DEVELOPMENT OF A DECENTRALIZED ARTIFICIAL INTELLIGENCE SYSTEM FOR DAMAGE DETECTION IN COMPOSITE LAMINATES FOR AEROSPACE STRUCTURES

A Dissertation by

Zachary Tyler Kral

Master of Science, Wichita State University, 2009

Bachelor of Science, Virginia Polytechnic Institute and State University, 2006

Submitted to the Department of Aerospace Engineering
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

December 2013

DEVELOPMENT OF A DECENTRALIZED ARTIFICIAL INTELLIGENCE SYSTEM FOR DAMAGE DETECTION IN COMPOSITE LAMINATES FOR AEROSPACE STRUCTURES

The following faculty members have examined the final copy of this dissertation for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Doctor of Philosophy with a major in Aerospace Engineering.

_____
Walter Horn, Committee Co-Chair

_____
James Steck, Committee Co-Chair

_____
Janet Twomey, Committee Member

_____
Pingfeng Wang, Committee Member

_____
Charles Yang, Committee Member

Accepted for the College of Engineering

_____
Vish Prasad, Interim Dean

Accepted for the Graduate School

_____
Abu Masud, Interim Dean

ACKNOWLEDGMENTS

I would like to thank the National Institute of Aviation Research (NIAR) for their contribution of equipment and testing articles. I would also like to thank the NIAS/Industry/Schools (NIS) foundation and the Department of Energy (DOE DE-FG36-08GO88149) for their funding to help make this research possible. All results and conclusions were obtained from my own independent experimentation and testing without consultation with the Department of Energy.

ABSTRACT

Because of economic impact that results from downtime, aircraft maintenance is an important issue in the aerospace industry. In-service structures will decay over time. Compared to low-cycle loading structures, aerospace structures experience extreme loading conditions, resulting in rapid crack propagation. The research involved in this dissertation concerns development of the initial stages of a structural health monitoring (SHM) system that includes a network of ultrasonic testing sensors with artificial intelligence capable of detecting damage before structure failure. A series of experiments examining the feasibility of ultrasonic sensors to detect the initial onset of damage on a composite laminate, similar in structure to that used in aerospace components, was conducted. The artificial neural network (ANN) with the best accuracy was found to be a hybrid of a self-organizing map (SOM) with a feed-forward hidden and output layer. This was used for a single actuator-to-sensor scans on a composite laminate with simulated damage. It was concluded that a decentralized network of sensors was appropriate for such a system. The small four-sensor system was proven to be capable of predicting the presence of damage within a scanning area on a composite laminate, as well as predict the location once damage was detected. The main experimentation for this dissertation involved four ultrasonic sensors operated in a pitch-catch configuration. Simulated damage, verified through experimentation, was placed at various locations in the scanning area of interest. Signals obtained from the ultrasonic sensors were analyzed by a multi-agent system in which each agent describes an ANN. The system was trained to determine damage size. A second multi-agent system was constructed to determine the location of the detected damage. The architecture was similar to the damage-sizing system. Results demonstrated that with the artificial intelligence post-processing of ultrasonic sensors, 95% confidence can be obtained for detecting and locating damage that is 0.375 in. in diameter, which was verified through a bootstrap method. This dissertation validated the initial stages of constructing such a network of ultrasonic sensors. Future research in this area could involve combining the four-sensor network into a larger network of sensors by means of multi-agent processing (i.e., developing scanning regions). The novel method presented here provides the basis for development of an SHM system for typical aerospace structures.

PREFACE

The safety of aircraft is an important issue in the aerospace field. Aerospace structures often experience large variations in loading and environmental conditions during operation, which can lead to fatigue damage in the structure and cause catastrophic failure. For wind turbines or satellites, this could mean a loss of power generation or communication capability. For aircraft, a failure could result in the loss of life. As a result, maintenance is important in the aerospace field.

One measure of improved design has been in the materials used for aerospace structures. Originally, aircraft were built of wood and fabric to produce a lightweight structure. As the aerospace field advanced, aluminum and titanium metals became the main structural materials of choice. With current materials research, modern composite laminates have been introduced. These materials consist of ceramic fibers in a polymer matrix. The resulting combination of materials has the potential to withstand loads close to the properties of ceramics, while maintaining some ductility of polymers. The high strength-to-weight ratio of composite laminates allows for a reduction of weight in aerospace systems without compromising the structure's designed properties. This system can be tailored to meet design requirements of the structure. For example, fibers can be oriented in a specific direction in order to achieve the desired strength. The fiber-to-matrix volume ratio can also be adjusted to alter material properties. Ply orientation can be combined for complex-loading situations. The number of plies can vary depending upon the structure. Current designs are incorporating an increasing amount of composite laminates into their structures. Wind turbine blades and spacecraft consist mainly of composite laminate structures. Aircraft such as the Airbus A380 and the Boeing 787 are being constructed with more composite components than other aircraft in production. However, these materials have complex modes of internal damage that are not present in metals. For metal structures with an isotropic, homogenous nature, cracks are the main source of damage. In composite laminates, internal damage can take the form of fiber breakage, matrix cracking, delamination between plies, or, more commonly, a combination of these damage modes. This complexity in composite materials requires the aerospace industry's careful attention to maintenance and damage detection in order to maintain safety and confidence in the structure.

Preventative maintenance measures developed for aerospace structures involve inspection routines based on analytical, experimental, and field report data. These usually consist of a threshold value for the initial inspection. In other words, cracks in the structure are expected after a set number of hours of operation. Since damage is

assumed to be present and continuously growing, an interval for subsequent inspections is determined. These inspections consist of proven and reliable nondestructive evaluation (NDE) methods, such as visual inspection and eddy currents. An operator performing a visual scan of a structure and damage must deal with many issues, including the detectable range of inspectability and human error. Eddy current technology involves sending an electric current through a metallic structure. Although more accurate than a visual inspection scan, this process can be labor intensive. Some sections of a structure must have its finish removed in order to detect cracks accurately. Some parts must be removed from the aerospace system to accurately assess for damage. To conduct these inspections, the structure or aerospace system must be removed from service. The current system of routine inspections consists of investing money to cover the damage inspection, along with removing a structure from operation for a period of time in order to ensure safety.

Research into NDE methods has advanced the technology over time. Modern techniques such as ultrasonic testing, X-rays, and magnetic particle infusion have been developed and implemented in the aerospace field, but these are used only for specific structures. Advanced systems can detect smaller damage with more accuracy than traditional inspection methods. However, the equipment used for these systems is large and cumbersome. In the case of ultrasonics, human error is still an element that is present when scanning. The objective of NDE methods is to improve damage-detection capabilities by increasing the detection abilities of each inspection. Alternatively, the time between inspections can affect the ability to detect damage. If more inspections are performed, then there is a higher probability of detecting any damage at all. However, this obviously increases the cost of operating the aerospace system.

Alternatives to ground inspections are currently being investigated. Researchers have developed methods of installing NDE systems in a structure for in-operation inspections. The goal here is a structural health monitoring (SHM) system, which would monitor for and detect damage, similar to routine ground inspections, while the aerospace structure is in service, not interfering with its operation. This could reduce the number of necessary ground inspections and thereby reduce maintenance costs while maintaining a suitable level of safety for the structure. The research for this dissertation focused on the development of an SHM system that could adapt to a variety of aerospace structures and materials.

The NDE method of choice for this research was a system of ultrasonic sensors that obtain information about the internal condition of a structure by sending signals through it in the form of strain waves. This NDE

method does not require metallic properties as do eddy current systems. Consequently, ultrasonic sensors are ideal for inspecting composite laminate structures. However, these signals are complicated by physical boundaries, which cause a strain wave to reflect and wave energy that will dissipate into the surrounding atmosphere, thus causing a reduced signal. Another important factor in ultrasonic sensors is wave dispersion. The travel speed of an ultrasonic strain wave will vary, based on the wave modes present. Similar to s-waves and p-waves in earthquakes, both symmetric and asymmetric wave modes are present—the higher a signal's frequency, the larger the number of wave modes present. Each mode travels at a different speed through a material. If the distance from the source is large enough, then the effect will be detectable in the measured signals. Research has been conducted to analyze these received signals to determine the presence of damage as well as size and location of the damage site. Signal processing, such as wavelets, has also been investigated. The research reported on in this dissertation used artificial neural networks as a means to interpret the measured ultrasonic signals.

An artificial neural network is a form of artificial intelligence, which uses an analysis system that emulates the human brain by learning to map a set of inputs to a desired output set. This method allows for an approximate, but rapid, analysis of complex problems and systems. Once the ANN has "learned" how to function properly, it can be fault and noise tolerant, and can account for unknown variables and errors in the data and still achieve a desired output. An ANN system seems to be the appropriate candidate for analyzing the measured ultrasonic waves, due to the complexities of these waves traveling through a material and the presence of non-related additional noise and other unknown variables not related to any damage that might be present.

The objective of this dissertation's research was to combine an ultrasonic sensor system with an ANN post-processing system to produce an adaptable SHM system capable of detecting damage within a composite laminate. In preliminary research undertaken in the initial stages of this project, a simple sensor system consisting of two piezoelectric disks bonded onto a composite laminate with a layup of $[0/\pm45/90]_S$ was constructed. One disk functioned as an actuator and introduced a controlled strain wave into the composite laminate. The second disk was a sensor that received the signal. Simulated damage was placed at various locations around the actuator-to-sensor path. A wafer, bonded to the surface of the laminate, disrupted the ultrasonic waves traveling through the material in a manner similar to that of actual damage. Before an artificial neural network was implemented, a statistical model was developed in an attempt to use the measured ultrasonic signals to locate the simulated damage. This method was judged to be too complex to use as a post-processing method. So the next series of experiments involved using an

actuator to release a strain wave to two sensors. Damage was simulated in a similar fashion as in the initial experiments. However, for these experiments, two different ANN architectures were tested, and both had promising results. The networks were found to be capable of learning the difference between large-size and small-size damage, and could predict its location. A third series of experiments expanded on the concepts of the previous results, with an actuator sending a strain wave out to three sensors. Large damage was detectable and differential, and a prediction of the location in polar coordinates was possible. These initial experiments proved that artificial neural networks are capable of post-processing signals of various actuator-to-sensor paths to determine damage size and location.

A final laboratory investigation of the effects of using simulated damage rather than actual physical damage on a test article was undertaken. The object here was to detect the initial stages of damage. With a simulated damage wafer, the initial position and size of damage could be controlled and varied. A comparison of ultrasonic signals of actual damage to those of simulated damage on the same test article confirmed that the simulated damage represented 150% of the actual damage size. Further testing revealed that the effects on ultrasonic waves traveling through the structure due to simulated damage on one side of a panel were identical to those for simulated damage on the opposite side of the panel. The bonded wafer was determined to be an adequate means to simulate the effects of damage at any depth through the thickness.

The research for this dissertation expanded on the ideas of these initial experiments. Since each piezoelectric disk can function as both an actuator and a sensor, pitch-catch scanning systems were considered. This research focused on a four-piezoelectric-disk system, each disk being able to send out a strain wave to be received by the remaining three disks. The process is not simultaneous; rather, one disk sends out a strain wave into the material, and the remaining three disks receive the signal. This process was repeated four times per scan as a strain wave was released by each piezoelectric disk. From this configuration, four actuator-to-sensor systems scanned approximately the same location. The four disks were equidistant from each other, forming a square scanning area. The results of the four systems were combined to form a larger sensor system and were found to be more accurate. Additional scans of varying frequencies were able to collect more information about a structure condition, which resulted in high accuracy in both damage detection and location. Collecting the results was done by a fuser network, which also consisted of an artificial neural network. The development of this four-sensor system, presented herein, can accurately detect and locate damage around a square-shaped area. The system demonstrated that it is capable of

detecting damage with a diameter of 3/8 in. or larger with a 95% confidence level. As well, the system can locate damage to within 1 in. with a 95% confidence level.

This four-sensor system model can be used to create a multi-agent network of sensors. For example, six sensors could be arranged to form two adjacent, square scanning areas by sharing two sensors between squares. Any damage on the border between the two squares would be detectable by both squares. Another layer of fusing results should lead to more accuracy. From this initial setup, the system could be expanded indefinitely to form a network of sensors to monitor a larger structure, with sensors arranged in small squares. The resulting system would use localized scanning to detect damage within each four-sensor square. Further research could include combining the results of the system of four-sensor squares into a global damage detection system for the structure, identifying critical damage sizes and widespread fatigue damage.

The research in this dissertation focused on damage detection and location of a complex material. The developed system could be used for metal structures as well. Current aircraft have complex geometries, so the system could be applied to current aircraft structures. The system is in the initial stages of development and could potentially become an adaptable structural health monitoring system for any aerospace structure in the near future.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

Chapter Page

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF ABBREVIATIONS / NOMENCLATURE

Acronyms

AE	Acoustic emission
AHM	Airplane Health Management
ANN	Artificial neural network
ANOVA	Analysis of variance
DAI	Distributed artificial intelligence
FAA	Federal Aviation Administration
FBG	Fiber Bragg grating
KQML	Knowledge query and manipulation language
MAS	Multi-agent system
NDE	Nondestructive evaluation
NIAR	National Institute for Aviation Research
PZT	Lead-zirconate-titanate
RMS	Root mean square
SHM	Structural health monitoring
SMART	Stanford multiactuator-receiver transduction
SOM	Self-organizing map
WT	Wind turbine

Symbols

A	Cross-sectional area of fiber
C	Wave speed
C_N	Stress concentration adjacent to N broken fibers
d	Distance between fibers
E_c	Young's modulus of bulk composite
E_f	Young's modulus of fiber
E_{RMS}	Root means squared error
G	Shear modulus of matrix

h	Thickness of laminate
k	Wave number
L	Length of fiber
L_0	Reference fiber length
L_n	Loading factor
m	Shape function
n	Number of broken fibers
O_i	Output from node i in artificial neural network
P_f	Probability of fiber failure
P_n	Load along n^{th} fiber
p_n	Normalized load along n^{th} fiber
P^{∞}	Remote loading
Q_i	Number of groups of i fibers
r_f	Fiber radius
r_m	Matrix radius
t_k	Desired or target values for output dataset k
U_n	Displacement of n^{th} fiber
u_n	Normalized displacement of n^{th} fiber
w_{ij}, v_{ij}	Weight for path from node i to node j in artificial neural network
y_{in}	Weighted sum of inputs for node in artificial neural network
X_i, Y_i, Z_i	Node i in row of input, hidden, or output layer, respectively, of artificial neural network
z	Axial direction along fiber

Greek Symbols

δ_i	Region length of additional stress due to i broken fibers
η	Shape function
ρ	Density of composite material
σ	Stress in material

$\overline{\sigma}$	Normal stress in material
σ^{∞}	Remote stress
σ_0	Reference stress in material
σ_f	Stress in fiber
Θ, Φ, Ψ	Shape function
θ	Angle from fiber orientation in unidirectional composite
ξ	Normalized axial direction along fiber
ω	Wave frequency

CHAPTER 1

INTRODUCTION

Maintenance and detection of damage in structures have remained important interests in the aerospace industry throughout its existence. Before the invention of aircraft, humans utilized windmills by capturing energy from wind. All of these structures have been placed in service for operation with a designed lifetime. However, in actuality, many of these devices, through maintenance and repair, have remained in use far beyond this expected time. As advances in structural materials have been made, repair techniques have improved. The first flight in 1903 by the Wright brothers blossomed into the aerospace industry. Manufacturers of windmills adopted aerospace techniques to become more efficient. Throughout history, materials and structural shapes have improved from metallic structures to more advanced composite laminates. These advancements have improved the structural properties of components. However, even more advanced structures have reduced performance as the operational time increases. All structures accumulate damage with time, which can lead to failure. Preventative maintenance measures have been enforced throughout the aerospace industry. Using inspection methods, mandated by the Federal Aviation Administration (FAA) or other regulatory organizations, most damage within structures is found before any catastrophic problem occurs. For example, after a set number of flight hours, aircraft are grounded in order to scan them for damage. Wind turbines are removed from the power grid and operation to scan for damage in the structures. These procedures usually result in high costs and are time extensive.

Recently a major area of focus has been in the development of a structural health monitoring (SHM) system, whereby a network of sensors is mounted onto a structural component. These sensors can record data, and through signal analysis and post processing, damage can be detected in the structure. In aerospace structures, these mounted sensors can be operated while the structure is in service, resulting in fewer ground inspections. Of the modern techniques of damage-detection sensors, ultrasonics and fiber optics appear to be the most promising in the near future. The later of these is being experimented with to act as a more accurate strain gage. The focus of this dissertation is on the use of ultrasonic sensors for detecting damage in an SHM system. Their small size and high accuracy in detecting and locating damage is promising. However, ultrasonic signals become difficult to analyze for structures with complex geometric boundaries or material composition, such as fiber-reinforced matrix composites. Additionally, metallic aircraft contain many fasteners and intricate geometries throughout, although they are still built mainly with metallic materials. The research in this dissertation consists of a novel method of artificial

intelligence techniques for the signal processing phase of damage detection using ultrasonics. Current research utilizes wavelets and analytical techniques to signal processes. The novel method of utilizing artificial intelligence to post-process signals is predicted to be a faster method of damage detection. This SHM system could aid in improving aircraft, wind turbine, and other aerospace systems in early damage detection and assessment without compromising the abilities of the structure in operation.

1.1 Aircraft Maintenance

The safety of aircraft has been a high priority since their invention, due to the possible harm to humans from structural failure. During regular flight, aircraft experience environmental and force loading, which greatly stress its structure. Damage can occur from micro-cracks and voids inherent in all structures, as in the Aloha Airlines Flight 243 incident [1]; from poor design choices, such as the Swissair Flight 111 incident [2]; or even from human errors, such as impact events from dropped tools in manufacturing or inspection processes. One of the outcomes of the Aloha Airlines disaster has been to decrease the flight hours between inspections and increase the scrutiny of each inspection. The number of ground inspections has been increased because various components within the aircraft have differing inspection hours. Current methods involve grounding an aircraft and disassembling it to examine it for damage using visual scanning [3] or coin tap tests [4]. Advances in eddy currents as a nondestructive evaluation (NDE) process have allowed for metallic components to receive a more precise level of damage detection. However, for some areas, the finish must be removed to properly use eddy current detection. All of these methods still have a human factor in interpreting results for the presence of damage. If damage is detected, then the appropriate action for repair is performed. The maintenance schedule for aircraft is also designed to detect damage initiation. Here the goal is to detect any damage as early as possible before any catastrophic failure occurs. However, due to past aircraft crashes and reported repairs, the time interval between inspections, by necessity, has become more conservative and is designed to be short enough to detect damage before failure of the structure. Due to the size of aircraft and human error, detection is ideally at 0.1 in. but in practice is at around 0.5 in.. However, often no damage is detected during these inspections, where aircraft are removed from service for a considerable amount of time. The aircraft industry has been seeking methods to ensure flight safety while reducing the downtime between service flights. The goal of the current research was to develop a structural health monitoring system to scan for damage during aircraft operation and reduce or eliminate ground inspection methods.

1.2 Wind Turbine Maintenance

Wind turbines (WTs) have similar issues in terms of maintenance. They are expected to operate for as long as 20 years before failing [5]. Environmental and loading effects experienced by the structure can greatly shorten the lifetime of a WT. Although human safety is not as large a concern for wind farm operators, the power generated can be. Small damage within the structure could grow to failure of the entire system. As WTs increase in size for greater power generation, the costs of manufacturing as well as inspections increase. The reliability and maintenance procedures of these structures are gaining more and more importance [6].

The current maintenance routine for wind turbines has been copied from the aircraft maintenance model. A WT is inspected after a designed number of hours in operation. Here, the system is removed from the power grid and brought to a halt. Two inspection routines are followed for the WT blades, one through a rope system, whereby a person or machine scales each blade and scans for damage, and the other involving a crane that is customized to handle WT dimensions and lower the hub and blades of the turbine to the ground. Once grounded, the blades are scanned for damage. The scanning process involves using techniques ranging from visual inspection to C-scans. The time interval between inspections is a conservative estimate, similar to that of aircraft. Although this type of damage detection has generally been used as the general inspection method for in-service aerospace systems, there remain disadvantages. For example removing a WT from service reduces the power generated by the wind farm. Effects from the interactions among wind turbines in neighboring wind farms can result in a reduction of power generated by neighboring wind turbines as well [7]. The cost of dismantling WTs is much greater than aircraft. The equipment and procedure can be time and cost extensive as well. Some wind farms constantly have several wind turbines out of service and under inspection [8]. Risk of impact damage from tools used to dismantle and reassemble WT structures is an issue of damage initiation, similar to that in the aircraft industry. These disadvantages affect how and when maintenance inspections are performed. Like aircraft, wind turbines could benefit from a structural health monitoring system, possibly reducing the number of inspections and keeping more of these structures in operation.

