW (A0 and S0) in a unidirectional multi-layered composite laminate.
- **Compare**: Compare and comment on the two different numerical methods used.
- **Creating the dataset**: Use the numerical models in the forward process to obtain the A0 and S0 responses for different sets of material properties representing different composites.
- **Dataset Check**: Sensitivity Analysis, Uniqueness check
- **Training (Inverse Process)**: Use the generated dataset, with A0 and S0 time histories as inputs and the material properties as the output labels to perform supervised regression using different deep learning architectures (1DCNN and LSTM)
- **Prediction**: Once trained, for any unseen A0 and S0 waveforms, the networks should be able to identify and estimate the material properties.
- **Robustness**: Use a model trained on noiseless data and predict on datasets without and with noise of various levels.

The main objectives of the ML based approach are as follows:

- **Dataset (Forward Process)**: Use Dispersion Calculator® software to generate group velocity dispersion curves for different sets of material properties representing different composites.
- **Training (Inverse Process)**: Use the generated dataset, with group velocity curves for A0, S0 and the Shear Horizontal mode SH0 and the material properties as the output labels to perform supervised regression using a simple multi-layer Dense Neural Network.

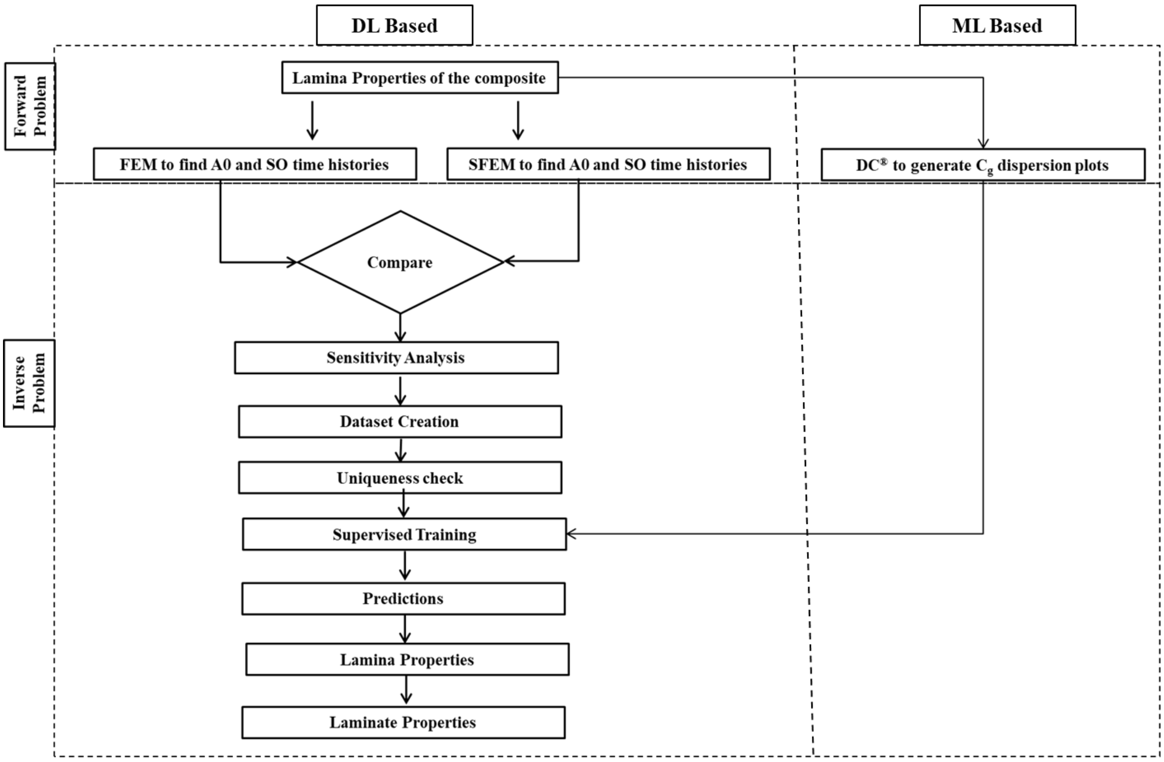


Figure 1.6 Overall Workflow of the Composite Property Estimation Framework

Figure 1.6 shows a flowchart describing the overall workflow of the composite material property identification framework. The organization of the thesis is as follows, Chapter 2 gives a basic introduction to composites and the Composite Laminate Theory (CLT). Chapter 3 then discusses guided waves in general before looking at specific cases of guided wave propagation in composite laminates. Chapter 4 covers the basics of Machine Learning, and describes in detail the Convolutional Neural Network, Long Short Term Memory Networks and the Dense Neural Networks. Chapter 5 describes in detail the two modelling techniques used in this thesis i.e. the Finite Element Method and the Spectral Finite Element Method. Chapter 6 outlines the data analysis algorithms used in this thesis, while Chapter 7 mentions the study parameters for the two different methods used. The results are presented and discussed in Chapter 8, and the thesis is concluded in Chapter 9.

CHAPTER 2

CLASSICAL LAMINATE THEORY

Composites can be broadly classified into three different types: *Fibrous Composites*, *Particulate Composites* and *Laminated Composites*. For structural applications, only the laminated composites are extensively used, and therefore the whole material property identification framework is built considering composite laminates. A composite laminate consists of many layers which are commonly referred to as laminae or plies. They are stacked together in particular order to form structures. Based on the strength and stiffness requirements, the number of plies is decided. Typically, each lamina consists of fibres oriented in a direction where maximum strength is required. The fibres are bonded by a matrix material which is mechanically inferior compared to the fibres. The laminate derives all its strength from the fibres. Some of the commonly used fibres are: Carbon Fibre, Kevlar, and Glass Fibre. The most commonly used matrix material is the Epoxy Resin. Laminated composites are assumed to have orthotropic properties at the lamina level, but when at the laminate level, they exhibit anisotropy. The anisotropic behaviour results in stiffness coupling, such as bending-axial-shear coupling in beams and plates, bending-axial-torsion coupling in aircraft's thin walled structures [19]. Therefore, the material property identification of these types of composites is complex compared to isotropic materials. The material properties of these composites typically can be determined at the lamina level (Micro Mechanics) or at the laminate level (Macro Mechanics). Section 2.1 below outlines the micro mechanical analysis while Section 2.2 describes the macro mechanical analysis.

2.1. Micro Mechanical Analysis

A 2D transversely isotropic lamina requires determination of 6 properties namely, Elastic modulus in two coordinate directions (E_1, E_2), Poisson's ratio's (ν_{12} and ν_{23}), shear modulus (G_{12}) and density (ρ). These properties are determined using the properties of the fibre and the binding matrix. The lamina and the derived laminate strength are strongly influenced by the type of fibres and their orientation. Another important parameter that has a direct influence on the laminate strength is the *Volume Fraction.* Volume fraction is the volume of fibre present with respect to the overall volume of the lamina. According to Jones [20], micro-mechanics is the study of composite material behaviour, wherein the interaction of the constituent materials is examined in detail as a part of the definition of the behaviour of the heterogeneous composite material. The overall elastic moduli of a composite material are expressed in relation to that of the fibres and the matrix. The properties of a lamina can be mathematically expressed as shown in Equation (2.1).

$$Q_{ij} = Q_{ij}(E_f, E_m, v_f, v_m, V_f, V_m) \tag{2.1}$$

Where E, v and V are the elastic (Young's) moduli, Poisson's ratio and the volume fraction respectively, while *f* and *m* in the subscript denotes the fibre and matrix respectively. The volume fraction of fibre and the matrix is given below in Equations (2.2) and (2.3).

$$V_f = \frac{Vol_f}{Vol_{com}} \tag{2.2}$$

$$V_m = 1 - V_f \tag{2.3}$$

Where vol_f and vol_{com} are the volume of fibre and the volume of the whole composite laminate respectively. In order to determine material properties of a lamina, some fundamental assumptions are required. The main fundamental one is that the fibre is mechanically superior compared to the matrix, and therefore is the main load bearing member in the constituent matrix. Also, the strains in the fibre and the matrix are assumed to be the same. In this work, a transversely isotropic laminate is considered for material property identification. In doing so, the analysis can be limited to a small representative volume. Such a volume is called *Representative Volume (RV).* A simple RV would be a fibre surrounded by a matrix as shown in Figure 2.1.

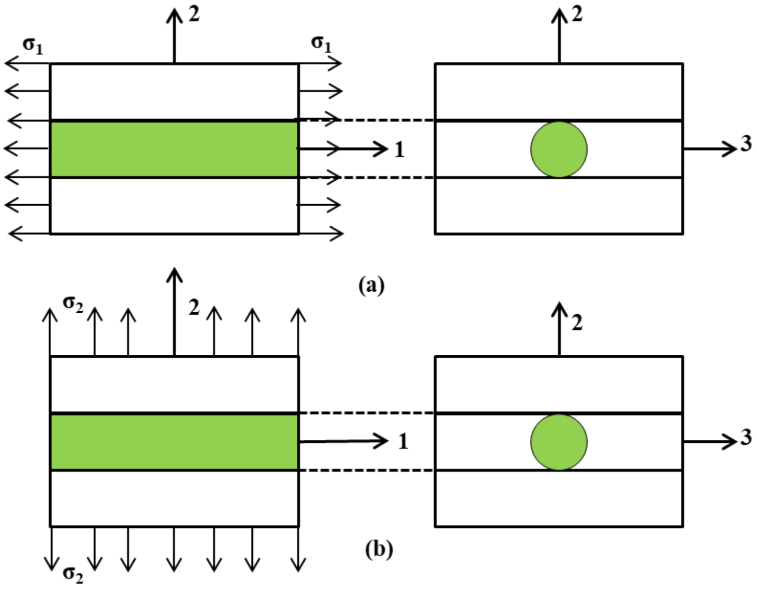


Figure 2.1 (a) RV for longitudinal material property determination (b) RV for transverse material property determination

By applying a stress σ_1 along direction 1, the elastic moduli (Young's modulus) in direction 1 are determined in [19] to be:

$$E_1 = E_f V_f + E_m V_m \tag{2.4}$$

Where E_f and E_m are the elastic moduli's of the fibre and matrix respectively. Equation (2.4) is more commonly well known as the *Rule of Mixtures*. Similarly, a stress σ_2 is applied along direction 2, and the elastic modulus in direction 2 is computed to be:

$$E_2 = \frac{E_f E_m}{E_m V_f + E_f V_m} \tag{2.5}$$

The Poisson's ratio is then computed to be:

$$\nu_{12} = \nu_f V_f + \nu_m V_m \tag{2.6}$$

The shear modulus is expressed in Equation (2.7), while the lamina density is expressed in Equation (2.8).

$$G_{12} = \frac{G_f G_m}{G_m V_f + G_f V_m} \tag{2.7}$$

$$\rho = \frac{\rho_f V_f + \rho_m V_m}{Vol_{com}} \tag{2.8}$$

Once the material properties have been determined at the lamina level, one can proceed then to perform a macro mechanical analysis of the lamina to effectively characterize the overall constitutive model of the laminate.

2.2. Macro Mechanical Analysis

Macro mechanical analysis involves the determination of the overall constitutive model i.e. the overall stress strain relations of the composite laminate consisting of individual laminas stacked together. This is done using *Classical Laminate Plate Theory (CLPT)*. As mentioned earlier, only necessary equations have been provided here. The theoretical detail of these equations can be found in [19]. Main assumptions made in CLPT are the following:

1. The composite material exhibits linear behaviour and the Hooke's Law along with the superposition principle are valid.
2. At the lamina level, the composite is homogeneous and orthotropic and has exactly two planes of symmetry, one along the direction of the fibre and other perpendicular to it.
3. The state of stress at the lamina level is predominantly plane stress.

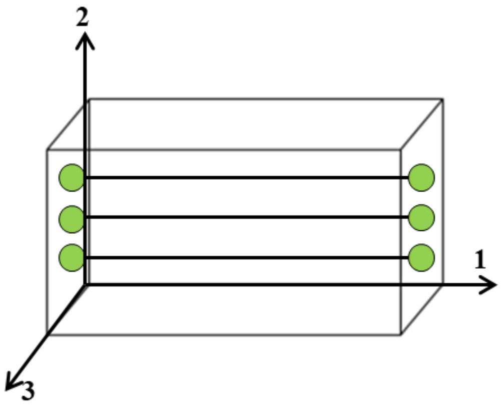


Figure 2.2 Co-ordinate system for a laminate

Looking at Figure 2.2, the principal axes are denoted as 1-2-3, where direction 1 is along the fibres, and direction 2 is transverse to it. . The lamina is assumed to be in 3-D state of stress with six stress components. For a transversely isotropic laminate, strain and stress is related by a compliance matrix as shown in Equation (2.9). All bold face variables denoted here are matrices.

$$\boldsymbol{\varepsilon}_{\mathbf{6\times1}} = \boldsymbol{S\sigma}_{\mathbf{6\times1}} \tag{2.9}$$

Where $\boldsymbol{\varepsilon}$ is the strain developed in the composite due to the applied stress $\boldsymbol{\sigma}$. The compliance matrix $\boldsymbol{S}$ is given by [21]:

$$\boldsymbol{S} = \begin{pmatrix} \frac{1}{E_1} & \frac{-\nu_{12}}{E_1} & \frac{-\nu_{12}}{E_1} & 0 & 0 & 0 \\ \frac{-\nu_{12}}{E_1} & \frac{1}{E_2} & \frac{-\nu_{21}}{E_2} & 0 & 0 & 0 \\ \frac{-\nu_{12}}{E_1} & \frac{-\nu_{21}}{E_2} & \frac{1}{E_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{2(1+\nu_{21})}{E_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & G_{12} & 0 \\ 0 & 0 & 0 & 0 & 0 & G_{12} \end{pmatrix} \tag{2.10}$$

The stiffness matrix $\boldsymbol{Q}$ is expressed in equation in Equation (2.11):

$$\boldsymbol{Q} = [\boldsymbol{S}]^{-\mathbf{1}} \tag{2.11}$$

On solving, the entries of matrix $\boldsymbol{Q}$ is derived in terms of the individual lamina properties determined in the micro mechanical analysis explained in section 2.1 [19] to be:

$$Q_{11} = \frac{E_1}{1-\nu_{12}\nu_{21}} \tag{2.12}$$

$$Q_{12} = \nu_{21}Q_{11} \tag{2.13}$$

$$Q_{22} = \frac{E_2}{1-\nu_{12}\nu_{21}} \tag{2.14}$$

$$Q_{66} = G_{12} \tag{2.15}$$

Through symmetry, the stiffness matrix $\boldsymbol{Q}$ can be reduced to a 3*3 matrix, and is expressed below in Equation (2.16).

$$\boldsymbol{Q} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \tag{2.16}$$

The above relations are for a unidirectional composite with fibres along direction 1 at 0°. Typically fibres of arbitrary orientations are used in structural applications. In most cases, the orientations of the global axes do not coincide with the principal axes of the composite. Hence a combination of translation (***T***) and rotation (***R***) operations is required to obtain the stiffness matrix of such composites. The reduced stiffness matrix is expressed in Equation (2.17) while Figure 2.3 shows the difference in the orientations of the principal axes and the global co-ordinate system.

$$\bar{Q} = [\boldsymbol{T}]^{-\mathbf{1}}[\boldsymbol{Q}][\boldsymbol{R}][\boldsymbol{T}][\boldsymbol{R}]^{-\mathbf{1}} \tag{2.17}$$

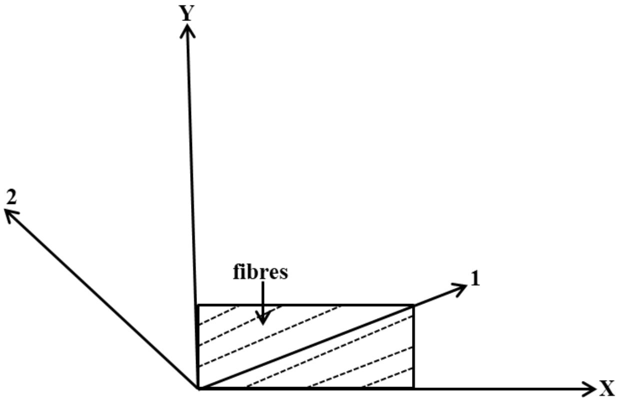


Figure 2.3 Principal axes of the laminate and global x-y axes

The matrix $\bar{Q}$ is fully populated. Hence, although the lamina in its own principal direction is orthotropic, in the transformed coordinate, it represents complete anisotropic behaviour. The elements of $\bar{Q}$ is given by

$$\bar{Q}_{11} = Q_{11}C^4 + 2(Q_{12} + 2Q_{66})S^2C^2 + Q_{22}S^4 \tag{2.18}$$

$$\bar{Q}_{12} = (Q_{11} + Q_{22} - 4Q_{66})S^2C^2 + Q_{12}(S^4 + C^4) \tag{2.19}$$

$$\bar{Q}_{16} = (Q_{11} - Q_{12} - 2Q_{66})SC^3 + (Q_{12} - Q_{22} + 2Q_{66})S^3C \tag{2.20}$$

$$\bar{Q}_{22} = Q_{11}S^4 + 2(Q_{12} + 2Q_{66})S^2C^2 + Q_{22}C^4 \tag{2.21}$$

$$\bar{Q}_{26} = (Q_{11} - Q_{12} - 2Q_{66})S^3C + (Q_{12} - Q_{22} + 2Q_{66})SC^3 \tag{2.22}$$

$$\bar{Q}_{66} = (Q_{11} + Q_{22} - 2Q_{12} - 2Q_{66})S^2C^2 + Q_{66}(S^4 + C^4) \tag{2.23}$$

$$*S = \sin\theta \; and \; C = \cos\theta$$

The reduced stiffness matrix $\bar{Q}$ can be full expressed as shown in Equation (2.24).

$$\bar{Q} = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{16} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{26} \\ \bar{Q}_{16} & \bar{Q}_{26} & \bar{Q}_{66} \end{bmatrix} \tag{2.24}$$

The axial stiffness for the whole constituent model is given in Equation (2.25), while the flexural (bending) stiffness is mathematically expressed in Equation (2.26). Equation (2.27) expresses the axial bending coupling i.e. it relates the in-plane forces with mid-plane curvatures.

$$\boldsymbol{A} = \sum_{k=1}^{N} (\bar{Q}_{\,k})(z_k - z_{k-1}) \tag{2.25}$$

$$\boldsymbol{B} = \frac{1}{2}\sum_{k=1}^{N} (\bar{Q}_{\,k})(z_k^2 - z_{k-1}^2) \tag{2.26}$$

$$\boldsymbol{D} = \frac{1}{3}\sum_{k=1}^{N} (\bar{Q}_{\,k})(z_k^3 - z_{k-1}^3) \tag{2.27}$$

Where *z* is the vertical position of the particular ply from the mid-plane of the laminate as shown in Figure 2.4 which shows a typical 8 layered composite laminate. The constituent equation of a composite laminate is concluded in Equation (2.28). The constituent equation relates the forces ***N*** and the moments ***M*** with the strains ε and curvatures k.

$$\begin{bmatrix} [\boldsymbol{N}]_{xy} \\ [\boldsymbol{M}]_{xy} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B} & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} [\varepsilon^0]_{xy} \\ [k]_{xy} \end{bmatrix} \tag{2.28}$$

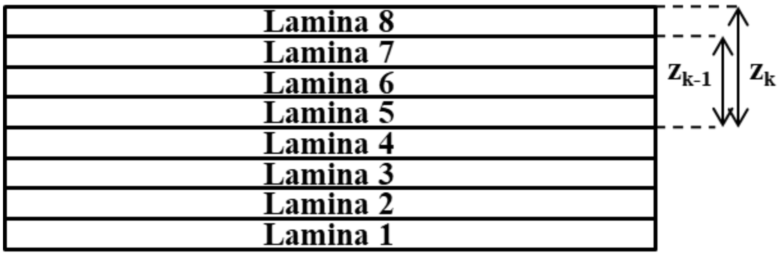


Figure 2.4 An 8 layered unidirectional composite laminate

CHAPTER 3

FUNDAMENTALS OF GUIDED WAVES

3.1. Introduction to Guided Waves

Guided waves are typically elastic waves carrying energy which is confined between the boundaries separating two mediums. Guided waves travel along the length of the sample. They are created by the interactions of bulk waves and the boundaries which results in several mode changes. These interactions also create standing wave modes normal to the boundaries which propagate parallel to the boundaries. In essence, guided wave modes are a constructive interference pattern created by the interaction of many bulk waves with the boundaries. They can be broadly classified into three types: *Rayleigh waves (Surface Acoustic Waves), Shear Horizontal Waves (SH)* and *Lamb waves*. Rayleigh waves are surface waves that can propagate in a half space i.e. it can propagate when there is only a single boundary present, while SH waves and Lamb waves need two boundaries to propagate. Hence, a lot of studies involve analysing the propagation of Guided Waves in plate like structures or pipelines.

Guided Waves can travel along surfaces, interfaces or even throughout the volume structures. This can be achieved by appropriately applying an incident pulse, and making sure the proper boundary conditions exist on the structures in question. Though the direction of propagation is always parallel to the boundaries, the particle motion can be longitudinal, transverse or elliptical depending on the mode that is generated. Because of the boundary effects, which induces the dispersiveness of the wave, analysis of guided waves are much more complex compared to Bulk waves. However, this complexity can be utilized with great effect in SHM. Compared to Bulk waves, Guided waves can travel long distances, and therefore it can be used to propagate along long structures (pipelines), and large areas can be monitored from a single location. Inspecting using bulk waves are cumbersome and time consuming and sometimes cannot inspect structures that are about few centimetre's thick due to impedance issues. Guided waves can also be used to detect very small defects depending on the frequency and the mode selection [22]. Figure 3.1 below roughly approximates the inspection areas of Bulk Wave Testing and Guided Wave Testing.

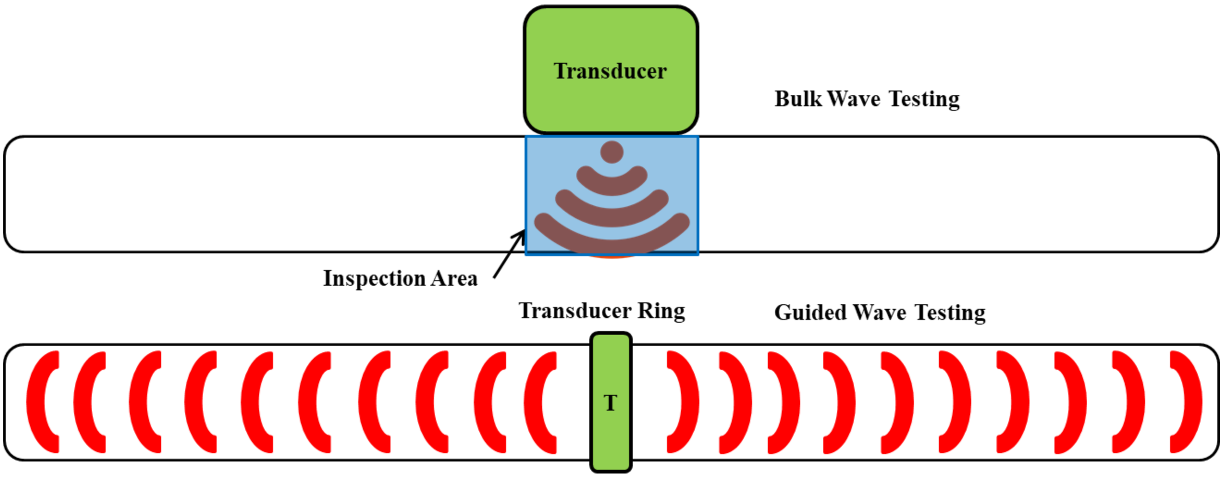


Figure 3.1 Bulk Wave Testing vs Guided Wave Testing

This section gives a general introduction to Guided waves, and describes the analysis of propagation of guided waves in isotropic plate (metal) like structures. It then talks about dispersion curves, before concluding with the analysis of guided waves in a 1D Laminated Composite. Since this is very well researched area and also well documented in literature, only the important equations relevant to this thesis is given. The in depth derivations can be found in the references cited.

3.2. Lamb Waves

Lamb waves are elastic stress waves that propagate between two traction free plates. They are also commonly known as *plate guided waves*. The governing equation of Lamb waves can be approximated to elastic wave propagation in doubly bounded media. The *Navier's* equation

[23] can be used to study Lamb waves in general in a three dimensional solid. Equation (3.1) describes it in a condensed form.

$$\rho \frac{\partial^2 \boldsymbol{u}}{\partial t^2} = \mu \nabla^2 \boldsymbol{u} + (\lambda + \mu) \nabla (\nabla . \boldsymbol{u}) \tag{3.1}$$

Where ρ is the density of the material, λ and μ are known as *Lame's* constants, ∇ is the *del* operator and ∇^2 is the Laplacian. $\boldsymbol{u}$ is the particle displacement vector and is defined as $\boldsymbol{u}=iu_x+ju_y+ku_z$.

The *Navier's* fundamental differential equations governing wave propagation are based upon three fundamental relationships from Linear Elasticity Theory. These include strain-displacement relation shown in Equation (3.2), generalized equation of motion by Newton in Equation (3.3) and constitutive stress-strain relations described previously in Chapter 2 and in Equation (3.4).

$$\varepsilon_{ij} = \frac{1}{2}(\boldsymbol{u}_{j,i} + \boldsymbol{u}_{i.j}) \tag{3.2}$$

$$\frac{d\sigma_{ij}}{dx_i} = \rho \frac{d\boldsymbol{u}_i^2}{dt^2} \tag{3.3}$$

$$\sigma_{ij} = c_{ijkl}\varepsilon_{kl} \tag{3.4}$$

ε_{kl} is the second order strain tensor, while c_{ijkl} is the fourth order stiffness tensor and σ_{ij} the second order stress tensor. The indices indicate summation and commas indicate partial derivatives where *i,j* and *k*=1,2,3. Basically, the *Navier's* equation combines three equilibrium equations, six stress displacement equations and six constituent stress strain relations. For an isotopic medium, the *Navier's* equation can be solved using *Helmholtz* vector by splitting the equation into two partial differential equations based on scalar potentials $\boldsymbol{H}$ and ϕ. For an isotropic medium, the governing equation can be effectively solved to give separate relations for the longitudinal (*P waves*) and the transverse waves (*S waves*). In essence, a single propagating guided wave can be decoupled to two of its fundamental components, the P wave and the S wave. Equations

(3.5) and (3.6) describe the *Helmholtz* decomposition, where Equation (3.6) is known as Gauge invariance.

$$\boldsymbol{u} = \nabla\Phi + \nabla \times \mathbf{H} \tag{3.5}$$

$$\nabla . \mathbf{H} = 0 \tag{3.6}$$

Substituting Equation (3.5) and (3.6) in Equation (3.1), we obtain two separate governing partial differential equation, which are given by:

$$(\lambda + 2\mu)\nabla^2\Phi - \rho\nabla\ddot{\Phi}=0 \tag{3.7}$$

$$\mu\nabla^2\boldsymbol{H} - \rho\ddot{\boldsymbol{H}} = 0 \tag{3.8}$$

Rearranging Equation (3.7) and (3.8), and then expressing it in the rectangular co-ordinate system, we obtain the classic equations of wave propagation in a 3D homogeneous solid in terms of scalar and vector potentials.

$$\nabla^2\Phi = \frac{1}{c_L}\frac{\partial^2\Phi}{\partial t^2} \tag{3.9}$$

$$\nabla^2\mathbf{H} = \frac{1}{c_S}\frac{\partial^2\mathbf{H}}{\partial t^2} \tag{3.10}$$

Where c_L is the longitudinal wave speed (or P wave speed) and is equal to *(2μ +λ)/ρ*, and c_S is the shear or transverse wave speed and is equal to *μ/ρ.* A complete guide to this derivation can be found in [17]. To find the longitudinal mode solutions, classic separation of variables is performed using the expressions of potentials presented in [24] where the continuity condition in circumferential direction is applied. For torsional or transverse mode solutions, alternative solutions are presented by [25]. The last step is the application of boundary conditions, and eigenvalues can be used to trace dispersion curves in general.

For any non 1D solid, the total displacement vector can be expressed as shown in Equation (3.11).

$$\boldsymbol{u} = \boldsymbol{u_L} + \boldsymbol{u_{SH}} + \boldsymbol{u_S} \tag{3.11}$$

Where u_L is the displacement due to the longitudinal or Pressure wave (P wave). The particle motion is parallel with respect to the direction of wave propagation in this case. u_S and u_{SH} are the displacements due to the transverse (S wave) and the *Shear Horizontal* modes, where the particle motion is normal to that of the direction of wave propagation. The interaction of shear and pressure waves in thin plates and shells give rise to guided waves.

In principle, there exist infinite guided wave modes. The first two fundamental of these wave modes are the *Symmetric (S0)* mode, and the *Anti-Symmetric (A0)* mode. These modes have “nascent frequencies” of zero i.e they simply exist even at zero frequency. In the lower frequency range where the wavelength is greater than the plate thickness, these modes are called the extensional mode and the flexural mode respectively. The names are derived based on the particle motion and the elastic stiffness’s that govern the propagation of these waves. These characteristics change as frequency increases. For simplicity, we refer to these modes as A0 and S0 in this work. These modes are particularly important as (a) they exist at all frequencies and (b) they in most cases carry more energy than the higher order modes. Point (a) is particularly important as they are the only modes to exist at lower frequencies until a certain cut off frequency. Therefore, these modes can be exploited for various applications in SHM, for if we use higher modes, multiple reflections from different wave modes will bury the defect signature thereby making the analysis of data more cumbersome and error prone. In the following section, we briefly show the symmetric and anti-symmetric solution for lamb wave propagation in two dimensional plates of infinite length.

3.3. Lamb Wave Propagation in 2D Plates

Let us consider an infinite 2D plate as shown in Figure 3.2. For simplicity of analysis it is possible to assume that the wave potentials are invariant to the z-direction along the wave front [23]. The direction of propagation of the wave is considered to be in the x direction. The plate here is considered to be isotropic in nature, as only then the *Helmholtz's* decomposition principle can be effectively applied to decouple the governing equation to its constituent longitudinal and transverse wave components.

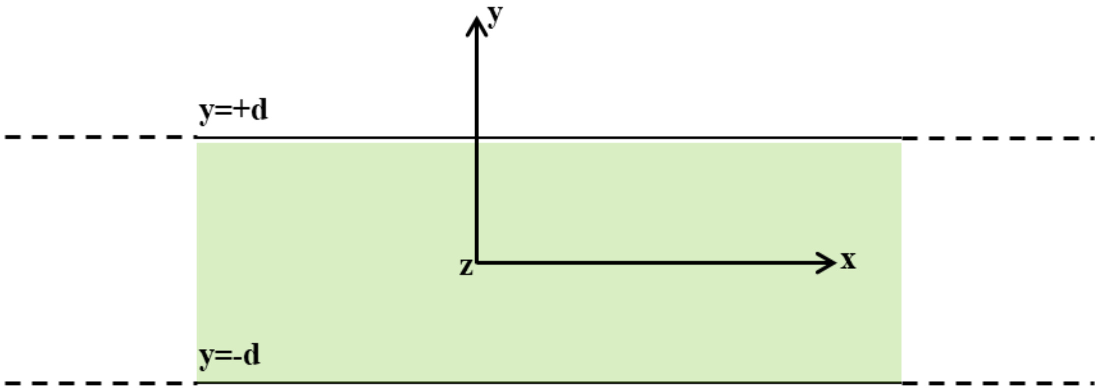


Figure 3.2 2D infinite plate of thickness 2d.

In the above case, since the wave potentials are assumed to be invariant along the z direction, it directly follows that ∂/∂z = 0 and $\boldsymbol{u_{SH}}$ has only the shear displacement component u_z. The displacement vector for the longitudinal mode and torsional mode $\boldsymbol{u_L}$ and $\boldsymbol{u_S}$ have both u_x and u_y components. These displacements only depend on the scalar potential Φ and z-component of a vector potential $\mathbf{H_z}$. Therefore the *Navier's* equations can be rewritten as:

$$\nabla^2\Phi = \frac{1}{c_L}\frac{\partial^2\Phi}{\partial t^2} \tag{3.12}$$

$$\nabla^2\boldsymbol{H_z} = \frac{1}{c_S}\frac{\partial^2\boldsymbol{H_z}}{\partial t^2} \tag{3.13}$$

By substituting simple notations for $\Phi=\phi$ and $\boldsymbol{H_z}=\varphi$**, and assuming plane wave solution of the form $\boldsymbol{\Phi = \phi e^{i\omega t}}$ and $\boldsymbol{\varphi = \varphi e^{i\omega t}}$**we can get a classic formulation for Lamb wave propagation in the potential form:

$$\frac{d^2\phi}{dx^2} + \frac{d^2\phi}{dy^2} + \frac{\omega^2}{c_L^2}\phi = 0 \tag{3.14}$$

$$\frac{d^2\varphi}{dx^2} + \frac{d^2\varphi}{dy^2} + \frac{\omega^2}{c_S^2}\varphi = 0 \tag{3.15}$$

Assuming a harmonic solution $e^{-\xi x}$, the above equations reduce to:

$$\frac{d^2\Phi}{dx^2} + \left(\frac{\omega^2}{c_L^2} - \xi^2\right)\phi = 0 \qquad (3.16)$$

$$\frac{d^2\varphi}{dx^2} + \left(\frac{\omega^2}{c_S^2} - \xi^2\right)\varphi = 0 \qquad (3.17)$$

In equations (3.16) and (3.17), $\xi=\omega/c$ is defined as the wavenumber. To simplify the analysis further and express the equations in a more compact form for convenience, the following substitutions are made:

$$p^2 = \frac{\omega^2}{c_L^2} - \xi^2 \qquad (3.18)$$

$$q^2 = \frac{\omega^2}{c_S^2} - \xi^2 \qquad (3.19)$$

Therefore, Equations (3.16) and (3.17) can be expressed as:

$$\frac{d^2\Phi}{dx^2} + p^2\phi = 0 \qquad (3.20)$$

$$\frac{d^2\varphi}{dx^2} + q^2\varphi = 0 \qquad (3.21)$$

A general solution for the above governing equations can be expressed as shown below:

$$\phi = A_1 \sin py + A_2 \cos py \qquad (3.22)$$

$$\varphi = B_1 \sin qy + B_2 \cos qy \qquad (3.23)$$

Where, A_1, A_2, B_1 and B_2 can be established by applying the appropriate boundary conditions. This can be done by applying symmetric or antisymmetric boundary conditions which gives rise to the symmetric and antisymmetric mode solution respectively. Equations (3.24) and (3.25) show the symmetric and antisymmetric mode solutions respectively. An in depth derivation of this can be found in Appendix A and B based on [25].

$$\frac{\tan pd}{\tan qd} = \frac{\left(\xi^2 - q^2\right)^2}{4\xi^2 pq} \qquad (3.24)$$

$$\frac{\tan pd}{\tan qd} = \frac{4\xi^2 pq}{(\xi^2 - q^2)^2} \qquad (3.25)$$

Where p and q are described in Equations (3.18) and (3.19) respectively. Therefore, using the above relations, one can trace dispersion curves.

3.4. Dispersion Principles

Dispersion in general signifies a changing relationship between velocity and frequency. Bulk waves aren't dispersive as at different frequencies, the velocity of that particular bulk wave does not change. The concept of phase and group velocity is critical to explain dispersion. In simple terms, group velocity is the velocity of the whole wave packet, or the whole wave front. A wave packet typically contains multiple sine waves of different frequency content packed within. The velocity of the sine waves that make up the envelope or wave mode is the phase velocity, while the velocity of the whole packet itself is the group velocity.

$$c_g = c_p^2[c_p - f\frac{dc_p}{df}]^{-1} \tag{3.26}$$

Since both p and q both depend on ω, the phase and group velocities should be evaluated numerically at each frequency step as a multiple of plate thickness. Figures 3.3 and 3.4 below show the phase and group velocity plots of a 1mm Steel Plate respectively. It follows that the solution is not unique, and at higher frequencies, multiple higher order modes of the Symmetric and Anti-Symmetric modes exist. Group velocity dispersion plots are more vital, for we can determine how fast the wave front propagates which helps us establish the *Time of Flight* (TOF). This information is extremely useful for damage detection and damage characterization in structures. Not only that, the group velocities is useful in establishing the material properties of a structure. Group velocity is derived from phase velocity, and the mathematical expression for it is expressed in Equation (3.26) [17].

Dispersion curves are critical in guided wave inspection. Depending on the wave mode that is suitable for testing, it helps in finalizing an operating frequency based on the requirements. It also gives an engineer the cut-off frequency up to which he can operate without involving higher order modes, which invariable complicates the structural analysis. The first step of any guided wave experiment or simulation is to plot the dispersion curves.

Performing any guided wave setup without doing so would result in wasted efforts, lost time and energies.

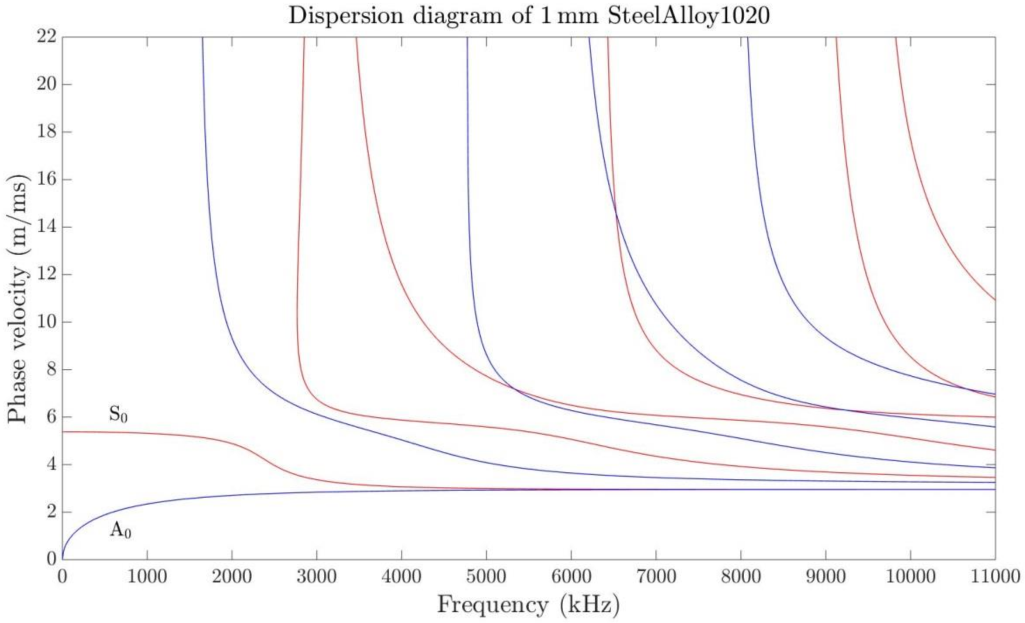


Figure 3.3 Phase Velocity Dispersion Plot for 1mm thick Steel Plate[1]

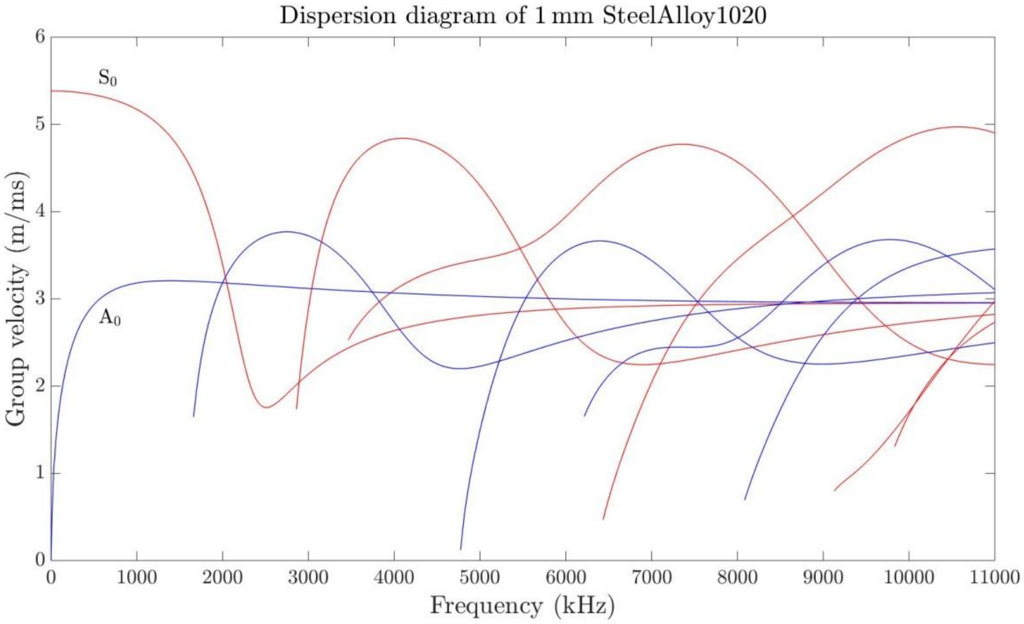


Figure 3.4 Group Velocity Dispersion Plot for 1mm thick Steel Plate[1]

Metals are mostly isotropic in nature. Though they contain a very small degree of anisotropy, it is neglected for most of the analysis in Wave Mechanics. Helmholtz decomposition principle can be used to solve the governing wave equations only when the

[1] Dispersion Plots are plotted with the help of DC software This footnote can go in Reference section
Link: https://www.dlr.de/zlp/en/desktopdefault.aspx/tabid-14332/24874_read-61142/#/gallery/33485

material is Isotropic. By applying it for any non 1D solid, we can basically decouple the equation into its constitutive longitudinal and transverse mode. But when it comes to composites, they are highly anisotropic in nature. There is not true P wave or true S Wave. Hence, anisotropy cannot be neglected in the analysis, and hence Helmholtz principal can no longer be used to solve a system that has more than one dimension. The analysis of 2D composites is achieved by using the *Partial Wave Theory* which is explained in depth in [19]. Though this concept is critical to understand in SHM, the scope of this thesis is to establish a reliable framework to estimate and identify material properties. Therefore, we consider a 1D composite, and the next section outlines briefly the guided wave propagation in 1D Laminated Composite. It is well known that using higher order 1-D theories, we can get to great extent the dispersion relations obtained from 2-D Lamb wave theory [19][49]. For example, if we use First Order Shear deformation theory (Timoshenko theory) and higher order Mindlin-Hermann Theory [19}, we can accurately get the dispersion relations obtained by the 2-D Lamb wave equations. Hence, we pursue this approach in this thesis.

A key point to note is that in a simple higher order 1D model, there will be only one wave mode present at a time i.e. the axial and flexural mode will not be coupled. The type of mode generated will depend on the direction of application of the incident tone burst signal.

3.5. Guided Waves in 1D Laminated Composite

In this derivation, we have considered a 1D composite waveguide that satisfies both elementary rod and beam theories. We begin with defining the displacement vectors fields.

$$u(x, y, z, t) = u^o(x, t) - z\frac{dw(x,t)}{dx} \tag{3.27}$$

$$w(x, y, z, t) = w(x, t) \tag{3.28}$$

Where u^o is the axial displacement along the mid plane, and w is the transverse displacement, and z is measured from the middle plane as shown in Figure 3.5.

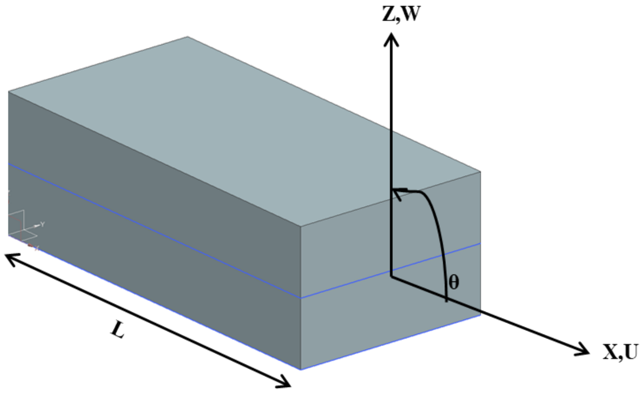


Figure 3.5 Elementary laminate Composite Waveguide with co-ordinate system

A constitutive 3D model for laminated composites was explained in Chapter 2 (Equations 2.18-2.23). The analysis of 1D composites can be achieved if we consider that the composite waveguide in in a 1-D state of stress, and therefore the layer-wise constitutive law is expressed in Equation (3.29).

$$\sigma_{xx} = \tilde{Q}_{11}\varepsilon_{xx} \tag{3.29}$$

Where σ_{xx} and ε_{xx} are the stress and strain applied in the x direction. The expression for $\bar{Q}_{11}$ as a function of ply angle θ is shown below.

$$\bar{Q}_{11} = Q_{11}C^4 + 2(Q_{12} + 2Q_{66}S^2C^2) + Q_{22}S^4 \tag{3.30}$$

Where *S=Sinθ* and *C=Cosθ*. Q_{ij} is the orthotropic elastic coefficients of the individual lamina, and its constitutive relations can be found in Chapter 2. The strain energy is defined in Equation (3.31).

$$U = \frac{1}{2}\int \sigma_{xx}\varepsilon_{xx}dv \tag{3.31}$$

The kinetic energy is then defined as,

$$t = \frac{1}{2}\int \rho({u^o}^2 + w^2)dv \tag{3.32}$$

Where ρ is the layer wise density, Applying Hamilton's principle, the governing differential equations are obtained, and they can be expressed as

$$\rho A\frac{d^2u^o}{dt^2} - A_{11}\frac{d^2u^o}{dx^2} + B_{11}\frac{d^4w}{dx^4} = 0 \tag{3.33}$$

$$\rho A\frac{d^2w}{dt^2} - B_{11}\frac{d^3u^o}{dx^3} + D_{11}\frac{d^4w}{dx^4} = 0 \tag{3.34}$$

The corresponding boundary conditions are,

$$A_{11}\frac{du^o}{dx} - B_{11}\frac{d^2w}{dx^2} = N_x \tag{3.35}$$

$$B_{11}\frac{d^2u^o}{dx^2} - D_{11}\frac{d^3w}{dx^3} = V_x \tag{3.36}$$

$$-B_{11}\frac{du^o}{dx} + D_{11}\frac{d^2w}{dx^2} = M_x \tag{3.37}$$

Where A_{11} is the axial stiffness, B_{11} the axial-bending coupled stiffness and D_{11} the bending stiffness. They are mathematically expressed in Equation (3.38).

$$[A_{11}, B_{11}, D_{11}] = \int_{-\frac{h}{2}}^{\frac{h}{2}} \check{Q}_{11}[1, z, z^2]b dz \tag{3.38}$$

Where h is the depth of the beam, b is the layer width. A is the cross sectional area while d^2u^o/dt^2 and d^2w/dt^2 are the longitudinal and transverse accelerations. N_x is the force acting in the axial direction; V_x is the shear force while M_x is the bending moment. In our studies here, we primarily consider only a symmetric laminate. Symmetric Laminates are composite laminates where the lay up above the mid plane is the mirror image of that of the layup below the mid plane. For a symmetric laminate, the axial-bending stiffness B_{11} is equal to 0. As a consequence, axial and flexural modes become uncoupled. Therefore for a symmetric laminate, we can rewrite equations (3.33) to (3.37) as

$$\rho A\frac{d^2u^o}{dt^2} - A_{11}\frac{d^2u^o}{dx^2} = 0 \tag{3.39}$$

$$\rho A\frac{d^2w}{dt^2} + D_{11}\frac{d^4w}{dx^4} = 0 \tag{3.40}$$

The boundary conditions can be rewritten as,

$$A_{11}\frac{du^o}{dx} = N_x \tag{3.41}$$

$$-D_{11}\frac{d^3w}{dx^3} = V_x \tag{3.42}$$

$$D_{11}\frac{d^2w}{dx^2} = M_x \tag{3.43}$$

From Equations (3.39) and (3.40), it is visible that the governing equations are uncoupled, i.e. solution to Equation (3.39) will give rise to the axial mode, while solution to the equation (3.40) will give rise to the flexural mode. An in depth derivation of the solution based on computation of wavenumbers is available in [19].

CHAPTER 4

OVERVIEW OF MACHINE LEARNING

Deep Neural Networks are an important subgroup of Machine Learning. *Machine Learning* (ML) is considered to be the natural evolution of statistical learning. It is one of the major dominant subset of *Artificial Intelligence* (AI). To summarize in a single sentence, ML is the science of developing statistical models that learn with time, and can perform a function/task of interest without necessary human intervention. Previously, the field of AI relied heavily on hard coded rules and pre fixed algorithms. This required enormous computational resources. Also, the systems depended on programmed intelligence, and were not capable of learning on their own. ML, whereas relies on learning using real time data instead of relying on hard coded rules. This is done by building a model that best describes patterns between the input and output data. In a nutshell, they can be seen as function approximators that fits the best possible function for a set of input and output vectors [26] [27]. ML algorithms have the ability to predict the output even on unseen data. It greatly reduces the computational effort, while the accuracy keeps improving as more real data is available to train. There are different types of algorithms available, some the common one's being *Decision Trees, Support Vector Machines, Radial Basis Functions* etc. Deep Learning Algorithms meanwhile are a subset of ML techniques that basically mimics a human brain. A deep neural network basically contains multiple layers that have nodes that resemble a neuron in the brain with associated weights. The weights can be learned using many optimization algorithms to predict the desired output. They are very flexible, and can learn complex and often nonlinear relationships. Though the amount of training data required is large, the type of tasks it can achieve is unparalleled, and is pretty much driving the field of AI these days. Figure 4.1 summarizes how these different techniques fit in. To summarize, Deep Learning techniques are one of the many different ML techniques, that is part of the broader AI field.

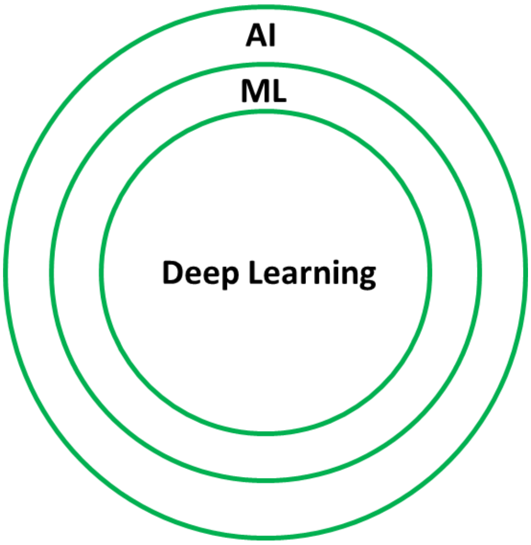


Figure 4.1 Machine Learning and Deep Learning

In this chapter, we introduce the fundamentals of an *Artificial Neural Network* (ANN) that include the basic theory, math and the algorithms behind it. Following this, more advanced deep learning models like *Convolutional Neural Networks* (CNN) are discussed in detail.

4.1 Artificial Neural Networks

An artificial neuron was formalized as initially the *Threshold Logic Unit* (TLU) by McCulloch and Pitts in 1943 [28]. For a single neuron shown in Figure 4.2, the output signal can be written mathematically as:

$$y_j = f\left(\sum_{i=1}^{n}\left(W_{ji}x_i + b\right)\right) \tag{4.1}$$

Where, x_i is the input to a neuron, y_j is the output from the neuron. W_{ji} is the associated weight matrix and b is the bias term, which is generally used to shift the decision boundary line or any hyper-plane in multidimensional problems [29]. The function f is the activation function which decides in which manner the neuron will be excited or fired for a particular input signal. As the name suggests, a heaveside step function is used in the TLU. The function is expressed mathematically in Equation (4.2).

$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & otherwise \end{cases} \tag{4.2}$$

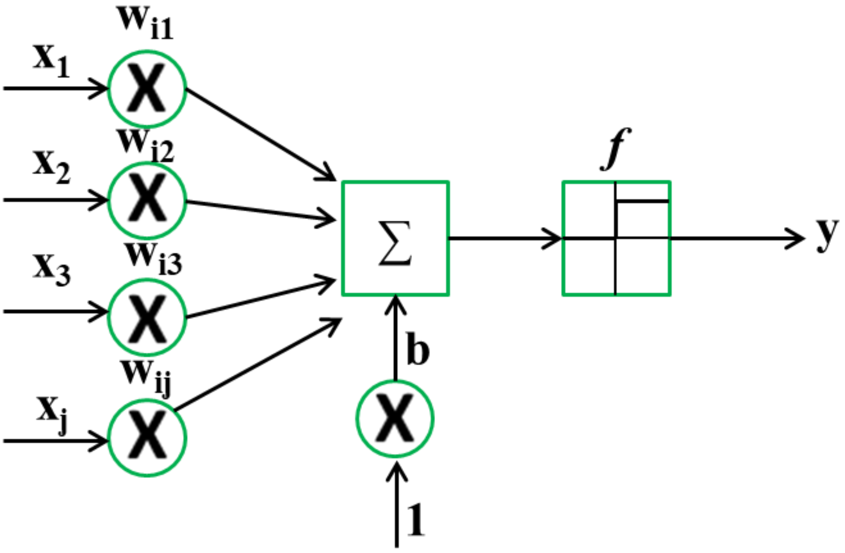


Figure 4.2 Pipeline of a Neuron

This activation is actually not used these days because it has a constant derivative, and more importantly the derivative is not defined at *x=0.* It must be noted that it is similar to how a biological neuron works. Another important point of note here is that, the weight matrix initially designed in the TLU does not have the capability to learn.

A TLU then evolved into the famous perceptron networks. Perceptron networks were similar to TLU, but the major difference was that there was now a learning rule to adjust the weight matrix. The delta rule, which uses the Vanilla gradient descent approach to minimize the error, is used as the learning rule. Multiple neurons stacked together in layers are now commonly known as *Multi-layer Perceptron Networks* (MLP) or as an *Artificial Neural Network* (ANN). Since each neuron is fully connected with neurons from the next layer, they are also referred to as *Fully Connected Networks* (FCN) or *Dense Neural Networks* (DNN). Perceptron Networks were first reported by Rosenblatt in 1958 [30].

4.2 Multi Layered Perceptron (MLP)

Figure 4.3 shows a typical MLP network with one input layer, one output layer and one hidden layers. The MLP is a feed forward network, where the flow of information is forward, while the flow of error information is backward. Each node *i* is connected another node *j* in the previous and following layers with an associated weight w_{ij}. In layer *k*, the weighted sum of is performed at each node *i* of all the signals $x_j^{(k-1)}$ from the preceding layer *k-1* giving the sum $z_i^{(k)}$ of the node. The sum is passed through a nonlinear activation function *f*, which gives the output of the MLP network. Mathematically, it can be expressed as,

$$y_i^{(k)} = f\left(z_i^{(k)}\right) = f\left(\sum_j w_{ji}^{(k)} x_j^{(k-1)}\right) \tag{4.3}$$

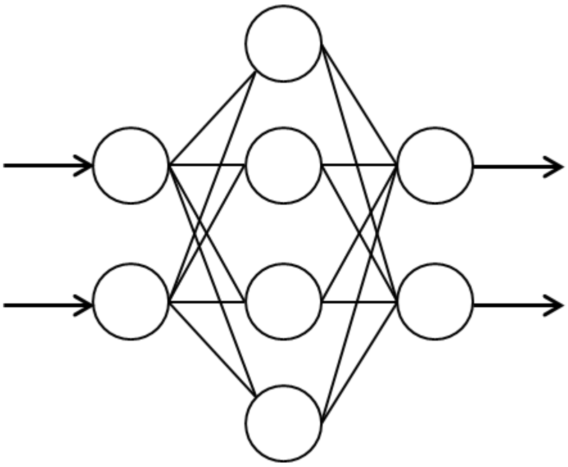


Figure 4.3 Three Layered MLP Network

There are different types of activation functions used. Each one has its own advantages and disadvantages which is not discussed in this thesis. The main nonlinear activation functions extensively used are the sigmoid, *tanh, softmax and ReLu* activation functions. The *ReLu* function i.e. the Rectified Linear Unit is computationally very efficient due to its simplicity. Equations (4.4) and (4.5) describe the *ReLu* function. The *ReLu* function introduces nonlinearity in the system that greatly improves the performance in most cases. One disadvantage of *ReLu* is that, when the inputs approach zero or negative, the gradient of the function becomes zero, therefore the network then cannot perform backpropagation and stops learning. This is known as the "*Dying ReLu*" Problem. This issue is solved by introducing a small positive slope in the negative area, and is known as the *Leaky ReLu.*

$$f(x) = \max(0, x) \tag{4.4}$$

$$f'(x) = \begin{cases} 1 & x \geq 0 \\ 0 & otherwise \end{cases} \tag{4.5}$$

Machine Learning can be broadly classified into three categories. *Supervised Learning* is where the model is task driven in order to predict/classify a desired output. *Unsupervised Learning* is data driven and includes clustering algorithms. The final type is the *Reinforcement Learning* where the model learns to react to an environment. In this work, we predominantly use Supervised Learning Algorithms. The core concept of Supervised Learning involves two passes, the first being the forward pass where an input signal is propagated through multiple layers to obtain a predicted output. The second being the backward pass, where the error obtained from the predicted and true output is propagated back up to the first hidden layer. Using the error information, the weights are adjusted based on some rules to move the predicted output close to the true output i.e minimize the error.

This is often achieved through the backpropagation algorithm. The beauty of the algorithm is that the weights are changed accordingly depending on how much they contribute to final error. Suppose we consider a Loss function as defined in Equation (4.6), where *n* is the number of samples, *y* and $\tilde{y}$ are the true and predicted outputs.

$$E(y, \tilde{y}) = \frac{1}{2}\sum_{j=1}^{n}(y - \tilde{y})^2 \tag{4.6}$$

A point to note is that this type of evaluating the loss using all samples at the end of every epoch is called batch or online training, where an epoch is defined as a measure of the number of times all of the training vectors are used once to update the weights. By back propagating through the network, we can obtain information on how the previous outputs influence the error using the chain rule. Assuming there are *k* hidden layers, for any particular neuron, we compute:

$$\frac{\partial \mathrm{E}}{\partial w_{ik}} = \frac{\partial \mathrm{E}}{\partial \breve{y}} * \frac{\partial \breve{y}}{\partial net_{ik}} * \frac{\partial net_{ik}}{\partial w_{ik}} \tag{4.7}$$

where net_{ik} is the total value entering the activation function in Equation (4.7). By looking at the individual derivatives, it becomes obvious to calculate all of them from the final output layer to the first input layer. By repeating the chain rule process, we have all the partial derivatives to update the weights according to Equation (4.8). This is typically the Vanilla version of back propagation. There are other versions that have been developed, which are more advanced, and computationally more efficient. The overall process of driving the loss to a minimum by means of backpropagation is known as gradient descent, which is given by

$$w_{ji}^{t+1} = w_{ji}^{t} - \eta \frac{\partial \mathrm{E}}{\partial w_{ji}^{t}} \tag{4.8}$$

Here, η is called the learning rate, which basically regulates the rate of the learning process. A general methodology to find an optimum learning rate is a very well posed problem, that doesn’t yet have a mathematical solution. There is no fixed way of determining this; rather mostly by trial and error. a suitable learning rate can be established. However, it is to be noted that establishing a suitable learning rate is fundamental to any ML study. Choosing too small a value can lead to premature convergence, while a big value can cause oscillations near the global minima which, is commonly known as overshooting.

The main purpose of a neural network is to classify or predict on unseen data, after being trained on seen data. In order to do that, a dataset is broken down in to three main subsets.

- **Training dataset**: This is the dataset that is used to train the model which basically fits for all the weights to get the desired output.
- **Validation dataset**: In order to verify how well the network is training, some part of the dataset is fed in at pre fixed time intervals and the validation loss is monitored. If the validation loss increases during training, then it is generally advisable to stop training. This is normally known as overfitting.

- **Test dataset**: Once training is completed, the weights are all updated and stored as a model. This is used on a dataset that consists of previously unseen data to predict or classify based on the task.

Typically, there is again no pre-set rule to determine the ratio in which these datasets are split from the main dataset. Having too many training examples will result in over generalization, while having a small training dataset will result in insufficient learning. Another important concept to consider is the problem of overfitting and underfitting. If the model is complex with multiple layers, and if a line is fit to pass through all the training samples, this results in under generalisation of the model, where the model has learnt so well on the training network and yet it fails to predict on any unobserved data. This is known as ***overfitting***. The exact opposite occurs when the model is too simple, where the model suffers from high bias and no matter how many observations is fed into it; the model always produces similar results. This is known as ***underfitting***. When the training loss is far lesser than the validation loss, this indicates overfitting while the opposite indicates underfitting. Therefore, achieving the right trade-off is one of the most challenging tasks in ML.