1.3 Composite Materials

Researchers in the aerospace industry have sought materials that reduce weight while maintaining structural capability. Metals, specifically aluminum and titanium, have been the major material in past designs of aircraft. For wind turbines, wood has been the major material of choice. However, as research and improvements in materials develop, advanced composite materials have begun to be used more frequently in aerospace structures. Composite

laminate materials allow for a structural component with properties tailored to an application's requirements. Complex-shaped structural components can be manufactured using this material as well. These materials have a high strength-to-weight ratio and long fatigue life, which make them an excellent choice for aerospace structures. However, compared to metals or other isotropic materials, their analysis increases in complexity. Composite laminates consist of at least two different materials to produce unique property effects. However, material interaction along with differing material properties can lead to various modes of failures [9, 10, 11]. Delamination, fiber breakage, and matrix cracking are the common forms of damage found in composite laminates. Two or more damage types can be initiated and grow simultaneously within the material at various locations. This results in complex damage formations, which become difficult to detect and locate. Current inspection methods, which, as mentioned earlier, use a form of human visual inspection that limits the size of damage detection, can be hindered. Furthermore, eddy current technologies cannot function properly on most composites laminates, since they do not contain metals. Composite laminates often have damage located below the surface of the material, which is also difficult to detect. Critical locations can be removed from a structure and scanned by means of phased array and through-transmission C-scans; however, this newer technique is capable of detecting damage the size of 0.5 in.. Damage could be present at critical sizes but not noticed by modern nondestructive evaluation. Similar to multiple-site damage in riveted lap joints, various tiny damaged areas could become unstable and link up too quickly, growing to catastrophic failure of an entire structure if they remain undetected. Composite laminate materials are slowly being integrated into the components of many aerospace structures. The ability to manufacture complex shapes with controlled material properties has made composite laminate material attractive. However, the stress and failure of these materials is complex, requiring extensive analysis in order to obtain information about the internal state. Thus, advanced methods of damage detection and location are required for these structures.

1.4 Structural Health Monitoring

A structural health monitoring system has been defined as a "system with the ability to detect and interpret adverse 'changes' in a structure in order to improve reliability and reduce life-cycle costs" [12]. Research has been undertaken in recent years to develop an integrated sensor system with structural components. The general SHM system concept consists of a network of sensors and an analysis package, which could scan for changes in a structure that is in service. Any changes in the structure would indicate the presence of damage. Any SHM system must not interfere with the structural behavior of the system. For aerospace systems, this would mean that the

structure would be able to withstand flight loads and not increase the weight or performance of the system. The result could allow for fewer ground inspections and better real-time damage analysis of structural components. Promising nondestructive evaluation methods for an SHM system include a variety of ultrasonic sensors as well as fiber optic cables with fiber Bragg gratings (FBGs) [13]. FBGs are sensitive enough to register dynamic strains in the ultrasonic frequency range, which would allow for fiber optics to act as passive ultrasonic sensors for some situations. The application of a large array of these types of sensors into an integrated vehicle SHM system would be costly, but as these sensors become more readily available, the cost should decline. Using known attributes of the specific type of NDE sensors, the optimum choice and location of sensors within the structural component have been investigated to obtain a fully integrated system of embedded sensors [14, 15, 16, 17, 18]. These instrumented structural components could resemble a human structure, with the network of sensors acting like a nervous system.

Many NDE techniques have been considered for an SHM system, but the method that became the focus of this dissertation centered on the instrumentation for ultrasonic testing. Ultrasonic sensors consist of piezoelectric ceramics, more commonly used today [16]. These sensors use atom alignment composition of lead-zirconate-titanate (PZT) and have demonstrated deformations on the micro-inch scale. The system functions with an actuator-to-sensor pair. The PZT actuator receives a voltage signal and translates this into a strain wave. The strain wave then travels through the material and is received by another PZT sensor. The receiving sensor will then translate the signal into a voltage signal again. Variations in a received signal result from the structural geometry and any occurring damages. The strain wave will also interact with any boundary conditions, which for composites would include the different inner-material boundaries. Furthermore, many aerospace structures have airfoil shapes, which can be complex in three-dimensional space. For aircraft or wing turbine blades, changing chords, twist angles, and airfoil shape across the span of the wing can result in complicated structures. Also, properties of the surrounding atmosphere of the structure affect the “leaking” of the wave from the structure. The wave itself has modes that travel at differing speeds based on frequency. The interactions between fibers and matrix cause all of these problems. Wave reflections, scattering, and “leaking” all occur at this region. The resulting received signal can become quite complex with considerable information. The complexities of composite laminates further alter a traveling strain wave, such as fiber volume fraction, ply direction, materials used, woven or unidirectional fibers, and other factors involved in designing a composite laminate. Thus, signal analysis of strain waves becomes difficult. Finite element and wavelet analysis is often the only method available for analytically solving any signal analysis [17, 18]. These

methods become complex and difficult to apply, can require considerable processing power, and are specific to the configuration and ply layup created for the model. The goal of this research was to produce an autonomous adaptable system for various ply layups, sensor network positioning, and materials used in the structure. The addition of artificial intelligence methods offers a different way to learn to detect a composite and strain wave signature and interpret alterations in the signal to detect if damage is present.

1.5 Artificial Neural Networks

An artificial neural network is a form of artificial intelligence that consists of an analysis system that emulates the human brain by learning to associate a set of inputs to a desired output set [19]. The method allows for an approximate, but rapid, analysis of complex problems and systems. Once the ANN has "learned" how to function properly, it can be fault and noise tolerant, and can account for unknown variables and errors in the data and still achieve a desired output. An ANN system seems to be an appropriate candidate for analyzing measured ultrasonic waves, due to the complexity of these waves traveling through a material and the presence of non-related additional noise and other unknown variables.

Similar to the human nervous system, an artificial neural network is a system of interconnected nodes, activated when sufficient incoming signals are received [19]. Each node has an activation level of 1 for active and 0 for not active, similar to switches in serial computers. However, an ANN differs in that the activation value can be a value between 0 and 1 to account for approximations, or "fuzzy logic." If a node is activated, fully or partially, it sends a signal to the next set of nodes. Each interconnection with these nodes has a weighted value that modifies the effect of the signals traveling to the receiving node. The system is trained by example to optimize the weights of the nodal connections to "learn" the correct input-to-output operation. The connection pattern of nodes (ANN architecture) varies with the specific application. For this research, two different architectures—feed-forward networks and self-organizing maps (SOMs)—were examined in various applications within an SHM analysis to utilize the strength of the neural network.

The ANN training operation consists of introducing a training dataset of input-correct output pairs to "learn" an approximate relationship between a dataset of input-outputs pairs, similar to the process by which a human learns tasks such as hand-eye coordination. The training dataset covers the entire domain of inputs to outputs that the neural network would encounter in operation. Once the ANN has been taught an input-to-output map, it can be used to interpret input data that lie within the domain of the learning dataset to approximate the appropriate

outputs rapidly. This rapid solution is the advantage of an ANN over other online, time-consuming, analytical methods. Damage detection in an SHM system must be fast and accurate so that components with damage can be recognized well before failure of the damaged part occurs.

1.6 Purpose of Research

Often the structural health monitoring system has been linked to bodily systems that monitor the body for any changes. The nervous and immune systems have been studied and replicated in algorithms for various applications. Algorithms associated with the nervous system and brain "learning" functions can be applied to ultrasonic sensors, forming a monitoring device for an aerospace structure. Ultrasonic sensors have the ability to convert strain wave energies in a material or structure into an electrical signal. These strain waves can be generated as the result of internal crack growth or corrosion in the structure, or these waves can be introduced into the material through actuators. The monitoring of crack growth is a passive system. Previous research has shown that signals obtained by passive listening to structural components can identify if damage growth is present [21]. Complexities of composite structures can make this process difficult to fully assess the results of passive listening, due to the presence of other noise [22]. A received sensor signal can become increasingly complex, with multiple damage types and locations growing simultaneously. Although difficult to analyze for detailed results on the presence of damage, the passive SHM system can act as an early warning device. An active ultrasonic testing system could be utilized for a more detailed analysis of damage severity and location. Thus, an active ultrasonic system was examined in this research. Once damage is detected by a passive system, then an active ultrasonic system could be activated. This active system would then be coupled with the passive, ultrasonic SHM system to obtain more accurate assessment of damage size and location.

The main focus of this research was on the development of an active ultrasonic system for this scenario. A decentralized multi-agent system (MAS) for signal analysis was determined to be the best approach. The research here focused on the development of an ultrasonic sensor network for a small area with a generic shape. The system utilized artificial neural networks for signal processing. Once developed, the small network could be replicated across an entire structure. A system of communication between neighboring networks was also developed to increase accuracy. The resulting system was a network of sensors that operates on a localized scale and detects damage with a high accuracy of severity and location.

CHAPTER 2

BACKGROUND

Since aircraft maintenance remains an important issue, the methods for detecting damage are an important research interest. In this dissertation, a structural health monitoring system with ultrasonic signals and artificial intelligence analysis for aerospace structures was created. Many different fields of study must be combined to form such a system. Composite laminates must be investigated for stress-strain and fiber-matrix interactions. The fracture mechanics of composite laminates was also researched. The effects of a broken fiber or a cracked matrix can alter how a structure will distribute stress. The dynamic movements of energy strain waves through such a material must be researched as well. How the energy wave travels through a composite laminate and how the presence of damage affects the wave must be investigated. These strain waves will be detected by sensors. The gathered signals are post-processed by artificial intelligent methods, including multi-agent systems and artificial neural networks. Much research has been performed in these various fields of study. This section details past advances in each field and describes how these different fields could be combined into a SHM system for an aerospace structure.

2.1 Materials and Damage Tolerance

Structures can contain damage throughout all stages of their lives. Damage can begin with manufacturing errors and mistakes and then grow through a structure's operational use. This can take the form of fatigue, corrosion, impact events, or other effects due to service conditions. If damage becomes extensive, then the strength of the structure can become compromised and failure could occur. Brittle fracture due to fatigue loading is a major concern in aerospace structures. Micro-damage or voids can grow with time through cyclic service loading. Critical damage within a structure can potentially lead to the catastrophic failure of an entire system. This has led to investigations into the nature of damage and damage growth.

2.1.1 Metallic Structures

Many methods for analyzing materials began with isotropic, homogeneous states. This has allowed for simplification of analytical modeling of the stress-strain or damage state of a material. As a result, many structures are built from metals, which have desirable material characteristics as well as isotropic properties.

The concept of flaws within a material can be dated as far back in history as Leonardo da Vinci (1450's to 1500's), who studied wire behavior [23]. Using different lengths of similar wire, he discovered that the tension required to break each wire segment varied. From this, he observed that longer lengths of wire required less tension

for failure than shorter lengths. He concluded that the wire had random internal flaws; thus, the longer the wire, the more likely a significant flaw would be present.

Significant advancement of the knowledge of structural failure was not reported until the 1800's, when the British Iron and Steel Company reported failures of structural components from brittle fracture at applied loads far below the determined failure loads [24]. During this time in history, structural failure was studied after the failure occurred. Very few predictions of any structure's lifetime were made.

The next major contribution to the study of brittle fracture failure occurred in the 1920's, when an engineer, A. Griffith, sought to develop a theory governing the failure of materials in structures. Through his experiments, he concluded that every material contains micro flaws. Since internal stresses within materials increase drastically in the vicinity of voids, any increase in stress around these cavities would increase the size of the voids and create cracks, even if the nominal stress within the material was far below the calculated critical stress. His theory related the elastic strain energy released by a crack as it extends in an elastic material to the energy required to create the additional surface area of the extended crack. The idea he presented later formed the basic inverse relation between the stress applied to a component and the square root of the crack length within the component [25].

In 1939, Westergaard further investigated the stresses at crack tips for elastic materials. From his research, the stress field surrounding the crack tip within an elastic body was found to be a function of the inverse of the square root of the distance from the crack tip. This helped validate the notion of stress around a crack tip being extremely large, compared to that of elastic material far removed from the crack tip. In 1948, Irwin and Orowan independently modified Griffith's theory to account for the energy of plastic deformation. In 1956, Irwin developed an equation that combined the influence of the applied stress, crack size, and geometry of the structural component with a crack into a single parameter, called the stress intensity factor, K. This concept allowed for easier assessments of cracks in elastic structures and led to a critical value for the stress intensity factor for which crack extension would occur [25].

The focus of much of the research of brittle failures during the 1960's and 1970's concerned crack extension resulting from cyclic loading. Failure due to fatigue loading is important, since all machines experience some cyclic loading while in service. In 1960, a group of researchers, led by Paris, developed a relationship between the rate of crack growth and the number of stress cycles applied to the component. This theory along with fracture mechanics developments relating the strength of material to crack length made it possible to predict when a flawed

component subjected to cyclic loading would fail. Thus, one could estimate the remaining useful life of the structural component [25].

2.1.2 Composite Material Structures

Composite materials, although having in use for many years, have grown in popularity as a major building material for engineering structures. Early examples in history include the reinforcement of mud bricks with straw by the Israelites in ancient Egypt and the reinforcement of bows to increase strength by the Mongols. The popularity of composite materials grew in the 1930's, when the Owens-Corning Fiberglass Company began advertising the material for more common use [26]. The ability to tailor material properties to desired values along with high strength-to-weight ratio has become ideal for the aerospace industry.

Structural components in aircraft, which have historically been metallic, are increasingly being replaced by composite materials. Many leading-edge flaps in aerospace structures are now made of Kevlar or fiberglass. These properties allow for impact events to be less harmful to this part of the aircraft. The elevators in fighter aircraft consist of uniform pieces of boron-fiber composite. This material is capable of handling the loads applied during supersonic flight. Many structures are being replaced with carbon-fiber and epoxy matrices. This combination offers the best performance for aerospace structures. The first all-composite aircraft, the Beechcraft Starship, had carbon-epoxy composites throughout its structure. The Boeing 787 contains approximately 23 tons of carbon fiber within its structure [27]. Because more research is being performed on composite materials, greater confidence in their performance applications is increasing in the aerospace industry.

Another common aerospace system is the wind turbine. WT blades consist primarily of composite structures in order to reduce weight [28]. They are typically made of woven or unidirectional E-glass fibers and a mixture of polyester and epoxy [28]. Recent trends in blade manufacturing have been to produce parts with epoxy due to better material properties and the use of prepreg layers for manufacturing processes. The advantages of carbon fiber are much greater than that of glass fiber. However, the main material of choice for wind turbine blades is glass, since it is much less expensive. As the cost of carbon fiber decreases, this material will be utilized more in WT structures.

Aerospace structures are an excellent field for composite laminates. Precise structural shapes with complex loading spectrums are abundant in aerospace systems. Composite laminates with woven or quasi-isotropic plies are used in such structures to reduce weight and the cost of manufacturing, and withstand the necessary loading.

However, the improvement in material properties also results in the complexity of any analysis or analytical modeling. Extensive research has recently been performed on the analysis of composite laminates, resulting in two different methods: one with analysis on a macro-scale, and the other with analysis on a micro-scale [9]. While some research investigates the internal structure and micro-voids in composite laminates, other research focuses on the overall performance of the material. Further studies have even begun on the fracture mechanics in composites. Most studies involve composites with reinforcement fibers, generally with a unidirectional orientation. Initial studies have consisted of investigating the ultimate strength of a composite material. A basic approach has involved looking at the fibers without a supporting matrix, using the principles of da Vinci's discovery. If a fiber is long, then there is a higher probability of failure at a lower load, compared to a short fiber. This process has used probabilities to determine if a bundle of fibers could withstand a specified load. However, this idea does not include the supporting factors of a matrix. A broken fiber can still transfer load to a surrounding matrix. Then a single fiber in a matrix was modeled. The single fiber could transfer load through shearing forces if the fiber was to break at a point. This "shear lag model" led to the research performed by Hedgepath and Van Dyke in the 1960's [29]. A single broken fiber would not only transfer load to the surrounding matrix but also nearby fibers. These nearby fibers could fail by additional stresses placed upon them by the additional loading, which could result in multiple fiber breakage. The researchers developed models to predict failure loads for unidirectional composites with a broken fiber present.

The shear lag model developed by Hedgepath and Van Dyke was expanded in the 1980's by several researchers [30, 31, 32]. Batdorf modified previous models with a three-dimension system. This model included using the effects of different fiber-breaking cluster geometries to transfer loads to surrounding fibers. Batdorf's objective was to use probability to determine the strength of a composite material [30, 31]. Another research group expanded the model to include short fibers. Their results include the effects of load transfer from both ends of a short fiber to surrounding continuous fibers [32]. These methods were investigated further and improved upon by later researchers. Research is currently being performed on cut-fiber and woven-fiber configurations. These composites contain many more intricacies in their analysis; however, the failure theory of these materials could be developed in time. Although current models were developed to determine the ultimate strength of a composite material, they could be expanded to look at the remaining life or performance condition of a damaged composite. If a fiber, or even multiple fibers, is known to be broken in a composite material, then these models could be applied to

determine new critical loadings. The remaining life of a composite could be determined, using a nondestructive evaluation method to identify and locate damage within a composite and predict failure probability.

2.2 Nondestructive Evaluation

Scanning for damage of structural systems is commonplace. This most basic form of detecting damage would be to disassemble a structure component to view and access all boundary surfaces of the component. A more efficient way to detect damage without disassembly is prefered. As a result, research has been performed to discover techniques to detect small cracks and damage in a material before the part fails. Current studies are seeking a system to conduct damage detection while the structural component remains in service. A wide variety of nondestructive techniques have been developed in order to create an optimum system for structural health monitoring. This section contains descriptions of some of the common nondestructive evaluation methods, including those being researched and those that are being practiced in routine maintenance.

2.2.1 Early Ultrasonic and Nondestructive Evaluation Methods

The earliest known technology for NDE was through visual inspection. Humans have been able to literally look at structural elements to determine if any problems are visible on the surface. For scheduled maintenance of an aircraft, some structural components are inspected visually by technicians. If a surface crack is visible, then the part must be repaired or replaced.

Another simple method of inspection is that of audio inspection, sometimes referred to as the “coin tap” test [4]. By tapping sharply on a structure, a trained person can detect changes in frequency from one region to another. If damage has occurred in a component, then some of the natural frequencies of the response to the tapping would have changed. If damage is significant, then the difference in the audible response would be distinguishable by a technician’s ears. Although crude, this method is still used today to find general faults in some structural systems, because of its ease and quickness. Some research on changes in frequency, or impedance monitoring, has been conducted in order to improve this method [17, 33]. Results from this technique are simplistic, limiting detection to large-scale damage. As a result, other technologies have been sought to better detect smaller forms of damage before failure occurs.

As shown in Figure 2.1, a system of strain gages and accelerometers has allowed for the measurement of many properties; however, damage determination has remained difficult to obtain with such a system. Some researchers have investigated a procedure for detecting damage in the vicinity of strain gages through a measure of

material property changes. That is, as a crack extends, the stiffness in the surrounding area of the structure changes [34]. This change is then detected with strain gage measurements, which discover those cracks. However, similar to visual inspection and coin tap testing, damage must be quite large to be detectable.

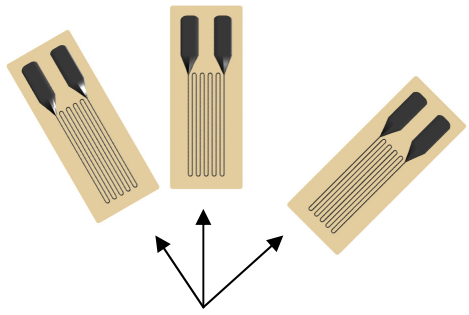
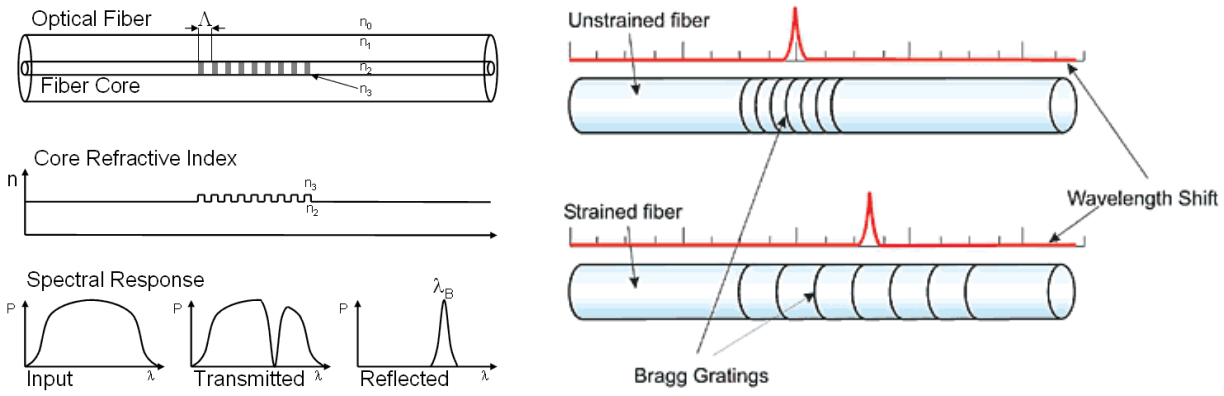


Figure 2.1 Strain gage rosette using array of three strain gages.

Strain gages have significant deficiencies that reduce their applicability for health monitoring applications. Apart from detecting only strain or a change of strain in local areas, they are temperature sensitive and susceptible to sensor drift and electromagnetic interference, thus requiring frequent calibration. An SHM system should be able to function without calibration for an extended period of time in order to reduce ground time and cost. Consequently, even though strain gages are inexpensive, they are not projected to be the key sensor for an SHM system.

A more promising method of nondestructive evaluation is in the field of fiber optics. The transfer of light through a glass tube was proven in the 1840's, but optical fiber was not developed until the 1950's. In 1978, Hill demonstrated the first fiber Bragg grating within an optic fiber [35]. With this development, optical fibers have been used to measure fiber elongation and thus strain within the structure to which it is bonded.

Fiber optic techniques center on the ability to measure a light wave pattern traveling through each optical fiber. Any deformation in the fiber will result in a change in wave shape of the light traveling through the fiber, which can be analyzed by a signal conditioner using an FBG pattern etched within the optical fiber. A Bragg wave is generated by a light source and travels through the optical wire. As it moves to the region containing the FBG, a small portion of the wave is reflected back. The wavelength of the reflection is altered by the elongation, or strain, applied to the FBG region of the fiber. Therefore, it can be used to measure longitudinal strain. This process is illustrated in Figure 2.2.