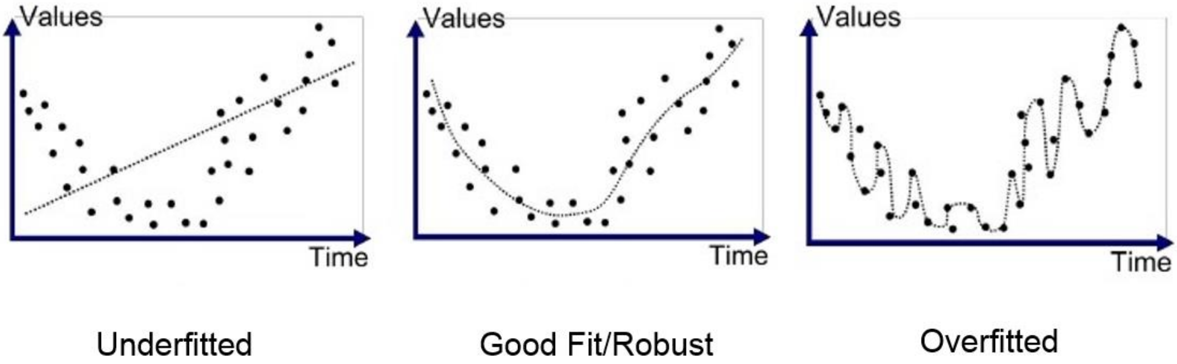


Figure 4.4 Overfitting and Underfitting [31]

By nature, deep neural networks are always complex models. Hence the problem of underfitting occurs much lesser than that of overfitting. Therefore, there is a need to avoid overfitting when designing deep neural networks. Typically this is achieved by many different approaches. The main methods use regularizers, which restrict the sudden changes in weights by adding a penalty term in the loss function. Another way to tackle overfitting is by adding dropout layers, which randomly prunes some neurons in the hidden layers thereby favouring some features over others. This greatly helps in making sure the model does not just generalize on the training dataset. Another commonly exploited technique is data augmentation. This simply refers to distorting some features of the data to create more data that can be fed in to the model to achieve better performance. Deep Neural Networks are always data hungry, and more the data, better is the performance in most cases. Typically the MLP/DNNs presented this section involves a time consuming feature selection process. The features are then fed to the neural network to perform an associated task. But deeper networks like Convolutional Neural Networks and Long Short Term Memory Networks eliminate this step by having a feature selection process incorporated within the network itself i.e. one would have to just feed in the raw signal, and the network would pick out features that best describe the input signals and complete an associated task.

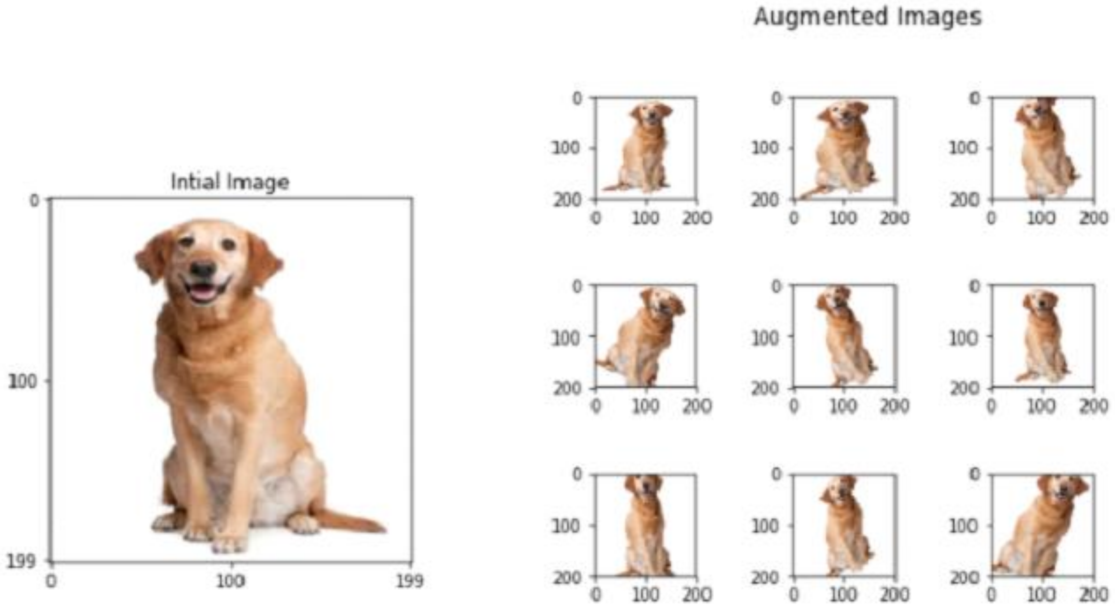
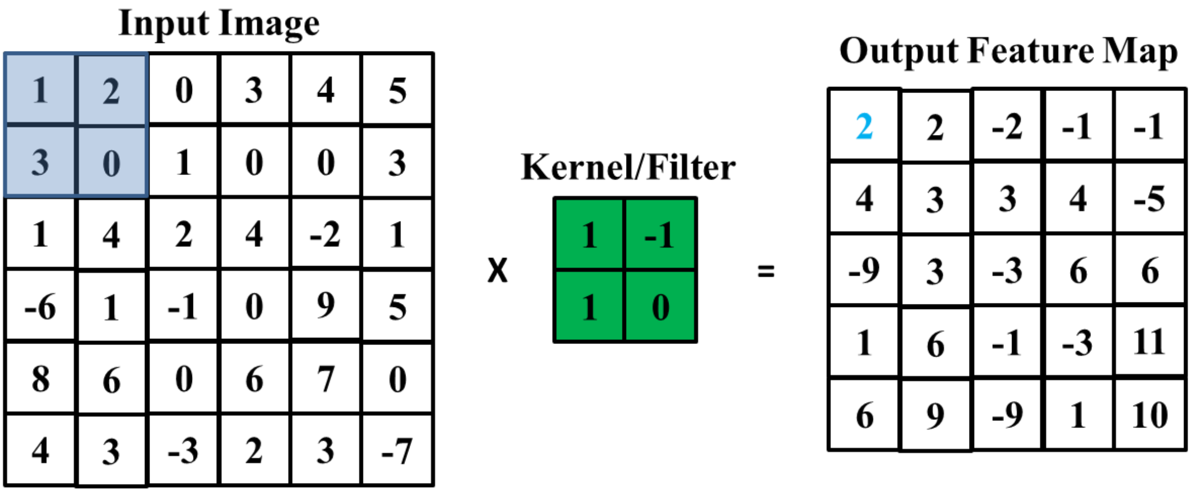


Figure 4.5 Example of Data Augmentation [32]

4.3 Convolutional Neural Networks (CNN)

CNNs take biological inspiration from the visual cortex. The visual cortex has small regions of cells that are sensitive to specific regions of the visual field. This idea was expanded upon by a fascinating experiment by Hubel and Wiesel in 1962 where they showed that some individual neuronal cells in the brain responded (or fired) only in the presence of edges of a certain orientation [33]. A CNN is a network that performs the Convolution operation (rather Correlation) in at least one of its layers instead of the general matrix multiplication and addition that takes place in a normal neural network. CNN's are used to capture spatial and temporal dependencies using relevant filters. CNN's is extensively used for datasets that contain images for training. Some of the most salient features of a CNN are spatial down sampling, shared weights and local reception.

A typical CNN has convolutional layers, pooling layers and fully connected layers. The convolutional layer is the first layer that is used to extract features from the input that best describes the input data. This is done by convoluting the input image by convolutional filters called as ***convolutional kernels***. The resulting output is typically known as feature maps. Figure 4.6 shows how a convolutional layer works.



1	2	0	3	4	5
3	0	1	0	0	3
1	4	2	4	-2	1
-6	1	-1	0	9	5
8	6	0	6	7	0
4	3	-3	2	3	-7

1	-1
1	0

2	2	-2	-1	-1
4	3	3	4	-5
-9	3	-3	6	6
1	6	-1	-3	11
6	9	-9	1	10

Figure 4.6 Working of a Convolutional Layer

Assuming an input image that is a 6 x 6 matrix, a convolutional filter of size 2 is applied to the input matrix. A simple multiplication of the elements in the filter and the input images is done to establish the first entry (@ (1, 1) of the feature matrix. The size of the output feature map depends on an entity called *stride*. The stride gives the spatial distance between the central pixels (both vertically and horizontally) on which the convolution operation takes place. The feature map in Figure 4.6 is computed with a stride of 1. Mathematically, the convolution operation can be defined as:

$$G[m,n] = (f * h)[m,n] = \sum_j \sum_k h[j,k] f[m-j, n-k] \tag{4.9}$$

The output G is the feature map produced by the convolution operation, while f is the input and h is the kernel/filter. m and n are the indices of the row and column of the output feature map. A convolutional layer enforces the idea of weight sharing. If every pixel in the image is imagined to be a neuron in the first layer, and if the output feature map is treated as the second layer where each pixel once again is treated as a neuron, it is clearly evident that at any single given computation, only few neurons from the first input layer is connected to the second output layer. As the kernel is moved across the image, different neurons are connected and this therefore ensures shared weights. Shared weights reduce the number of parameters to be learnt, and it becomes computationally more efficient.

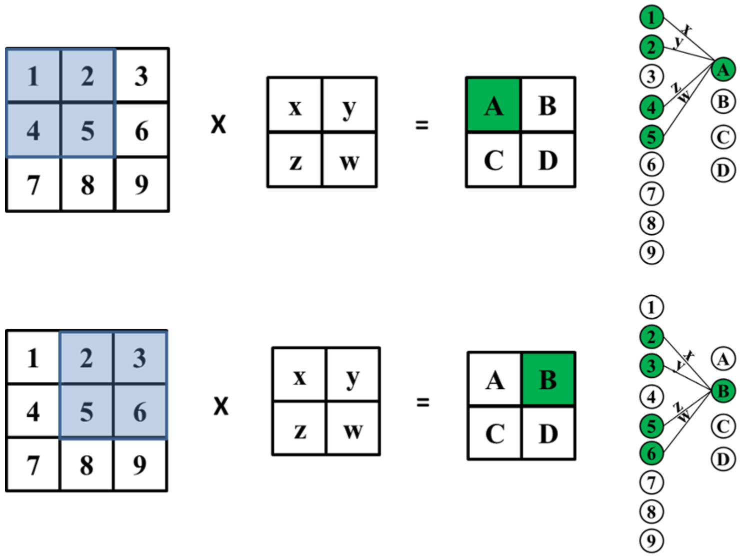
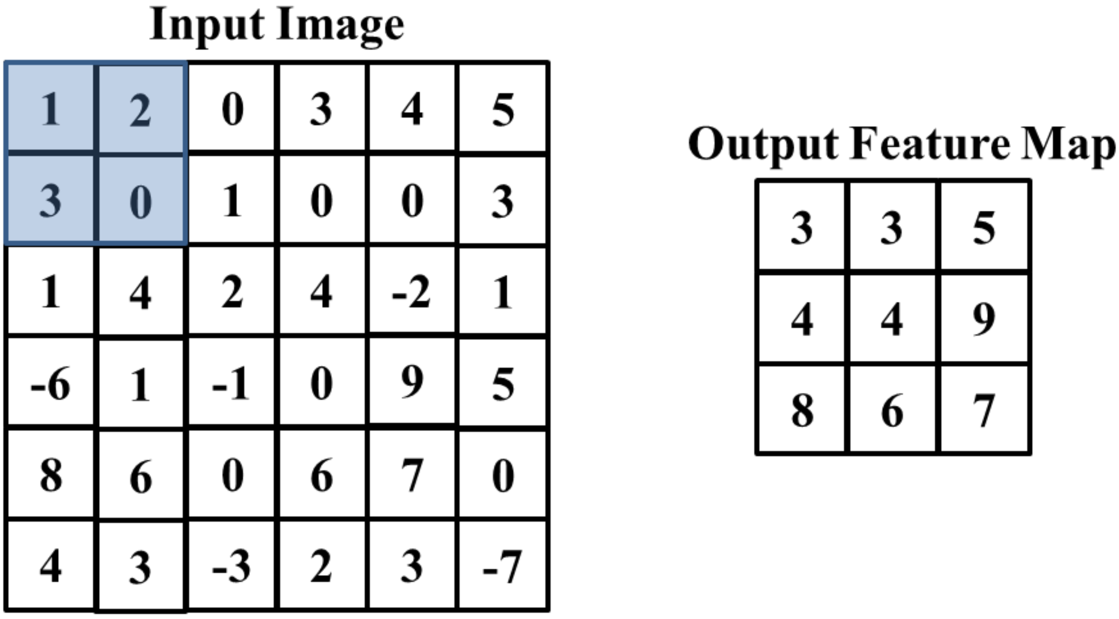


Figure 4.7 Example of weight sharing in Convolutional Layers

Another fundamental component of CNN’s is the pooling layers. These layers are used to sub sample the feature maps spatially. They reduce the dimensions that we are working with and make it computationally more efficient. Similar to convolutional layers, the input image is stride over by a kernel, wherein operations like max pooling, average pooling takes place. A point to note is that pooling layers have no learnable parameters. Figure 4.8 below shows the working of a max pooling layer for a stride length 2.



Input Image

1	2	0	3	4	5
3	0	1	0	0	3
1	4	2	4	-2	1
-6	1	-1	0	9	5
8	6	0	6	7	0
4	3	-3	2	3	-7

Output Feature Map

3	3	5
4	4	9
8	6	7

Figure 4.8 Working of a Max Pooling Layer (stride=2)

The final architectural idea involved in a CNN is the fully connected layer (FCN). This is very similar to the MLP networks discussed before, and is basically a classifier/predictor model that uses the features extracted by the kernels in previous layers. All the parameters of this layer are learnable and is generally the last layer of a CNN. Convolutional Neural Networks are generally used only on images, but a lot of recent research has gone into using one dimensional CNN’s (1D-CNN) for time series data. 1D-CNN’s are known to extract inherent features from long time signals. 1D-CNN’s work similar to traditional CNN’s, the only difference being the operations is done in single dimension. In 1DCNN’s the selection of kernel size is very critical and can be treated as a hyper parameter.

4.4 Recurrent Neural Networks (RNN)

The traditional neural networks and convolutional neural networks all work with fixed input and output lengths. But that is never the case in many practical applications (for example, finding the number of vowels in a sentence). Problems like speech recognition and time series predictions or forecasting require a system to store and use context information. More so, if we consider a human brain, one of the most salient features is the persistence present in our system i.e. we remember something we learnt/saw days or weeks before. Recurrent Neural network (RNN) typically brings about persistence in a traditional neural network. Recurrent Neural Networks take the previous output or hidden states in most cases as the inputs for the current computation [44]. The composite input at time *t* has some historical information about the happening at time *T*<*t*. RNNs are useful as their intermediate values (state) and can store information about past inputs for a time that is not fixed a priori. RNN's in general hold information over a certain time that is not fixed. Figure 4.9 shows how a RNN is different from a typical feed forward network.

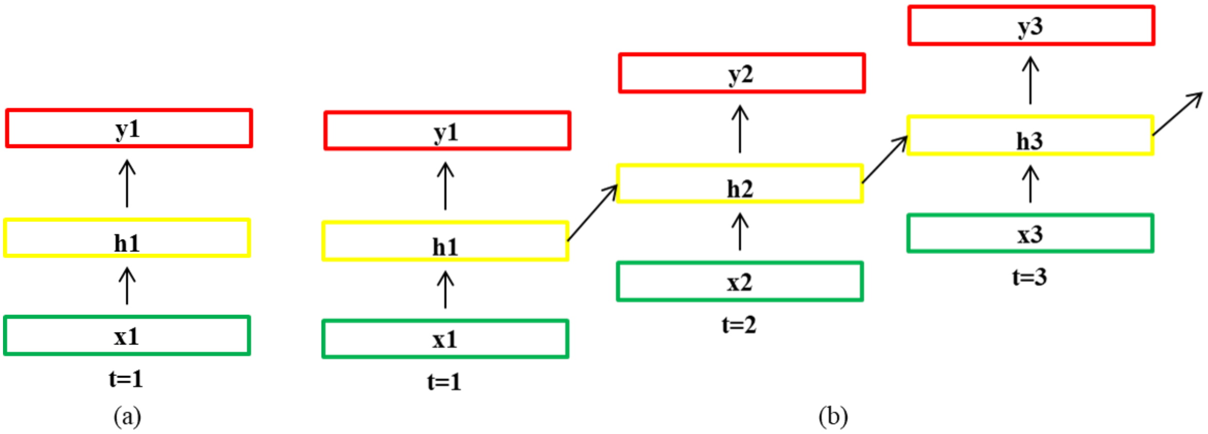


Figure 4.9 (a) Simple Feed Forward Network (b) A RNN with previous hidden states as input

Recurrent Neural Networks though in theory are capable of handling long-term dependencies fall short when it comes to practical applications. This problem was very well explored in depth by Hochreiter (1991) and Bengio, et al. (1994) [45]. It was seen that remembering information over long periods requires calculating the distances between distant nodes that involves multiple multiplications of the Jacobian Matrix. Problems with the more commonly occurring vanishing gradients and lesser frequent exploding gradients caused the performance of these models to be not satisfactory. It was seen that a trade of between gradient descent based learning and the time over which the information is held was required. In order to overcome this, Hochreiter and Schmidhuber (1997) [46] introduced the Long Short Term Memory networks usually called LSTM's. The LSTM's accumulates long-term relationships between distant nodes by designing weight coefficients between connections. These networks have shown unbelievable applications in speech processing, Natural Language Processing and image captioning among other applications.

4.5 Long Short Term Memory Networks (LSTM)

A LSTM unit utilizes a "memory" cell (denoted by c_t) that decides whether the 'information' is useful or not and a gating mechanism that contains three non-linear gates: (i) an input (denoted by i_t), (ii) an output (denoted by o_t) and (iii) a forget gate (denoted by f_t). The gates regulate the flow of signals into and out of the cell to adjust long-term dependencies effectively and achieve successful RNN training. The standard equations for LSTM memory blocks are given as follows.

The three gates:

$$i_t=\sigma(U_i h_{t-1}+W_i x_t+b_i) \tag{4.10}$$

$$f_t=\sigma(U_f h_{t-1}+W_f x_t+b_f) \tag{4.11}$$

$$o_t=\sigma(U h_{t-1}+W x_t+b_o) \tag{4.12}$$

Memory Cells and hidden units:

$$c_t=i_t * c_{t-1} + tanh\ (U_c \mathrm{h}_{t-1}+\mathrm{W}_c \mathrm{x}_t+b_c) \tag{4.13}$$

$$h_t=o_t * tanh\ (c_t) \tag{4.14}$$

Where x_t is the external input vector, *tanh* is the hyperbolic tangent function, and the parameters are the matrices $\boldsymbol{W}$ and $\boldsymbol{U}$, and vector bias $\boldsymbol{b}$, with appropriate sizes for compatibility.

The output layer:

$$y_t = Vh_t + d \tag{4.15}$$

The internal dynamic “state” captures the essential information of an input’s time-history profile. In any time series signal processing scheme in recurrent systems, repeated multiplication of internal states beyond the external input sequence isn’t required or can be avoided. Figure 4.10 shows a LSTM network. Due to the higher number of trainable parameters in a LSTM network, the training time is typically very high. Therefore, LSTMs though very powerful sometimes are very painful to train in many cases. CuDNNLSTM is a inbuilt tensorflow API that almost speeds up training by 10 times while using LSTM layers. This is achieved by skipping repeated activation function at each cell. A vanilla based LSTM cell is shown in Figure 4.11. Alternative methods called Slim LSTMs have been investigated where the number of adaptive parameters at each gate is reduced [47].

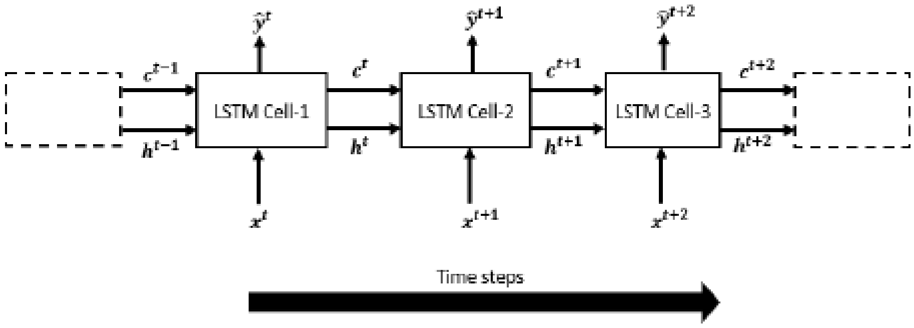


Figure 4.10 Architecture of a simple LSTM Network [48]

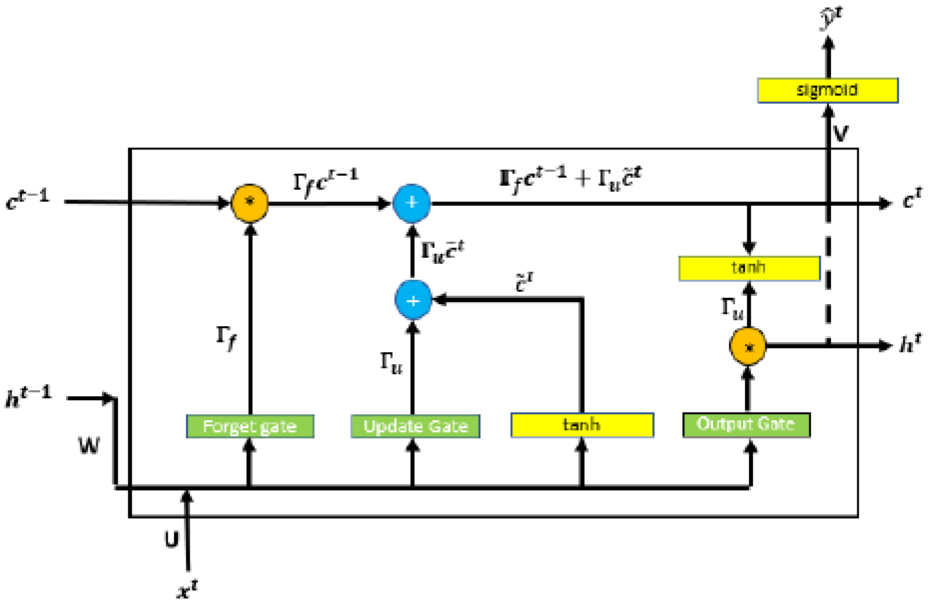


Figure 4.11 Vanilla based LSTM Cell [48]

In scope of this thesis, 1DCNN's and LSTMs are known to extract inherent features in long temporal signals. This allows us to directly feed raw waveforms (A0 and S0 time histories) to the neural networks, thereby eliminating the time consuming feature engineering process. This is exploited in the DL based approach, while in the ML based approach, we pick out three features (group velocities of three wave modes as a function of frequency) and feed it into a multi-layer DNN. This is done to address one of the limitations of the DL based approach that will be discussed in Chapter 6 in detail.

CHAPTER 5

MODELLING TECHNIQUES FOR GUIDED WAVES IN COMPOSITES

5.1 Why do we need modelling?

The behaviour of any wave in any dynamic system is generally governed by a set of *Partial Differential Equations* (PDE), which is normally called the '*equilibrium equations'* in mechanics. Apart from the simplest cases, most PDE's do not have an exact analytical solution, or at best have solutions that can be reached after an enormous computational effort. This makes PDE's almost impossible to solve without alternate solution strategies. This typically involves assuming a solution involving many constants and we then determine these constants such that the governing PDE is satisfied in an approximate sense. To put it in simpler words, it is like finding one of the solutions for a problem that has infinite solutions. However, the assumed solution should satisfy certain conditions called the boundary conditions and initial conditions. This can be thought of as conditions that drive the particular solution to the closest possible approximation. Typically there are two types of boundary conditions: the natural boundary conditions and the kinematic boundary conditions. The former is also called sometimes as the force boundary conditions or in general Nuemann boundary conditions that give the value of the derivatives of the unknown variable at the boundary. Meanwhile, the kinematic boundary conditions are also called the essential boundary conditions or in general Dirchelet boundary condition that gives the value of the unknown variable at any boundary [34]. The governing PDE is not amenable for this solution philosophy due to higher order continuity requirements of the assumed solutions. Therefore, one requires an alternate statement of the governing PDE that is generally most suited for the numerical solution. This is normally provided by the variational statement of the problem. Finite Element Method (FEM) is one such method of numerical solution which is further explained in this chapter. In this chapter we also discuss yet another semi-analytical method called Spectral Finite Element Method (SFEM), which can be described as FEM but formulated in frequency domain. There are similarities and differences between these methods which will be discussed in detail in this chapter.

5.2 Finite Element Method

The finite element method is one of the most common numerical methods, which essentially assumes a solution that only solves the kinematic boundary conditions. It is a stiffness based method, wherein the domain of the problem is broken down to its constituent elements that are connected to each other by nodes.

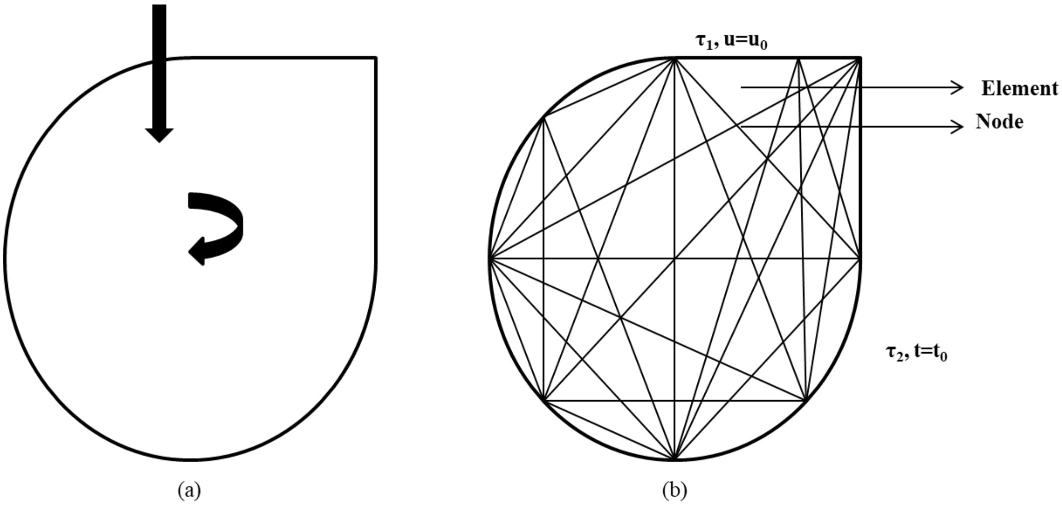


Figure 5.1 (a) Overall domain of the problem (b) Discretized domain set of the problem with boundary conditions

The FEM procedure begins with the assumption of the displacement field in each of the elements, and the assumed field is then substituted into the weak form of the differential equations, which is basically the integral form of the original differential form of the governing PDE. The number of elements is driven directly by the element size, or also known as mesh size. Changing the mesh size changes the number of elements. The mesh size is a very important parameter when it comes to the accuracy of the solution. For example, if the stress and strain gradients are high at a particular location (this is especially true at the edges of a crack or a defect); the mesh discretization should be very fine. Also, when performing high frequency studies (wave propagation problems) where the wavelength is very small, mesh sizes that are in the scale of the wavelength can act as potential boundaries and can reflect waves, which complicates the problem in hand even further. Suppose *u(x,y,z,t)* is the displacement vector (in mechanics), the objective here is to estimate a solution of the dependent variable *u* in the form:

$$u(x,y,z,t) = \sum_{n=1}^{N} a_n(t)\psi_n(x,y,z) \tag{5.1}$$

Where $a_n(t)$ are the unknown time dependent coefficients that are to be determined, and ψ_n are the spatial dependent functions that normally satisfy the kinematic boundary conditions. The displacement vector *u(x,y,z,t)* is a function of all three co-ordinates and time. The formulation of the different approximate methods is the *Weighed Residual Technique* (WRT), where in the error or the residual obtained by substituting the assumed approximate solution in the governing PDE is weighted with a weight function and integrated over the domain. This process has similarities to the backpropagation algorithm in neural networks discussed previously. The error obtained by the difference in the approximate target solution to the true solution is propagated back to each neuron (node) and the weights (weighted function) are updated accordingly. The final integration is basically the adding up of the contribution of each neuron (node).

In complex structures, the different approximation techniques are very difficult to use. The main difficulty lies in determining functions ψ_n, which are called the *Ritz functions*. Typically methods like *Rayleigh-Ritz* method [35] cannot be used in complex structures. Hence by dividing the domain into sub domains (which constitutes an element in the finite element mesh), one can apply different approximation techniques over all the elements which are then pieced together to obtain the total approximate solution. This in essence is the Finite

Element Method. To summarize, the flow chart in Figure 5.2 gives an overall perspective of the different steps involved in the FEM.

The Figure 5.2 can be explained as follows: FEM method fits the approximate solution for the dependent variable as given in Equation (5.1) over each element. Using this equation under the WRT framework, reduces the governing PDE into a set of coupled *Ordinary Differential Equations (*ODE) of the form as shown below

$$[M]\{\ddot{u}\} + [K]\{u\} = \{F\} \tag{5.2}$$

Where ***[M]*** is the assembled mass matrix, **[K]** is the assembled stiffness matrix, $\{\boldsymbol{\ddot{u}}\}$ is the nodal acceleration vector $\{\boldsymbol{u}\}$ is the nodal displacement vector and $\{\boldsymbol{F}\}$ is the nodal force vector. After enforcing displacement boundary conditions, the above equation is time integrated by replacing the differential operators with finite difference coefficients. The details of the assembly of elemental matrices to obtain the assembled ***[M]*** and ***[K]*** is given in [34] and not repeated here. Although there are many time integration scheme is reported in the literature**,** *Newmark-β central difference method* [34] is the most common method *and* the Figure 5.2 explains this method of obtaining dynamic displacements and velocities.