(a) Basic setup of Bragg wave and reflected wavelength of light source

(b) Wavelength shift due to elongation of FBG

Figure 2.2 Visual of fiber Bragg grating as strain gage.

Because only light waves are transmitted through the fiber, electromagnetic interference is avoided and little to no sensor drift occurs. This system can also be used in a manner similar to strain gages, in that an array of fibers and FBGs can be arranged to form a rosette pattern, similar to that shown previously in Figure 2.1, in order to obtain longitudinal strain in three directions at a single point. The fiber optic system has a high data-acquisition or sampling rate, so it could be coupled with an ultrasonic system. This arrangement could produce a lightweight SHM system composed of optical fibers and an array of ultrasonic actuators/sensors.

However, the fiber optic system does have some deficiencies. Optical fibers operate in detecting oscillating strains on the lower end of the ultrasonic frequency range, which leads to them being poor acoustic emission sensors with lower quality data when compared to other forms of ultrasonic sensors. The price of the fiber optic system is high relative to strain gage systems, but this price could decrease with increasing use in civil and aerospace structures. Research is ongoing to increase the abilities of fiber optics and FBGs [36, 37, 38]. This work includes fiber optic cable within composite laminates. Although some strength is compromised from embedded fibers, the structure would not compromise any surface boundary conditions [39].

In conclusion, fiber optics provides an alternative to strain gages and possibly has the ability to be used with an ultrasonic system, despite the currently relative high price. This method has been researched and used in practice for hot-spot locations. The focus of this dissertation research was on ultrasonic testing techniques, which are described in the following section.

2.2.2 Piezoelectric Materials

Many other nondestructive methods involve the analysis of strains in a material. Since these strains are small, an accurate sensor must be used. Strain gages are not expensive to make, but their measurements have low accuracy. Thus, other forms of sensors are in development. Current research has investigated active and passive ultrasonic techniques with the use of a piezoelectric material sensor. When such a sensor is created, it is cured so that the grain alignment within the material is unidirectional, resulting in a unique trait. Piezoelectric materials produce a voltage when deformed, or they inversely deform when subjected to an applied voltage. The first demonstration of this piezoelectric effect in a material was in 1880 by Pierre Curie and Jacques Curie. Woldemar Voigt published a book in 1910 on crystal physics [40], detailing 20 crystal classes of piezoelectricity and determining constants and comparable traits of these materials, which are common in today's microphones and speakers. In the 1950's, a ceramic crystal of lead-zirconate-titanate was discovered. This PZT crystal became the primary material for acoustic emission and ultrasonic sensors, since it is sensitive enough to produce a change in voltage as a result of its deformation when subjected to a strain wave of the level associated with crack extensions [41, 42]. However, as a result of the crystal's sensitivity, it also detects the superfluous noise present in all dynamic structures. Much of this noise can be reduced with the aid of signal filters; however, some noise will be present in the signal and produce an error in the final measured strain waves.

2.2.3 Passive Acoustic Emission

The concept of passive acoustic emission (AE) testing has existed for many years and recently has been extended to include an even larger range of materials. It is known that severe cracking of structures produces an audible sound that can be heard by the human ear. Early man listened to pottery cooling for sounds that would denote structural damage. As early as 3,700 BC, tin smelters in Asia termed the phrase "tin cry," which described noises produced by deforming tin. As early as the eighth century, books documented the changes in sounds that different materials produce as a result of deformation. By the 1800's, acoustic emissions from tin, iron, and other materials, audible to human ears, were described in books. Czochralski, and later improved upon by Anderson, correlated the audible sounds and the materials that produced them. Tensile tests of metals during the mid-1900's were published by Kaiser, who is credited as the principal source of modern day AE techniques [43].

Current acoustic emission testing methods make use of a variety of ultrasonic sensors to detect emissions of energy that are too small to be audible. These sensors allow for the detection of minute increments of crack

extension on the order of 0.01 in. within metals. Therefore, it is possible for a system to detect crack initiation well before the failure of structural components. However, this system is extremely sensitive. The passive system does gather vast amounts of information for damage detection. It can become complicated in composite structures, due to the nature of damage in this type of material. Previous work has shown the degree of difficulty in interpreting the results for an AE system [44]. Thus, for the research in this dissertation, an active ultrasonic system was investigated.

2.2.4 Active Ultrasonic Testing

During World War I, a pulse transit-time method was devised for ships [16]. This idea came about as a way to detect icebergs at a distance that was sufficient for routine avoidance. The technique was soon adopted on surface ships and submarines for detecting other ships in the vicinity, thus forming a simple radar/sonar system. This system was an early form of a pulse-echo system, where pulse waves were sent out through the water, and waves were reflected off of any obstructions in the water and returned to the transmitted ship. However, this concept of pulse-echo waves was not adapted to a nondestructive evaluation method until 1940 by Firestone, who recognized its importance in the field, and shortly after in 1942 by Sokolov [16]. Both researchers used previous discoveries of the effects of sound waves in materials to create a pulse-echo system to send out sound waves into materials and then analyze any returning response waves. The differences between response waves in the material and those within an undamaged structure would indicate any internal cracks or voids. Thus, the beginnings of nondestructive, ultrasonic testing were formed [16].

Discoveries in seismic exploration led to knowledge in the nondestructive evaluation field. These waves could also be used to study effects in the Earth. During the 1950's, researchers developed models for waves traveling through plane-layered media. These models predicted wave speed through materials and consisted of long wavelengths traveling through layered anisotropic media. It was not until much later that frequency dependence became important, as the wavelength size became comparable to layer thickness because of thinner layers. During the 1960's and 1970's, various studies were conducted in developing the theory of waves in periodically laminated media. These included stiffness effects on the wave and focused on determining wave speed throughout the material. The studied frequency became shorter and shorter, and the effects of frequency on the wave speed were studied as well. The dispersion of wave modes in isotropic materials was modeled. The interaction with surface boundaries led to modeling in layered media. Studies of fiber-reinforced composites in the 1980's focused on the interaction

between fiber and matrix, along with wave travel in cylinders and plates with shells. Research on a combination of layered fiber-reinforced laminates was studied by researchers in the 2000's. The study of wave travel in composite laminates is a recent idea, and much study is still ongoing in order to fully model wave travel speeds in all media [17, 45].

Due to the limitations of sound-producing materials available in the past, low frequencies have been used. As advancements in ceramics and piezoelectric materials have occurred, today's actuators and sensors that are used to pulse and receive strain waves in materials are able to produce frequency sound wave pulses in the ultrasonic frequency range of 20 kHz and higher. A known energy wave is sent through a material and received by a sensor. The energy wave is affected by boundary geometries and any damage present. A received signal contains considerable information, and the process of analyzing a signal wave to extract this information has become a critical research effort.

2.3 Signal Processing and Artificial Intelligence Analysis

Many natural phenomena can be modeled mathematically to predict resultant actions if the phenomena should occur again. The solution of many problems lies in the form of differential equations, which often require assumptions and specific methods to solve. An example would be a bar pivoting about one of its ends. The position of the bar can be defined in terms of the angle of the bar relative to a horizontal line as well as the angular velocity. If the problem is expanded to the human arm, then the mechanism contains multiple members and joints. All of the individual members have multiple degrees-of-freedom in three-dimensional space, thus adding complexity. Although a solvable problem, the task of predicting and controlling the movement would require many time-intensive calculations. But for humans, the tasks of walking or picking up objects with arm and finger movements are quickly performed within the brain, with accuracy and precision. Thus, researchers have searched for a mathematical procedure to replicate the thought process of the brain and implement it to solve difficult problems with no defined analytical solutions.

Debates on what defines intelligence have even ensued [20]. Early conception has focused on the computation of complex problems, resulting in the popularity of the serial computer. These systems have worked well in quickly and accurately determining an answer with given input. However, they could not perform the same action if a problem statement was presented to them in words. Furthermore, these computers would not be capable of reading words off a page. Current methods in artificial intelligence are seeking ways to achieve these goals. A

structural health monitoring system could benefit from these new developments in artificial intelligence roles, since it is a complex problem itself.

2.3.1 Artificial Neural Networks

Artificial neural networks, or similar concepts with different names, have been studied in various fields. For example, most artificial intelligence learning procedures fall under the category of ANNs, since the principles of the learning system are the same.

The concept of an artificial neural network was introduced in the 1940's by McCulloch and Pitts. Through specific network architectures, they created simple two-neuron systems, which could solve basic logic problems, such as AND and OR gates. Hebb suggested the first rules for a learning algorithm for a simple network in 1949. Using these laws, simple networks could "learn" how to solve basic logic problems by being trained with examples. In 1962, Rosenblatt created simple one-layer networks, called perceptrons. In addition, Rosenblatt developed a learning scheme more powerful than Hebb's learning laws, which led to the creation of better networks. The limitations of these networks were merely the range of problems they could solve. Only lower-order problems could be solved with this style of network. It was not until much later that higher-order problems could be solved by ANNs [19].

In the late 1950's, Von Neumann developed the serial computer, which involved basic sequential-lined coding and computations. Since the serial computer performed much better than ANNs in solving problems, the main focus of research shifted to this type of computer in the 1970's. As a result, current computers and most programming software languages follow this process, initialized by Von Neumann, and research on neural networks has been greatly diminished [19].

It was not until the 1980's that neural networks became a popular focus of research again. In 1986, Rumelhart et al. developed a procedure to train neural networks with multiple layers of neurons, or nodes. The result was a procedure to solve higher-order problems, which were impossible for simple perceptrons. Since ANNs are massively parallel processors, they have the quickness to solve very complex problems, compared to the serial computer. Around this same time, Kohonen developed a novel form of network architecture, known as a self-organizing map, which categorized datasets into different groups and placed them onto a two-dimensional topological map. Networks based on this architecture can group objects without any prior knowledge of the category sets and often discover dataset patterns that the human mind would not have detected [19].

This resurgence of artificial neural networks continues. Recent and classical problems are being researched with neural networks to examine their learning ability to solve complex problems. While ANNs may not have a use in all applications, the field of signal processing and pattern recognition for structural health monitoring certainly appears to be a good match with neural networks.

2.3.2 Artificial Immune Systems

Alternative methods have used algorithms in the modeling of the immune system. These systems operate on the basis of continuously scanning for anomalies in the sensing range. This is similar to how the immune system detects a virus in the body [46]. Once a disturbance has been detected by the system, the results are compared to a database of known anomalies. Some fuzzy logic is applied in order to match the current problem with past problems. Once identified, appropriate actions are taken. The problem of damage detection is similar to the initial stages of an immune system. The system continuously scans for damage. Once something is identified as abnormal, then the resulting signal changes could be matched to a database of known damage signals. Problems with this include requiring a database of damage positions and signals. Laminates are highly customizable for each application. Furthermore, composites allow for complex shapes and curves for aerodynamic and aesthetic choices. A database may be difficult to form due to the infinite possibilities. The idea of an artificial immune system is mentioned in this dissertation. However, artificial intelligence methods have shown more promise in identifying and locating damage within a composite aerospace structure.

2.3.3 Multi-Agent Systems

Multi-agent systems are a sub-topic of distributed artificial intelligence (DAI). This new concept was developed in the 1980's and gained interest in the 1990's. Many studies have been conducted to model human behavior and intelligence. Artificial neural networks are models of an individual brain. The idea of shared information between multiple minds has gained interest in another layer of intelligence. An example of DAI is the swarming of animals, or swarm intelligence. A fish in the ocean can have a random swimming behavior; however, a school of fish will act as a unit and respond appropriately to any threats encountered [20]. That is, each individual fish has intelligence of its own, but the school of fish also behaves intelligently. Early interest in this area began in the 1980's with distributed decision-making systems. Each individual system was termed an agent, capable of independent intelligent decisions. However, each system could also share information with each other, in order to

make more-informed decisions. The goal here was to have each agent work independently, while maintaining the overall objective of the system [47, 48].

This concept was not explored much further until the 1990's, when several researchers began to develop DAI systems. Demazeau et al. studied the ideas of each agent as a knowledge-based system [49, 50]. Each individual agent was capable of making a decision, based on previous, learned knowledge, similar to artificial neural network methods. Chaudron et al. developed the concept of each agent being organized in a group system as well as being an expert at least at the local level [51]. Each agent has the ability to perceive information and make an informed decision on an action to perform. The work of Bond and Gasser focused on the coordination of these agents with each other, including methods of relaying information between agents and performing actions in a decentralized manner. No system would be required to control the individual agents, but instead they would be motivated to act for both individual and group goals. Together all of this work formed the modern form of multi-agent systems. DAI systems have the potential for scanning for damage within systems. Signals and predictions of neighboring sensor networks could discuss predictions to increase the accuracy of a final prediction. This concept is new, and techniques are still being discovered to apply such a method to systems. However, there is potential in applying the method to a structural health monitoring system [20, 47, 48, 52].

2.4 Structural Health Monitoring

Various methods of nondestructive evaluation are being researched and used in practice. Some NDE systems have used current technology to form a basic structural health monitoring system, while others are under development and utilize artificial intelligence signal processing techniques. Research in the field of structural maintenance has addressed the potential for in-service structural health monitoring in recent years. Studies have continued to improve all methods of monitoring for damage within all forms of structures, although the main focus has been in aerospace, biological, and civil engineering projects.

For aircraft systems, these studies have ranged from improving the methods of current ground maintenance schedules to scanning for in-flight damage. The Boeing Company has investigated a method of utilizing strain gage readings along with other sensors currently installed on aircraft. They have built a database of measurements taken during flight, which are then used as a statistical baseline for analysis of aircraft during flight. The system, called the Airplane Health Management (AHM) service, takes measurements from sensors on aircraft during flight, compares readings to the database of similar aircraft models, attempts to determine if a structural problem exists, and then

identifies the location of the problem. Previous problem areas and instrument measurements have been incorporated into the AHM system to analyze the aircraft in-flight. This method of statistical analysis has shown promise and could also be applied to technologies other than structural health monitoring [53].

Further research has examined the possibility of embedding an entire sensor system as part of the structure itself. Studies of embedding optical fibers [39] or piezoelectric actuator/sensors [54] into composite panels have been conducted, since these sensors are able to withstand the curing process and remain functional during and after the manufacturing process of the structural component. As a result, the structure would operate normally but with internal sensors constantly scanning for damage. Another research focus has been on creating a quick processing health monitoring system. Kirikera et al. have developed a structural neural system, which utilizes acoustic emissions and a specialized data collection process to determine damage location in a flat structure [55].

Similar research has involved the investigation of the potential of artificial neural networks as a means to post process complicated ultrasonic signals. Strain waves from a point source were detected by a series of piezoelectric strips. The signals from these strips were used in a feed-forward ANN to determine location. The system was proven to locate point sources within the area of interest on the structure. This research demonstrated that there is a possible use of ANNs coupled with NDE techniques to identify damage within the structure [56]. Another example is the research work performed by Crupi et al., where an artificial neural network was trained to know normal operating conditions. Any deviation from this would be from the result of damage. The outlier in the data would be a signal that damage was present within the system, and further investigation would be required [57]. Artificial neural networks have also been employed for detection in buildings. By analyzing the natural frequencies of a building's frame, an ANN learned to estimate damage severity on a scale from 0 to 1. The network was proven to predict the presence with low error [58]. Previous research studies have demonstrated that ANNs are applicable in the field of nondestructive testing.

The research in this dissertation had a goal of combining the knowledge of composite materials, using Batdorf's discoveries in fracture mechanics [30, 31], active ultrasonics using the studies of wave theory by Rose [17] and Datta and Shah [45], and artificial neural networks with advances made by Rumelhart et al. [19]. This research contains the initial stages of the development of a structural health monitoring system, capable of monitoring an aerospace structure in near real-time, taking advantage of the abilities of each of these individual elements. In this research, a composite laminate had simulated damage introduced. Through an active ultrasonic

sensor system and artificial intelligence as post-processing, an SHM system can be formed. The system developed here shows that artificial neural networks have promising possibilities in the field of nondestructive evaluation by allowing quick analysis of results, gathered by an NDE system. Using fracture mechanics procedures, a quick assessment of the remaining life of a structure could be obtained. This research also shows that combining various fields of study can provide the necessary tools for a functional SHM system. For further details on previous research and topics, see Appendix A for additional references.

CHAPTER 3

ANALYTICAL METHODS

The research in this dissertation utilizes different areas of study to develop a structural health monitoring system. This chapter discusses the analytical methods used in this research: (1) fracture mechanics of composites, (2) ultrasonic wave theory, and (3) signal processing with artificial intelligence methods.

Composite laminates have the capability of being tailor made to each structural application. Any change in a structure's geometry will affect how ultrasonic waves travel through it. Any complex ultrasonic signal, obtained from sensors, could then be used in an artificial neural network to determine the difference between a damaged and non-damaged state. In this way, a structural health monitoring system is formed. This chapter details the various stages in the SHM development, from composite laminate fracture mechanics to artificial neural networks.

3.1 Fracture Mechanics of Composite Laminates

Historically, aerospace structures have consisted of metals. Their material properties have been suitable for many applications, and their isotropic nature has allowed for easy analysis of these materials. Specifically, aluminum and titanium alloys are used throughout aerospace structures. Recently more attention has been given to composite laminates as a substitute for metallic components. Their high strength-to-weight ratio and easy tailoring of directional material properties has resulted in composites replacing metals in wind turbines, aircraft, and other aerospace systems, resulting in lighter systems without compromising structural strength. However, a side effect of using composites is the complex analysis of the structure at the microscopic level. Elastic analysis methods for isotropic materials and applications of composite laminates are detailed in this section. Analytical modeling of basic fracture mechanics of composites, including a description of fiber breakage, matrix cracking, and other damage modes, is also developed.

The main focus of studying fiber-reinforced composites is their ultimate strength. Usually, fibers are composed of brittle materials, which typically break before yielding. The fibers within a composite support most of the load. If a fiber breaks, the surrounding matrix must transfer the load to other fibers in the area. Thus, an increase in stress occurs near the area of the broken fiber. If loading becomes too great, then neighboring fibers may not be able to support the increased load and break as well. If a load to a composite is too great, then the fibers within the composite will break in grouped areas. These areas can connect to form greater damage areas and lead to a failure of the entire material. The connection of damage areas due to increased stress concentration is similar to that of

multiple-site damage in riveted joints. Cracks that form between rivet holes can rapidly link up through unstable crack growth and cause failure of the joint.

Consider a group of several single-strand fibers, each having a different length, and a similar tensile stress is applied. All fibers obey the same material property rules, but as the fiber length increases, the probability of micro-flaws also increases. Thus, the longer a fiber, the less stress a fiber can withstand before breaking. Similarly, the higher the applied stress, the greater the chance that ultimate stress is obtained and the greater the probability of breaking. Researchers have developed an equation, based on a Weibull model, to determine the probability of fiber failure, P_f, as a function of fiber length, L, and strength, σ:

$$P_f(L,\sigma) = 1 - \exp\left[-\frac{L}{L_0}\left(\frac{\sigma}{\sigma_0} \right)^m \right] \tag{3.1}$$

For the equation, a shape function parameter, m, is used, along with a reference length, L_0, and corresponding strength, σ_0. These reference parameters are obtained by experimental testing of fibers of similar material and diameter, but with varying lengths and stresses. The reference parameters then become a mix of geometry and material properties.

Several methods are used to model fibers in a matrix. For fracture mechanics, these include a shear lag model and the Batdorf tensile strength model, both researched previously and entail examining uniform directional fibers in a matrix. These are two- and three-dimensional models but maintain a uniform direction of fibers. If a nondestructive evaluation technique was capable of determining damage size, then the estimated number of fibers broken could be determined. From this, an estimation of the composite structure's strength could also be determined, and a modified form of equation (3.1) could be used to determine the probability failure for the damaged structure. Thus, a damage tolerance probability assessment could be conducted, based on any detected damage.

3.2 Ultrasonic Wave Theory

Ultrasonic wave theory applies methods derived from sonar to solid materials. Wave theory can be used to examine dynamic properties of an energy wave pulse traveling through a solid material from a known point of origin. The general form of the one-dimensional equation of a wave motion takes the following form:

$$\frac{\partial^2 p}{\partial x^2} - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0 \tag{3.2}$$

where p is deformation or disturbance, x is the position in a one-dimensional field, t is time, and c is the wave speed.

The disturbance for solid mechanics would relate to deformation of a material. The general solution to equation (3.3) is D'Lambert's wave solution:

$$p = p(x,t) = f(x-ct) + g(x+ct) \tag{3.3}$$

where f and g are general functions.

D'Lambert's wave solution has been proven to be the sum of functions f and g, which are both functions of position, x, and time, t, along with the speed of the wave, c. In this solution, it is assumed that none of the wave is transferred to the surrounding area, and the material is assumed to be homogenous and linearly elastic. These disturbances can be the model of strain waves in a material. Strain waves are either shear or compressive. Shear waves travel on the boundary surface of structural components, causing strains and thus stresses on the surface of the material. Compressive waves travel within a structural component. For thick structures, both waveforms are present, with compressive waves (Rayleigh waves) receiving greater attention in the structure. However, for thin-walled structures, shear waves (Lamb waves) dominate, due to the small thickness. These waves can travel greater distances in an object, using both surfaces of the material, which are relatively close to one another in thin structures. Thin-walled structures are the dominant configuration for most aerospace structures in order to conserve weight, so these were the focus of this research [18].