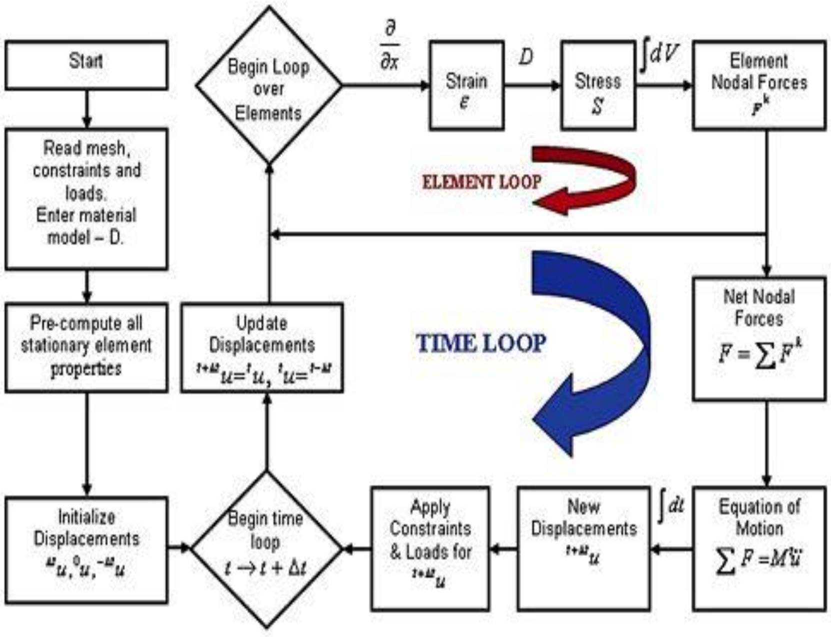

Start
Read mesh, constraints and loads. Enter material model – D.
Pre-compute all stationary element properties
Initialize Displacements $^{\Delta t}u, ^{0}u, ^{-\Delta t}u$
Begin Loop over Elements
$\frac{\partial}{\partial x}$
Strain ε
D
Stress S
$\int dV$
Element Nodal Forces F^k
ELEMENT LOOP
TIME LOOP
Net Nodal Forces $F = \sum F^k$
Equation of Motion $\sum F = M\ddot{u}$
$\int dt$
New Displacements $^{t+\Delta t}u$
Apply Constraints & Loads for $^{t+\Delta t}u$
Begin time loop $t \rightarrow t + \Delta t$
Update Displacements $^{t+\Delta t}u = {}^{t}u,\ {}^{t}u = {}^{t-\Delta t}$

Figure 5.2 Finite Element Method Procedure [34]

For modelling guided waves using FEM on thin plates as used here in this thesis, we need to enforce zero stress (both normal and shear stresses) on the top and bottom surface. This can be accomplished by not enforcing any displacement boundary conditions on these surface nodes. The details of how this is modelled using commercial software COMSOL® is explained in the next section.

5.3 FEM Modelling using COMSOL ® Multiphysics

COMSOL ® Multiphysics is a software package that among its many functionalities lets us model guided waves in structures. The waves are modelled using the FEM described in the last section, where the governing equations to be solved are discussed previously in Chapter 3. In this section, the details of the simulation model are discussed. A 2D FEM model is developed to mimic guided wave propagation in thin composite plates. A 16 layered unidirectional symmetric laminate has been considered in this case. A zoomed in version of the cross section of the geometry is shown in Figure 5.3. The material properties of the composite used here is given in table 5.1. The length of the composite is 2.5 m, while it is 2 mm thick, with each layer 0.125 mm thick.

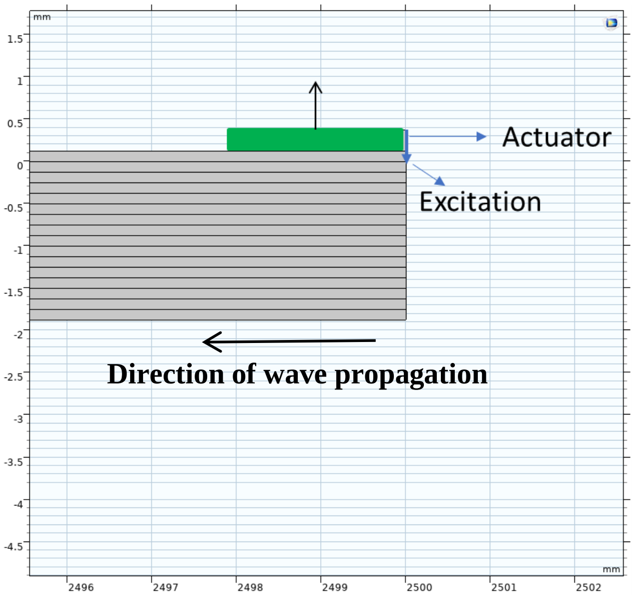
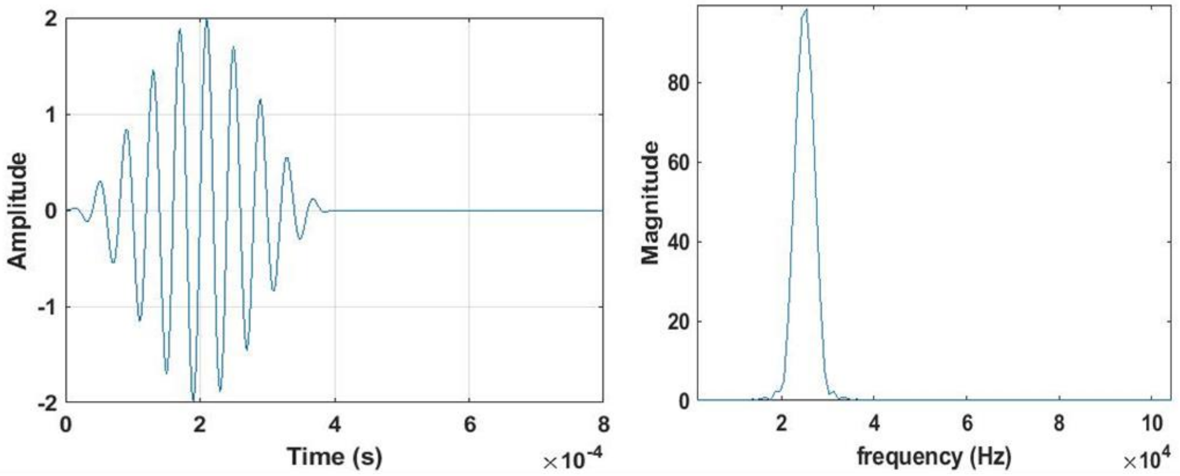


Figure 5.3 Cross sectional view of the sample geometry

As seen in Figure 5.3, an input disturbance is given at the top right of the sample. The disturbance is given normal to the direction of wave propagation. The input signal is a 7 cycle cosine function passed through a hanning window. The central frequency of the signal is 25 KHz. Figure 5.4 shows the time domain and frequency domain representation of the input disturbance *e*, while Equation (5.2) describes the disturbance mathematically.

$$e = Sin(\omega t) * \left(1 - \frac{cos(\omega t)}{n}\right) \tag{5.3}$$



(a) (b)

Figure 5.4 (a) Time domain representation of the input tone burst (b) Frequency domain representation of the input tone burst

As discussed before, the finite element mesh and its related parameters are important parameters to any numerical study. In this particular study, we have used the quadrilateral (QUAD) to mesh the whole domain [36]. A fine mesh is used where each mesh element is of the size 0.75 mm.

Figure 5.5 shows the meshing of the whole domain for solving the governing PDE using FEM. A time dependent analysis using the direct linear solver MUMPS available in COMSOL ® is used to simulate the ultrasonic guided wave in the composite. The total time interval for the simulation is 1200 μs while the time step is 2 μs. The stress free boundary on the top and bottom surfaces are enforced as explained in the previous section.

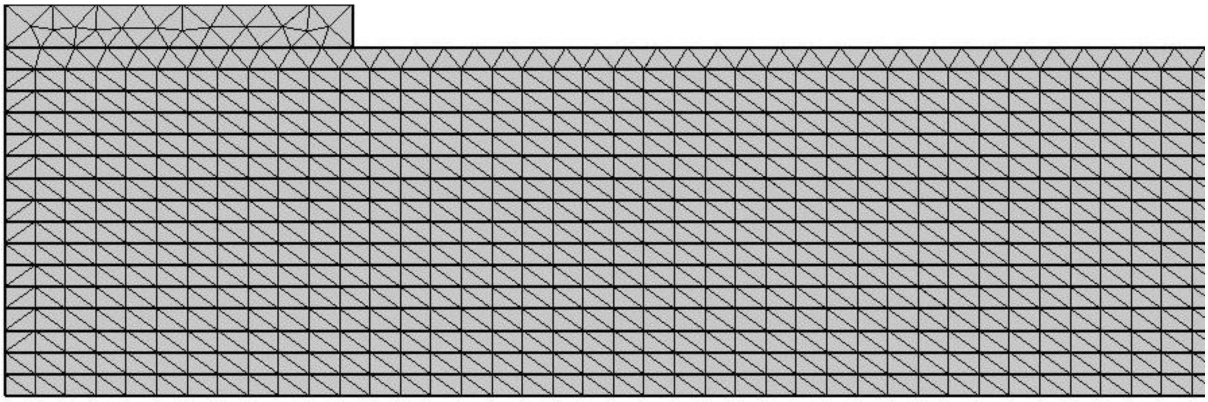
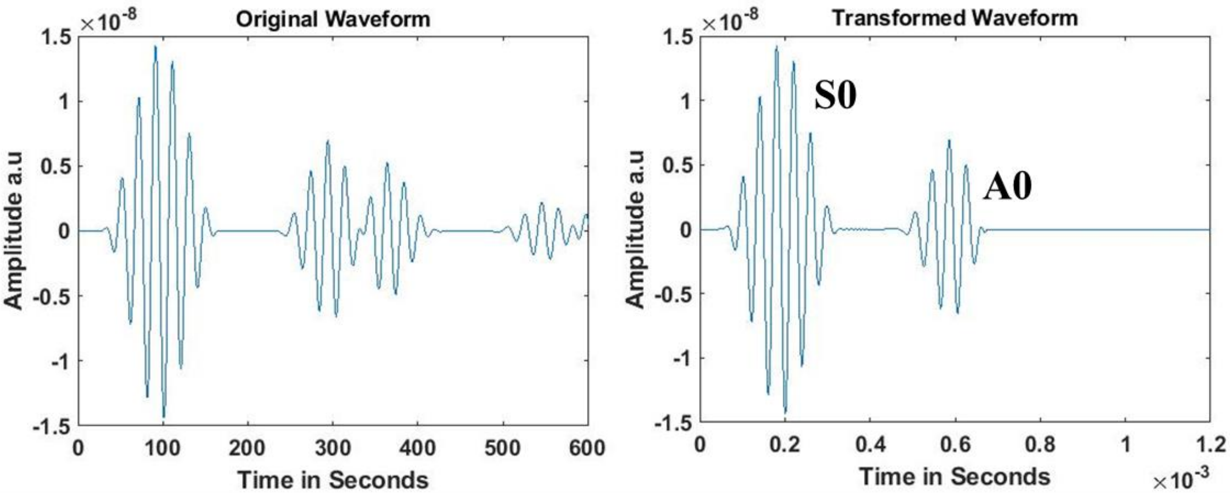


Figure 5.5 Finite Element Mesh for the composite laminate

Table 5.1 Material Properties of a transversely isotropic lamina

ρ (kg/m^3)	E_1 (GPa)	E_2 (GPa)	ν_{12}	ν_{23}	G_{12} (Gpa)
1750	173.1	12.5	0.31	0.38	2

The responses are received at a location 500 mm away from the actuator. When the composite laminate is excited, the two fundamental modes A0 and S0 are generated and propagate through the structure. The edges of the composite are artificially loaded with low reflecting conditions to avoid boundary reflections to the best extent possible. Figure 5.6a shows the received waveform and Figure 5.6b shows the filtered waveform where the fundamental wave modes are isolated. The filtering is done using the knowledge of the group velocities of the fundamental wave modes in the composite laminate.



(a)

(b)

Figure 5.6 (a) Received response at 500 mm from the actuator (b) Filtered received response to isolate the fundamental wave modes

Through multiple research advancements and increase in computational capabilities, the numerical FEM approach can produce very accurate solutions that are close to the ground truth. They indeed find multiple uses in various applications. But due to the way FEM is built upon, the computational requirement are way too high. For example, if one is dealing with a very fine mesh (for problems involving in high frequency excitations), thereby increasing the number of elements which directly increases the computation time and memory. Therefore, it was important to work a way around this and Spectral Finite Element Method (SFEM) provides very accurate alternative but with very small computational requirements. The next section describes the basic working principles of SFEM and some fundamental derivations based on it

5.4 Spectral Finite Element Method

The *SPECTRAL FINITE ELEMENT METHOD* (**SFEM**) is a method based on FEM principle. This method is typically used to solve dynamic wave propagation problems. In this method, the field variable in the governing PDE is transformed into frequency domain using *Fast Fourier Transforms* (FFT), which is a numerical version of the *Discrete Fourier Transforms* (DFT). The DFT is performed by a FFT algorithm popularly known as the *Cooley Turkey* algorithm [37] and is expressed as described in Equation (5.3). When transformed to the frequency domain, for a 1D realization, the PDE's become a set of ODE's (Ordinary Differential Equations) having frequency (sampling frequency in FFT) as the primary variable and in most cases the resulting equations are ODEs with constant coefficients. The finite element procedure is then applied to the transformed governing equations by solving them in the wavenumber space (k space) to find the exact solution for the ODEs. SFEM employs this exact solution as an interpolating function for element formulation. The constants of integration are driven to satisfy the boundary conditions, and therefore all the requirements are satisfied at every discrete frequency. Using the Inverse Fast Fourier Transform (IFFT), the time domain representation of the solution can be obtained. The SFEM procedure is shown as flow chart shown in Figure 5.7

There is more to SFEM to than just transforming the governing PDE into frequency domain and solving the ODE. In essence, it can be categorized as a type of FEM as methods like the Ritz method that can be employed in frequency domain to obtain the structural stiffness matrix similar to the stiffness matrix obtained in FEM. In essence, it can be categorized as a type of FEM [38].

$$u(x,y,z,t) = \sum_{n=0}^{N-1} \hat{u}(x,y,z,\omega_n) e^{-j\omega t} \tag{5.4}$$

Where N is the number of FFT points, ω_n is the discrete circular frequency. The formulation of SFEM for 2D structural waveguides is more complex, because the governing PDEs can no longer be reduced to a set of simple ODEs when converting to the frequency domain. Here we will need the second Fourier Transform in one of the spatial direction to reduce the dimensionality of the problem to one dimension. In the next two sections, the formulation of spectral elements for a isotropic rod and a higher order Timoshenko beam is outlined, as these are the two formulations used in the thesis.

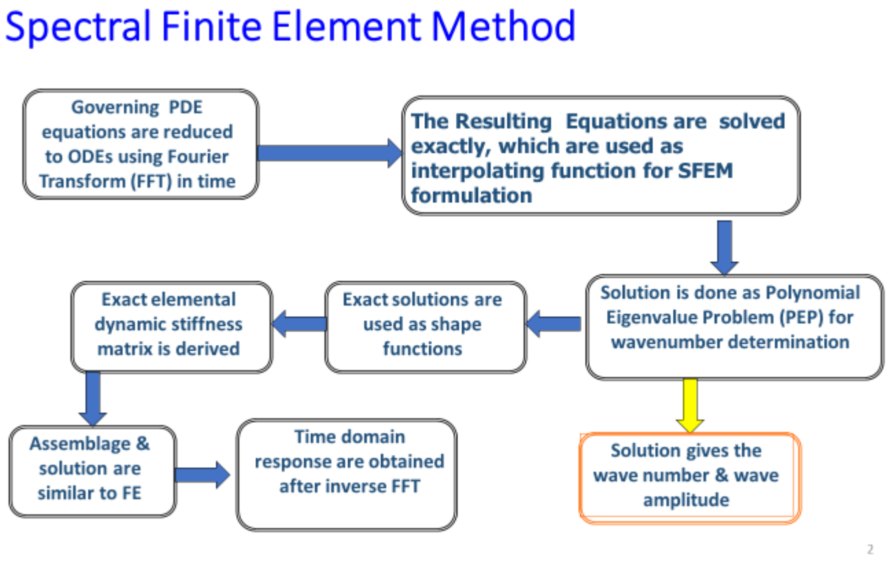

Spectral Finite Element Method
Governing PDE equations are reduced to ODEs using Fourier Transform (FFT) in time
The Resulting Equations are solved exactly, which are used as interpolating function for SFEM formulation
Solution is done as Polynomial Eigenvalue Problem (PEP) for wavenumber determination
Exact solutions are used as shape functions
Exact elemental dynamic stiffness matrix is derived
Assemblage & solution are similar to FE
Time domain response are obtained after inverse FFT
Solution gives the wave number & wave amplitude
2

Figure 5.7 SFEM Procedure of solving PDE

In the previous section we described time domain 2-D FEM procedure where in we got the low frequency guided waves (A0 and S0 modes) in a single analysis. SFEM procedure is normally used to analyse 1-D wave propagation (both A0 and S0 propagation). The question is how the equivalence of these two methods can be used to estimate guided wave modes and thereby use these results for material property identification. According to [34], the 2-D guided wave modes can be accurately obtained using higher order 1-D wave theories. Hence, we use here higher order Timoshenko beam theory to obtain accurate estimate of S0 mode, while we use elementary theory to obtain A0 mode. Reference [34] also shows that the accuracy of the A0 mode does not suffer much by using elementary rod theory. Hence, in the next section we will discuss in detail the formulation of Spectral Elementary Rod and Timoshenko beam element.

5.5 Non-dispersive Isotropic Rod: FFT Based Spectral Element Formulation

A rod element of length *L* model has two degrees of freedom u_1 and u_2 which define the axial the two axial boundary conditions at *x=0* and *x=L*. The corresponding forces at the two nodes are F_1 and F_2. The governing differential (PDE) equation for an isotropic homogeneous rod of density ρ and Young's Modulus *E* is

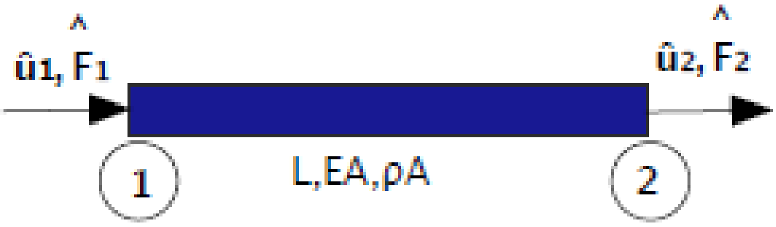


Figure 5.8 Spectral Isotropic Rod Element [19]

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \tag{5.5}$$

Where $u=u(x,t)$ is the axial displacement and $c^2=E/\rho$ is the square of the wave speed of the material. The force boundary condition is given by

$$F(x,t) = AE\frac{\partial u}{\partial x} \tag{5.6}$$

Where A is the cross sectional area of the rod and $F(x,t)$ is the axial force due to the axial displacement. The displacement (essential) boundary condition is the specification of the displacement u at the boundaries. Assuming a solution of the form

$$u(x,t) = \sum_{n=1}^{N} \hat{u}(x,\omega_n)e^{-j\omega_n t} \tag{5.7}$$

Substituting Equation (5.7) in Equation (5.5), the reduced governing ODE becomes

$$c^2\frac{\partial^2 u}{\partial x^2} + \omega_n^2\hat{u} = 0 \tag{5.8}$$

Whose solution is of the form $u_o e^{-jkx}$. Upon substitution in Equation (5.8), the discretized form of the ODE becomes

$$(-c^2k^2 + \omega_n^2)u_o = 0 \tag{5.9}$$

The wavenumber can be computed trivially in this case as $k_n=\pm w_n/c$ and for both mods, wave amplitude can be taken as 1. Thus the complete solution is,

$$\hat{u}(x,\omega_n) = c_1 e^{-jk_n x} + c_2 e^{-jk_n(L-x)} \tag{5.10}$$

Where c_1 *and* c_2 are coefficients to be determined using the boundary conditions, and L being the length of the element (isotropic rod). Following a derivation as shown in [1], we obtain a relationship between the nodal forces and nodal displacements as given by

$$\begin{Bmatrix} \widehat{F_1} \\ \widehat{F_2} \end{Bmatrix} = T_2 T_1^{-1} \begin{Bmatrix} \widehat{u_1} \\ \widehat{u_2} \end{Bmatrix} \tag{5.11}$$

Where T_2 and T_1 are defined as

$$T_1 = \begin{bmatrix} e^{-jk_n x_1} & -e^{jk_n x_1} \\ -e^{-jk_n x_1} & e^{jk_n x_1} \end{bmatrix}$$

$$T_2 = \begin{bmatrix} e^{-jk_n x_2} & -e^{jk_n x_2} \\ -e^{-jk_n x_2} & e^{jk_n x_2} \end{bmatrix} \tag{5.12}$$

Hence, the dynamic stiffness matrix (DSM) for the rod at frequency w_n is

$$D_{SFEM} = T_2 T_1^{-1} \tag{5.13}$$

FFT based SFEM requires the formulation of one noded infinite segment called the throw off element for good time resolution. This is obtained by removing the reflected coefficients from the solution. The solution to the throw off elements and the dynamic stiffness for the throw off elements is given in Equations (5.14) and (5.15)

$$\hat{u}(x, \omega_n) = c_1 e^{-jk_n x} \tag{5.14}$$

$$\widehat{F_1} = EAjk_n\widehat{u_1} \tag{5.15}$$

5. 6 Spectral Element Formulation of Timoshenko Beams

The guided wave propagation in a 1D composite laminate was shown in Section 3.5. There, we assumed a 1D waveguide that satisfies both the elementary rod and beam theory. In the beam theory, the slope *θ* is equal to *dw/dx*. However, in the Timoshenko theory, *θ* is assumed as an independent variable. Timoshenko beam theory captures the beam dynamics more accurately with the introduction of shear deformations. It is therefore preferred to generate responses that mimic the S0 mode. Spectral element formulation requires governing PDE, which for Timoshenko beam is given by

$$GAK\frac{\partial}{\partial x}\left[\frac{\partial w}{\partial x} - \theta\right] = \rho A\ddot{w} \tag{5.16}$$

$$EI\frac{\partial^2\theta}{\partial x^2} + GAK\left[\frac{\partial w}{\partial x} - \theta\right] = \rho I\ddot{\theta} \tag{5.17}$$

where *w(x,t)* is the transverse deformation, *θ(x,t)* is the rotation of the mid-plane of the beam, *G* is the shear modulus, *A* and *I* are the area and area moment of inertia of the cross section, *K* is the shape factor that takes care of anomalies in the shear stress deformation across the depth of the cross section. For rectangular cross section, *K=0.85.* All the quantities with two dots on them represent the acceleration. These equations are supplemented by the following force boundary conditions

$$V = \; GAK\left[\frac{\partial w}{\partial x} - \theta\right],\; M = EI\frac{\partial \theta}{\partial x} \tag{5.18}$$

Where *V* is the shear force and *M* is the bending moment. Assuming the solution of the form

$$w(x,t) = A_0 e^{i(kx-\omega t)},\; \theta(x,t) = B_0 = e^{i(kx-\omega t)} \tag{5.19}$$

Substituting Equation (5.19) in Equation (5.18), we get fourth order equation for wavenumber *k*, which is given by

$$k^4 - k^2\left[\frac{1}{c_s^2} + \left(\frac{Q^2}{c_{0q}^2}\right)\right] - \left[\left(\frac{\omega^2}{c_{0q}^2}\right) - \left(\frac{Q^4}{c_s^4 c_{0q}^4}\right)\right] = 0 \tag{5.20}$$

Where, $c_{0q} = \sqrt{\frac{EI}{\rho A}},\; c_s = \sqrt{\frac{GAK}{\rho A}},\;\; Q = \sqrt{\rho A/\rho I}$

Solving Equation (5.20), we get four modes of the form $k_1, k_2, -k_1, -k_2$ and the complete solution at frequency ω_n is given by

$$\begin{Bmatrix} \widehat{w}\,(x,\omega_{n)} \\ \hat{\theta}(x,\omega_n \end{Bmatrix} = \textstyle\sum_1^4 C_m \begin{Bmatrix} R_{1m} \\ R_{2m} \end{Bmatrix} e^{-ik_m x} \tag{5.21}$$

Where C_m is an unknown coefficient to be determined with four boundary conditions at the two nodes of the Timoshenko beam. Evaluating Equation (5.21) at the two ends of the beam situated at *x=0* and *x=L*, where *L* is the length of the beam with $R_{1m}$and R_{2m} being the amplitude ratios at four wavenumbers$k_1, k_2, -k_1, -k_2$, we get $[\boldsymbol{T_1}]$

$$\mathbf{T}_1 = \begin{bmatrix} R_{11}e^{-jk_1x_1} & R_{12}e^{-jk_2x_1} & R_{13}e^{-jk_3x_1} & R_{14}e^{-jk_4x_1} \\ R_{21}e^{-jk_1x_1} & R_{22}e^{-jk_2x_1} & R_{23}e^{-jk_3x_1} & R_{24}e^{-jk_4x_1} \\ R_{11}e^{-jk_1x_2} & R_{12}e^{-jk_2x_2} & R_{13}e^{-jk_3x_2} & R_{14}e^{-jk_4x_2} \\ R_{21}e^{-jk_1x_2} & R_{22}e^{-jk_2x_2} & R_{23}e^{-jk_3x_2} & R_{24}e^{-jk_4x_2} \end{bmatrix} \tag{5.22}$$

This matrix gives relationship between unknown coefficients C_m and nodal degrees of freedom, $\{\widehat{w}_1 \quad \hat{\theta}_1 \quad \widehat{w}_2 \quad \hat{\theta}_2\}$. Similarly, Forces and Moments given by Equation (5.18) is evaluated at two nodes located at x=*0* and *x=L* as $V(x=0) = -\hat{V}_1, V(x=L) = \hat{V}_2, M(x=0) = -\widehat{M}_1, M(x=L) = \widehat{M}_2$

Evaluating the forces, we can express unknown coefficients $C_m$in terms of nodal degrees of freedom as

$$\begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{Bmatrix} = \begin{bmatrix} T_{2_{11}} & \cdots & T_{2_{14}} \\ \vdots & \ddots & \vdots \\ T_{2_{41}} & \cdots & T_{2_{44}} \end{bmatrix} \begin{Bmatrix} \hat{V}_1 \\ \hat{M}_1 \\ \hat{V}_2 \\ \hat{M}_2 \end{Bmatrix} \tag{5.23}$$

Where **[$\boldsymbol{T_2}$]** matrix is given by

$$T_2(1,m) = -GAK\big(-ik_mR(1,m) - R(2,m)\big), \qquad T_2(2,m) = -EI\big(-ik_mR(2,m)\big)$$

$$T_2(3,m) = GAK\big(-ik_mR(1,m) - R(2,m)\big),\ T_2(4,m) = EI\big(-ik_mR(2,m)\big) \tag{5.24}$$

The Dynamic stiffness matrix$[\hat{K}]$ of the Timoshenko beam is then given by

$$[\hat{K}] = [T_2][T_1]^{-1} \tag{5.25}$$

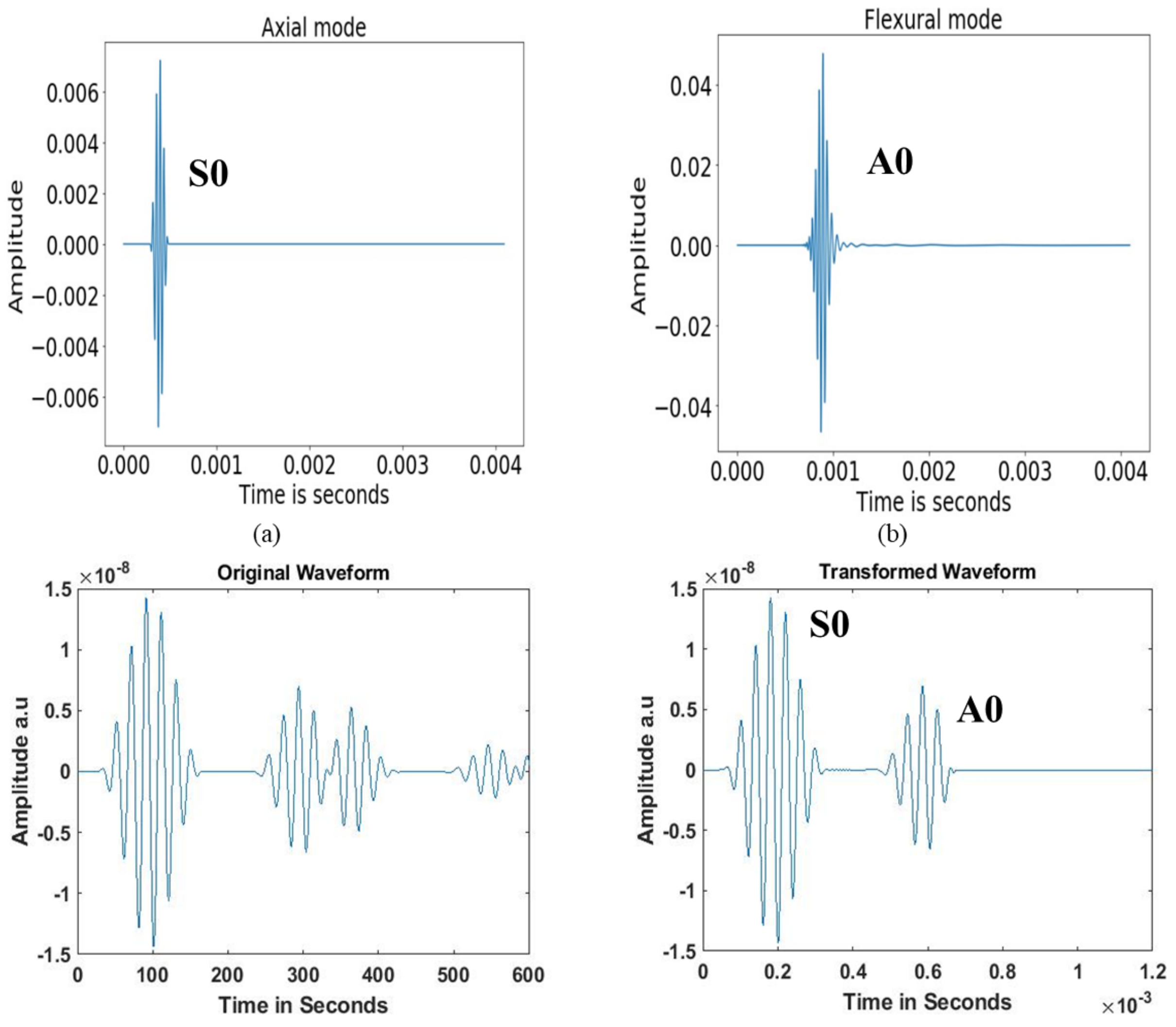
5.7 FEM vs SFEM: A comparison

A thin 2D composite plate was modelled using the FEM approach, while 1D waveguides were modelled using the SFEM approach. By using the elementary rod element in SFEM, we could mimic the A0 mode similar to the ones generated in the FEM approach, while the higher order Timoshenko Beam was used to generate the S0 mode. Since the FEM model was in two dimensions, both the modes are coupled. Using the SFEM on 1D waveguides, the modes are decoupled, and A0 and S0 modes can be obtained individually depending on the direction of incident loading (axial loading or transverse loading). This can be seen in the Figure 5.9. The velocities of the two modes obtained using the two different methods are calculated and is outline in Table 5.2.