Since these waves frequencies are in the ultrasonic range, other complexities make wave analysis difficult. Any smooth continuous wave, such as a strain wave, can be broken down into components of shear and compressive waveforms, called phase modes. This is similar to the Fourier transform of waves. These phase modes consist of symmetric and asymmetric waves, which represent the compressive and shear waves, respectively, and are usually denoted as s_i for the symmetric mode i , and a_i for the asymmetric mode i. These are called symmetric and asymmetric phase modes, due to the general shape produced by the wave in a material. An example of some of the first few wave modes are illustrated in Figure 3.1. Each of these phase modes travels at a different velocity within a material. Plots, called dispersion curves, can be created for specific materials to map the speed of the phase modes as a function of their frequency and thickness (Figure 3.2) [18]. However, due to their complexity, only numerical methods can be used to obtain these plots. No analytical solution currently exists [18].

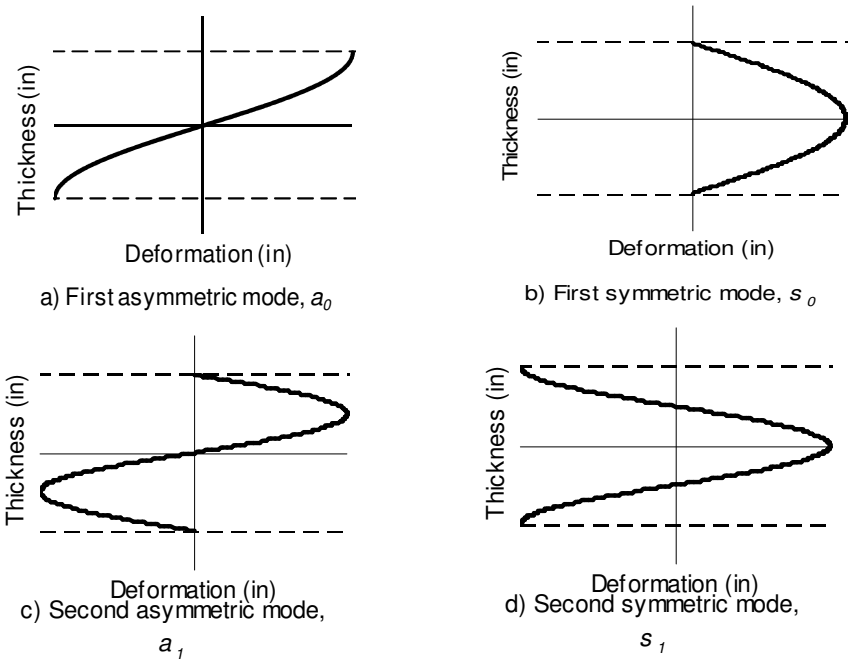


Figure 3.1 First- and second-phase mode shapes for strain waves in a structure.

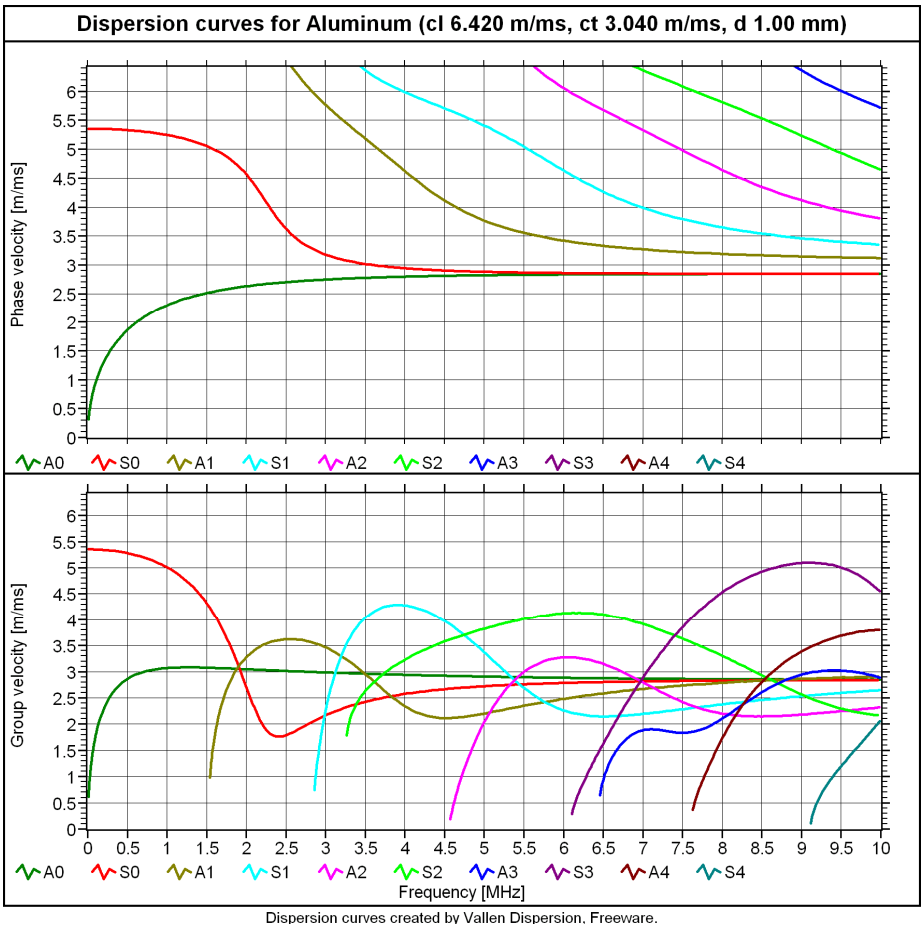


Figure 3.2 Example of dispersion curves for complex strain wave.

For general practice, a low-frequency range is used for scanning, since only one asymmetric and one symmetric wave are expected to be present. The experimentation presented in this dissertation consisted of using frequencies of 50 kHz, 100 kHz, 150 kHz, and 200 kHz, which fall in this range. These dispersion curves are material and structure specific.

Composite materials with unidirectional fibers can be modeled as anisotropic material. For composites with multiple plies, some deflection can travel between plies. By applying stress equilibrium and strain equalities, a method of deriving a wave equation in composite laminates can be obtained. The work of Reismann and Pawlik [59] contains information about where dispersion curves can be determined for composite materials.

The foundation of the ultrasonic wave theory is based on an infinite plate with no boundaries. Analyses of these waves produce multiple strain wave modes with different frequencies. These wave modes travel with different speeds, which can be determined using analytical methods. Using analytical methods, time of flight of the wave from its source to a sensor can be calculated.

All real plates have boundaries that will reflect a strain wave, causing a super-position of a virgin wave and reflected wave at a receiving sensor. Thus, the sensors of the experimental tests in this dissertation were placed at a significant distance from all boundaries, to decrease any influence on the received signals. Damage of the plate will cause an effect similar to that of an edge boundary. Due to the uncertainty of the boundary shape of the damage areas, a wave interacting with the boundary of a damage area can be scattered in all directions, decreasing the overall energy of the wave received by the sensor. Ultrasonic waves traveling through a solid material will disperse over a distance. The different wave modes travel at various speeds along with dispersing energy into the surrounding environment. Furthermore, the manufacturing process can produce voids and flaws in the finished composite structure. These additional factors are assumed to be negligible in any modeling of ultrasonic waves in materials. A realistic model of a strain wave traveling in a composite laminate becomes much more complicated to analyze. Therefore, for actual experimentation, another method of signal processing of strain waves was investigated.

3.3 Artificial Neural Networks

Artificial neural networks are numerical methods for learning solutions to complicated problems in a manner that emulates human brain functions. Due to the complexities of strain waves and the behavior of damage in composites, it was fitting that the measured strain wave data of the ultrasonic sensor system be analyzed using ANNs.

Created around the same time as serial computers, artificial neural networks are composed of hardware and algorithms to mimic the ability of an organic brain to analyze a set of inputs to obtain a desired output set. The human thought process was emulated mathematically using a network of connected nodes with adjustable weighted values on the connecting paths. Similar to a human brain, this network can be "taught" the relationship of inputs to outputs using example datasets. After a sufficient number of example datasets are presented, the neural network can be used to determine the output from a new input dataset that lies within the domain of the examples from training [19]. Artificial neural networks are also able to cope with "noisy" data. Many uncontrollable variables exist in most practical engineering problems. For composite laminates, many flaws and micro-voids could be produced as early as during manufacturing, and they could all grow simultaneously. This is a time- and processor-consuming endeavor for many other analytical procedures.

The algorithms for ANNs rely on massive parallel processing of simple mathematics to solve problems. This ability to analyze complex problems with reduced computation power and time is a particularly useful property for applying it to a structural health monitoring system where the need exists for quick assessment of strain wave signals generated by ultrasonic sensor systems. This could lead to an accurate, real-time assessment of damage to structural components while in service. In this dissertation, artificial neural networks were the method of choice to analyze the measured strain wave data of an active ultrasonic system to determine the location and severity of damage.

3.3.1 Neuron to Node

The basic principles and rules of an artificial neural network are based on the operations of an organic brain [19]. The nervous system of a human contains cells, called neurons. These biological neurons pass information in the form of electrical signals from one neuron to another. The electrical signals transmitted between cells are the information of thought, or intelligence. As illustrated in Figure 3.3, an electrical signal is received by a neuron through the dendrites, the red sections of the cell in this figure. A small gap, or synapse gap, between the connected neurons modifies the electrical flow. The received energy is then stored in the main body of the cell, or soma (purple section of the cell in this figure). After enough energy has been stored within the soma, it is released through the axon (green section of the cell in this figure) to be sent to another neuron. The connection patterns of the cells and the traits of the synapse gap are the key to intelligence. The configuration of connections between cells can change, depending on the nature of the information and the destination. Within a true neural network, many neurons may be

actively sending or receiving information simultaneously, resulting in vast parallel processing in the system. This provides quicker calculations, resulting in close to real-time evaluations of time-dependent input and selection of appropriate reactions to the input data.

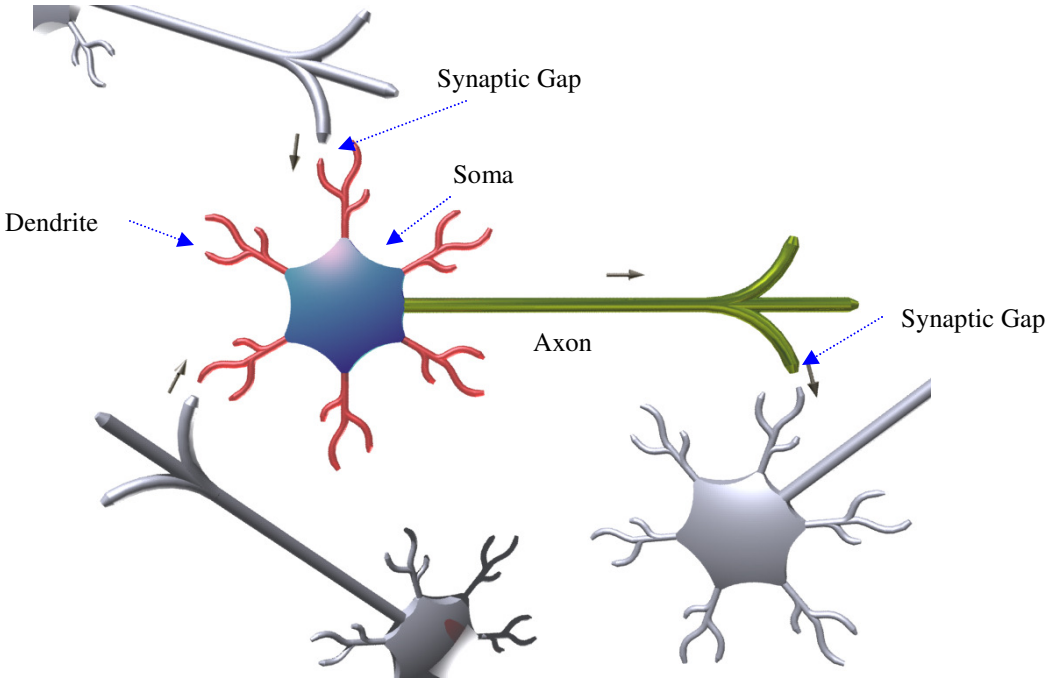


Figure 3.3 Biological neuron cell with three attached neurons (adapted from Fausett [19]).

The basic concept of a biological neuron cell can be translated into a mathematical model, as shown in Figure 3.4. Here the cell can be reduced to a simple circle, a node, with weighted inputs and a single output transmitted to the next node of the network. Inputs consist of numerical values from other nodes and are multiplicatively weighted in a manner similar to the synaptic gap. All inputs are summed within the node, and an activation function is implemented, similar to the process that occurs in the soma. The output is transmitted to the next connected node and is weighted before being received by the next node.

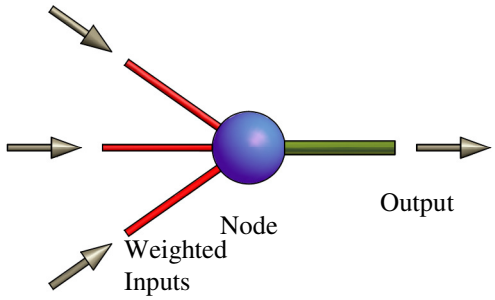


Figure 3.4 Single node for neural network model with similar coloring to Figure 3.3.

The conceptual model of the simple node can be translated into a mathematical form, using the notation in Figure 3.5. The added weights are indicated by w_{ij}, where the subscript represents the path from node i to node j. The inputs to the node, labeled as $X_1 \ldots X_m$, are values from a previous layer. These values are then multiplied by a weight and summed together in node Y_1 (see equation (3.4)). The output of node Y_1 is then the result of an activation function applied to the weighted sum, as shown in equation (3.5). This function is the activation function, since it "activates" the node to produce an output value.

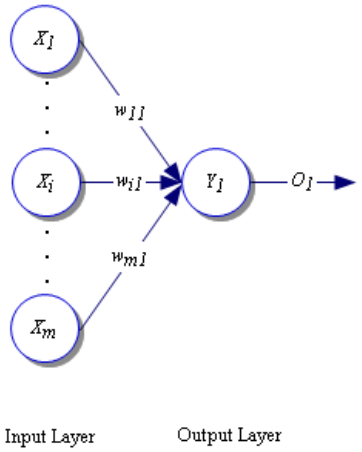


Figure 3.5 Single node network of m inputs, placed in nodes X_i, to obtain output O_1, through node Y_1.

$$y_{in} = w_{11}X_1 + \ldots + w_{i1}X_i + \ldots + w_{m1}X_m$$

$$y_{in} = \sum_{i=1}^{m} w_{i1}X_i \tag{3.4}$$

$$O_1 = f(y_{in}) \tag{3.5}$$

The network itself works best in a binary or bipolar setting, consisting of the inputs and outputs of all nodes operating in a similar manner as serial computers from 0 to 1 domain for binary (-1 to 1 for bipolar domain), where 0 means off and 1 means on. For computers, a definitive switch occurs between these two values, or hard threshold, where within its hardware, bits are stored as either 0 or 1 to represent all data. The difference in artificial neural networks lies in the parallel processing action as well as the function applied to y_{in} in equation (3.5). Unlike serial computers, the activation function in nodes can contain a soft transition between on and off. Four examples of the most common activation functions are shown below in Figure 3.6 and subsequent equations (3.6) to (3.9).

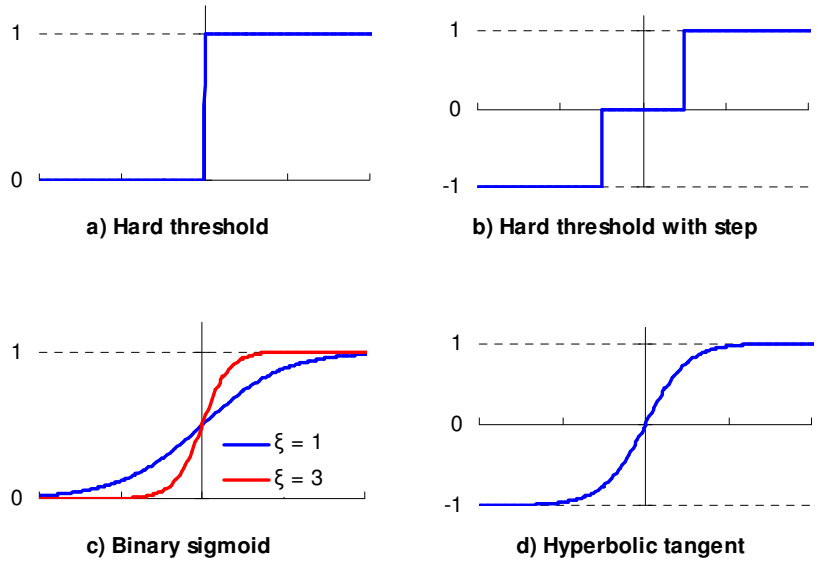


Figure 3.6 Common activation functions with each plot corresponding to: a) equation (3.6), b) equation (3.7), c) equation (3.8), d) equation (3.9)

$$f(x) = \begin{cases} 0 & x < 0 \\ 1 & x > 0 \end{cases} \tag{3.6}$$

$$f(x) = \begin{cases} -1 & x < -c \\ 0 & -c \le x \le c \\ 1 & x > c \end{cases} \tag{3.7}$$

$$f'(x) = \xi \cdot f(x) \cdot [1 - f(x)]$$
$$f(x) = \frac{1}{1 + e^{-\xi x}} \tag{3.8}$$

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$
$$f'(x) = [1 + f(x)] \cdot [1 - f(x)] \tag{3.9}$$

These activation functions produce output values between the solid boundaries of on or off. Consequently, output approximations between 0 and 1 are possible, thus removing the limitations of serial computing. The connections can lead to many different types of for network architecture, including feed-forward and recurrent networks with feed-back loops. A secondary approach to artificial neural networks includes a novel architecture, called a self-organizing map. This network is formed differently and is discussed later in this chapter. The research

in this dissertation employs primarily feed-forward networks and self-organizing maps and their applications for an SHM system.

ANN learning then comes in the form of supervised and unsupervised training. Similar to training the brain, supervised learning of the neural network is accomplished by providing the neural network with multiple examples of matching pairs of input and target data. During each iteration, the weighting values, w_{ij} in equation (3.4), are adjusted, and hence, to reduce the output error the relations between the datasets, are "learned" by the network. Once an input-to-output trend has been learned, the ANN can compute an approximation to the output from the input data that it has never encountered. If the input data is similar in value to any of the data used during the training exercise, the ANN will produce accurate outputs, when compared to the target output.

For unsupervised training, which is used in the self-organizing map architecture, an ANN is provided sets of inputs, and with no assistance, it will categorize the input sets. Thus, the learning of the network, much like a brain, is achieved by adjusting the weights between connections of nodes. Each of these ANN architectures has its advantages and disadvantages. For the SHM system created in this dissertation, both ANN types were utilized in different manners to maximize their performance.

3.3.2 Feed-Forward Network

The node-to-connections process, described in the previous section, is applicable to most ANN architectures. A network of nodes can have paths connecting any one node to another. Connections can be chosen to form a simplified network of nodes by allowing information to flow through the network in only one direction. Similar to control theory, this network, as illustrated in Figure 3.7, is called a feed-forward network. The network acts as a mapping from an input to a desired output. From the work of previous researchers, an iteration method was determined to adjust the weights within the network in order to learn an output set when an input set is provided. The notation shown in Figure 3.7 is used throughout the remainder of this paper [19].

The type of architecture of the networks illustrated here is feed-forward, as described above, where there are layers of nodes that affect the next layer of the hierarchy. No connections, or information, go backward through the network; therefore, it is not time dependent. The input variables form their own, first set (or layer) of nodes in the network. Then several sets of nodes, called hidden layers, are placed next. Finally, the last layer of nodes is the output set, which is the output of the entire network.

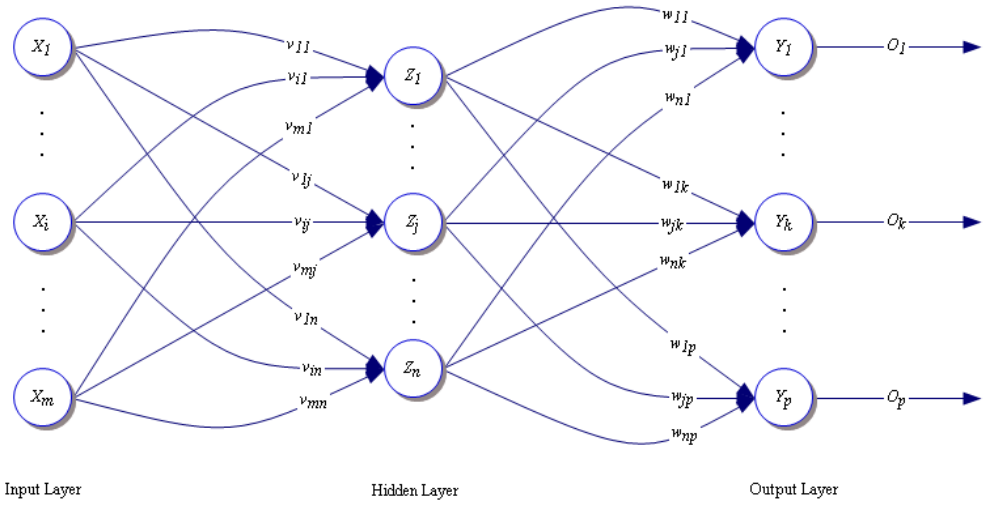


Figure 3.7 Generic feed-forward architecture for artificial neural network.

The training of this network involves supervised learning, which means using known input-to-output datasets to adjust the weights within the network. Adjustment of the weights comes about through a method developed by Rumelhart et al. [19], which involves taking the error between the desired outputs, t_k, and the output obtained by the network, O_k, and slightly adjusting the weights, which in Figure 3.7 are weights w_{jk}, using equation (3.10). This process, which is based on an optimization method of adjustment by way of steepest descent, uses a learning rate, α, to adjust the weights slowly. The error value for the output layer, shown in brackets in equation (3.10), then proceeds backwards through the network in a way similar to that described in equations (3.4) and (3.5) to determine the error values for the first node of a hidden layer. Once the errors values are calculated for this node, the weight adjustments can be obtained for other connections within the network. From equation (3.10) each weight value can be updated from w_{jk}^{old} to w_{jk}^{new}. The adjusted weight will result in less error between the target output and predicted output by the network. Through many iterations of the training dataset, a reasonable optimization of the weights within the neural network can be determined.

$$\Delta w_{jk} = \alpha \cdot Z_j \left[(t_k - O_k) \cdot f' \left(\sum_{j=1}^{n} w_{jk} \cdot Z_j \right) \right]$$
$$w_{jk}^{new} = w_{jk}^{old} + \Delta w_{jk} \qquad (3.10)$$

Common practice using neural networks follows a method in which the learning is conducted by repeatedly showing the neural network a training set until an RMS error, E_{RMS}, reaches a minimum value. Using q datasets within the training set, the error is found through equation (3.11).

$$E_{RMS} = \sqrt{\frac{1}{q}\sum_{i=1}^{q}\sum_{k=1}^{p}\left(t_k - O_k\right)_i^2} \tag{3.11}$$

After the entire training set is used in adjusting weights once, also called an epoch, an RMS error is found. The network is constrained to learn for a set number of epochs before ending the learning process. This number of epochs is chosen large enough to result in a set of weights that gives a small RMS error for the training sets.

Along with this method, there are more advanced techniques for learning, including better optimization methods. Among these, a common addition is a momentum factor, which helps to avoid local minimum errors, relative to the desired global minimum, E_{RMS}, value for the neural network. Along with this is an adjustment of the learning coefficient, α, after a set number of iterations or epochs. Again, this is used to avoid local minimum error values but also allows for increased precision on the global minimum error value.

3.3.3 Self-Organizing Maps

Another common architecture for artificial neural networks is a self-organizing map, which functions much differently than the feed-forward network described above. This network type has the ability to categorize datasets into clusters, using an unsupervised method that eliminates the need for a known output prior to learning. This type of network has nodes, which store group types within them by matching similar traits. An input set is compared to each node, and the node most similar will absorb that input set. The groups can then be used as identifying categories. A basic example using three nodes is illustrated in Figure 3.8. In this example the nodes are arranged in a single row. The following text in this section is from Fausett's work on artificial neural networks [19].

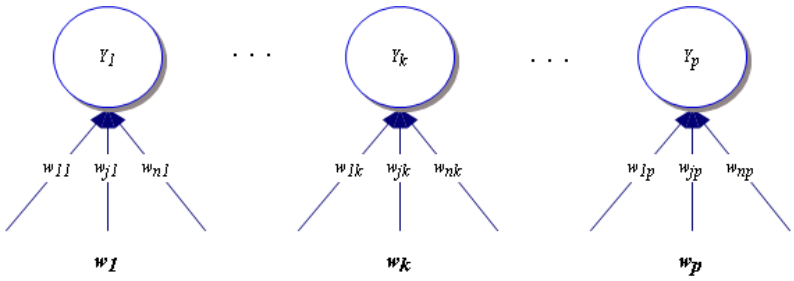


Figure 3.8 Generic self-organizing map architecture for an artificial neural network.

In order for this network to learn, it is first shown a new input set of data from a training set. The weight vectors belonging to each node $\left(w_{1k},\cdots,w_{nk}\right)$, called exemplar sets, are initialized to small positive random values for an untrained network. Then these are compared to the input vector. A general method of comparison between an exemplar, w_k, and an input, X_i, is to use the Euclidean distances between the two, expressed in equation (3.12).

$$ED_k = \sqrt{\sum_{j=1}^{p} \left(w_{jk} - X_j\right)^2} \tag{3.12}$$

The network then works competitively, whereby the node with the smallest distance will win the input. For example, let node k "win" the input set. Then, the exemplar w_k is adjusted in a direction to more closely resemble the new input set, using equation (3.13), where again a learning coefficient, α, is used.

$$w_{jk}^{new} = (1-\alpha) \cdot w_{jk}^{old} + \alpha \cdot X_j \tag{3.13}$$

Through iterations of all the input sets, this network will adjust itself and group the input sets into like categories without any prior knowledge of the datasets.

This concept can be expanded further, such that the exemplars of neighbors of the winning node are updated. Consequently, a winning node k will adjust its exemplar values, as well as those of neighboring nodes $k \pm 1$ for a small neighborhood, or larger surrounding nodes (i.e., nodes $k \pm 2$, $k \pm 3$, …). The concept of neighborhoods allows for better grouping of input sets, since similar sets, but with slight differences, will be placed in nodes close to one another but not necessarily in the exact same node. This neighborhood concept is improved upon by reducing the size of the neighborhood after a set number of iterations. This allows for more sharply defined categories.

A better SOM structure increases the number of nodes used and places them into a two-dimensional network field of rows and columns, called a Kohonen layer, named for the creator. The adjustment of weights through learning categories and the winning neighborhood concept apply here as well. However, the neighborhood shapes are two-dimensional. An example of a common square neighborhood is shown in Figure 3.9, where the neighborhood radius is three nodes away from the winning node. The exemplars of all of the nodes within the box are then adjusted.

Once completed, a two-dimensional map of groups is formed. This map can then be used with new datasets by placing them into groups on the map. Note that once trained on example datasets, the exemplars can be adjusted again or not, depending on the desired results. That is, the network can be adjusted further to redefine the group boundaries, using newer data presented to the network, if desired. Generally, the exemplars stay constant and are not adjusted further for any new datasets introduced to the network. The larger number of nodes used allows for a better defined map; however, a larger amount of computing time is required as well. So an optimum ratio of number of nodes to processing time is desired when using this form of neural network. This architecture is elegant in using simple equations for learning categorizing datasets, while remaining powerful enough to discover similarities not

easily noticeable in other grouping algorithms. The uses for this network architecture include categorizing detected strain waves.

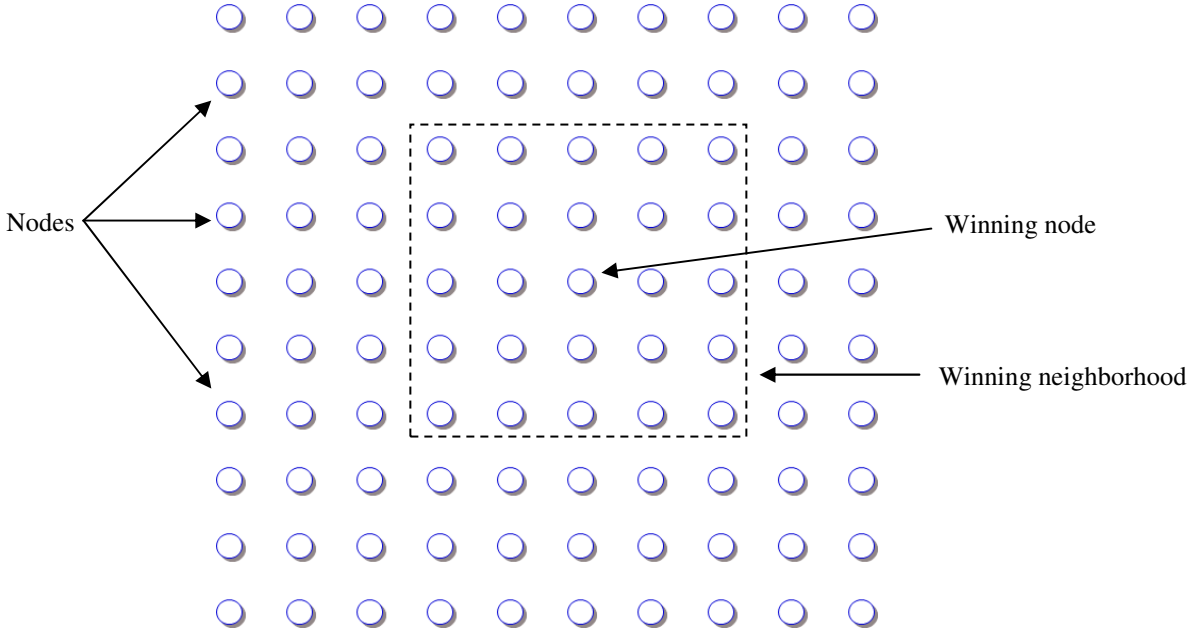


Figure 3.9 Kohonen layer in self-organizing map with square neighborhood of radius 2.

3.4 Multi-Agent System

Multi-agent systems are more commonly being developed for control systems. Many current control systems act through serial programming, usually with a predefined set of orders to follow. However, current systems are become more complex and difficult to account for all situations. An intelligent controller should be capable of making decisions itself and act accordingly, based on the situation. This intelligent controller is also known as an agent. Figure 3.10 illustrates the basic performance of an agent. It will perceive the surrounding environment, which consists of anything exterior to the agent itself, through sensors. Based on the results witnessed and a database of internal knowledge, appropriate actions will be undertaken to change the environment.

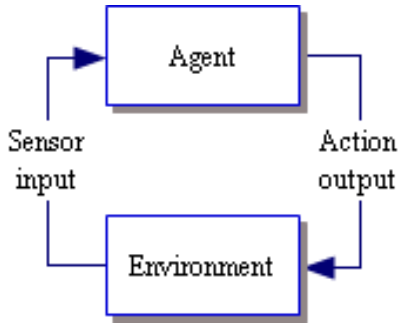


Figure 3.10 Single agent performing in an environment.

A simple example of a single agent system is a room thermostat. The system will record and monitor the current temperature of the room. If a certain threshold is crossed, then heat or air conditioning is applied to change the temperature back to the desired range. A major component of an agent is an internal knowledge base. The decisions made by the agent are based on this knowledge. In this example, the agent has a range of appropriate temperature levels to maintain. Although simple in this example, an agent can be more complex but still has a goal to obtain.

Suppose there is a domain space of two agents. Agent #1 would see Agent #2 as part of the environment. It could output information to connect with Agent #2, and collectively both agents could form a cooperative output response to outside environmental actions. This could be repeated, such that many agents could operate together to form a multi-agent system (MAS). Consider a domain space with several agents acting towards obtaining their individual goals but also communicating with one another to reach a shared objective. Each agent could be modified in such a manner that the inputs consist of sensor input from the environment and knowledge from neighboring agents. Agents could communicate with neighboring agents on basic knowledge in their individual datasets to gain increased performance. Some systems only communicate vital information to aid and teach neighboring agents, with the purpose of obtaining a global goal [47].

In electrical engineering and computer programming fields, a knowledge query and manipulation language (KQML) has been developed. This KQML has become a standardized format for communication between agents, which could operate in controlling the flow the information between agents. Within these messages between agents, a variety of information can be distributed. For cooperative multi-agent systems, information could include helpful information and compromise in tasks in order to share responsibilities and obtain a global objective. In competitive MASs, some information could be false or negative in order for one agent to benefit above others. This format is similar to the self-organizing map or competitive network in artificial neural networks. [47]

Whichever structure is used, an MAS should have an overall objective to accomplish. Each individual agent is working towards its own goal, using the aid of neighboring agents. However, the global view of the entire system usually has a larger objective. An example of this would be the Internet [60]. A multitude of information is sent around to different Internet protocol addresses through wiring and storage servers, while not obstructing the flow of other data packets. The servers or agents could work with neighboring servers to obtain local solutions to data flow. The global system would have all data flow be as fast as possible. Researchers have investigated the

interaction between players on a sports team as well [61]. Through collaboration among locations in a domain space, a team of robots can communicate with each other to accomplish the goal of scoring goals on an opposing team. These methods could be applied to other fields to develop autonomous systems of gathering information and responding accordingly to control their surrounding environment.

For this research, these multi-agent system concepts are applied in combination with artificial neural networks to form a committee or fuser network with better controlled sharing of knowledge. Individual ANNs can act as agents. By communicating with neighboring agents, the resulting system is predicted to be highly accurate in the signal processing of strain waves received by ultrasonic sensors on composite laminates. Furthermore, by limiting the communication between agents to a neighborhood, the system could become modular, allowing for a sensor network of infinite sensors to be used in detecting damage on an infinitely large structure. The various topics of damage detection and signal processing could be combined to form an autonomous structural health monitoring system for a complex aerospace structure.

CHAPTER 4

PRELIMINARY EXPERIMENTS

The application of artificial neural networks to post-process ultrasonic wave signals has been investigated in the past. For example, self-organizing maps were used to monitor vibrations on refinery machines. The SOM would train to normal operating conditions. If damage occurred, the vibration pattern would be altered, and the SOM would identify it as outside the training region [57]. In another example, the vibration of a steel support produced a specific operating frequency. By using deflections at specific locations on the structure, a feed-forward network was trained to be able to detect damage in the steel structure [58]. For this dissertation, some initial preliminary experiments were conducted to explore the feasibility of using these methods for composites. Early experiments consisted of the investigation of using artificial neural networks as signal post-processing for ultrasonic sensor systems on composite laminates. The experimental setup for this dissertation research was developed based on the results of these initial experiments. All of these experiments were performed on composite laminates with quasi-isotropic characteristics. The composite laminate materials, sensor equipment, and damage wafers remained constant throughout all preliminary experimentation as well as the research for this dissertation. More detail on these materials and equipment can be found in Chapter 5.

4.1 Half-Factorial Design

Initial experimentation consisted of examining the abilities of an ultrasonic testing system to detect damage on a composite laminate. The objective was to use statistical methods to determine if an artificial neural network system, rather than a statistical model, was necessary in damage detection. An experiment was conducted to examine seven different factors involved with active ultrasonic sensors to detect damage. Three piezoelectric disks were bonded onto the surface of a quasi-isotropic composite laminate. One location of damage was placed between the sensors in a specific location. Damage was simulated using a wafer, described in later experiments. A half-factorial setup was chosen to determine the significance of these factors, along with any significant interactions between them. Table 4.1 contains a description of each factor chosen for this experiment.

TABLE 4.1

SEVEN FACTORS OF ULTRASONIC TESTING UNDER INVESTIGATION

Factor	Range	Notes
A – Sensor Angle	0°–45°	Angles relative to fiber orientation of top ply.
B – Damage Horizontal	1–4 in	Measured along the direct actuator-to-sensor path, relative to actuator.
C – Damage Vertical	0–0.5 in	Measured as the normal away from the direct actuator-to-sensor path.
D – Damage Size	0.5–1 in	Diameter of simulated damage.
E – Signal Frequency	50–100 kHz	Frequency of strain wave released by actuator.
F – Signal Amplitude	20–50 dB	Recommended strain wave amplitude range by Acellent Technologies.
G – Signal Gain	20–40 dB	Recommend range by Acellent Technologies for their equipment.

Figure 4.1 illustrates some of the factors involved in these experiments. Here, "damage distance" is factor *B*, the damage horizontal, and "damage height" is factor *C*, the damage vertical. The two sensors used for this experiment were at 0° and 45°. The angle orientation was based on the fiber direction of the top ply.

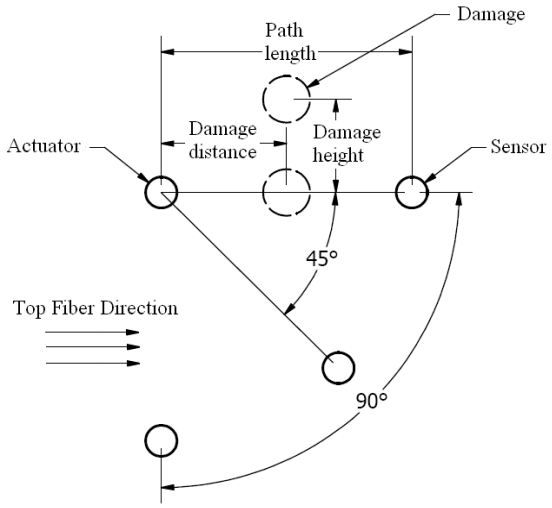


Figure 4.1 Ultrasonic actuator and sensor configuration on composite laminate.

4.1.1 Experimental Setup

For ultrasonic testing, the comparison method was utilized. This involved comparing a current state to a baseline state, as detailed in previous chapters. Only a single asymmetric strain wave was released by the actuator to remove wave-dispersion effects. The change in amplitude of a strain wave between the current and baseline signals was calculated and used in the analysis.

A half-factorial design of 2^{7-1}_{VII} was used as the design setup. Each of the seven factors was given two values, as listed in Table 4.1. In order to obtain all combinations of factors involved, 128 scans were required.

However, only half of the total factor combinations used in the experiment, or 64 combinations of these factors, were used, thus reducing experimentation time. The generator for the half-factorial design was $G = ABCDEF$, which produced the highest resolution for this factorial design, thus ensuring that even though half of the experiment was performed, the results were statistically accurate.

4.1.2 Results of Experiment

Table 4.2 displays the analysis of variance (ANOVA) model and results for the half-factorial design. The analysis involved used statistical means to determine if there was a link between factors and an output, which in this case was the change in amplitude. From this analysis, relationships between factors can be determined. The objective would be to create a system to determine the damage location, factors *B* and *C*, based on the remaining factors and change in amplitude.

However, many interactions were involved between the factors in this ultrasonic system. Based on the *p*-values in the table, many of the interactions were deemed significant. If the *p*-value was less than 5%, then the interaction between those factors was concluded to be a significant. There are multiple third-level interactions in this model. For example, the damage in the vertical location, C, the damage size, D, and the signal amplitude, F, have a third-level interaction, CDF, with a probability of < 0.0001, according to Table 4.2. This means that these three elements are dependent on each other. If one of these factors changes, then the other two factors will change as well. Table 4.2 provides those interactions and determines their significance levels. For this study, a 95% significance rate was used, so any p-value interaction below 0.05 was determined to be a significant interaction. The half-factorial design did not allow level-four or higher interactions to be determined. However, the number of interactions between three factors was large.

The model produced was significant and explains 98.95% of the scatter in the data. The ANOVA model, although complex, accurately describes the data. This singular model consisted of those interactions with low *p*-values, which for this study were factors and interactions with a p-value less than or equal to 0.05. The equation to model the system has the following generic form:

$$(Amplitude\ Change) = n_1 A + n_2 C + \ldots + n_{22} CFG + n_{23} DFG \tag{4.1}$$

where n_1 through n_{23} have a value based on the ANOVA model, and the capital letters are values for the different factors (see Table 4.1 previously). In practice for a damage detection system, factors *B*, *C*, and *D* would be unknowns. This model has one equation and three unknowns with significant relationships among them, so none

could be considered negligible and removed from the model. Due to the larger number of factors and interactions involved, other methods were pursued. Further experimentation investigated the effectiveness of artificial neural networks to model the system. An artificial neural network was chosen to use a "fuzzy logic" means to solve this complex model.

TABLE 4.2

ANALYSIS OF VARIANCE MODEL OF HALF-FACTORIAL DESIGN

Source	Sum of Squares	df	Mean Square	F Value	P-Value Prob > F
A – Sensor Angle	8480.62	1	8480.62	7.87	0.0084
B – Damage Horz	1044.06	1	1044.06	0.97	0.3320
C – Damage Vert	6.31E+05	1	6.31E+05	586.04	< 0.0001
D – Damage Size	1.83E+05	1	1.83E+05	169.87	< 0.0001
E – Sig Freq	22147.13	1	22147.13	20.56	< 0.0001
F – Sig Amplitude	1.08E+05	1	1.08E+05	100.12	< 0.0001
G – Sig Gain	86585.83	1	86585.83	80.38	< 0.0001
AB	2160.66	1	2160.66	2.01	0.1661
AC	1564.25	1	1564.25	1.45	0.2368
AD	1676.94	1	1676.94	1.56	0.2209
AE	11621.5	1	11621.5	10.79	0.0024
AG	44423.49	1	44423.49	41.24	< 0.0001
BC	56.14	1	56.14	0.052	0.8208
BD	1455.91	1	1455.91	1.35	0.2534
BE	2371.38	1	2371.38	2.2	0.1474
CD	1.62E+05	1	1.62E+05	150.43	< 0.0001
CE	39139.6	1	39139.6	36.33	< 0.0001
CF	80297.56	1	80297.56	74.54	< 0.0001
CG	41347.04	1	41347.04	38.38	< 0.0001
DF	31945.72	1	31945.72	29.65	< 0.0001
DG	41278.46	1	41278.46	38.32	< 0.0001
FG	3.71E+05	1	3.71E+05	344.61	< 0.0001
ABD	14845.27	1	14845.27	13.78	0.0008
ABE	82390.05	1	82390.05	76.48	< 0.0001
ACG	7855.36	1	7855.36	7.29	0.0108
BCD	6349.82	1	6349.82	5.89	0.0208
CDF	31281.63	1	31281.63	29.04	< 0.0001
CDG	32702.62	1	32702.62	30.36	< 0.0001
CFG	3.44E+05	1	3.44E+05	319.3	< 0.0001
DFG	75616.77	1	75616.77	70.19	< 0.0001
Error	35549.36	33	1077.25		
Total	2.50E+06	63			

4.1.3 Conclusions

A screening experiment was performed for active ultrasonic testing with seven major factors, all of which were shown to be significant. Additionally, there were two- and three-order interactions among them. Due to the design of the experiment, higher-order interactions were confounded. A complex ANOVA model can be created by

statistical methods. However, the difficulty in implementing such a system resulted in the investigation of artificial neural networks to model the system.

4.2 Expanded Factorial Design

The previous set of experiments concluded that a statistical model was difficult to develop for a damage detection system. The many factors resulted in a complex model with too many unknowns. The next investigation focused on modeling an ultrasonic testing system using artificial neural networks instead of statistical modeling.