Table 5.2 Wave velocities: SFEM vs FEM

Method	S0 Velocity (m/s)	A0 velocity (m/s)
SFEM	8696	1048
FEM	8602	1064
Error %	1.07	1.52

From Table 5.2, it is observed that the velocities of the two wave modes match very well for both the methods. Therefore, we can conclude that, at this particular frequency and by using just the fundamental wave modes A0 and S0, it is possibly to scale down a thin 2D laminate plate to a 1D waveguide. 1D waveguides can easily be modelled in SFEM to generate wave modes that mimic the A0 and S0 modes generated in the FEM scheme. This is particularly important as SFEM is computationally many orders better than FEM. The computation time for an FEM simulation in this case is about 30 mins, compared to a SFEM simulation that takes seconds to execute.



(a)

(b)

(c)

(d)

Figure 5.9 (a) and (b): A0 and S0 response using the SFEM method. (c) Overall raw waveform obtained from the COMSOL FEM simulation (d) Filtered waveform isolating the individual A0 and S0 reflections (filtering done using knowledge of group velocities of these two modes)

There are of course methods to optimize the FEM model to reduce the computation time, but even at the best performing level, any COMSOL simulation takes couple of minute's best which is still more compared to that of SFEM. It is hence important to be able scale down geometrically, in order to model guided waves using SFEM. This enables us to generate a much larger dataset that in turn can be used to train data hungry networks like 1DCNN and LSTMs to achieve much improved performance levels. Therefore, for the DL based approach, we use the SFEM approach to generate a large dataset to solve the inverse problem of materials characterization.

CHAPTER 6

SENSITIVITY ANALYSIS AND DATA UNIQUENESS

In this thesis, one of the main objectives was to build an overall framework for material property identification in composite structures, complete with signal processing, and other data handling algorithms. In this section, the different methodologies used are discussed. The overall flow of the research is also described in detail. Figure 1.6 (see Section 1.7) shows a flowchart detailing the flow of analysis methodology. Initially as part of the forward problem for the deep learning based approach (*DL based*), we model guided waves in a composite laminate using two different methods (Numerical FEM and Semi Analytical SFEM). For this process, the priori knowledge of the material properties of different composites used typically in the industry was considered. In the previous chapter, the advantages and disadvantages of the two different modelling techniques were discussed.

It was concluded that, for the problem of material characterization, it is most advisable to use SFEM to model guided waves. This allows us to collect or generate a large database of A0 and S0 waveforms for different combinations of material properties at a very small time. A large training dataset is always preferred to train deep neural networks in order to achieve better performance levels. Figure 6.1 shows how the A0 wave mode change for two different sets of material properties.

Note that unlike in typical wave propagation and structural health monitoring problems, which necessarily deals with high frequency loading (which in turn requires very very fine finite element mesh for analysis), material property identification analysis can be performed using only two guided wave modes (A0 and S0), which are low frequency modes. Hence, in all the analysis performed in this thesis, we use an input tone burst signal of 25 kHz. The main advantage of using such low frequency signals is that the FE analysis does not require too fine a mesh to capture the two guided wave modes.

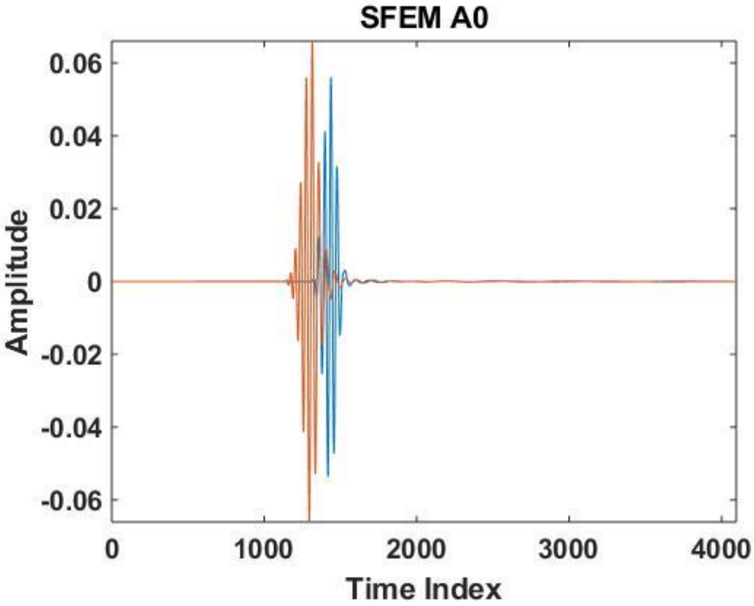


Figure 6.1 A0 waveforms for two different sets of material properties (**Blue**: ρ= 1000 kg/m^3, E_1= 50 GPa, E_2= 5 GPa, v_{12}= 0.25, v_{23}= 0.25,G_{12}= 2 GPa, **Orange:** ρ= 1750 kg/m^3, E_1= 111.53 GPa, E_2= 5 GPa, v_{12}= 0.25, v_{23}= 0.25, G_{12}= 2 GPa)

Using the two numerical models, and at the frequency range of 25 KHz, it is not possible to estimate or identify the shear modulus G_{12} (Section 6.1). Therefore, we propose an alternative strategy to predict shear modulus G_{12}. This is done by using software called *DC*® to generate and create a dataset using group velocity dispersion curves for different sets of material properties [39]. The dispersion plot data for three modes A0, S0 and the *shear horizontal mode SH0* are directly fed into a Dense Neural Network to predict the material properties. By incorporating the SH0 mode, the framework extends its capability to predict shear modulus G_{12}. Similarly the deep models like the 1DCNN and LSTMs are trained on a large dataset collected using the SFE method to predict on the lamina properties. Two key challenges that come into picture in the process of data collection are:

- What material properties to change every forward pass?
- How unique is each set of A0 and S0 waveform? (SFEM Method)

In the next few sections, we try to answer the two above questions by using Sensitivity Analysis, and data uniqueness checks.

6.1 Sensitivity Analysis

Sensitivity analysis is an important part of any inverse problem. Several previous work revolving around solving materials characterization as an inverse problem have used sensitivity analysis to different effects [15][40][41][42]. The primary reason of this analysis with respect to this thesis is:

1. To verify whether the five material properties and density independently have any effect on the A0 and S0 wave velocities.
2. If they do have an effect, quantify it using some means to gain a better understanding.

This is important because the training datasets are collected by perturbing material properties every forward pass to collect the respective A0 and S0 wave forms. By this analysis, we aim to establish which parameter changes wave velocities the most, thereby adjusting our strategy to collect data accordingly. This is useful from the point of view of training a Convolutional Neural Network to learn the inverse problem effectively. The wave velocities are focussed as the central part of this study, as they would be the most likely feature to change for different sets of material properties.

One of the parameter is varied in a range and the effect on the velocities is calculated while taking the mean, minimum or maximum of other non-involved parameters. For example, density is varied from 400-3200 kg/m^3 while other parameters are set at E_1 = 220 GPa (mean of the range 40-400 GPa), E_2 = 41.5 GPa (mean of the range 3-80 GPa), v_{12} = 0.35 (mean of the range 0.2-0.5), v_{23} = 0.35 (mean of the range 0.2-0.5), G_{12} = 6 GPa (mean of the range 2-10 GPa). Secant Sensitivity is used to quantify the sensitivity of a parameter to the wave velocities. It is described as:

$$\delta = \frac{C_e - C_s}{P_e - P_s} \tag{6.1}$$

Where, C_e is end value of velocity in m/s for lamina property P_e and C_s in the start value velocity for P_s.

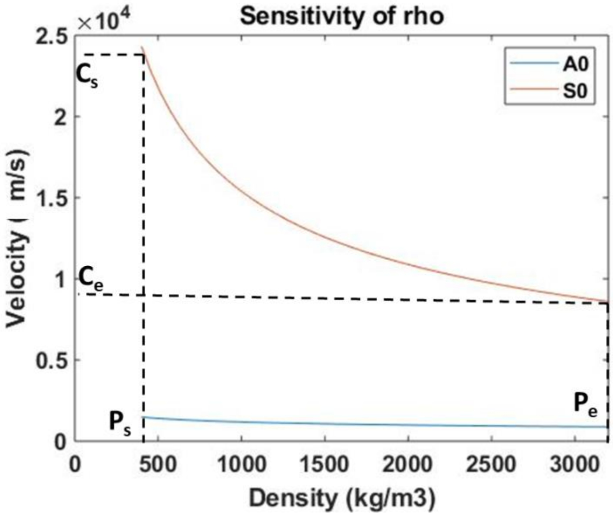
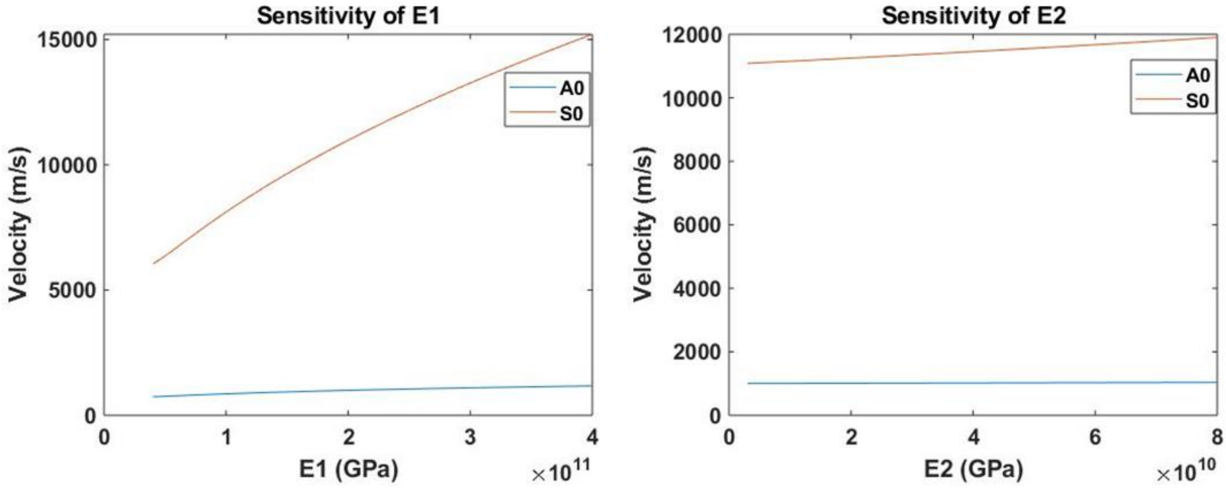


Figure 6.2 Sensitivity of density (rho) on A0 and S0 wave velocities

Figure 6.2 shows the effect of density on A0 and S0 wave velocities. Figure 6.3 shows the sensitivity of Young's Modulus E_1 and E_2 and Figure 6.4 shows the effect of Poisson's ratio and the shear modulus G_{12} on the wave velocities. The sensitivity values are summarized in Table 6.1.



(a)

(b)

Figure 6.3 (a) Sensitivity of Young's Modulus E1 on A0 and S0 wave velocities (b) Sensitivity of Young's Modulus E2 on A0 and S0 wave velocities

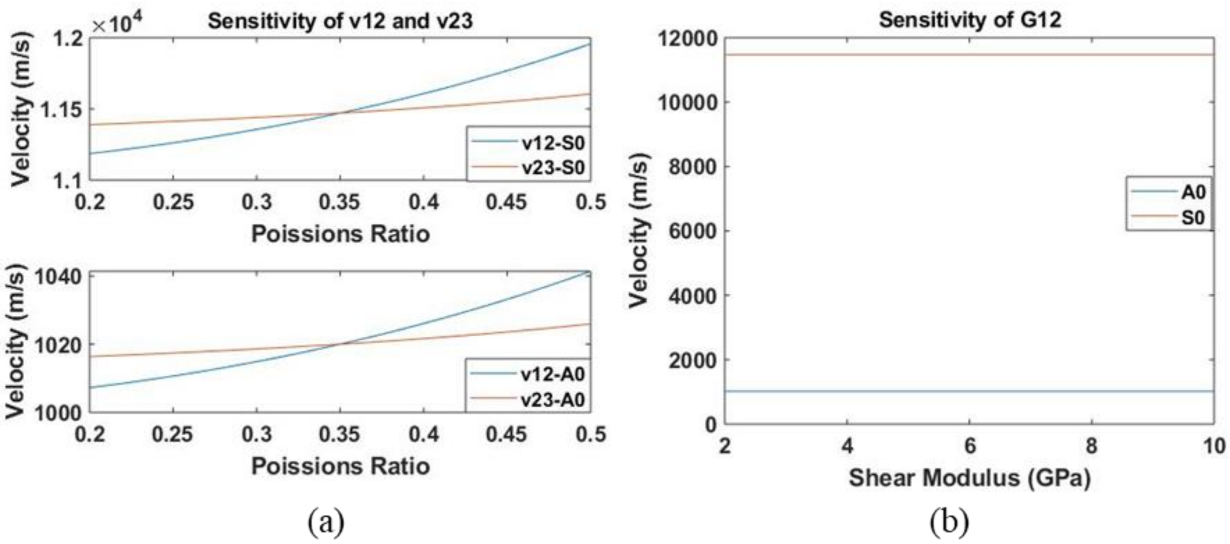


Figure 6.4 (a) Sensitivity of Poisson's ration v_{12} and v_{23} on A0 and S0 wave velocities (b) Sensitivity of Shear Modulus G_{12} on A0 and S0 wave velocities

Table 6.1 Secant Sensitivity Values

Parameter	Secant Sensitivity
ρ	**δ_{S0}=5.61** **δ_{A0}=-0.215**
E_1	**δ_{S0}=24.8** **δ_{A0}=1.20**
E_2	**δ_{S0}=10.61** **δ_{A0}=0.47**
v_{12}	**δ_{S0}=2569** **δ_{A0}=113.7**
G_{12}	**δ_{S0}=0** **δ_{A0}=0**

The following conclusions can be made from the sensitivity analysis conducted above:

1. The parameters are more sensitive to the S0 mode compared to the A0 mode. This can be seen both in the above figures, and in Table 6.1.
2. The wave speeds are more sensitive to E_1 compared E_2.
3. The wave speeds are most sensitive to the Poisson's ratio. Particularly, v_{12} seems to have the biggest effect on the wave speeds
4. G_{12} has no effect on the wave velocities and therefore, it is conceptually not possible to predict G_{12} from a neural network using S0 and A0 time histories as inputs.

Therefore, after this analysis it was decided to not perturb shear modulus G_{12} as it has no effect on the speeds. The four material properties and density were perturbed and its respective A0 and S0 wave modes were collected using the SFEM model outlined in the previous chapter.

6.2 Data Uniqueness

The forward system is a non-unique system. For different sets of material properties, it is possible to get similar A0 and S0 waveforms. This can adversely affect the performance of any Deep Learning algorithm, for there exists similar A0 and S0 waveforms for different sets of material properties. This is similar to one when there is a single equation but two different solutions to it. In these cases, it becomes impossible to find one unique solution. Therefore, it is important to clean the training dataset. This is achieved by having a uniqueness check before training. Once again, the velocity of the wave modes are analysed and training sets (A0 and S0 waveforms) where A0 and S0 waves with same velocities are same are deleted. This then ensures that every resulting training dataset is unique to a particular set of material properties.

Originally, more than 10000 training sets were generated for more than 10000 different sets of material properties using the SFEM model. After the uniqueness check, only 2719 unique training sets were found. Figure 6.5 shows the A0 and S0 velocities of an example training set that is 512 samples big. After the uniqueness check, the training dataset reduced to a size of 88 unique A0 and S0 pairs.

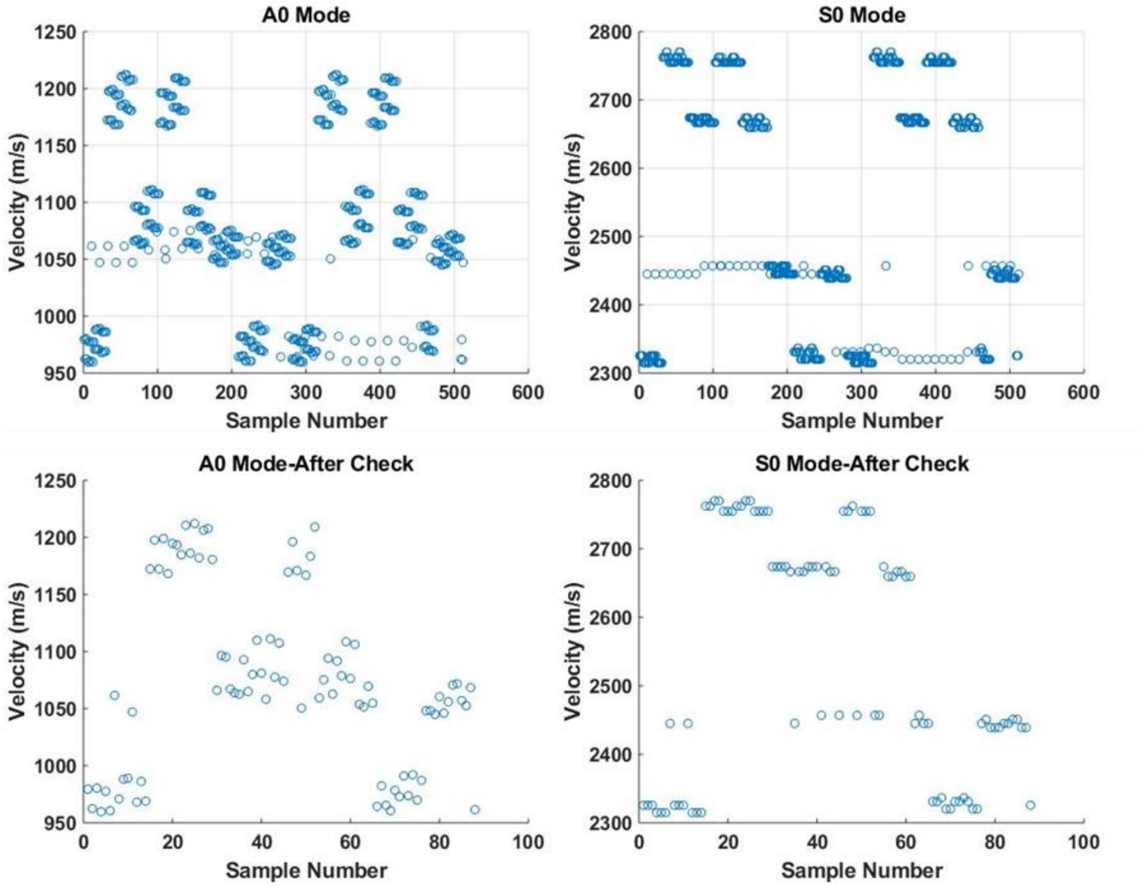


Figure 6.5 Uniqueness check for a dataset that had 512 samples originally but trimmed down to 88 after the check

CHAPTER 7

STUDY PARAMETERS FOR THE IDENTIFICATION OF ELASTIC PROPERTIES OF A COMPOSITE

In this thesis, a complete framework with inter linking data analysis algorithms and different learning models are explored for solving the inverse problem of material property identification. In particular, two deep architectures i.e. there 1D Convolutional Neural Network and Long Short Term Memory Networks are used, while a much shallower ML based Dense Neural Network is also used in the framework. The data generation and collection, and the study parameters for the DL based and ML based approach are different and are highlighted in the next two sections.

7.1 Study Parameters: DL Based Approach

The DL based approach uses a dataset created by the Spectral Finite Element Method. The SFEM method was finalized based on the advantages it has as discussed in section 5.7. Sensitivity Analysis (Section 6.1) is used to dictate the way data is generated. A dataset consisting of more than 10,000 pairs of A0 and S0 histories are generated for 10,000 different sets of material properties. The data uniqueness check is used to filter the created dataset of repeated and redundant data (Section 6.2). After this process, the size of the dataset is reduced to 2719 unique pairs of A0 and S0 time histories. The composite considered is a 16 layered unidirectional laminate that is transversely isotropic. The thickness of the composite laminate is 2 mm and each of its 16 layers has the same material properties and same thickness. The created dataset is used to train the two main deep architectures commonly explored in the field i.e. the 1DCNN (Section 4.3) and the LSTM (Section 4.5). The model trained on noiseless data is used to predict the five material properties *(ρ, E_1, E_2, v_{12}, v_{23})* on data both in the presence and absence of noise of different levels.

7.2 Study Parameters: ML Based Approach

In the previous chapter (section 6.1), the limitation of the current DL based models in predicting Shear Modulus G_{12} is highlighted. Hence, we adopt an alternative Machine Learning based (ML based) approach that utilizes group velocity dispersion data directly as inputs to a multi layered Dense Neural Network (DNN). To obtain the group velocity dispersion data, an open source software called the DC® is used. This software is very similar to the very popular *DISPERSE*® software, which are both fundamentally developed to study the dispersion phenomena of guided waves in different materials and configurations. The basics of the dispersion phenomena and group velocity are introduced in Chapter 3 (Section 3.4).

In order to increase the complexity of the whole framework, three different composites layups ($[0\ 0]_s$, $[0\ 0\ 0\ 0]_s$, $[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]_s$) of three different thicknesses (1mm, 2mm and 3mm) are considered. All the composite layups considered are symmetric, unidirectional and transversely isotropic in nature (similar to the DL based approach). Eight commonly used composites are considered. The material properties of the eight composites are given in Table 7.1.

The data collection is done using the DC® software as previously mentioned. In particular the group velocity data as a function of frequency is taken for three different wave modes. The first two modes are the fundamental wave modes of a lamb wave i.e the A0 and S0 mode as used in the DL based approach. The third new mode incorporated in this method is the Shear Horizontal (SH0) mode. Including the physics of this particular mode enables us to also predict Shear Modulus G_{12}. The shear horizontal mode is another type of guided wave similar to lamb waves. A lamb wave acts in plane while the SH0 mode acts out of plane i.e. if the lamb waves act in x-y direction, the SH0 mode acts in the x-z direction. The shear mode is typically generated due to a transverse loading that brings about a bending motion, thereby affecting the shear modulus of the composite. A typical group velocity dispersion curve involving the three different modes is shown in Figure 7.1.

Table 7.1 Material Properties of the composites used for the ML based study

Composite	**ρ (g/cm^3)**	**E_1 (Gpa)**	**E_2 (Gpa)**	**G_{12} (Gpa)**	**v_{12}**	**v_{23}**
AS4M8502	1.55	144.65	9.636	6	0.299	0.289
SAERTEX7006919	1.454	119.99	7.25	6.02	0.322	0.447
SigrafilCE125023039	1.5	128.555	6.873	6.1	0.330	0.375
T300M914	1.56	139.817	10.052	5.7	0.313	0.478
T700M21	1.571	125.5	8.7	4.135	0.37	0.45
T700PPS	1.6	149.956	9.988	4.5	0.291	0.368
T800M913	1.55	152.137	6.653	4.2	0.252	0.543
T800M924	1.5	161	9.25	6	0.34	0.41

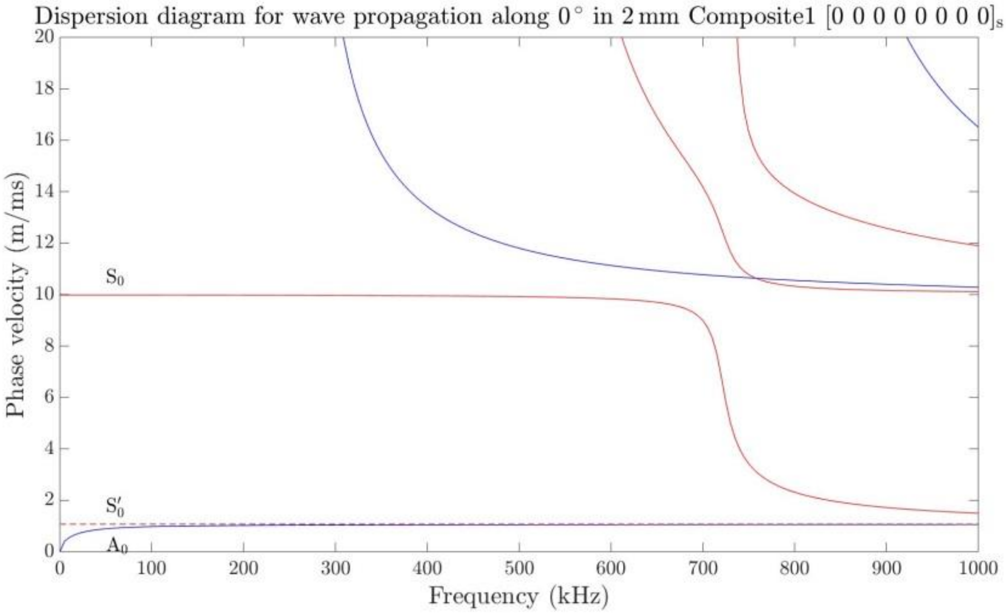


Figure 7.1 Group Velocity Dispersion Curve for a random composite with three modes (A0, SH0 and S0) highlighted

CHAPTER 8

RESULTS AND DISCCUSION

As previously described in the previous chapters, a large dataset containing sets of A0 and S0 waveforms were collected by perturbing the material properties every iteration. This constituted the forward problem, and in the inverse problem we reverse engineer to obtain the material properties using the collected dataset. In this chapter, the training and testing results of three different ML architectures (DNN, 1DCNN and LSTM) are discussed. The benefits and disadvantages of the models are highlighted and related comparisons are also established. The performance of these models in the presence of noise of various levels in the data is also studied. Finally, the limitations of the frameworks used are also discussed in detail. Figure 8.1 summarizes the research flow for the deep learning architectures (1DCNN and LSTM) used in this thesis. In the case of DNN, the data collection process is different as we have used the DC® software to generate group velocity curves for different sets of material properties (or different composites).

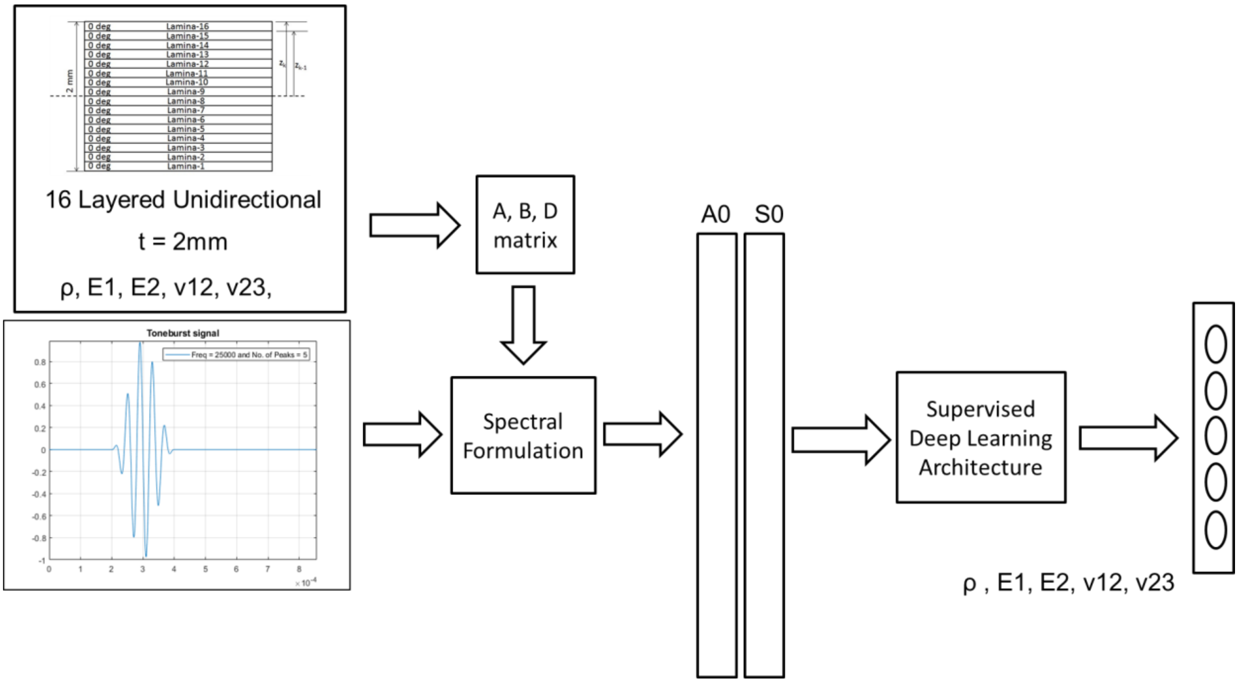


Figure 8.1 Research Flow for the two Deep Learning Models (1DCNN and LSTM)

The metrics used to evaluate the performance of the learning algorithms are the Mean Absolute Error (MAE), and the coefficient of determination (R^2 coefficient). Since the problem in hand can be viewed as a regression task, wherein we are trying to regress two inputs with reflections of certain amplitude to five absolute material property values. We are training our models on Google Colaboratory, which is a free cloud-based Python platform to train ML models. We are provided with Tesla T80 GPUs, Intel- Xeon CPU with 2.30 GHz clock speed (2 Cores), RAM of 12 GB and storage of 312 GB.

8.1 Training Results with 1D Convolutional Neural Network

The dataset collected using the SFEM consisted of 2819 unique A0, S0 waveforms for 2819 different composites. Training and validation samples are split in 9:1 ratio which gives 2448 training examples and 282 test (validation) examples. S0 and A0 are fed as two different input features (Figure 8.1) into the network and 5 lamina properties act as the output labels. A supervised regression based 3- Conv. + 2-Dense layered 1D-CNN (see Table 8.1) is trained to learn the relationship between ultrasonic guided waves and corresponding lamina properties.

Table 8.1 1DCNN Architecture

Layer	Units	Output	Parameters
Conv1D (LRelu)	(3,16)	(4096,16)	112
Maxpool	(2,)	(2048,16)	0
Conv1D (LRelu)	(3,32)	(2095,32)	1568
Maxpool	(2,)	(1022,32)	0
Conv1D (LRelu)	(3,64)	(1020,64)	6208
Maxpool	(2,)	(510,64)	0
Flatten	-	-	32,640
Dense (LRelu)	64	-	522,256
Dropout (0.05)	-	-	0
Dense (Linear)	5	-	85
Total parameters			530,229

The hyper parameters like learning rate, batch size, number of epochs etc. were set after some trial and error testing where a trade-off between the best performance and quick training time was desired. A learning rate of 0.01 and a batch size of 4 were found most suitable for this architecture. An Adam optimizer (β1=0.9 and β2=0.999) with MSE as loss function is used. The network is trained for 500 epochs. The training is performed keeping in mind the factors like:

1. Under fitting and overfitting.
2. A smoother loss curve
3. Generalization on test set.

The loss curve, MAE and R^2 coefficient curves are given in Figures 8.2, 8.3 and 8.4 respectively. It is seen from the above figures that the network is able to obtain a MAE of 32 and a R2 coefficient of 0.98. The model takes 4s/epoch to train and 1ms/sample for prediction. The MAE and the loss is a high value, because the target labels (material properties) aren't normalized. The material properties range from a scale of 10^{11} to 10^{-1}. If normalized, the loss and MAE values will be low and closer to zero.