4.2.1 Experimental Setup

A composite laminate similar to the one in the last set of experiments was used in this one. One piezoelectric actuator and three piezoelectric sensors were arranged, as shown previously in Figure 4.1. Sensors were placed at 0°, 45°, and 90° locations, relative to the actuator. The sampling rate of the piezoelectric sensors was 12 MS/sec for 8,000 sampling points. Additional signals received by the piezoelectric sensors were truncated in the analysis process. The change in amplitude between the current scan and the baseline scan were used as outputs to this system. Similar to the previous experiments, only one asymmetric mode of the strain wave was produced at this frequency range, removing any dispersion effects. This was chosen as a method to simplify the experiment.

The objective of this study was to examine six different factors used in damage detection with active ultrasonic sensors. These are listed in Table 4.3. Damage was simulated through wafers. Previous experiments had revealed that the damage horizontal distance did not have as many significant interactions with other factors. Therefore, for this series of experiments, the damage distance, as labeled in Figure 4.1, was held constant, with damage equidistant between actuator and sensor as a means to simplify the experiment. The last experiment suggested that there were higher-level interactions among factors, so a general factorial design was used in this experiment, which allowed for some of the factors to increase the number of values instead of a binary design.

TABLE 4.3

SIX ULTRASONIC TESTING FACTORS UNDER INVESTIGATION

Factor	Values	Notes
A – Damage Size	0.75, 1.25 in	Simulated damage.
B – Damage Height	0 in, 1 in	See Figure 4.1 for illustration
C – Sensor angle	0°, 45°, 90°	Angle relative to fiber orientation of top ply.
D – Signal Frequency	50 kHz, 100 kHz	Frequency of strain wave released by actuator.
E – Signal Amplitude	20 dB, 40 dB, 60 dB	Recommended strain wave amplitude range by Acellent Technologies
F – Signal Gain	20 dB, 40 dB	Recommend range by Acellent Technologies for their equipment

Experiments totaling 144 different configurations were performed. This series of experiments was capable of determining up to the sixth-level interaction among factors. Furthermore, for each physical damage location, a signal was collected from each of the three sensors for each frequency, amplitude, and gain listed in Table 4.3, resulting in 5,184 individual signals to scan for damage across various repeating places on the composite laminate. These were used to determine the capabilities of artificial neural networks.

4.2.2 Results of Experiment

A statistical model was again created to examine the significant interactions among factors. Table 4.4 shows the results of the ANOVA model. As in the previous experiment, the change in amplitude between current and baseline scans was used.

TABLE 4.4

ANALYSIS OF VARIANCE RESULTS FOR SIGNIFICANCE

Source	Sum of Squares	df	Mean Square	F Value	P-Value Prob > F
Model	767148	47	16322	11.387	< 0.0001
A – Damage Size	58923	1	58923	41.105	< 0.0001
B – Damage Height	28918	1	28918	20.173	< 0.0001
C – Sensor Angle	214176	2	107088	74.705	< 0.0001
D – Signal Frequency	12832	1	12832	8.952	0.0035
E – Signal Amplitude	1006	2	503	0.351	0.7049
F – Signal Gain	1213	1	1213	0.846	0.3600
AB	345	1	345	0.241	0.6248
AC	39497	2	19748	13.777	< 0.0001
BC	130282	2	65141	45.443	< 0.0001
BD	3025	1	3025	2.111	0.1496
BE	1778	2	889	0.620	0.5400
BF	2581	1	2581	1.800	0.1828
CD	18775	2	9387	6.549	0.0022
CE	5177	4	1294	0.903	0.4655
CF	20747	2	10373	7.237	0.0012
EF	23259	2	11629	8.113	0.0006
ABC	66541	2	33271	23.210	< 0.0001
BCD	26795	2	13398	9.346	0.0002
BCE	1670	4	418	0.291	0.8830
BCF	5565	2	2783	1.941	0.1491
BEF	6248	2	3124	2.179	0.1187
CEF	70025	4	17506	12.212	< 0.0001
BCEF	27769	4	6942	4.843	0.0013
Error	137613	96	1433		
Total	904761	143			

From this ANOVA table, it can be seen that there many higher-order interactions among factors are present. There is one fourth-level interaction and several third-level interactions, which are significant. These have a confidence in significance of 99.8%, which is above the aerospace standard of 95%. For a model of an actual ultrasonic damage scanning system, the amplitude change would be an input, along with factors *C*, *D*, *E*, and *F*, leaving damage size and height to be determined. Similar to before, the model is an undetermined system; therefore, artificial neural networks were investigated as an alternative method.

The amplitude change data can be modeled from the statistical data. However, an undetermined system is formed. As an alternative method, the data was introduced to an artificial neural network to investigate the capabilities of the ANN to "learn" the data. In other words ANNs were trained in the dataset of the 144 combinations to map the input facts, change in amplitude, and factors *C*, *D*, *E*, and *F*, to the output factors, damage size, and height.

Several artificial neural network architectures were investigated to determine which was best suited for this problem. The main focus was on a combination of a self-organizing map and feed-forward ANN. The architecture consisted of a SOM connected to a hidden layer of nodes and then to an output. This allowed for a desired output to be trained by supervised learning, while the SOM was trained in an unsupervised learning style. In this dissertation, this is referred to as an SOM feed-forward hybrid network. Since signal changes are a function of the other factors, it was assumed that a pattern would be present in the data to link the change in amplitude to damage size and position. The ultrasonic sensor system data was then used in two different ways. The first method focused on a local coordinate set, local to each sensor. These were created using NeuralWorks II software [63]. For this ANN dataset, the input set was comprised of the fiber angle (factor C), amplitude change, and signal properties (factors D, E, and F) for an output of damage size and position relative to the actuator-to-sensor path. The ANNs with this configuration were not capable of predicting damage sizing or location. The local coordinate set could not be recognized by a self-organizing map. There was no recognizable pattern that the network could determine.

The second method of investigation was on a global scale. The amplitude changes for the three actuator-sensor pairings were used together to determine the damage size and position. Initially Cartesian coordinates were used to describe the position output with the origin at the actuator. However, this was altered to be in polar coordinates. The origin was set to the actuator location with the angle value aligned to the top ply of the laminate, similar to how the sensor angle was defined. The most promising architecture for this configuration was self-

organizing maps. The global network consisted of using the amplitude changes for all the different signal combinations as an input set. An ANN that performed well was an SOM with a Kohonen layer of 20 x 20 nodes and 5 output nodes for back propagation. Figure 4.2 shows the results of using the global scale. Here the values predicted values by the ANN are close to the actual values. The root mean squared (RMS) error for this particular network once trained was 0.08 with a correlation of 89%. These results demonstrate that artificial neural networks and specifically self-organizing maps are capable of learning damage size. However, the scope and range of the factors in this experimentation were limited. Further testing was required.

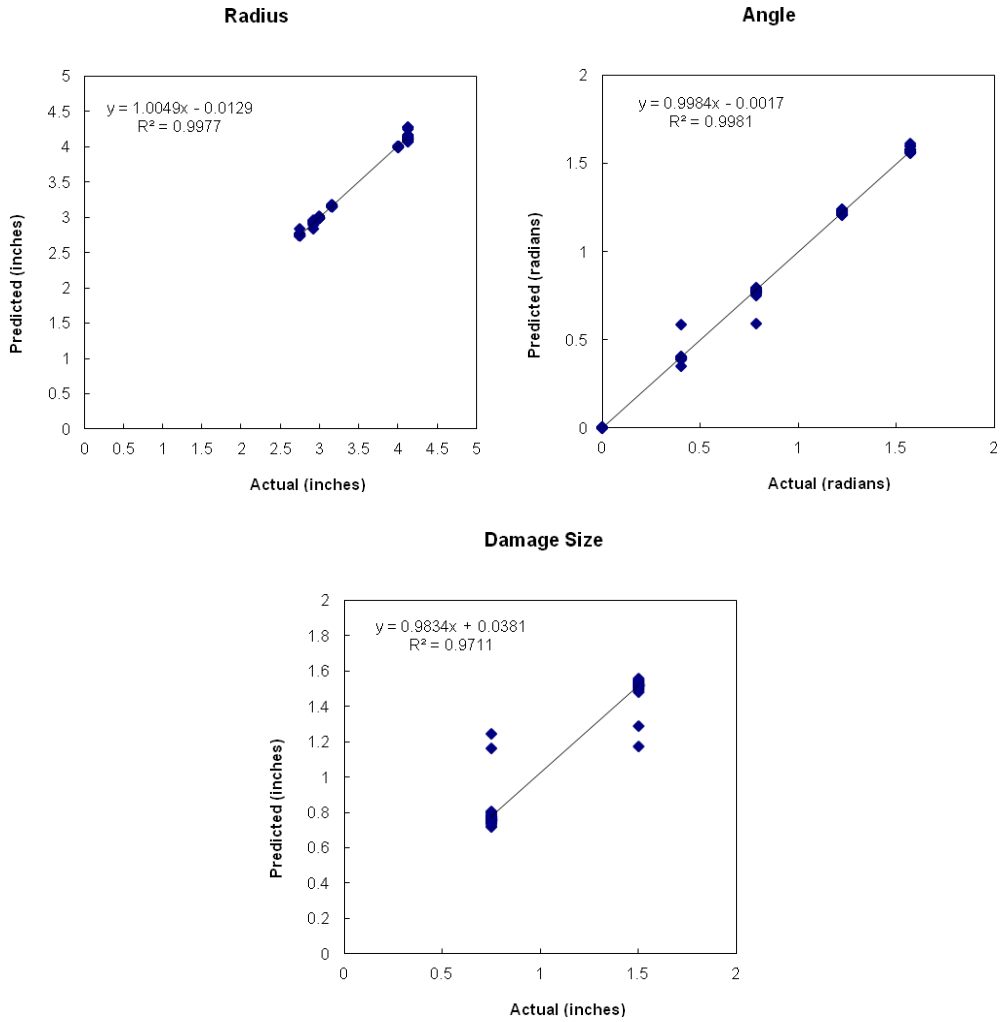


Figure 4.2 Output predicted values as functions of the actual values.

4.2.3 Conclusions

It was concluded from the initial experimentation that a statistical model is effective in an ultrasonic testing system. However, many significant interactions among factors exist, resulting in the model being complex. Therefore, artificial neural networks were introduced as an alternative model. This model consisted of a network of

actuator-sensor pairs with various signal-generating parameters. An SOM feed-forward hybrid network was demonstrated as being effective in detecting and locating damage. The damage location was modeled in polar coordinates for the ANN to train well. ANN models could be used to interpret ultrasonic sensor signals for damage assessment; however, the domain used in these experiments was determined to be too small.

4.3 Extended Global Network

The next experiment consisted of expanding the ranges of damage positioning beyond those used in the previous experiment. A sensor setup similar to the previous experiment was used here. Again, only one PZT disk was used as an actuator, and the remaining disks were sensors. Damage positioning was measured in polar coordinates with the actuator at the origin. This setup is illustrated in Figure 4.3. The angle used is based on angle deviation from the surface ply direction. As shown, the ply direction is parallel to the actuator-sensor #1 linear path. This experiment focused on creating a training set to investigate the capabilities of artificial neural networks.

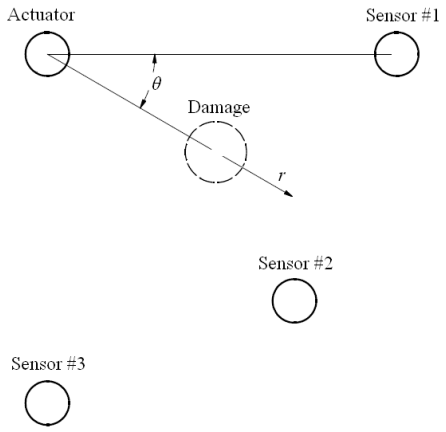


Figure 4.3 Actuator and sensor configuration with sign convention with horizontal surface ply fiber orientation.

4.3.1 Experimental Setup

For these series of scans, two responses from each sensor signal were used for the input series. Similar to previous experimentation, strain waves with a frequency range of 50 kHz to 100 kHz were generated by the actuator. For this experiment, the gain was held constant at 40 dB, and the amplitude was held constant at 60 dB. Since the goal was to increase the quantity of damage locations, the number of combinations of factors was reduced to simplify the experiment, so these factors were held constant. To further simplify the experiment, damage simulated with wafers, as in the previous experiments, had a constant diameter of 0.75 in. throughout these experiments. The measured responses from the sensor signals were the amplitude and time-of-flight differences

between current and baseline scans. Since many factors were held constant, the time-of-flight change between baseline and current signals was measured as well. This experimentation was to determine if more information, beyond amplitude change, was necessary from signals. A series of 90 experiments, including replicates, were performed at varying radii and angles in a randomized order, as shown in Figure 4.4. For this experiment, the sensors were placed 4 in. away from the actuator at 0°, 45°, or 90° degrees relative to the fiber direction (see Figure 4.3). Simulated damage was placed at a combination of three radii locations (1 in., 2 in., 3 in.) and five angles (0°, 22.5°, 45°, 67.5°, 90°) in polar coordinates. The positions of damage relative to the ultrasonic sensors is illustrated in Figure 4.4 The three received signals were used as an input set to various artificial neural networks to predict damage position.

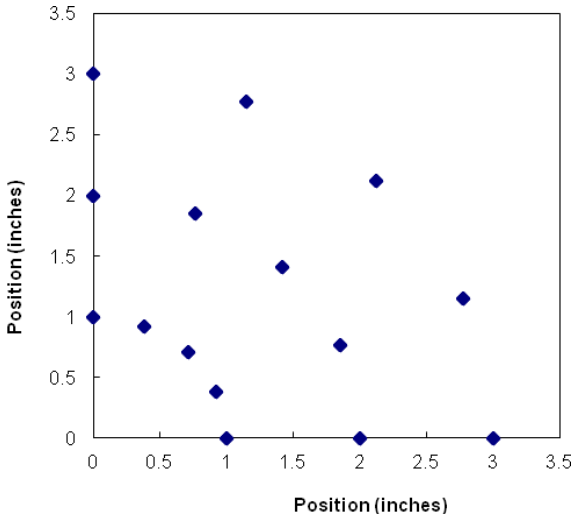


Figure 4.4 Damage testing position on composite laminate (all units in inches).

4.3.2 Artificial Neural Networks with Ultrasonic Signals

Various artificial neural networks were investigated for damage-positioning analysis. First, a feed-forward network with back-propagation was studied. An initial architecture of three hidden layers of 10 nodes each was chosen. The neural network was created in NeuralWorks II [63], with a hyperbolic tangent function as the activation function. The results are presented in Figure 4.5, where the predicted and actual values of the polar coordinates are compared. The prediction of radius had a correlation of 0.94, and the polar angle had a correlation of 0.98. There was some spread present, but the system trained well. The angle of the damage was trained with higher confidence, compared to the distance away from the actuator, the radius. Paths with angles of 0° and 90° were more accurate,

suggesting a special case at these angles. These results validated the conclusions of previous experimentations, since damage on the linear actuator-sensor path tended to be more accurate than damage away from the path.

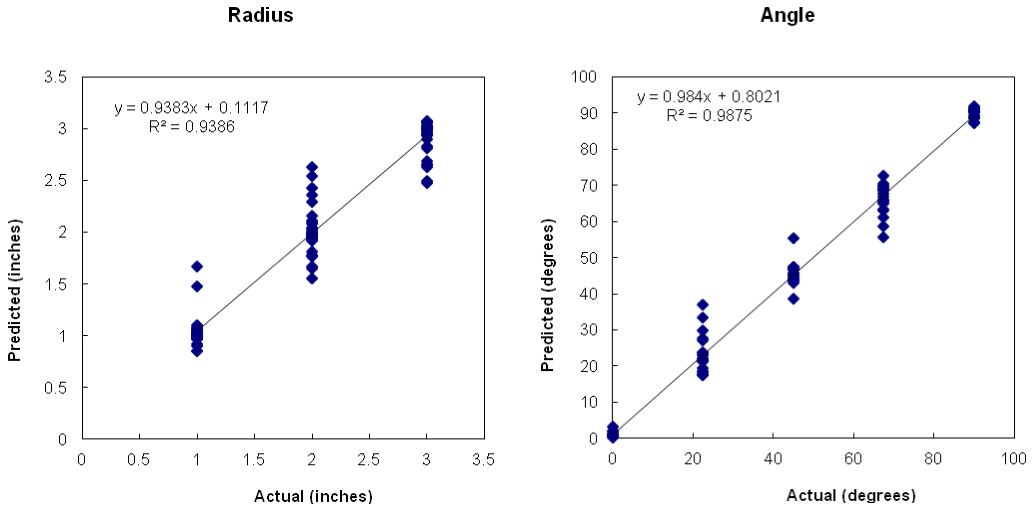


Figure 4.5 Results from feed-forward artificial neural network (all units in inches).

The next architecture examined was an SOM feed-forward hybrid, similar in architecture to the previous experiments. These networks were also constructed in NeuralWorks II with a Kohonen layer of 10 x 10 nodes. The maps generated by the training process of the networks are shown in Figure 4.6. The SOM had two back-propagation outputs. Each plot shows the points in the map, colored based on either the radius distance or the angle position of the damage.

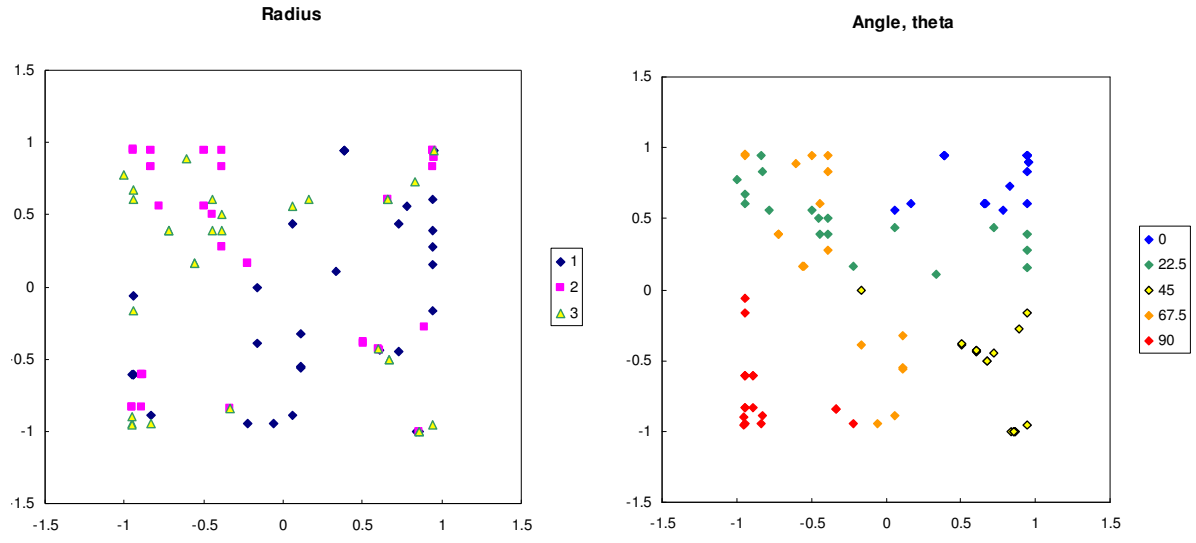


Figure 4.6 Results of self-organizing map using NeuralWorks software.

Results from a radius prediction are difficult to determine if a pattern is present. However, the angle has a distinct grouping. The angles where damage was at 0°, 45°, or 90° were all distinctly separated into groups. However, the angles between actuator-sensor paths were more mixed together. The results of angle position being more accurate compared to distance from the actuator mirrored the results of the back-propagation neural network.

Different self-organizing map software, Viscovery, became available in the middle of experimentation and was implemented [64]. This software created a Kohonen layer of 1,000 nodes and trained on the created dataset. The maps developed by this software are shown in Figure 4.7. The radius (labeled as Attribute 19), was difficult to determine, while the positioning angle (Attribute 20) had three distinct groups associated with 0°, 45°, and 90°. Damage from positions with 22.5° and 67.5° angle appear to be more mixed together in these results. This trend matches the trend formed by NeuralWorks II software (see Figure 4.6).

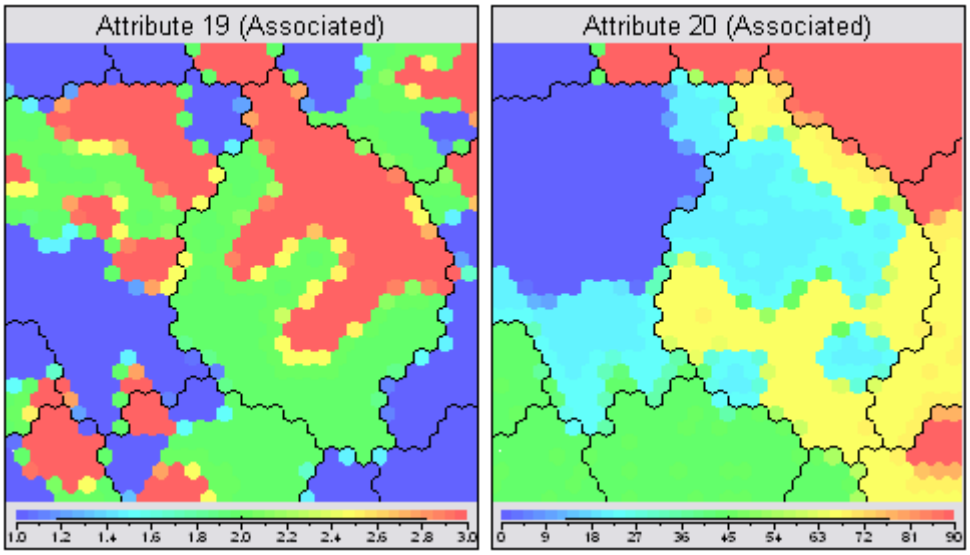


Figure 4.7 Results of self-organizing map with Viscovery software, where Attribute 19 is the distance from the actuator, or radius, and Attribute 20 is the angle from fiber direction**.**

4.3.3 Conclusions

This experiment investigated the use of various artificial neural network architectures to predict the damage location within an area around a PZT actuator. Three sensor signals, received from the initial actuator strain wave, were used together as inputs for ANNs to be mapped to damage location in the polar coordinate form. The different architectures could train to predict the polar angle location of a damage presence but could not estimate the radius position as accurately. These initial experiments validated the supposition that artificial neural networks of multiple architectures could be trained to locate damage, based on strain wave signals of ultrasonic frequencies in the range

of 50 kHz to 100 kHz. The most promising architecture was determined to be the hybrid combination of a self-organizing map and feed-forward ANN.

Experimentation for the research in this dissertation was a continuation of the three experiments described previously. The objective was to develop a system to identify and locate damage using the signal response of three sensors and one actuator. Instead of limiting each piezoelectric disk in the system to one function, four piezoelectric disks each acted as both actuator and sensor. Thus, the ultrasonic testing sensors formed a pitch-catch method of scanning. The results of each actuator-to-three-sensor system are expected to coincide with one another. The results were combined to develop a four-sensor system with higher accuracy and better damage size and location predictions than those presented in these initial experiments.

4.4 Comparison Between Actual Damage and Simulated Damage

The previous experimentation was performed with simulated damage consisting of small wafers, which were temporarily bonded onto the surface of a structure. The structures for experimentation were composed of composite laminates with carbon fibers and epoxy matrices. Piezoelectric disks, acquired through Acellent Technologies, were bonded onto one surface of a composite laminate with an epoxy, similar to the bonding agent used for applying strain gages to a structure surface. Simulated damage was temporarily bonded onto the same surface of the material for experimentation. The composite laminate was supported as a simply supported plate. The testing range on the plate was within the center region, around 4 in. away from any corner support. All scanning performed examined a single wave component, first asymmetric wave, with all reflected waves excluded for post-processing. The effect was a free-surface boundary condition in the testing region of the plate. See Figure 4.8 for an example of this setup.

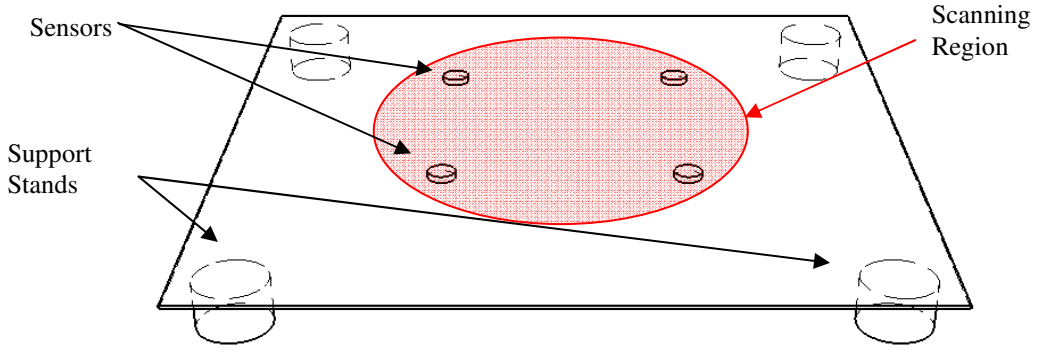


Figure 4.8 Generic configuration of simply supported composite laminate with ultrasonic sensors (scanning range has boundary-free surfaces).

4.4.1 Overview

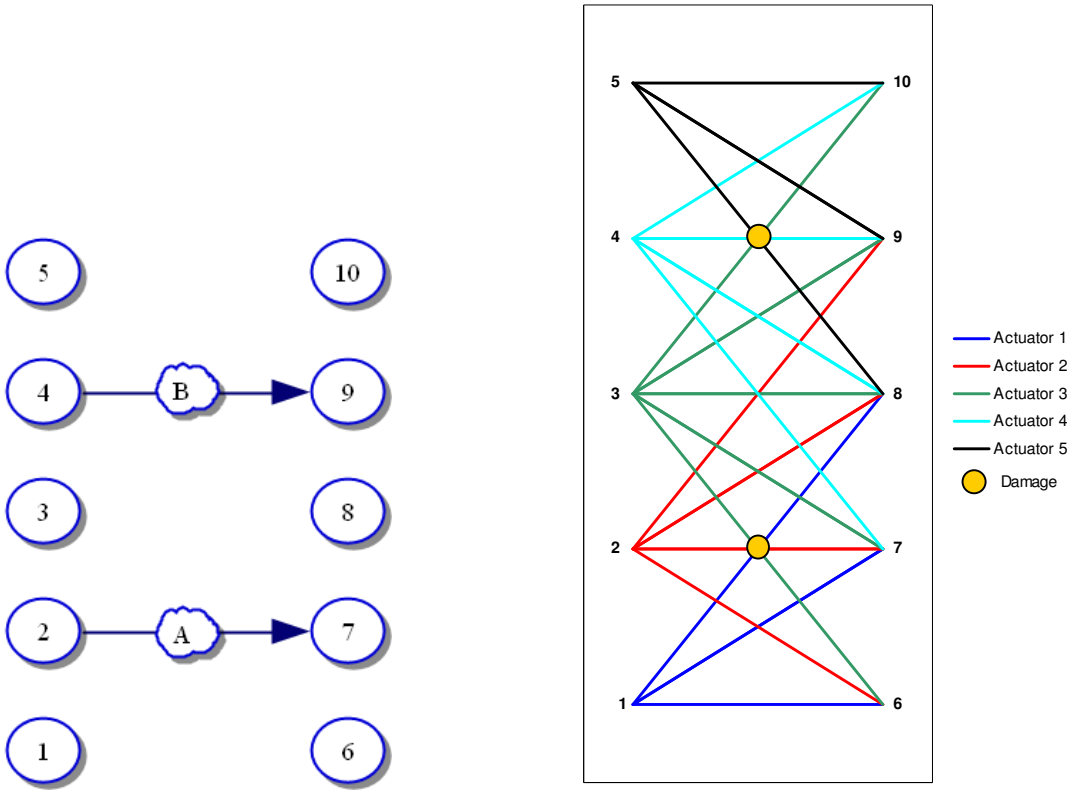
The wafers placed on the surface affected those free-surface boundary conditions. The wave signal was altered as a result of the placed wafer. In previous experimentations, an assumption that the wafer would scatter a traveling strain wave in a similar manner to that of actual physical damage was made. This method has been validated by equipment suppliers to demonstrate the effectiveness of their products, including Acellent Technologies. These wafers have an adhesive strong enough to acoustically bond to the surface of a structure but can be removed to return the structure to its initial state. For ultrasonic testing applications, an elastic wave traveling through a material will become disturbed by the bonded pad. Some of the energy of the elastic wave will be transferred to the wafer material. Since the wafers have high porosity, much of the transferred wave will be dissipated and removed from the structure. The resulting signal received from an elastic wave that has traveled past a wafer pad will be reduced in amplitude. This phenomenon is similar to the effects of a damaged location on an elastic wave. The objective of this last set of preliminary experiments was to determine the accuracy of damage wafers as a simulator of actual damage in a composite structure.

Actual damage could be applied during testing, but this would result in too many complexities for experimentation. Composite coupons with bonded sensors would have to be replicated to compare signal results from one damage position to another. This would include eliminating differences in fiber fraction volume, fiber place, and laminate manufacturing voids for each coupon. The sensors would be required to be bonded into the exact same location on each coupon. The cost to produce all of these coupons would increase drastically. However, a wafer can be removed by hand, thus returning a structure to its original state before the piece was attached. This is an advantageous characteristic for a system such as an artificial neural network for detecting the first onset of damage in a material. The damage-wafer system allows for an unvaried baseline signal to be used to determine the presence of damage in any desired location on a structure. Artificial neural networks rely on a domain of situations to train or learn how to handle any configuration, as long as it is within the domain. For damage detection, this domain would consist of damage initiation at any location on the surface of a thin-walled structure.

4.4.2 Experimentation

This experiment consisted of a set of ten piezoelectric sensors, arranged into two rows of five sensors each. The sensors were bonded onto a composite laminate having the same properties as described later on in Chapter 5. Simulated damage was placed in two different locations on the panel. This configuration is illustrated in Figure 4.9.

Simulated damage was placed in the direct paths between sensors 2 to 7 and sensors 4 to 9. Piezoelectric sensors 1 to 5 were used as actuators to send signals to piezoelectric sensors 6 to 10. Path definitions, shown in Figure 4.9b, are colored to the actuator of the signal. Not all possible paths were scanned for this experiment. It was determined to be sufficient to gather enough data from the paths presented in this figure. The frequencies used throughout the experiments were 50 kHz, 100 kHz, 150 kHz, and 200 kHz. This range was used in previous experiments and was recommended by Acellent Technologies for this configuration on a composite laminate [42]. In random order, configurations of damage sizes of 0, 0.19 in., 0.5 in., 0.75 in., and 1.5 in. were placed at each location to simulate damage initiation. The baseline for this was an undamaged laminate.



(a) Sensor and damage notation (b) Colored paths from each actuator 1–5 to sensors 6–10

Figure 4.9 Sensor and damage definitions for experimentation.

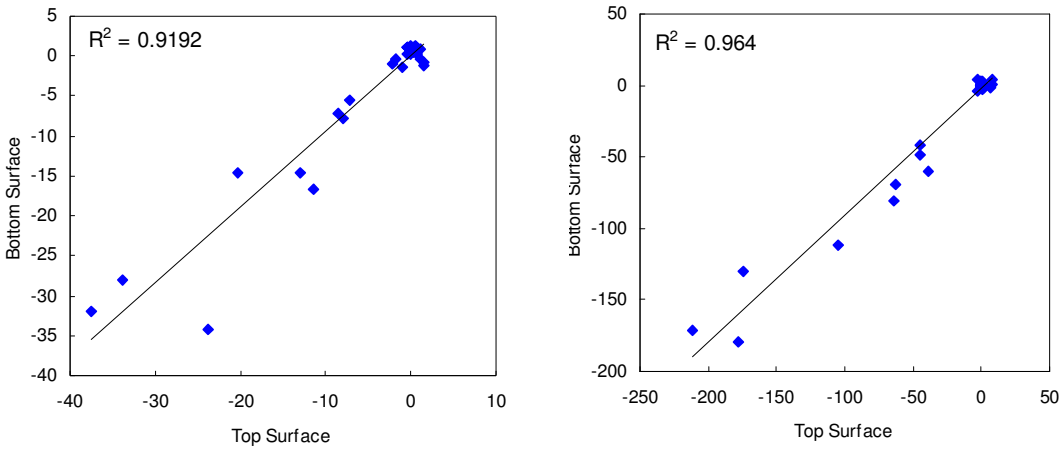
This experiment was then repeated for the backside of the laminate. In theory, the elastic waves were expected to travel between the two surfaces of a thin-walled material. If damage was present on either surface, then some of an elastic wave's energy, traveling in the vicinity of the damage, would be absorbed and affect the wave. That theory was tested in this series of experiments. The final set of experiments involved scanning with actual damage. Physical damage holes were drilled into the laminate. One location had a through hole of various diameters,

while the second location had a partial through hole. Both represented damage that could be present on aerospace structures. The composite panels were already made, so a delamination situation was not possible to be created with this experimentation set. The two damage locations were those points where simulated damage was placed. A correlation between signal changes between simulated and actual damage was drawn.

Through post-processing in MATLAB, undamaged, baseline state, and current state were compared. The four main traits of signal wave, amplitude, time of flight, duration, and rise time were compared between the baseline and current states. The signals produced from wafers placed on the same surface as the sensors were used as a comparison to the other conditions.

4.4.3 Results of Laminate Backside

Figure 4.10 shows the results from a sample in this experiment. The signal changes from the result of wafers placed on the backside of the laminate were compared to those for a wafer placed on the main surface of the composite. In this context, the "main" surface is the one containing the ultrasonic sensors. The signal comparison of the amplitude change for signals with damage placed in the direct path is displayed in Figure 4.10. It was assumed that the simulated damage would not be affected by the through thickness.



a) Change in Amplitude for 50 kHz
Simulated Damage between Sensor: 4 --> 9

b) Change in Amplitude for 100 kHz
Simulated Damage between Sensor: 4 --> 9

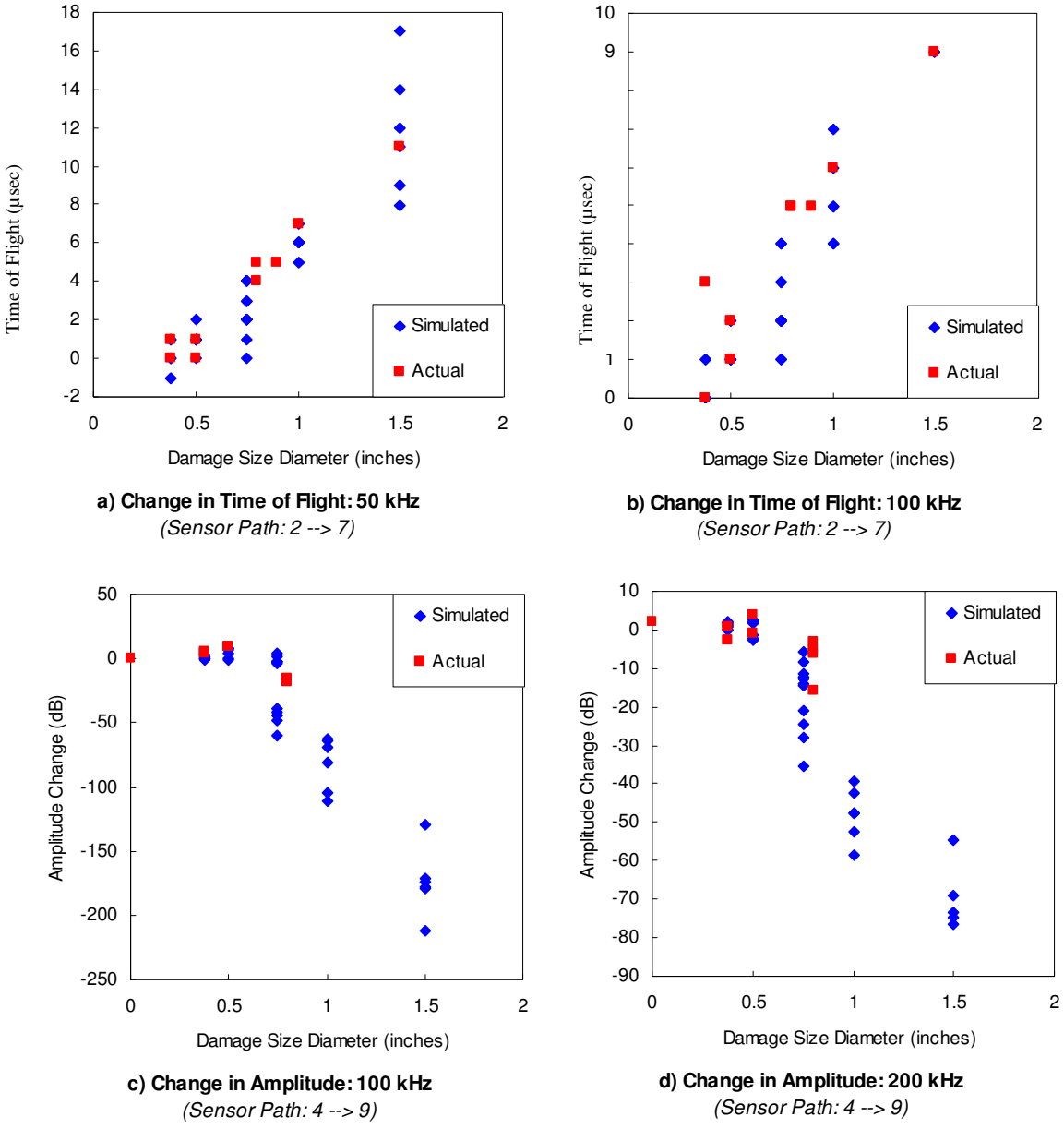
Figure 4.10 Comparison of signal changes due to simulated damage for either surface of composite laminate.

This data is a comparison of amplitude changes between baseline and current signals for various damage sizes. A comparison of all actuator-sensor path combinations was conducted, but only a sample of the results is shown. The results of all actuator-sensor paths confirm this assumption, as the correlation between the signals between both surfaces is close to 1. Least-squares regression lines placed on the data showed a linear correlation

between the resulting signals from the top and bottom surfaces. The lines are close to a 1:1 ratio between signals as the result of damage from either surface of the material. This verifies that simulated damage placed on one surface will have the same change in signal to that of simulated damage placed on the other surface. As long as the simulated damage accurately models physical damage, any internal damage is assumed to exhibit the same signal-changing characteristics as those on the surface of a structure.

4.4.4 Results of Actual Damage

The signal changes from the presence of actual damage to simulated damage are presented in Figure 4.11.



a) Change in Time of Flight: 50 kHz
(Sensor Path: 2 --> 7)

b) Change in Time of Flight: 100 kHz
(Sensor Path: 2 --> 7)

c) Change in Amplitude: 100 kHz
(Sensor Path: 4 --> 9)

d) Change in Amplitude: 200 kHz
(Sensor Path: 4 --> 9)

Figure 4.11 Comparison of actual to simulated damage of ultrasonic signal changes on composite laminate.

Different diameter holes were drilled to half thickness and then through thickness into the surface of the composite laminate, with scans between. The hole diameters used for actual damage were 0 in., 0.078 in., 0.125 in., 0.188 in., 0.4 in., and 0.5 in. Current detection methods are rated to detect damage between 0.1 in. and 0.5 in. for ultrasonic inspections with C-scans in the aerospace industry. For this experimentation, a damage range of 0 in. to .75 in. was used to encompass current inspection abilities. The two property traits of time of flight and amplitude change are presented with an adjusted scaling of the actual data. The duration and rise-time properties were also investigated. However, these traits were determined to not have a significant change in signals between damaged and undamaged states for both simulated and actual damage scenarios. Therefore, only the former properties are to be used for signal processing of ultrasonic signals. The difference between a through-thickness damage hole and a partial-through damage hole was found to be negligible for damage holes with a diameter less than or equal to 0.188 in. Therefore, the following results apply to both damage locations.

The amplitude decreases as some of the wave energy is scattered by damage. The time-of-flight is increased. When physical damage is present, some material from the surface would be gone. This results in a decreased thickness between the two surfaces. If this occurs, then the strain seems to increase its speed through the material. Damage then alters the time of flight of the wave. The simulated damage was discovered to have a similar effect on strain waves. There is a correlation between the simulated damage and actual damage for alterations to an ultrasonic signal. As shown previously in Figure 4.11, the damage diameters of the actual damage are adjusted to match the trending curve of the simulated damage. This method was optimized by visual inspection of the plots to obtain a general fit of the data. The resulting trend is a linear relationship between simulated damage wafers and actual damage, presented as

$$\left(Simulated\ Damage\ Diameter\right) = 1.502 \times \left(Acutal\ Damage\ Diameter\right) + 0.344 \tag{4.2}$$

These equations show that the effect of simulated damage is roughly 1.5 times greater than actual damage. Previous experiments have been conducted with simulated damage and have related results in simulated damage diameter sizes. However, the true size of actual physical damage detectable is proven to be smaller. Therefore, the detectable range of damage using ultrasonic testing with simulated damage wafers is smaller than the wafer diameters used.

4.4.5 Conclusion

Through this series of experiments, previous experimentation was verified. Damage positioning in the thickness direction was determined to have no effect for the ultrasonic testing configuration. Since strain waves travel through a thin-walled structure, relying on the surfaces to remain in the material, any damage at any thickness of the composite is assumed to be the same. Further investigation of a damage item with other NDE methods would be required to locate the through-thickness direction of the damage. The simulated damage was concluded to be adequate for the experiments conducted. Further research presented in the following chapters also relates results to simulated damage sizing. Equation (4.2) is an adjustment factor that can be applied to relate simulated damage results to actual damage results. Since this research focused on the proof of concept, the simulated damage diameter was determined to be sufficient. Only an additional post-processing step is required to relate the simulated damage severity to actual damage severity.

CHAPTER 5

EXPERIMENTAL SETUP

In Chapter 4, ultrasonic testing signals were determined to be too complex for statistical modeling. Therefore, artificial neural networks were implemented. Some damage severity was determined, but the focus shifted to damage positioning in a polar coordinate system, using one piezoelectric disk as an actuator and three disks as sensors. The main focus of this dissertation involved using piezoelectric disks as a means of detecting damage in a selected area on a composite laminate using artificial neural networks as a signal processing method. This chapter contains the configuration of equipment that would be used and the initial setup to create an acceptable ANN input-to-output dataset. Chapter 6 details the results of these experiments.

5.1 Testing Coupon and Equipment

Experimentation consisted of three items: a composite laminate, an ultrasonic testing sensor system, and a process to simulate damage on the composite. These descriptions are valid as well for all previous experiments performed in Chapter 4.

5.1.1 Laminate Composite Material

This research experiment used a laminate composite configuration with similar layups to those currently used in aerospace structures. Composite laminates often are constructed with woven fabrics or quasi-isotropic states in order to account for unknown variables in operation. Although these materials are replacing metallic components in a structure, not enough knowledge is available to allow full confidence of safety or performance in full composite structures. Thus, many components are made with 0°, 45°, and 90° ply configurations, with many plies to increase thickness and increase human assurance.