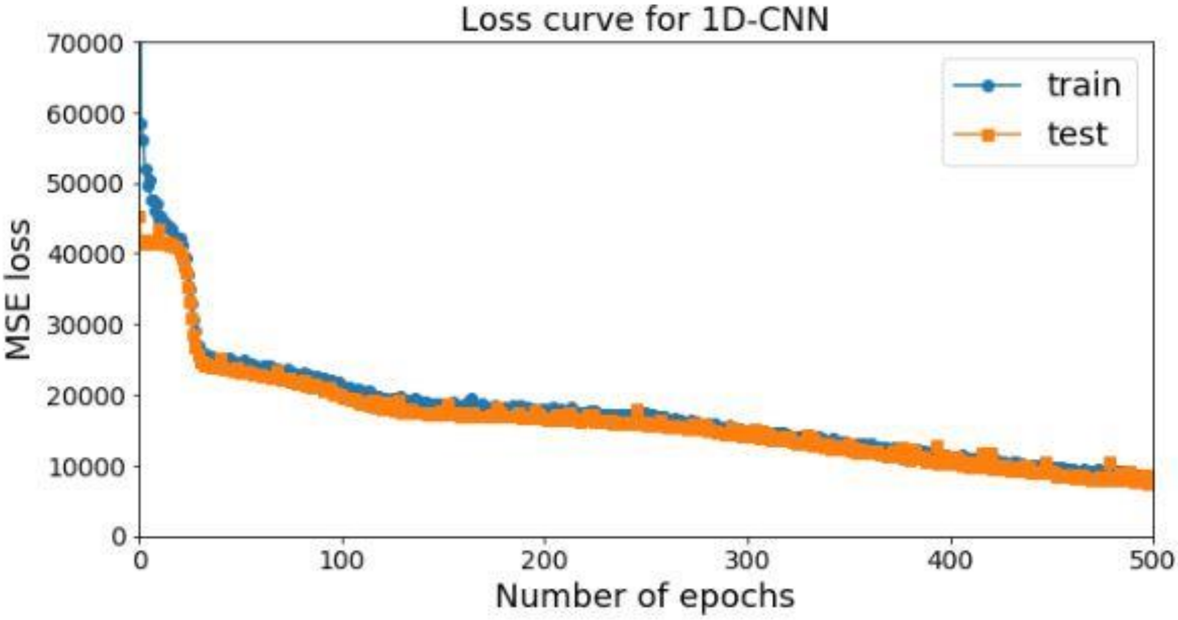


Figure 8.2 Loss curve: MSE vs number of epochs for 1DCNN

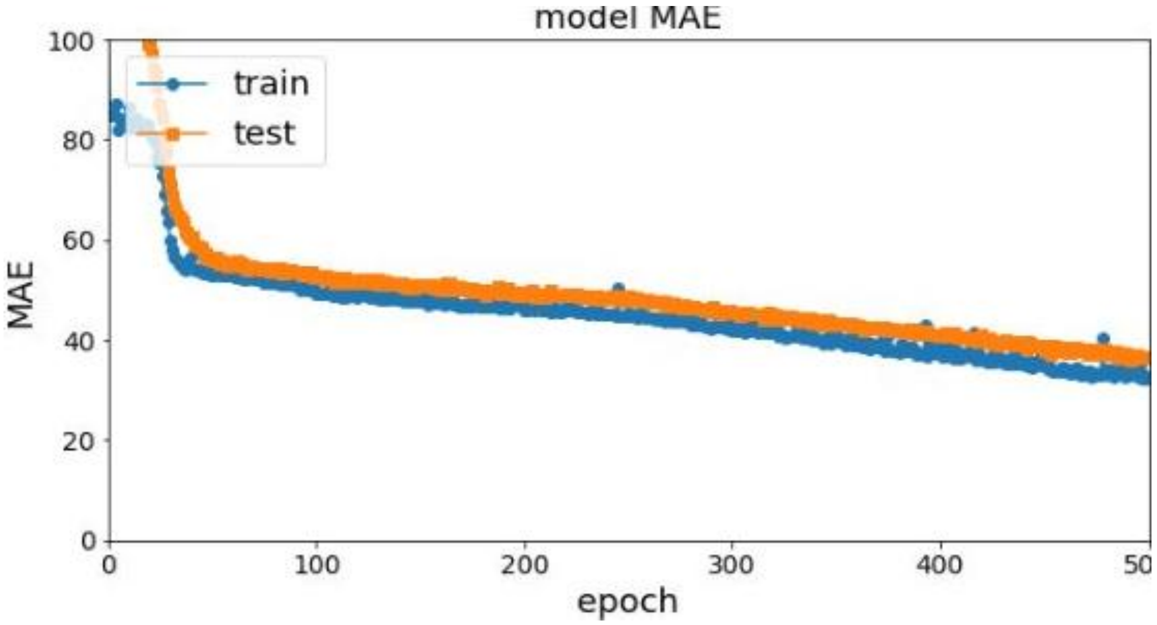


Figure 8.3 MAE curve: MAE vs number of epochs for 1DCNN

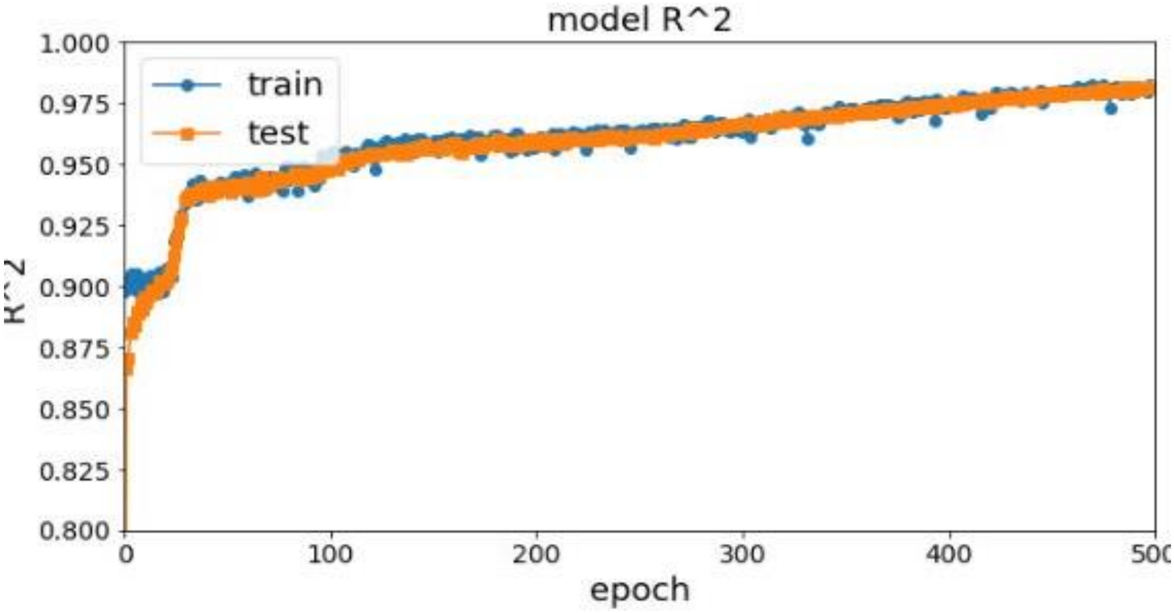


Figure 8.4 R^2 coefficient curve: R^2 coefficient vs number of epochs for 1DCNN

Once the 1DCNN is trained, a dataset containing previously unseen examples are fed in to the network to predict its respective material properties. The prediction results are presented in the Figure 8.5 and Table 8.2. Looking at Figure 8.5, the ideal scenario would be all scatter points lying on the linear $y=x$ line (green dotted). The network performs very well in predicting the material properties, but they are certain outliers or poor predictions. One such prediction is highlighted in red (see Table 8.2 for values). For the material property E_2 in the 2nd example, the predicted value is 28.8 GPa but the true value is 9 GPa. We have tried to use the model to predict on samples with almost similar material properties (only one/two parameter(s) are different) to verify the robustness of our architecture. All the prediction results are plotted on normalized values (to accommodate bigger modulus values and the smaller Poisson’s ratio)

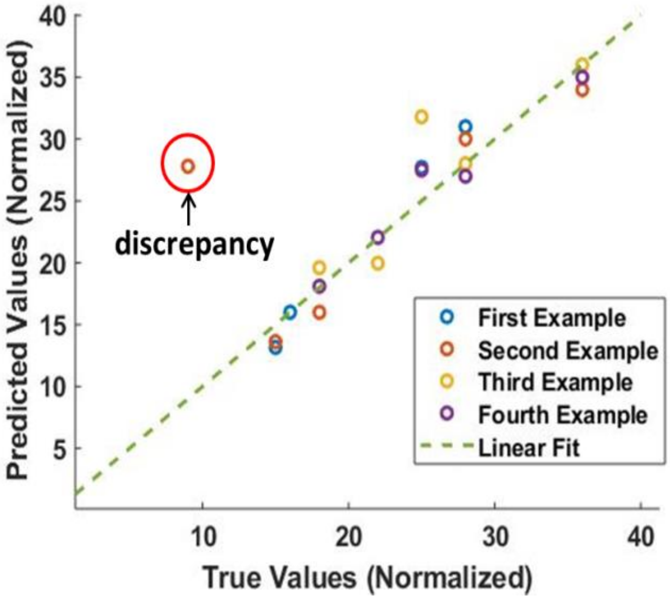


Figure 8.5 Prediction Results: True vs Predicted Plot for 1DCNN

Table 8.2 Prediction Results from the 1DCNN

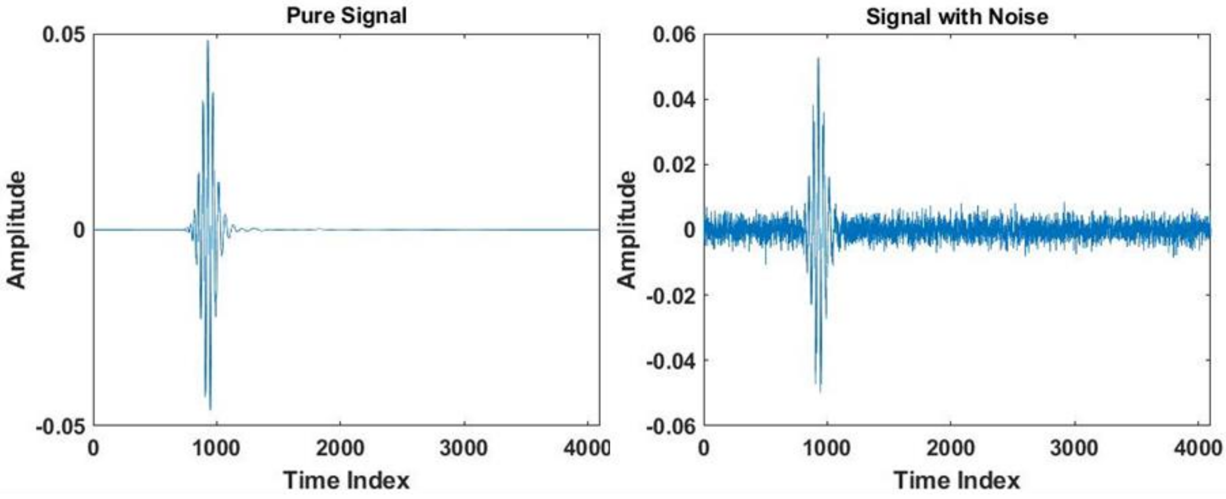
Example		ρ (kg/m^3)	E$_1$(Gpa)	E$_2$(Gpa)	v$_{12}$	v$_{23}$
Example 1	**True**	1500	160	25	0.28	0.36
	Predicted	1316	160	28.8	0.31	0.35
Example 2	**True**	1500	180	9	0.28	0.36
	Predicted	1362	160	28.8	0.30	0.34
Example 3	**True**	2200	160	25	0.28	0.36
	Predicted	1998	196	31.8	0.28	0.34
Example 4	**True**	2200	180	25	0.28	0.36
	Predicted	2205	181	28.5	0.28	0.35

The poor predictions can mainly be due to two reasons:

- The model is trained on about 2500 samples approximately, and as per our understanding, it is still a small dataset considering how data hungry convolutional neural networks are. The waveforms change very little physically for different composites (i.e. different material properties) and a larger dataset might be a viable solution to avoid poor predictions.

- One of the facts which are not highlighted in deep learning literature is the limitation of regression-based neural networks in extrapolation. It is observed that neural networks do not predict well on a data point that lies outside the range of training or towards the extreme points of the training range. For example, the range of E_2 is between 5-50 GPa in training and the predictions are not good on 9 GPa (near to one the extremes of the range i.e. 5 GPa) as compared to the prediction on 25 GPa.

Currently, the training and predictions have been on noiseless data. But in no practical consideration, one can expect no noise. Experimental data from guided wave testing typically have many uncertainties that manifest itself into noise and corrupt the original signal. Therefore, any framework that aims to work with such signals should be robust to noise. There are two ways to make a framework/model robust.



(a) (b)

Figure 8.6 (a) Original Signal (b) Original signal corrupted with a noise of SNR5

The first method should be to use a model trained on noiseless data to predict on noisy data. If the performance is satisfactory, the trained model is already robust to noise. If not, one would have to include noisy data in the training process itself. In this thesis, we use the former and try to leverage a model trained on noiseless data to predict on data with different levels of noise. Additive White Gaussian Noise (AWGN) of different levels (SNR5, SNR10 and SNR20) was added to the original signal to create three separate datasets on which the trained model will predict on. Figure 8.6 shows how the original signal is corrupted with a noise of SNR5. Figures 8.7, 8.8 and 8.9 show the prediction results on data with noise levels of SNR20, SNR10 and SNR5 respectively.

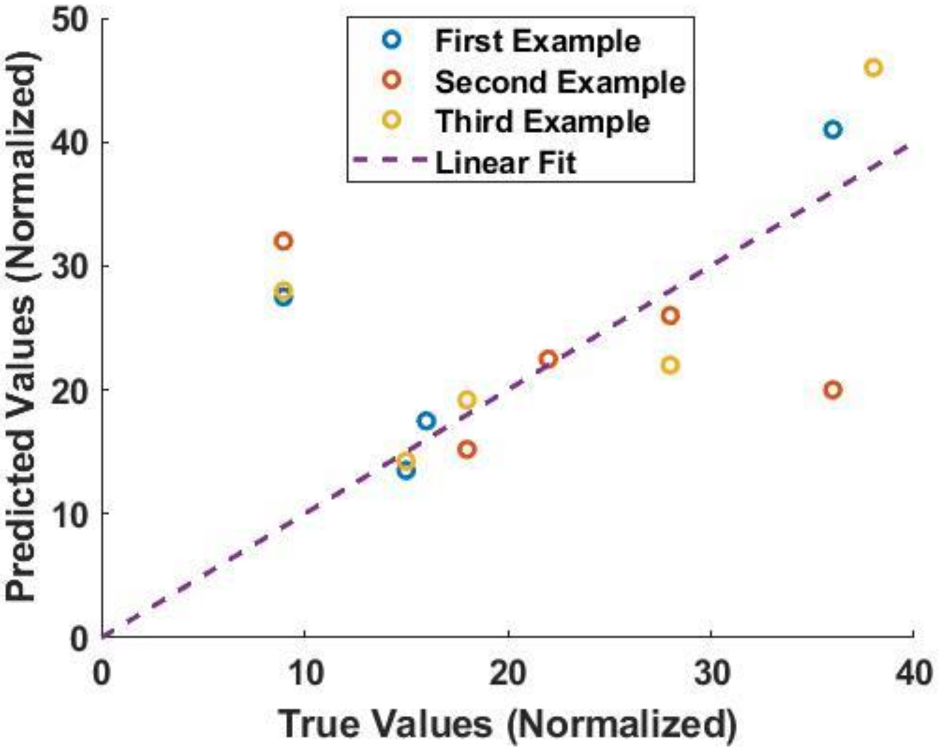


Figure 8.7 Prediction results on data with noise of SNR20 for 1DCNN

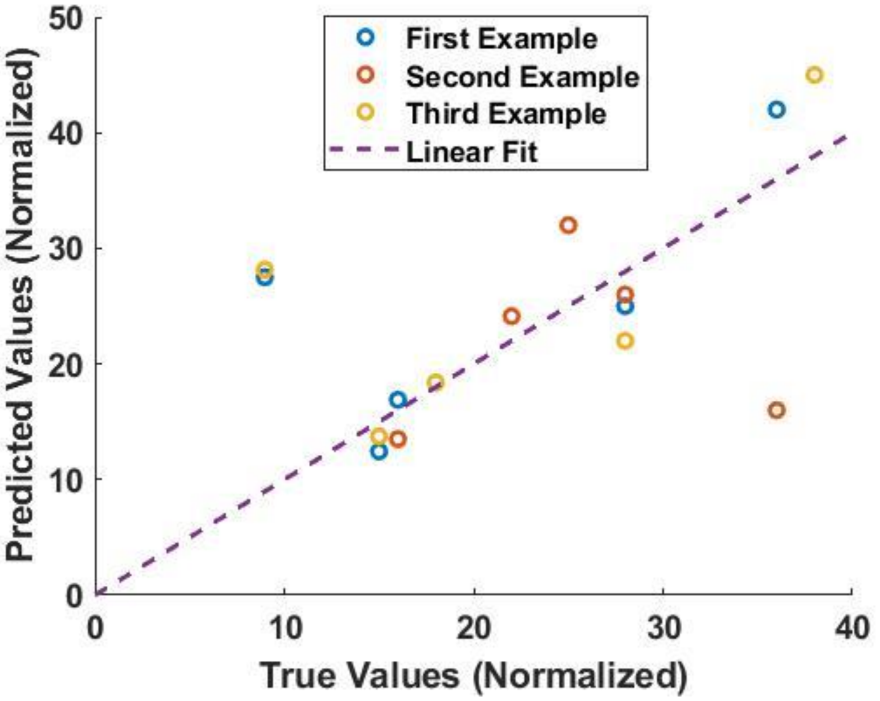


Figure 8.8 Prediction results on data with noise of SNR10 for 1DCNN

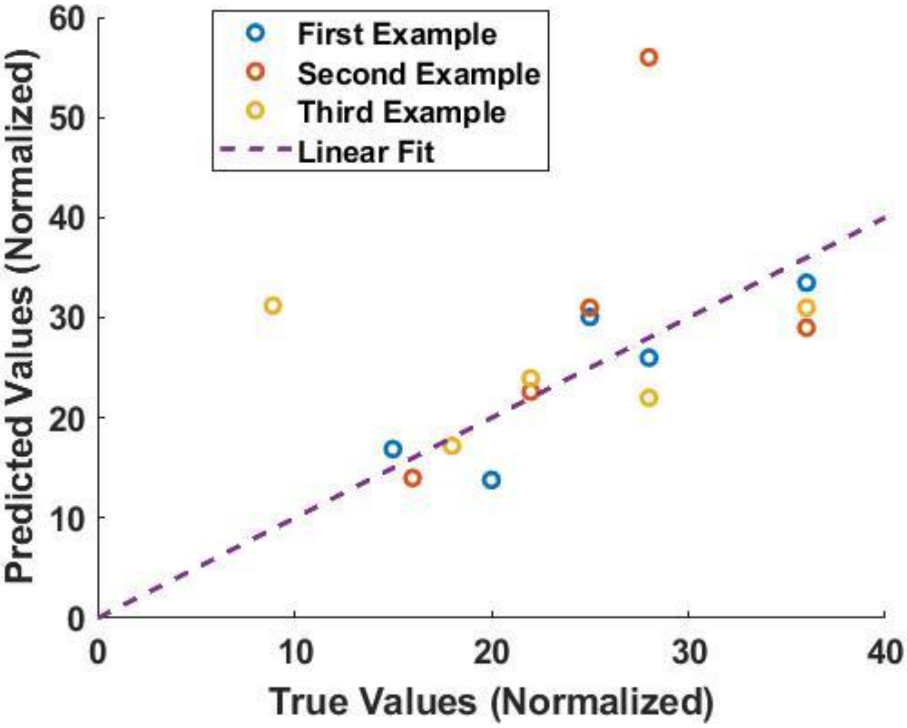


Figure 8.9 Prediction results on data with noise of SNR5 for 1DCNN

The noise levels are very high in these cases in order to test the limit of the algorithms, but such high noise levels (i.e. SNR5) typically do no occur in guided wave testing. From

Figures 8.8 to 8.9, we observe that the predictions are much poorer on noisy data compared to noiseless data. This was expected, but as long as the predictions are within a certain error range, they can be acceptable. In order to quantify the prediction performance of the 1DCNN architecture, a new parameter called *Number of Wrong Predictions (NOWP)* is introduced. Any prediction is termed a wrong prediction if the predicted value lies outside the 15% error range of the true value. Figure 8.10 shows the NOWP for the prediction on noiseless (baseline) and different levels of noisy data. It is clearly visible that the performance deteriorates in the presence of noise, but an interesting observation can be made for the SNR5 dataset . SNR5 signifies that the noise level is highest compared to SNR10 and SNR20, but the predictions on the SNR5 dataset is better compared to that of SNR10 and SNR20. In fact, the predictions on the SNR5 dataset are second best after the predictions on the noiseless data. There is certainly no linear trend in terms of performance of the model and increasing noise levels.

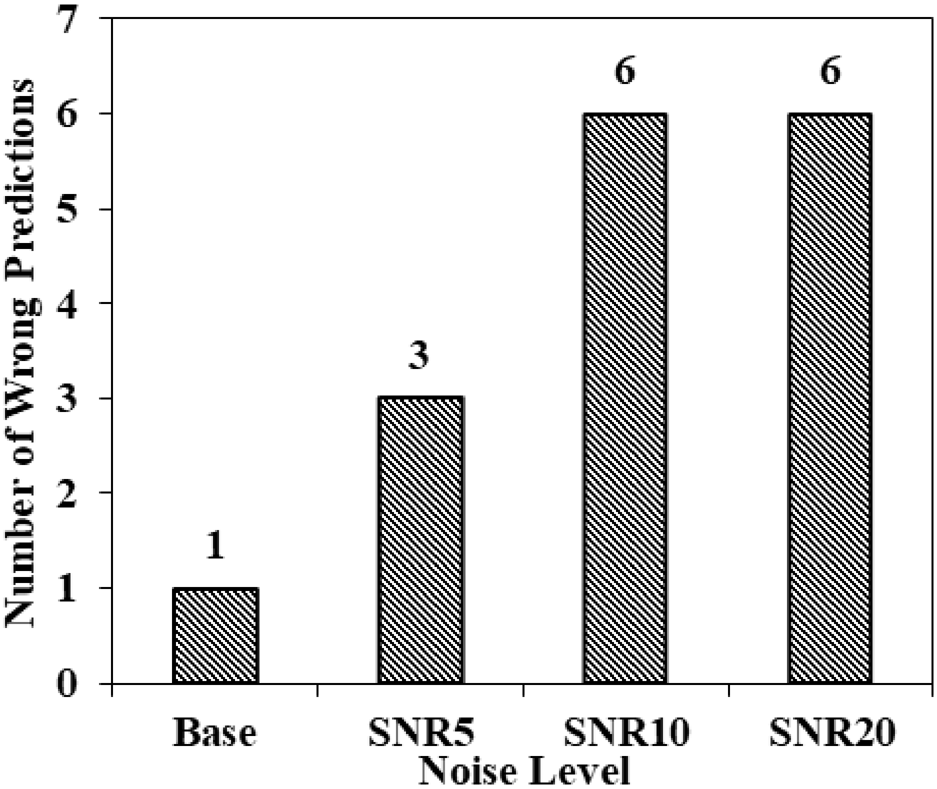


Figure 8.10 NOWP for the predictions on noisy and noiseless data for 1DCNN

8.2 Training Results with Long Short Term Memory Networks (LSTM)

A LSTM network is used to map the A0 and S0 waveforms to the target material properties in this case. The most important difference with respect to 1DCNN's apart from the working of the model is that the LSTM layer takes way too long to train. Therefore, in this thesis an inbuilt tensorflow API called CuDNNLSTM (in place of LSTMs) is used which accelerates the training process of LSTMs. The CuDNNLSTM layer can easily ramp up the training speed by more than 10 times. This is done by skipping repeated activation functions at every gate. They have shown to be equally accurate and have been used in various applications in signal processing, biomedical image processing etc. The architecture of the LSTM model used is shown in Table 8.3. A learning rate of 0.9e-3 and a batch size of 8 are found suitable. The model is run over 50 epochs.

Table 8.3 LSTM Architecture

Layer	Activation	Output (Feature map, Units)	Parameters
Batch Normalization	-	(4096,2)	8
CuDNNLSTM	-	(4096,10)	560
Dropout (0.2)	-	(4096,10)	0
Flatten	-	40960	0
Dense	ReLU	16	655386
Batch Normalization	-	16	64
Dropout (0.05)	-	16	0
Dense	Linear	5	85
Total parameters			656,093

Figures 8.11, 8.12 and 8.13 show the loss curve, MAE and R^2 curves respectively. The training performance of the LSTM model is very similar to that of the 1DCNN. It reaches a MAE of 37 and a R^2 coefficient of 0.98. The network takes 120 ms per epoch to train which is almost 30 times more than that of the 1DCNN. This increase is even after using the CuDNNLSTM layer.

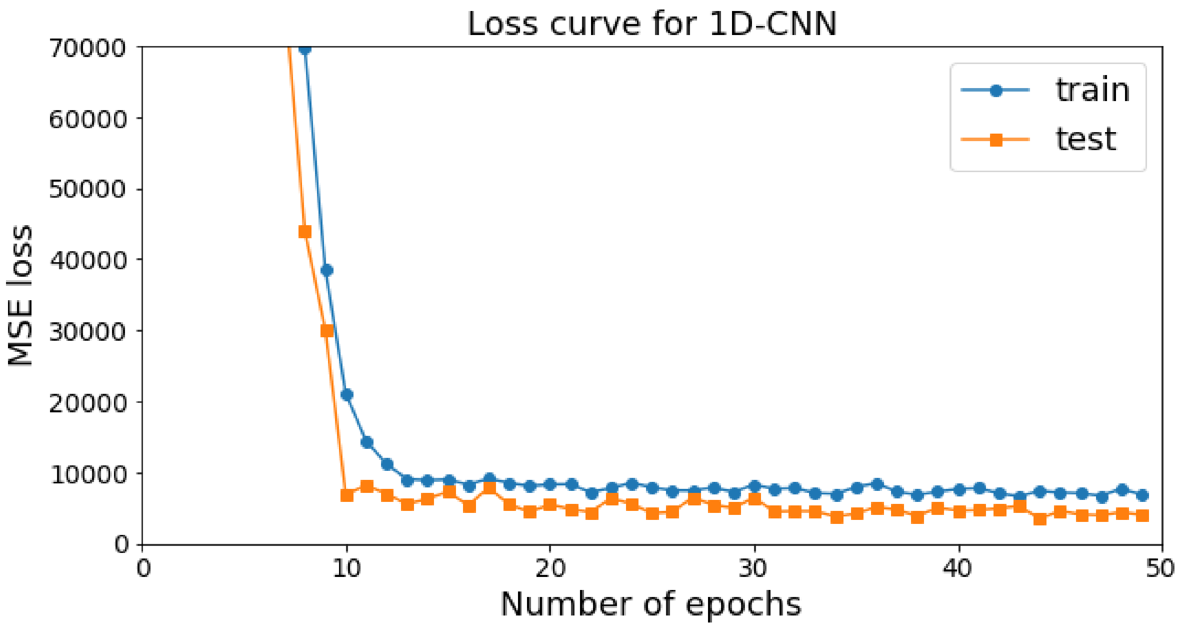


Figure 8.11 Loss curve: MSE vs number of epochs for LSTM

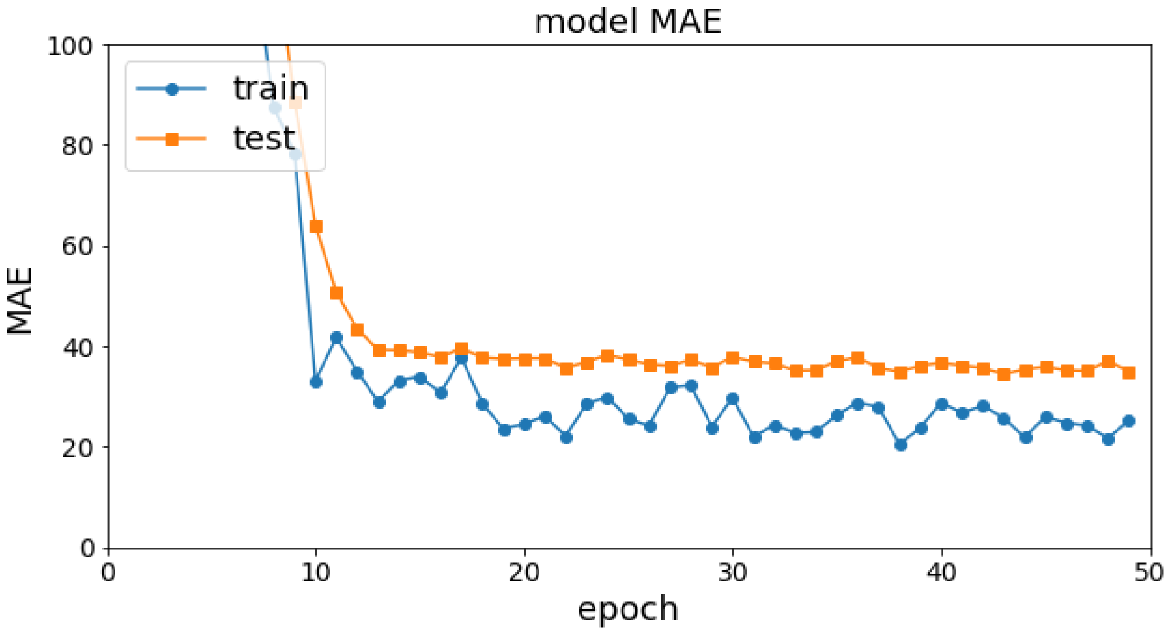


Figure 8.12 MAE curve: MAE vs number of epochs for LSTM

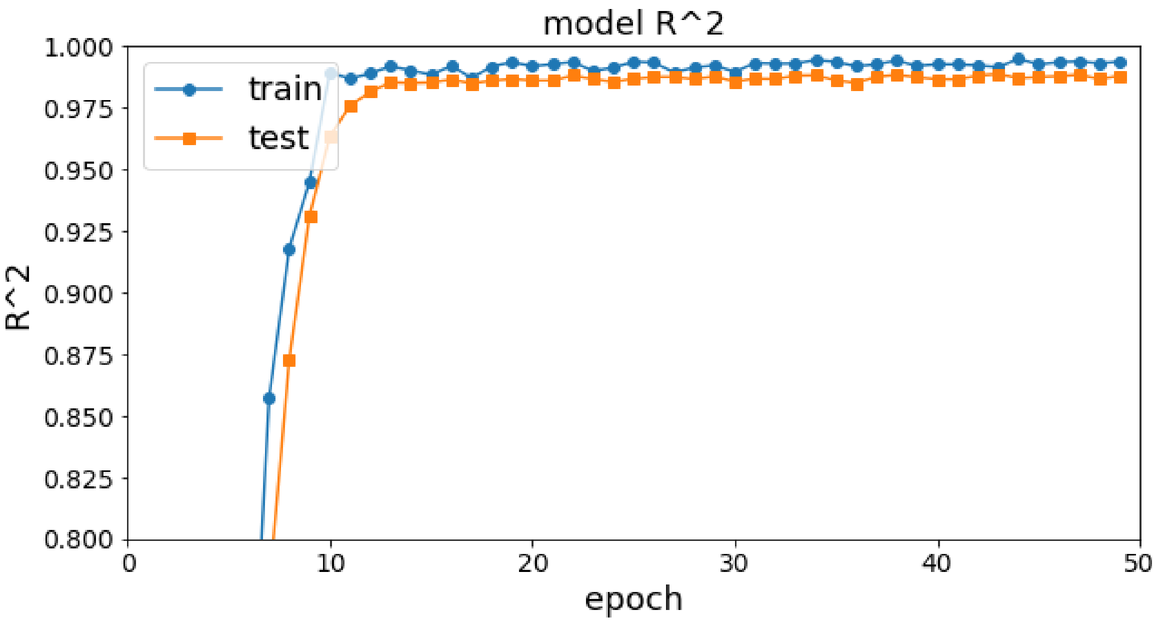


Figure 8.13 R^2 coefficient curve: R^2 coefficient vs number of epochs for LSTM

The trained model is then fed with previously unseen examples. The data here is noiseless and will serve as the baseline for evaluation of the performance on noisy data. Figure 8.14 outlines the prediction results of the LSTM network. Similar to the 1DCNN, the trained LSTM network was utilized to predict on noisy data of three different levels. The performance of the network on noisy and noiseless data is quantified using the NOWP parameter. Figures 8.15, 8.16 and 8.17 show the performance of the trained model on noisy data of SNR20, SNR10 and SNR5 respectively.

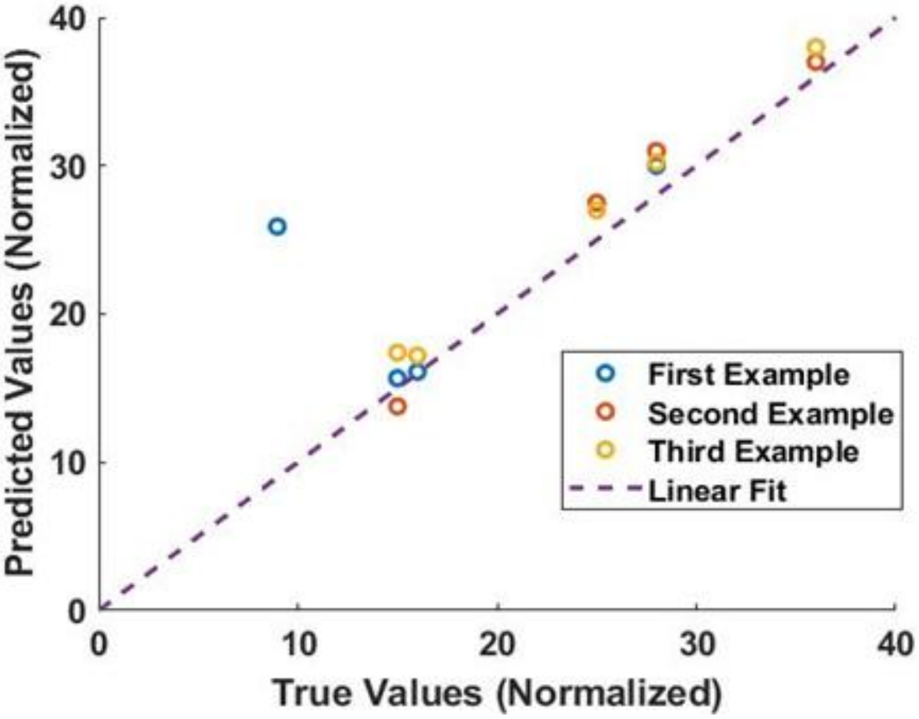


Figure 8.14 Prediction Results: True vs Predicted Plot for LSTM

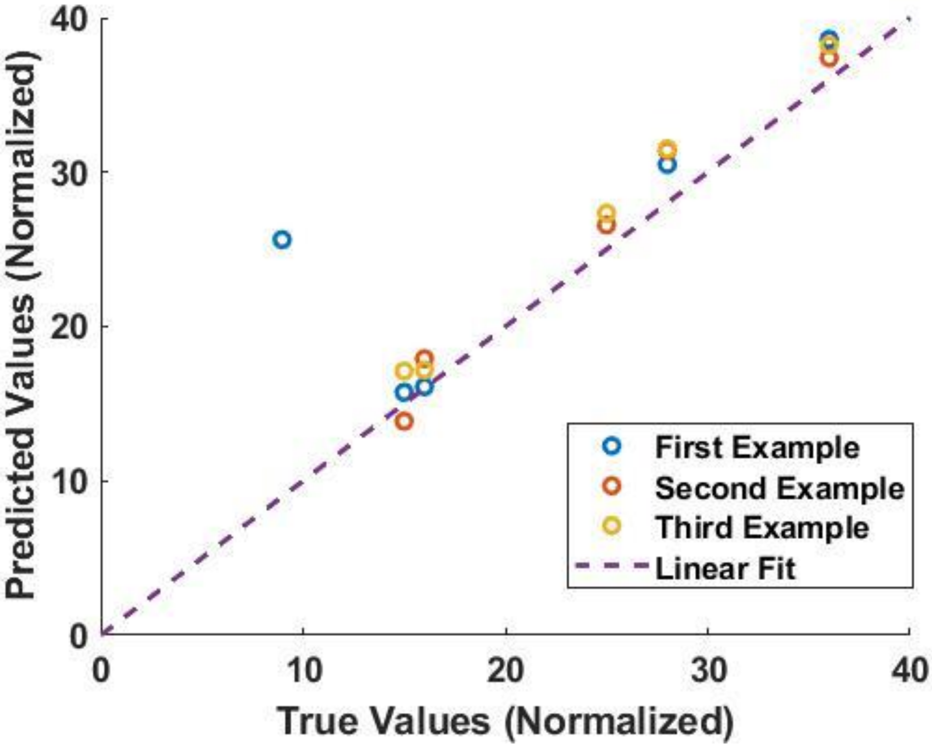


Figure 8.15 Prediction results on data with noise of SNR20 for LSTM

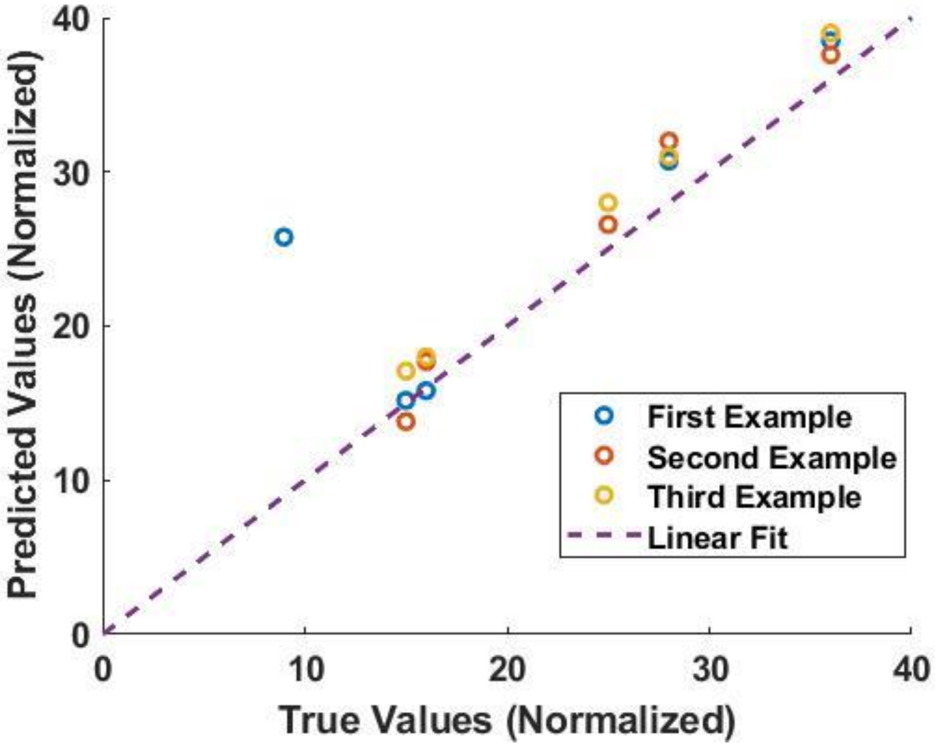


Figure 8.16 Prediction results on data with noise of SNR10 for LSTM

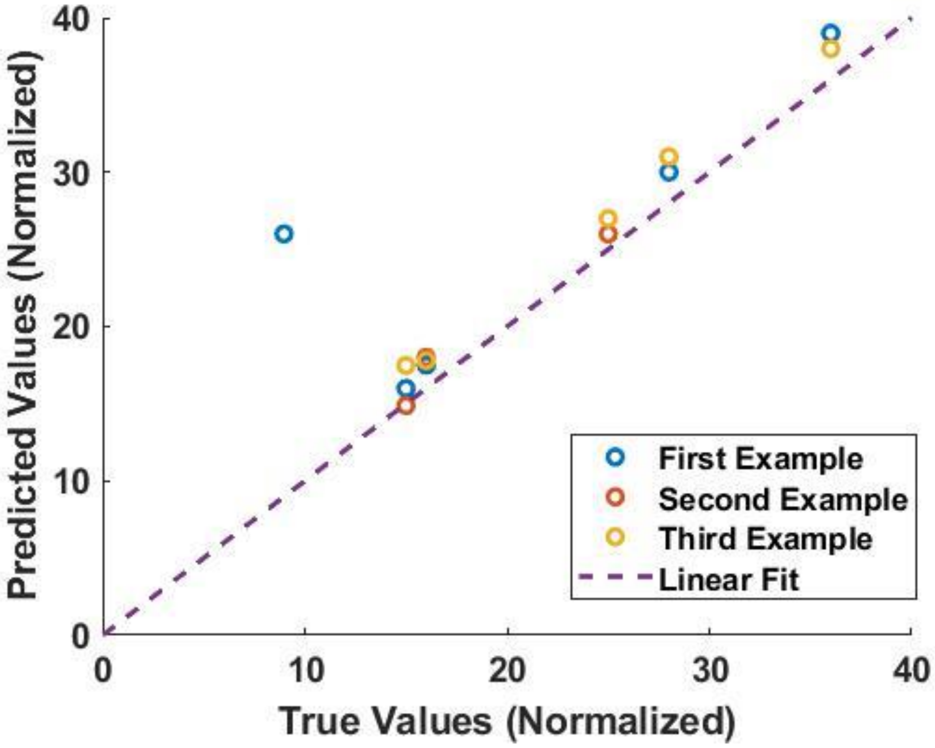


Figure 8.17 Prediction results on data with noise of SNR5 for LSTM

The limitations of the LSTM network are similar to that of the 1DCNN architecture. Therefore the reasons might be similar to the ones explained for the 1DCNN architecture. The performance of the network on noisy and noiseless data is quantified as mentioned before using the NOWP parameter. The NOWP plot is shown in Figure 8.18. The performance of the LSTM network on noisy data is seen to be similar to the performance on noiseless data. Unlike the 1DCNN, the predictions do not deteriorate in the presence of noise of various levels.

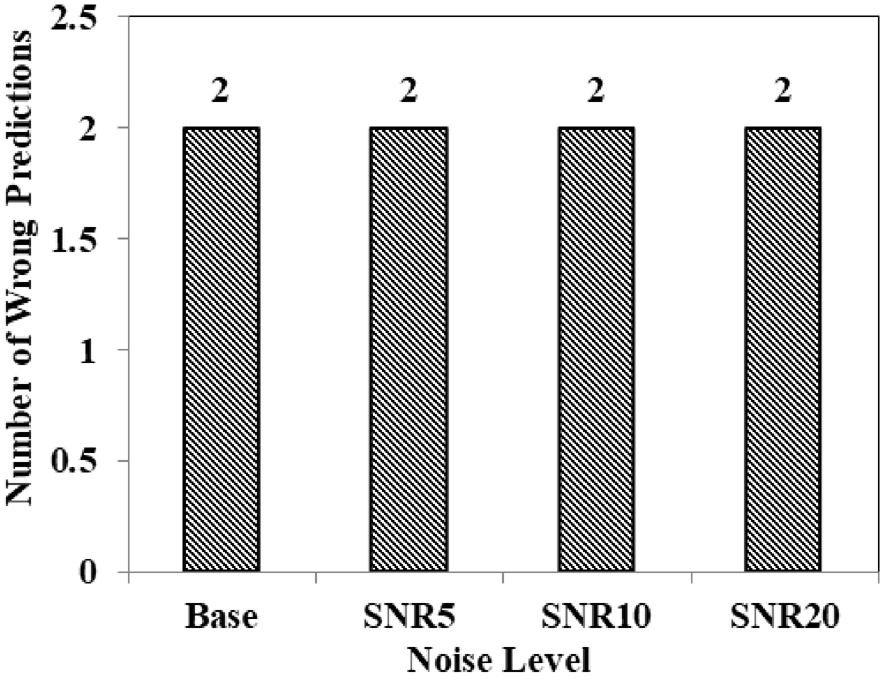


Figure 8.18 NOWP for the predictions on noisy and noiseless data for LSTM network

The performance of the two deep models i.e. 1DCNN and LSTM on noiseless data is comparable as seen in Figure 8.19 where the NOWP is plotted for both in presence and absence of noise in data. But the LSTM model has the edge when it comes to predictions on noisy data. It comfortably outperforms the 1DCNN model in the presence of noise. For the 1DCNN model, the performance clearly deteriorates as noise is added, but the performance of the LSTM network is consistent even as the noise levels are increased. The downside of the LSTM architecture is that the training time is very high compared to the 1DCNN network. This is evident in Figure 8.20 where the training time per epoch for the LSTM is almost 30 times that of the 1DCNN network. Even though convergence is relatively quicker in a LSTM network, the overall training time is naturally quite high when compared to 1DCNN. Higher accuracy of LSTM comes with longer training time which may not be feasible with big datasets. There should be a trade-off between training time and accuracy of the results depending upon the complexity of the problem in hand. It may be useful to start with both 1DCNN and LSTM with small subsets of the training set and compare the results and decide the trade-off. Another way of solving this dilemma is to develop a hybrid network like the *Convolutional Recurrent Neural Network* (CRNN). [43]

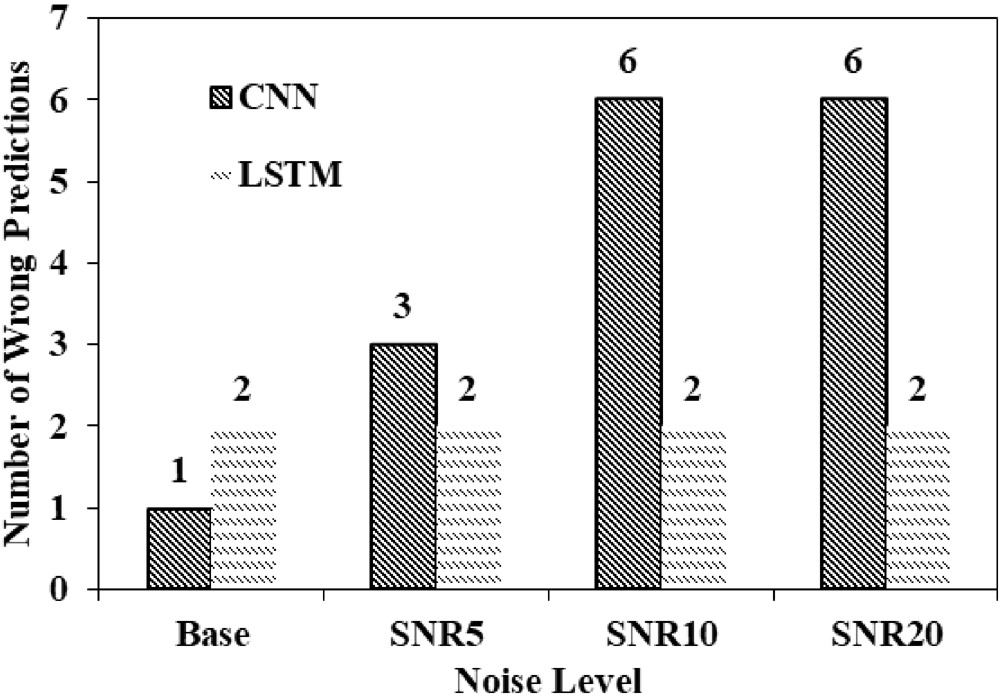


Figure 8.19 NOWP for the two deep architectures in the presence and absence of noise in the signal

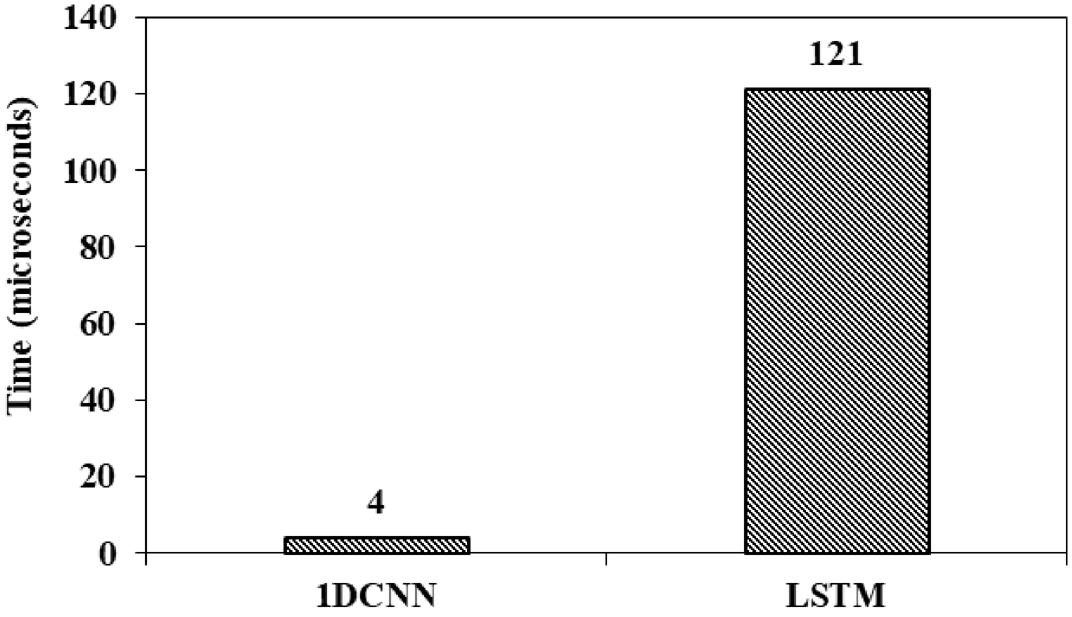


Figure 8.20 Training time per epoch for the two deep architectures (1DCNN and LSTM)

8.3 Training Results with Dense Neural Network

In the previous two sections, we presented the training and prediction using two Deep Learning architectures i.e. the 1D Convolutional Neural Network and the Long Short Term Memory Networks. Though the models perform very well even in presence of high noise levels, the current framework is incapable of predicting the Shear Modulus (*G_{12})* (See Section 6.1). In this section we present the results of a DNN trained on data collected using the DC® software. The data collection process is explained in detail in Chapter 7. This framework is capable of predicting the shear modulus due to the incorporation of the SH0 mode physics in our dataset. As mentioned in Chapter 7, we have considered 8 composites widely used in the industry. We have generated and collected group velocity (*C_g)* dispersion curves for the eight unidirectional composites with three different layups ($[0\ 0]_s$, $[0\ 0\ 0\ 0]_s$, $[0\ 0\ 0\ 0\ 0\ 0\ 00]_s$) and three different thickness (1mm, 2 mm and 3 mm). This results in generating a total of 72 (8*3*3) different dispersion plots. Different layups and different thickness of the composites were introduced to increase the complexity of the problem in hand. The training procedure is illustrated in the Figure 8.21. We use the dispersion curve data (i.e. C_g v/s frequency) for the A0, S0 and SH0 mode as our input features, while the output labels are the six material properties (ρ, E_1, E_2, ν_{12}, ν_{23}, G_{12}). The C_g information for all A0 modes of all the 72 combinations as a function of the frequency vector is concatenated into a single input vector. Similarly, all S0 and SH0 modes are concatenated respectively to form three long input feature vectors.

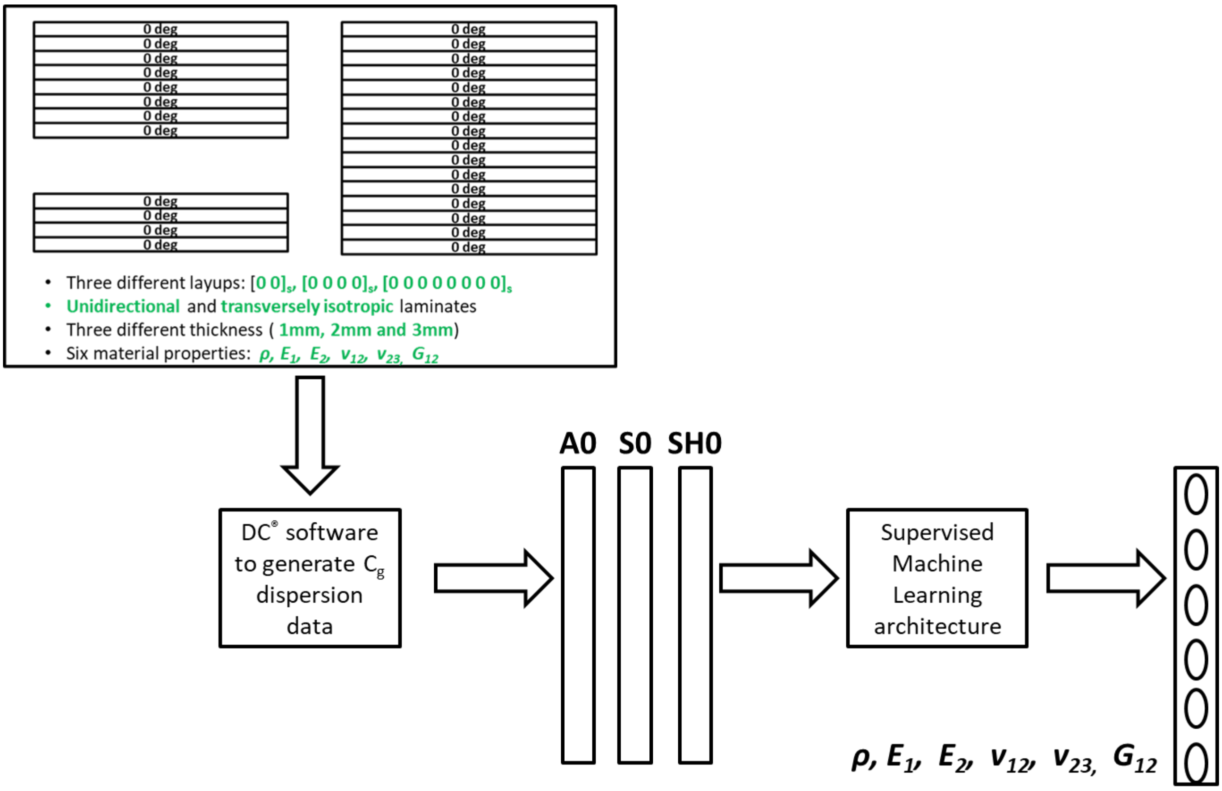

0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
0 deg
• Three different layups: [0 0]$_s$, [0 0 0 0]$_s$, [0 0 0 0 0 0 0 0]$_s$
• Unidirectional and transversely isotropic laminates
• Three different thickness (1mm, 2mm and 3mm)
• Six material properties: ρ, E_1, E_2, ν_{12}, ν_{23}, G_{12}
DC® software to generate C_g dispersion data
A0
S0
SH0
Supervised Machine Learning architecture
ρ, E_1, E_2, ν_{12}, ν_{23}, G_{12}

Figure 8.21 Research Flow for the ML based approach

The architecture of the dense neural network used is shown in Table 8.4. The training and testing dataset is split in a 39:1 ratio. The training dataset is further split in to a separate training and validation dataset with a 4:1 ratio. A learning rate of 1e^{-3} and a batch size of 8 are found suitable for this architecture. The network is run over 100 epochs. We have used similar metrics and loss functions to the ones used for 1DCNN and LSTM to evaluate the performance.

Table 8.4 DNN (MLP) Architecture

Layer	Activation	Output (Units)	Parameters
Dense	ReLU	512	2048
Dropout (0.0001)	-	-	0
Dense	ReLU	256	131328
Dense	ReLU	64	16448
Dropout (0.0001)	-	-	0
Dense	Linear	6	590
Total parameters			150,214

The loss (MSE) curve is shown in Figure 8.22, while the MAE curve is shown in Figure 8.23. The R^2 coefficient is shown in Figure 8.24. It is seen that the model trains very well and attains a MAE value of only 0.84 and a very high R^2 coefficient of 0.98. The spikes in the loss and MAE curve are an unavoidable consequence of Mini-Batch Gradient Descent in Adam (batch size=8). Some mini-batches have 'by chance' unlucky data for the optimization, inducing those spikes that are seen in the cost function.

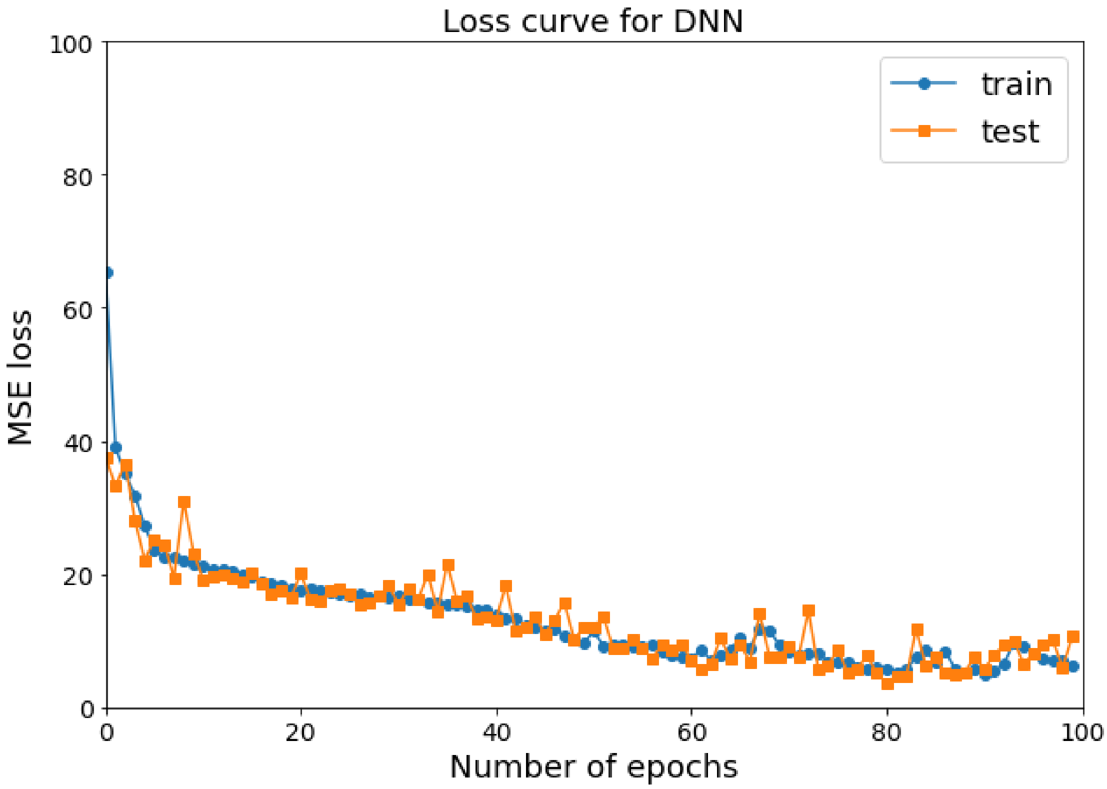


Figure 8.22 Loss curve: MSE vs Epochs for the DNN

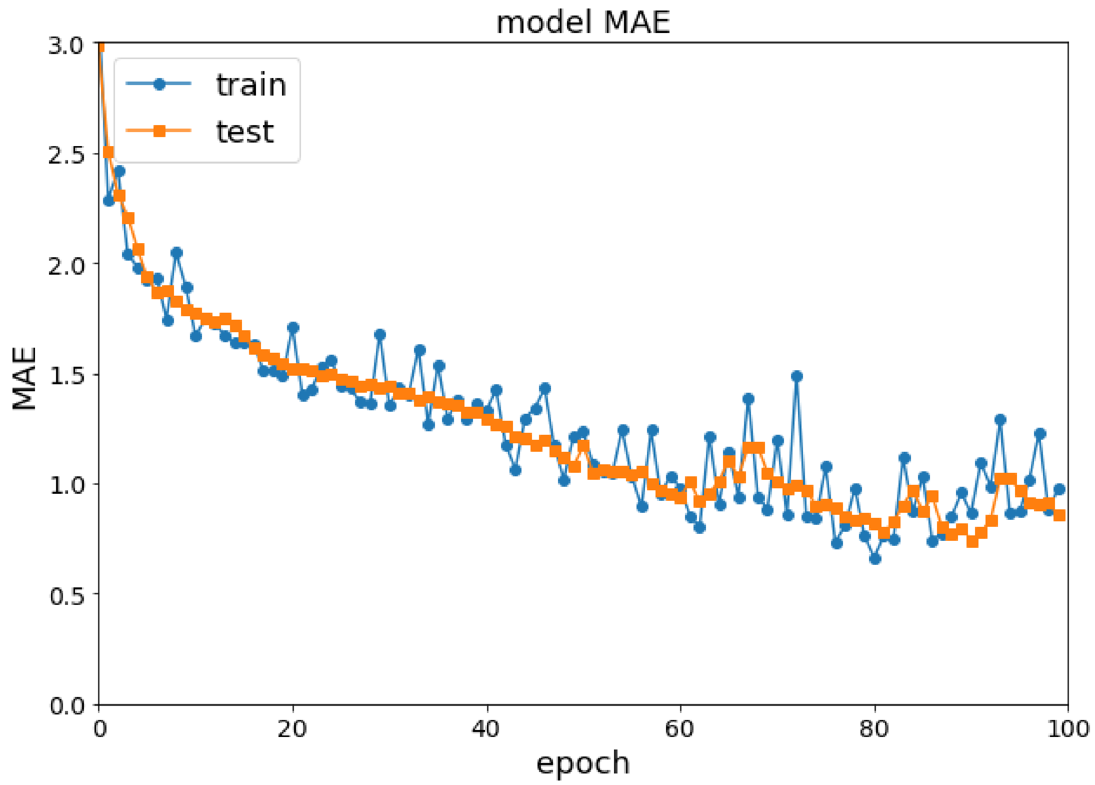


Figure 8.23 MAE curve: MAE vs Epochs for the DNN

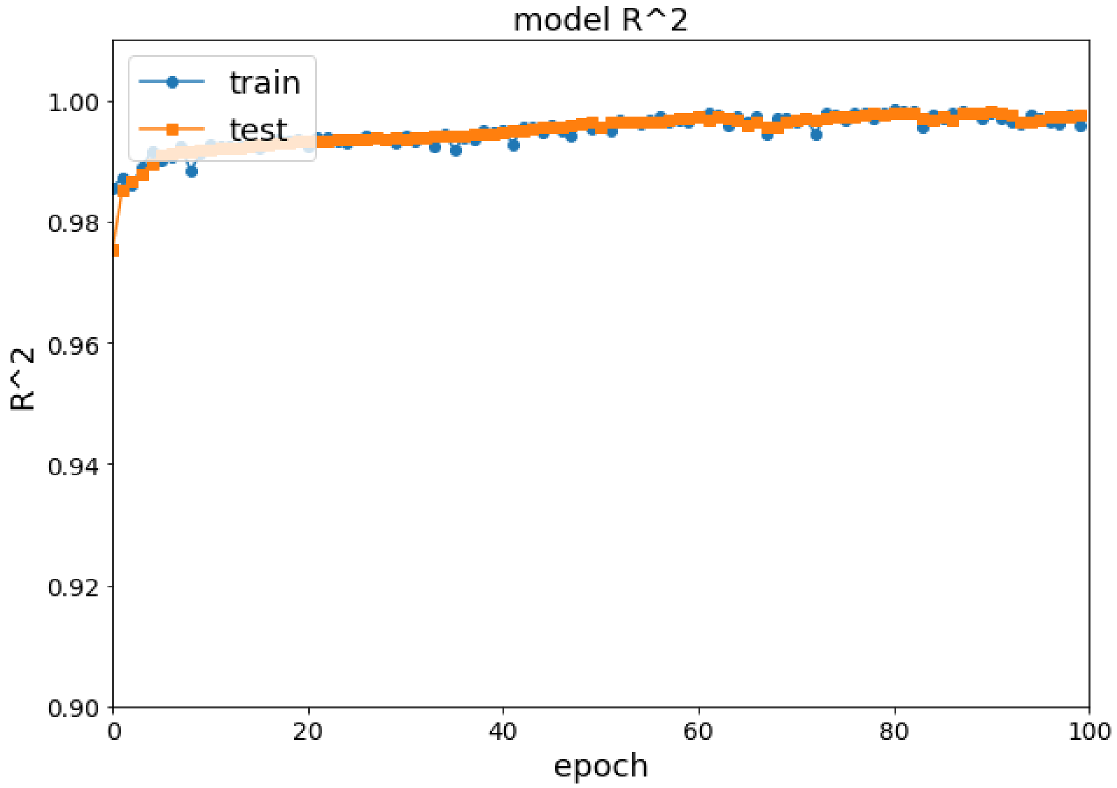


Figure 8.24 R^2 coefficient curve: R^2 coefficient vs Epochs for the DNN

For the testing or predictions process, group velocity dispersion curves of previously unseen composites are fed in to the trained model. The prediction results are shown in Figure 8.25 and in Table 8.5. It is seen that the predictions are very satisfactory for almost all cases. Unlike the prediction by 1DCNN and LSTM, the DNN can predict any value of E_2 accurately. Moreover it is also able to accurately predict the shear modulus G_{12}. The drawback here is the poor predictions on Poisson's ratio v_{23} (highlighted in red in the Table 8.5 and Figure 8.25). In most cases (except the Poisson's ratio v_{23}) , the network was able to predict the material properties accurately. The accuracy of the predictions on the Poisson's ratio v_{23} is only about 55% compared to the accuracies of the other material properties that are more than 95%.

Figure 8.26 shows the NOWP for all three models we have used in the study. A0 and S0 waveforms were used as inputs for the deep models, while velocity dispersion data was used for the DNN. 1DCNN has the least NOWP and DNN the most. Though, all the NOWP for the DNN are resulting from the poor predictions on the Poisson's ratio $v_{23.}$ In terms of predicting other material properties, the DNN seems to do a much better job as it can effectively predict E_2 and G_{12} accurately. But the dispersion data has been collected using the DC® software that is not developed for the purpose of generating lots of dispersion curves. It is rather a software to generate a couple of dispersion curves to study the dispersion phenomenon of the guided wave. Hence data collection and then training using that data is not probably a solution that would be viable for real time use. We have only used it in this thesis, for we wanted our framework to be capable of predicting all the material properties including the Shear Modulus.

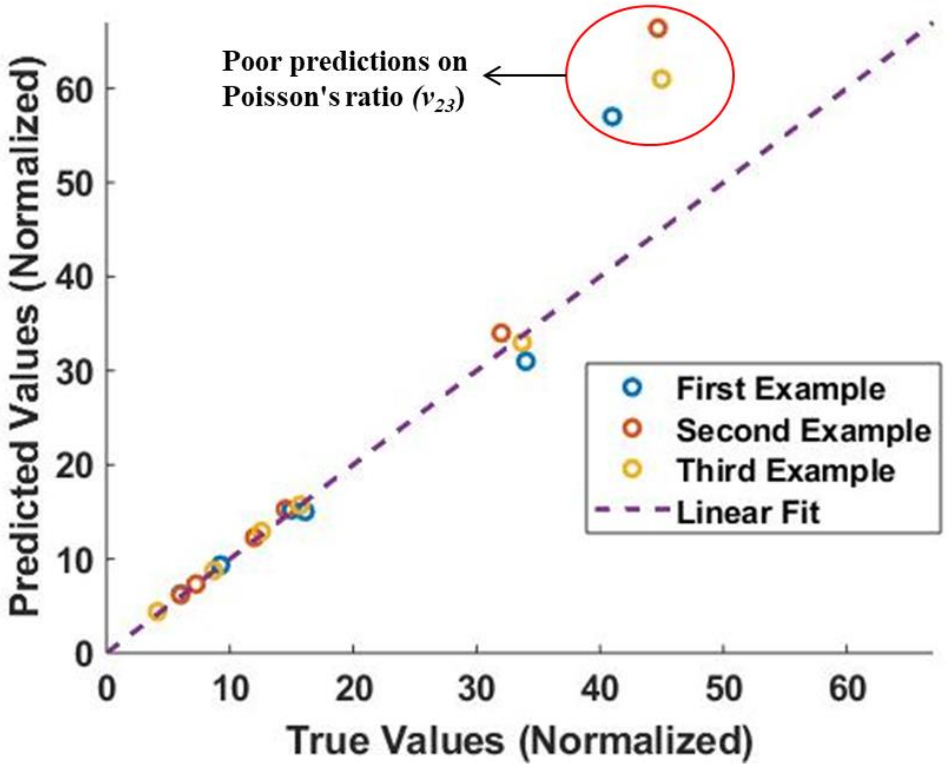


Figure 8.25 Prediction results using DNN

Table 8.5 Prediction Results using DNN

Example		ρ (kg/m^3)	E$_1$(Gpa)	E$_2$(Gpa)	G$_{12}$(GPa)	v$_{12}$	v$_{23}$
Example 1	**True**	1500	161	9.25	6	0.34	0.41
	Predicted	1520	150.28	9.35	6.28	0.31	0.58
Example 2	**True**	1450	119.99	8.25	6.02	0.32	0.45
	Predicted	1530	122.84	8.32	6.18	0.34	0.65
Example 3	**True**	1581	125.5	8.8	4.13	0.34	0.45
	Predicted	1582	129.3	8.85	4.40	0.33	0.61

The training time per epoch for the DNN is 3 ms. Figure 8.27 shows the training time per epoch for the three models. The overall training time is the least for the DNN, while 1DCNN also takes similar computational time. The LSTM lags behind the other two architectures in this regard as it takes about 120 ms per epoch which is 30 times more than the 1DCNN and 40 times more than the DNN. Do note that the comparisons are made for the deep architectures using only the noiseless data. It does not make sense to add noise to the dispersion data as they are purely simulated, and aren’t subject to a lot of uncertaininity.

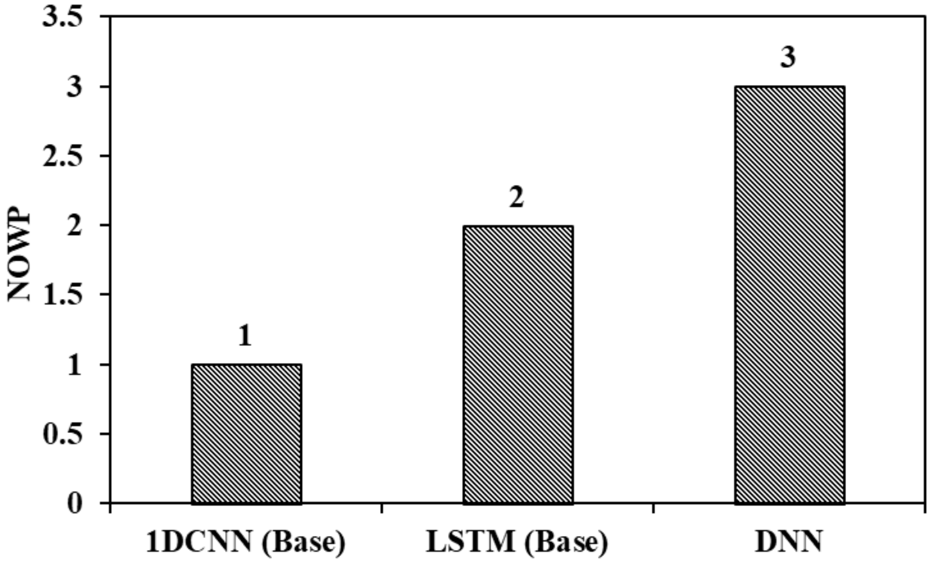


Figure 8.26 NOWP using DNN

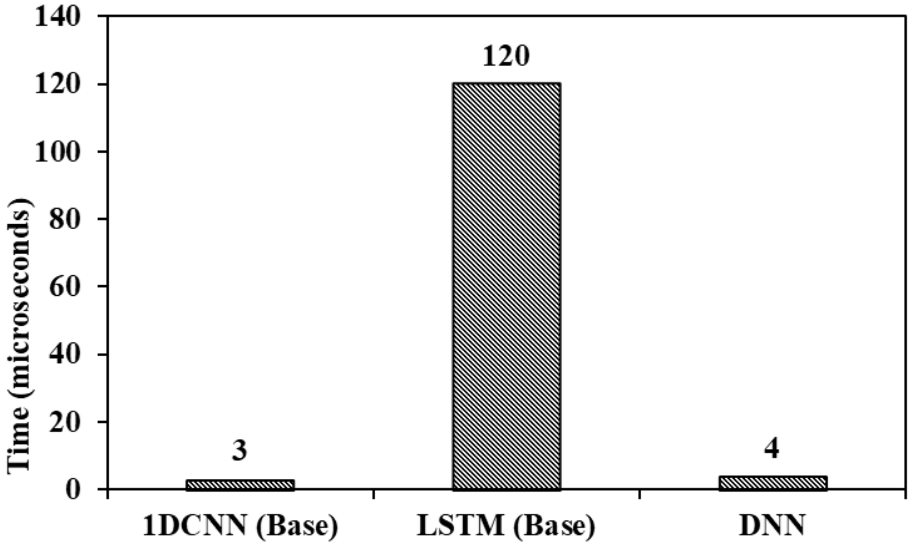


Figure 8.27 Training time per epoch using DNN

CHAPTER 9

CONCLUSIONS AND FUTURE WORK

9.1. Conclusions

In this thesis, we have explored different learning models to estimate and predict the elastic properties of composites using Ultrasonic Guided Waves. We have evaluated the performance of the models by predicting on data in the presence and absence of noise. Some of the major conclusions that can be drawn from this thesis are:

1. We used two different modelling methods to simulate guided waves in a composite, and showed how at the operating frequency range, and by using fundamental modes we can geometrically scale down a 2D model to a 1D waveguide.
2. It is seen that with A0 and S0 time histories as inputs and five properties as outputs, and in this frequency range, the inverse problem of property identification is limited by its uniqueness. Due to this, it is not possible to measure shear modulus using fundamental modes.
3. This being a conceptual work, and with a limited amount of dataset, the predictions from the 1DCNN and the LSTM models on unseen examples are very satisfactory. Further, the performance of these models on noisy data is also satisfactory. It can be concluded that the two deep neural networks can learn the inherent relationship between ultrasonic guided wave signal and material properties. It can also be concluded that the deep models are robust to noise of various levels.
4. It is seen that the LSTM network is better in terms of performance when it comes to noisy data, but LSTM’s also take longer time to learn. Therefore, there is a trade-off required depending on the complexity of the problem in hand.

5. ML based DNN performs very well generally but is limited on its predictions on Poisson’s ratio. Moreover, the data collection for this particular method is time consuming and not viable for real time use.
6. Some of the other research works have demonstrated the use of a single transmitter and multiple sensors. Using multiple sensors may not be a feasible option for a lot of applications. On the other hand, we have demonstrated the use of UGW for property identification using a single transmitter and a single sensor.
7. Previous research is more inclined to solve the inverse problem of materials characterization using global optimisation schemes like GA, SA etc. With this research, we have obtained good results with deep learning networks trained using Adam optimization (variant of stochastic gradient descent with added momentum and adaptive learning rate) on mean square loss function.
8. Heuristics based GA and SA takes a long time with very high computation resources. With neural networks and stochastic gradient descent, the training can be accelerated with the use of GPUs.
9. The learning is necessary to automate a materials characterization problem. Neural networks provide that automation since it learns the inverse problem by minimising the loss function on the dataset.
10. It is also required the estimation is real time and quick. Unlike previously explored methods like GA or SA, any new prediction here takes only 1 ms which enables the deployment of the algorithms in real time scenarios. As per the author’s knowledge, there is no literature available on the application of deep learning techniques for the identification of materials properties. Our research demonstrates the application of this concept clearly, and even in the absence of experimental results, the performance is very

good even with high noise content. Our future research will be focused towards implementation on real experimental data.

9.2. Future Work

The results presented in this thesis can be looked at as a sort of starting step towards a bigger problem that is using Deep Learning for Material Property Identification. Though our models perform very well and predict with accuracies more than 85%, they are still subject to improvements. One approach to improve the existing framework would be to further refine our FEM and SFEM models. For example, if one can optimize the FEM model better to reduce the computational time, a dataset can be created using the FEM model instead of the SFEM model. The FEM model incorporates more physics compared to the SFEM model as the wave modes are coupled. This might lead to better predictions. Another possible direction to pursue would be to incorporate experimental data in the framework. There are two way to do this. The first being using a model trained on simulation data to predict on experimental data. If the performance is not satisfactory, some of the experimental data can even be included in the training process itself.

The current work presented is limited to composite laminates. The framework can be extended easily to other materials like metals, soft polymers etc. Experimental determination of material properties of soft polymers is specifically hard due to the flimsy nature of the materials which make it sometimes impossible to conduct experiments on.

9.3. Thesis Contributions

There is little or no literature available on the application of deep learning techniques for the identification of materials properties. The thesis demonstrates the application of this concept clearly, and even in the absence of experimental results, the performance of all the learning models used is very good even with high noise content. We aim to propose a complete framework complete with intermediate data analysis algorithms and different learning models for solving the problem of material property identification. The proposed framework can possibly be extended to characterize other materials used in NDE/SHM

APPENDICES

APPENDIX A

Symmetric Solution

For a symmetric solution, the u_x component of the particle displacement vector and shear stresses are symmetric about the mid-plane across the thickness of the 2D infinite plate.

$$u_x(x,-d) = u_x(x,d) \qquad \tau_{yx}(x,-d) = -\tau_{yx}(x,d)$$

$$u_y(x,-d) = -u_y(x,d) \qquad \tau_{yy}(x,-d) = \tau_{yy}(x,d) \qquad \text{(A.1)}$$

Symmetric boundary conditions also include traction free surfaces, therefore a part of Equation (A.2) becomes:

$$\tau_{yx}(x,-d) = -\tau_{yx}(x,d) = 0 \qquad \text{(A.2)}$$

$$\tau_{yy}(x,-d) = \tau_{yy}(x,d)\text{=}0$$

Using the general solution for potentials in Equations (3.22) and (3.23), and expressions for stress and displacements, Equation (A.2) can be expressed as:

$$-2i\xi A_2 \sin pd + B_1(\xi^2 - q^2)\sin qd = 0 \qquad \text{(A.3)}$$

$$A_2(\xi^2 - q^2)\cos pd - 2i\xi B_1 \cos pd = 0 \qquad \text{(A.4)}$$

A non-trivial solution for the above linear system of equations exists if the determinant in the above Equation vanishes.

$$\Delta_S = (\xi^2 - q^2)^2 \sin qd \cos pd + 4\xi^2 pq \cos qd = 0 \qquad \text{(A.5)}$$

Rearranging, the dispersion relation for the symmetric mode is obtained.

$$\frac{\tan pd}{\tan qd} = \frac{\left(\xi^2 - q^2\right)^2}{4\xi^2 pq} \qquad \text{(A.6)}$$

Where p and q have been described in Equation (3.18) and (3.19) respectively.

APPENDIX B

Anti-Symmetric Solution

For an anti-symmetric solution, the u_x component of the particle displacement vector and shear stresses are anti-symmetric about the mid-plane across the thickness of the 2D infinite plate.

$$u_x(x,-d) = -u_x(x,d) \qquad \tau_{yx}(x,-d) = \tau_{yx}(x,d)$$

$$u_y(x,-d) = u_y(x,d) \qquad \tau_{yy}(x,-d) = -\tau_{yy}(x,d) \tag{B.1}$$

The anti-symmetric boundary conditions also include traction free surfaces, therefore a part of Equation (B.1) becomes:

$$\tau_{yx}(x,-d) = \tau_{yx}(x,d) = 0 \tag{B.2}$$

$$\tau_{yy}(x,-d) = -\tau_{yy}(x,d){=}0$$

Using the general solution for potentials in Equations (3.22) and (3.23), and expressions for stress and displacements, Equation (B.2) can be expressed as:

$$2i\xi A_1 \cos pd + B_2(\xi^2 - q^2)\cos qd = 0 \tag{B.3}$$

$$A_1(\xi^2 - q^2)\sin pd - 2i\xi B_2 \cos pd = 0 \tag{B.4}$$

A non-trivial solution for the above linear system of equations exists if the determinant in the above Equation vanishes.

$$\mathit{\Delta}_S = (\xi^2 - q^2)^2 \sin qd \cos pd + 4\xi^2 pq \cos qdsin\ qd = 0 \tag{B.5}$$

Rearranging, the dispersion relation for the symmetric mode is obtained.

$$\frac{\tan pd}{\tan qd} = \frac{4\xi^2 pq\left(\xi^2 - q^2\right)^2}{(\xi^2 - q^2)^2} \tag{B.6}$$

Where p and q have been described in Equation (3.18) and (3.19) respectively.

APPENDIX C

LIST OF PUBLICATIONS

1. Karthik Gopalakrishnan, Mahindra Rautela and Yiming Deng (submitted and accepted), "Deep Learning Based Identification of Elastic Properties Using Ultrasonic Guided Waves", European Workshop on Structural Health Monitoring, July 2020.

2. Rautela M, Karthik Gopalakrishnan, Yiming Deng and Gopalakrishnan S (in press). Ultrasonic Guided Waves Based Identification of Elastic Properties Using 1D-Convolutional Neural Networks. 2020 IEEE International Conference on Prognostics and Health Management (ICPHM).

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Gopalakrishnan, Srinivasan, Massimo Ruzzene, and Sathyanaraya Hanagud. 2011. *Computational Techniques for Structural Health Monitoring*. Springer Series in Reliability Engineering. London: Springer-Verlag. https://doi.org/10.1007/978-0-85729-284-1.

[2] Doebling, S. W., C. R. Farrar, M. B. Prime, and D. W. Shevitz. 1996. "Damage Identification and Health Monitoring of Structural and Mechanical Systems from Changes in Their Vibration Characteristics: A Literature Review." LA-13070-MS. Los Alamos National Lab., NM (United States). https://doi.org/10.2172/249299.

[3] Worden, K., and J. M. Dulieu-Barton. 2016. "An Overview of Intelligent Fault Detection in Systems and Structures:" *Structural Health Monitoring*, August. https://doi.org/10.1177/1475921704041866.

[4] Gopalakrishnan, Srinivasan. 2009. "Modeling Aspects in Finite Elements." In *Encyclopedia of Structural Health Monitoring*. American Cancer Society. https://doi.org/10.1002/9780470061626.shm064

[5] Gopalakrishnan, Srinivasan, Abir Chakraborty, and Debiprosad Roy Mahapatra. 2008. *Spectral Finite Element Method: Wave Propagation, Diagnostics and Control in Anisotropic and Inhomogeneous Structures*. Computational Fluid and Solid Mechanics. London: Springer-Verlag. https://doi.org/10.1007/978-1-84628-356-7.

[6] Xu, Luyao, and Y. Frank Cheng. 2017. "A Finite Element Based Model for Prediction of Corrosion Defect Growth on Pipelines." International Journal of Pressure Vessels and Piping,June. https://doi.org/10.1016/j.ijpvp.2017.05.002.

[7] Demma, A., P. Cawley, M. Lowe, A. G. Roosenbrand, and B. Pavlakovic. 2004. "The Reflection of Guided Waves from Notches in Pipes: A Guide for Interpreting Corrosion Measurements." NDT and E International 3 (37):167–80. https://doi.org/10.1016/j.ndteint.2003.09.004.

[8] Qian, Qianyue, Houman Hanachi, Jie Liu, Junjie Gu, Fai Ma, Ashok Koul, and Avisekh Banerjee. 2016. "Simulation of Ultrasonic Testing for Resolution of Corrosion Detection in Pipes." In 2016 IEEE International Conference on Prognostics and Health Management(ICPHM),1–5. https://doi.org/10.1109/ICPHM.2016.7542836.

[9] Poddar, Banibrata. 2016. "Physics Based Modeling Of Guided Waves For Detection And Characterization Of Structural Damage In NDE and SHM." *Theses and Dissertations*, January. https://scholarcommons.sc.edu/etd/3923.

[10] Sanguineti, Marcelo. 2008. "Universal Approximation by Ridge Computational Models and Neural Networks:" *The Open Applied Mathematics Journal* 2 (1). https://benthamopen.com/ABSTRACT/TOAMJ-2-31.

[11] Telgarsky, Matus. 2015. "Representation Benefits of Deep Feedforward Networks." *ArXiv:1509.08101 [Cs]*, September. http://arxiv.org/abs/1509.08101.

[12] Cai, Jian, Lei Qiu, Shenfang Yuan, Lihua Shi, PeiPei Liu, and Dong Liang. 2012. “Structural Health Monitoring for Composite Materials.” *Composites and Their Applications*, August. https://doi.org/10.5772/48215.

[13] Campbell FC. Structural Composite Materials. Ohio: ASM International; 2010.

[14] Tam JH, Ong ZC, Ismail Z, Ang BC, Khoo SY. Identification of material properties of composite materials using nondestructive vibrational evaluation approaches: A review Mechanics of Advanced Materials and Structures. 2017 Sep 10;24(12):971-86.

[15] Cui R, di Scalea FL. On the identification of the elastic properties of composites by ultrasonic guided waves and optimization algorithm Composite Structures. 2019 Sep 1;223:110969.

[16] Staszewski W, Boller C, Tomlinson GR. Health monitoring of aerospace structures: smart sensor technologies and signal processing John Wiley & Sons. 2004 Sep 10.

[17] Rose JL. Ultrasonic guided waves in solid media Cambridge University Press. 2015.

[18] Giurgiutiu V. Structural health monitoring (SHM) of aerospace composites Polym. Compos. Aerosp. Ind. Elsevier 2017.

[19] Gopalakrishnan, Srinivasan. 2016. *Wave Propagation in Materials and Structures*. 1 edition. Boca Raton: CRC Press.

[20] R.M. Jones. Mechanics of Composite Materials. Scripta, Washington, DC, 1975.

[21] Reddy JN. Mechanics of laminated composite plates and shells: theory and analysis CRC press; 2003 Nov 24.

[22] Waller, Matthew David. 2016. “Defect Growth Detection Potential Using Guided Waves,” July. https://etda.libraries.psu.edu/catalog/t435gc96m.

[23] V. Giurgiutiu. Structural Health Monitoring with Piezoelectric Wafer Active Sensors. Academic Press, 2008

[24] Gazis, D.C., 1959, “Three Dimensional Investigation of the Propagation of Waves in Hollow Circular Cylinders,” Journal of the Acoustical Society of America 31: 568- 73.

[25] Karpenko, Oleksii. n.d. “Signal Analysis in Guided Wave Structural Health Monitoring,” 94.

[26] Barron AR. Universal approximation bounds for superpositions of a sigmoidal function IEEE Transactions on Information theory. 1993 May;39(3):930-45.

[27] Telgarsky M. Benefits of depth in neural networks arXiv preprint arXiv: 1602.04485. 2016 Feb 14.

[28] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5 (4), 115-133.

[29] Karypidis, Dimitrios. 2019. “Structural Health Monitoring of Concrete Elements Using Deep Machine Learning.” https://odr.chalmers.se/handle/20.500.12380/256959.

[30] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65 (6), 386. 4

[31] Bhande, Anup. 2018. “What Is Underfitting and Overfitting in Machine Learning and How to Deal with It.” Medium. March 18, 2018. https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76.

[32] “Image Data Augmentation- Image Processing In TensorFlow- Part 2.” 2020. *Mc.Ai* (blog). January 5, 2020. https://mc.ai/image-data-augmentation-image-processing-in-tensorflow-part-2/.

[33] LeCun Y., Haffner P., Bottou L., Bengio Y. (1999) Object Recognition with Gradient-Based Learning. In: Shape, Contour and Grouping in Computer Vision. Lecture Notes in Computer Science, vol 1681. Springer, Berlin, Heidelberg

[34] Ananthasuresh, G. K., K. J. Vinoy, S. Gopalakrishnan and K. N. Bhat, and V. K. Aatre. 2010. *MICRO AND SMART SYSTEMS*. Wiley India Pvt. Limited.

[35] MacDonald, J. K. L. 1933. “Successive Approximations by the Rayleigh-Ritz Variation Method.” *Physical Review* 43 (10): 830–33. https://doi.org/10.1103/PhysRev.43.830.

[36] Ghose, Bikash, Krishnan Balasubramaniam, C V Krishnamurthy, and A Subhananda Rao. 2011. “Comsol® Based 2-D FEM Model For Ultrasonic Guided Wave Propagation In Symmetrically Delaminated Unidirectional Multi-Layered Composite Structure,” 5.

[37] Cooley, James W., and John W. Tukey. 1965. “An Algorithm for the Machine Calculation of Complex Fourier Series.” *Mathematics of Computation* 19 (90): 297–301. https://doi.org/10.2307/2003354.

[38]Gopalakrishnan, S., A. Chakraborty and D R Mahapatra, *Spectral Finite Element methods*, Springer-Verlag, UK, January 2008

[39] “Center for Lightweight-Production-Technology - The Dispersion Calculator: A Free Software for Calculating Dispersion Curves of Guided Waves in Multilayered Composites.” n.d. Accessed August 11, 2020. https://www.dlr.de/zlp/en/desktopdefault.aspx/tabid-14332/24874_read-61142/#/gallery/33485.

[40] Balasubramaniam K. Inversion of the ply lay-up sequence for multilayered fiber reinforced composite plates using genetic algorithm. Nondestructive Testing and Evaluation 1998 Nov 1;15(5):311-31.

[41] Hosten B, Castaings M, Tretout H, Voillaume H. Identification of composite materials elastic moduli from Lamb wave velocities measured with single sided, contactless ultrasonic method InAIP Conference Proceedings 2001 Apr 30 (Vol. 557, No. 1, pp. 1023-1030). American Institute of Physics.

[42] Vishnuvardhan J, Krishnamurthy CV, Balasubramaniam K. Genetic algorithm based reconstruction of the elastic moduli of orthotropic plates using an ultrasonic guided

wave single-transmitter-multiple-receiver SHM array Smart materials and structures 2007 Aug 8; 16(5):1639.

[43] Choi, Keunwoo, György Fazekas, Mark Sandler, and Kyunghyun Cho. 2017. “Convolutional Recurrent Neural Networks for Music Classification.” In 2017 IEEE International Conference on Acoustics,

[44] F. Chollet, Deep Learning with Python, 1st Edition, Manning Publication Co., Greewich, CT, USA, 2017.

[45] Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," in IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157-166, March, 1994.doi: 10.1109/72.

[46] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[47] Gopalakrishnan, Karthik, and Fathi M. Salem. 2020. “Sentiment Analysis Using Simplified Long Short-Term Memory Recurrent Neural Networks.” *ArXiv:2005.03993 [Cs, Stat]*, May. http://arxiv.org/abs/2005.03993.

[48] Rautela, Mahindra, and S Gopalakrishnan. n.d. “Deep Learning Frameworks for Wave Propagation-Based Damage Detection in 1D-Waveguides,” 11

[49] Doyle, James F. 1997. *Wave Propagation in Structures: Spectral Analysis Using Fast Discrete Fourier Transforms*. 2nd ed. Mechanical Engineering Series. New York: Springer-Verlag. https://doi.org/10.1007/978-1-4612-1832-6.