The research in this dissertation employed a composite laminate structure closely resembling a part in service. The testing coupon consisted of a quasi-isotropic layup of 32 plies with a layup of $\left[\pm 45/0/90/0/\mp 45/0/90/0/\pm 45/0/90/\mp 45\right]_s$. The composite remained symmetric and balanced, since this is common in aerospace structures to prevent any unwanted warping or asymmetric deflection effects throughout the material. The part was created with prepreg laminates with a closed tooling, or two-mold, system, resulting in a smooth surface for both surfaces. Composites are often manufactured with an open-mold setup, leaving a rough surface on one side. However, for this research, a two-mold system was utilized to remove additional unknowns from the experimentation. This testing coupon had a fiber volume fraction of around 0.7 to 0.75, which is the

common value for most composite laminates in any structure. The size of the laminate used in actual experimentation was a 14.5-in. square. The sensor placement allowed for space between the scanning region and the edge boundaries of the material. Reflections of the strain wave were minimized by using this setup. Post-processing for removing reflections in signals obtained was performed before analysis in order to reproduce an infinite plate setup. The laminate was suspended by simple support at four corners away from any scanning region. This simulated an infinite panel with free-edge boundaries on both sides, which reduced environmental effects on the laminate. Figure 5.1 below is an illustration of the setup of the composite, including the placement of sensors and damage scanning area. The testing coupons for all experiments were provided by the National Institute for Aviation Research (NIAR).

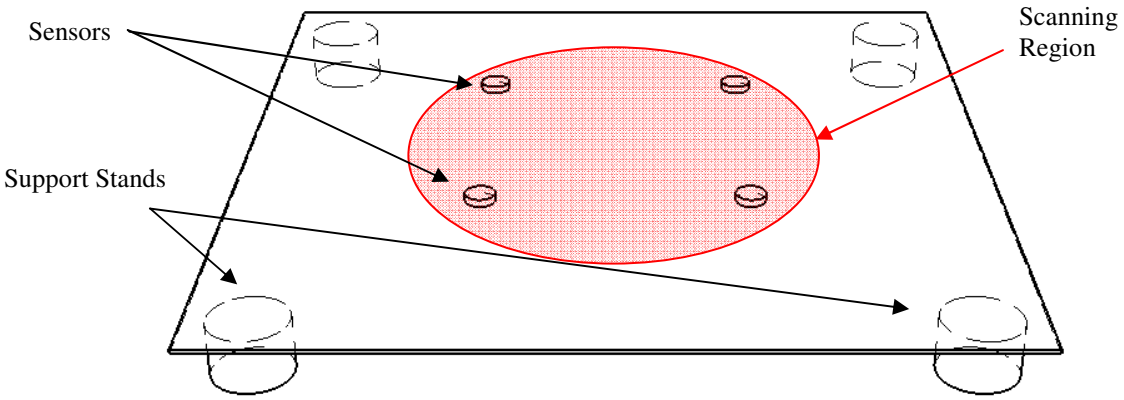


Figure 5.1 Composite laminate in simply supported configuration (scanning area is away from any boundary in center of laminate).

5.1.2 Ultrasonic Testing System

The ultrasonic testing system consisted of an equipment bundle, obtained from Acellent Technologies. Contained within the bundle were piezoelectric disks 0.2 in. in diameter and 0.001 in. thick. These disks functioned as either actuators or sensors. The disks were bonded onto the surface of the laminate composites by epoxy. The bonding process was similar to bonding a strain gage onto the surface of a structure. The piezoelectric disks themselves were similar to the company's Stanford multiactuator-receiver transduction (SMART) layer technology due to the size and wiring of the sensors. The SMART layer is a fabricated polymer sheet with piezoelectric disks embedded within it at designed locations. Two wires were connected to each piezoelectric disk. A voltage applied to the disk would result in radial expansion, and when bonded onto a structure, the deflection would cause a shear deformation. For an ultrasonic frequency oscillation, an asymmetric strain wave was released into the structure. The

reception of strain waves in a structure would function in the same manner, detecting shear deflection on the bonded surface of the component.

The piezoelectric disks were connected by a wire to a signal conditioner unit. The signal conditioner, called a ScanGenie, converted the desired signal into a voltage for the piezoelectric disks and performed the reverse process for any received signal by the sensors. The conditioner had a signal amplifier and several gains to alter the signal. As a result, the ultrasonic system was sensitive enough for the evaluation of signals in the ultrasonic frequency range. The ScanGenie was connected via a USB to a PC and was controlled by MATLAB-based software, called ACESS [42]. This software allowed for the control of signal amplitude, frequency, and gain. It could generate the raw signal form of a strain wave and convert this into a MATLAB form for post-processing. Additionally, the system was capable of directing any connected piezoelectric disk to function as either an actuator or sensor. Thus, a pitch-catch ultrasonic testing system was possible with this equipment bundle.

This Acellent Technologies sensor system was used in the initial experiments, detailed in Chapter 4. For those experiments, piezoelectric disks were used as only actuators or sensors exclusively. For this last set of experiments, the discs alternated between actuators and sensors to obtain more data and information of the laminate.

The positioning of the piezoelectric disks on a structure was required to be at least 3 in. apart [42]. When the signal conditioner sent out a voltage to an actuator, the receiving sensor also received the sent signal, due to reverberation in the equipment. To ensure that the received signal was separated in time from this crosstalk, the distance between the actuator and sensor must be greater than 3 in. Thus, the piezoelectric disks were placed 4 in. apart for this experimental research. The sampling rate of the piezoelectric sensors was 12 MS/sec for 8,000 sampling points for all experimentation, including those detailed in Chapter 4. These sampling values allowed enough time for the wave to be received by the sensor and reduce the amount of data points stored by the system.

5.1.3 Simulated Damage

Damage was simulated by the placement of wafers on the composite laminate surface. Simulated damage enabled a moveable initial damage location on the same structure. These circular-shaped wafers had diameters ranging from 0.375 in. to 1.50 in. Experimentation included a no-damage scenario. The wafers consisted of a porous material with a bonding surface on one side. The bond was strong enough to hold the wafer in place; however, a researcher was capable of removing the wafer from the surface of the laminate without special tools, returning the structure to a non-damaged state. Strain wave energy was scattered once in contact with the wafer region in a similar

manner to that of delamination or multi-site damage [42]. This method of simulated damage in a composite has been proven to be effective, based on previous experimentation (see Chapter 4). Only one damage location was used per scan for this experiment. Thus, this research included the assumption that only one damage area is present within any four-sensor system. Studies into multiple damage locations within the scanning area could be developed in future studies.

5.2 Experimental Configuration

The objective of this research was to lay out the ground work for a structural health monitoring system. An experiment involving a modular design of a sensor network was created and tested. Chapter 4 contains several initial experiments in validating the use of using artificial neural networks as signal processors for ultrasonic testing signals. These next two chapters refine the methods presented in that chapter into a more complete SHM system.

The overall goal of this research topic was to have sensor networks capable of scanning for damage installed onto thin-walled structures. Consider a sensor network that creates a grid layout. Each sensor is placed an equal distance from one another. An illustration of this is shown in Figure 5.2, where $d_1 = d_2$.

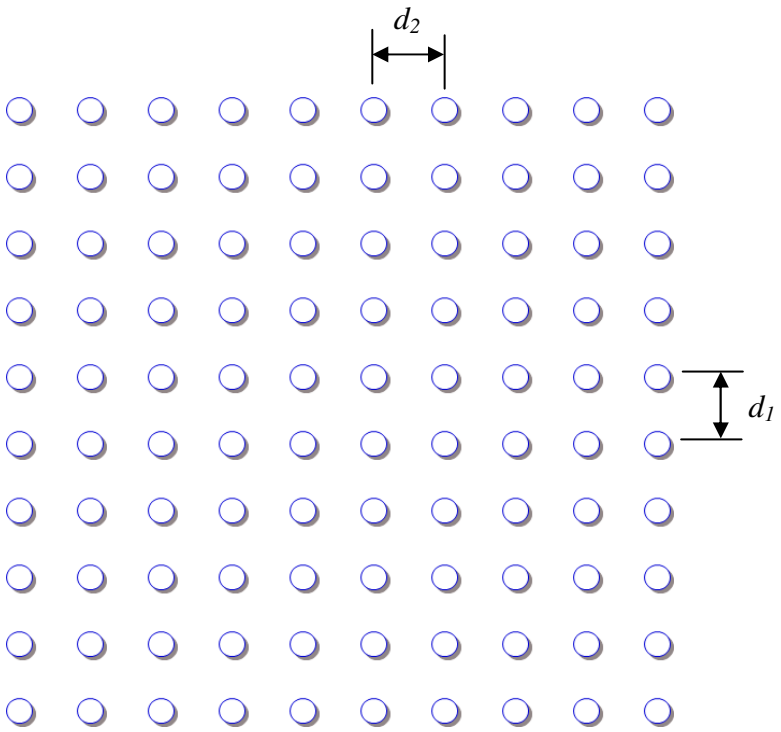


Figure 5.2 Network of sensors on structural component, where all sensors are equidistant from one another.

Although the structure does not have to be isotropic, assume that structural material is repeated across the structure. In other words, there are no internal flaws or changes in property, but the material properties can be translated throughout the structure. For a composite laminate, if a sample point was examined for through-thickness properties, then it would be the same for any point in the laminate. Using this assumption, four sensors with fixed orientation to the fiber layup should produce the same signals for any arbitrary location on the surface. If the material remains uniform throughout, then a small sensor network designed and tested for one region could be replicated over the entire structure, while maintaining the same damage scanning process. This allows the system to become modular. Assuming uniformity of the material in a four-sensor network, the validation of four sensors to detect damage would result in verifying the ability of the entire network to scan for damage.

If the four-sensor network operated either independently or only relied on neighboring systems for information, then a multi-agent system is created. Figure 5.3 and Figure 5.4 illustrate a four-sensor network and neighboring networks. In Figure 5.3, four red sensors make up a local network of four sensors. While this network scans for damage in a small area, other local networks could scan for damage simultaneously. If the networks have some distance apart, then no communication between then local networks would be necessary, so they would effectively be working independently. However, some communication between neighboring local networks could occur. The results could be compiled to a global damage assessment for the entire structure, creating a decentralized sensor network for detecting damage. The goal of this research is to validate a single, local four sensor network. The objective of this dissertation was to have a single four sensor network identify a single damage in its scanning range. Thus, with multiple four sensor networks, multiple damage areas could be detected throughout the entire structure.

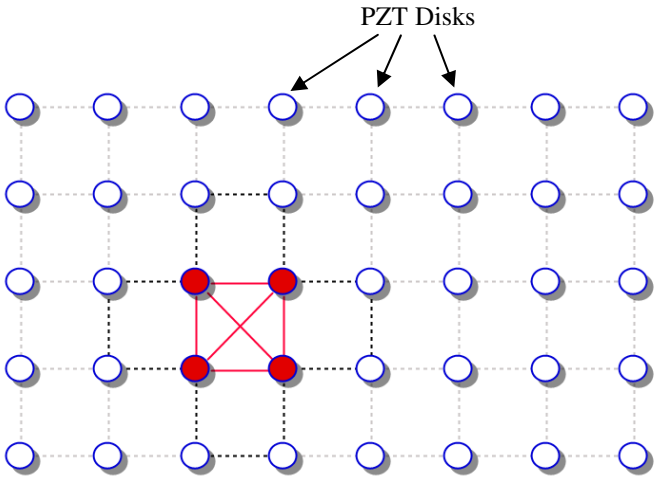


Figure 5.3 Four-sensor network for validation, highlighted in red.

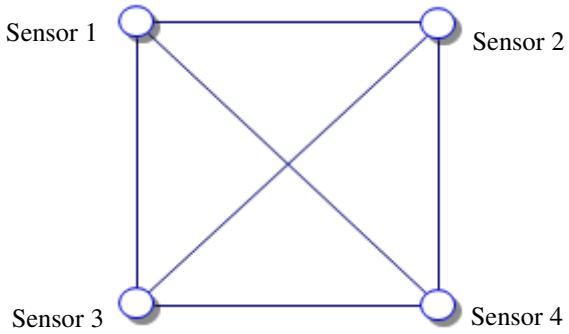


Figure 5.4 Four-sensor network consisting of four piezoelectric disks.

5.2.1 Actuator to Sensor Path Analysis

The general method of analysis for ultrasonic testing is through signal comparison. Due to complexities in wave signals, a scan of the material with a known undamaged state is recorded as a "baseline scan." This signal is used as a comparison for any scans with an unknown state, or "current scan." The current signal can be compared to the baseline signal. If any damage is present, then this is usually from the result of damage to the structure. Current research has multiple methods of comparing a current scan state to a baseline scan state. For this research, two methods of comparison will be performed.

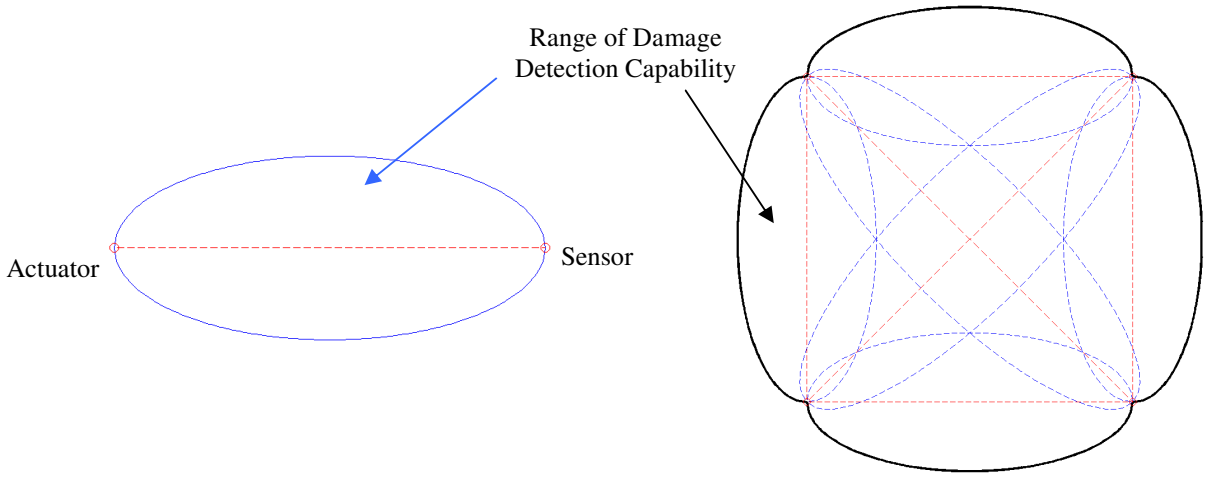
Each received ultrasonic signal was pre-processed using two methods. The first method is common in acoustic emission system analyses in factory and company inspections. For this method, certain characteristic measures of wave signals instead of the entire signal are used. Examples include the duration, rise-time, amplitude, and time of flight of wave pulses or beats in the received signal. This more crude method of gathering information reduced the amount of data that was needed to be saved. The second method, a damage index method, looks at taking the energy stored in a wave. Energy was determined to be the integral of a strain wave signal voltage envelope as a function of time. This was then compared to the damage index of the baseline scan. The method was based on the work of Rose [17]. In Acellent software analysis, a crude form of using wave properties from the scatter signal is employed [42]. However, this method relies on a large number of actuator-to-sensor paths. One of the goals of using artificial intelligence methods was to reduce the number of scanning sensors and paths, while maintaining the same accurate damage assessment of the material.

The frequency of strain waves in the experimentation ranged from 50 kHz to 200 kHz. The strain waves consisted of only a single asymmetric mode. Previous experiments, discussed in Chapter 4, along with aid from Acellent Technologies, had validated this scanning frequency range [42].

For any comparison method examined, the difference between a baseline signal and a current or unknown state signal was placed as input into a processing system to determine the presence of damage. Artificial neural networks were implemented at this stage in the analysis. Different architectures, based on previous experimentation, were investigated. The variations in sign between current and undamaged states were shown to artificial neural networks in order to train and determine if and where damage was present in the material. Programming was done through MATLAB to connect the software involved [62]. This included Acellent Technologies for signal acquisition [42] and NeuralWorks II for the generation of artificial neural networks [63]. Copies of some of the programs written for this research are provided in Appendix B.

5.2.2 Multi-Agent System

Consider a four-sensor square as an agent. This agent has twelve paths to scan for damage within a square area, as illustrated previously in Figure 5.4. The agent must be capable of accurately detecting and locating damage within this square area and possibly surrounding location, as shown in Figure 5.5, which illustrates that based on a single actuator-to-sensor path, there should be a potential damage detection area around the direct path. For an agent of four sensors, this area encompasses the square area formed by the sensors and some of the surrounding area.

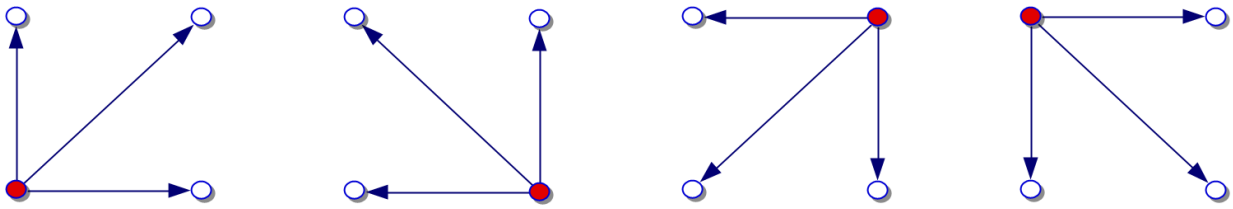


(a) Two-Sensor System (b) Four-Sensor System

Figure 5.5 Potential detectable range for (a) two-sensor system and (b) four-sensor system.

The method of detecting and locating damage within an agent was based on previous experimentation. Each agent could be sub-divided into four sub-agents. Each sub-agent used a single actuator to send a strain wave to the three remaining sensors. The three signals received by sensors from a single actuator point were combined in a single artificial neural network. This system would produce three signals—0° path, 45° path, and 90° path—for each

ANN. Figure 5.6 is a visual representation of the four sub-agents. The initial stage was to examine for damage presence or severity. These four sub-agents acted independently and determined if damage was present, based on an ANN. Each of the four sub-agents was trained independently but simultaneously on the same damage placement. Inputs consisted of the three signals, compressed into properties detailed in the follow chapter, and were trained to the output of damage size. The output set was a single value of the diameter of the simulated damage piece. The outputs of these agents, once trained, were then combined into a fuser network. This network was then trained to predict the size of the damage on the testing coupon. Simulated damage ranged from no damage present to damage 1.5 in. in diameter. This range enveloped current damage detection NDT methods in the aerospace industry. Current inspection programs rely on damage being detected before 0.5 in. is reached, so this range was deemed suitable for experimentation. A flow chart of this system is presented in Figure 5.7. The purpose of creating a fuser network was to eliminate any bias that one particular piezoelectric disk would alter the system.



Actuator 1 Actuator 2 Actuator 3 Actuator 4

Figure 5.6 First layer actuator to sensor paths (each actuator, or red disk, sends out a signal to three sensors, or white disk, and each actuator has its own artificial neural network system to determine the damage location).

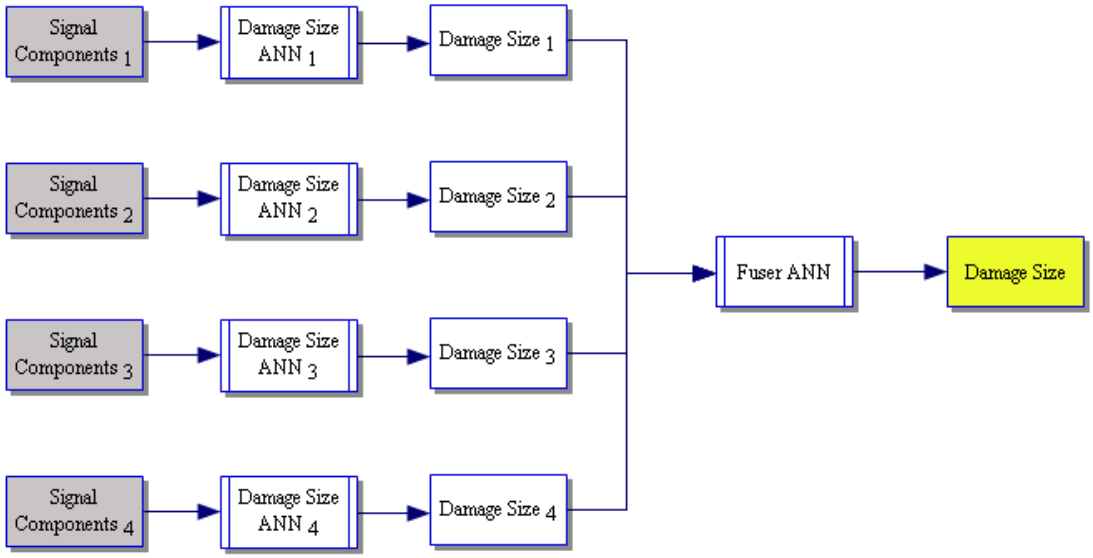


Figure 5.7 Flow chart of damage-severity system.

The next step was to add a filter. The filter chosen was a hard threshold on results obtained from damage severity. After training, a value produced by the four-agent system was chosen as either "damage is present" or "undamaged state." This value was conservatively chosen to be low, since false readings about damage present are more desirable than the alternative of predicting no damage within the system when there is actual damage. This value, determined to be 0.2 in., is detailed in the Chapter 6, relating the results of training of the various ANNs in the damage-severity system.

If a set of signals obtained was determined to be damage, then a second system of ANNs was developed. This system was similar to the configuration of damage severity but modeled from previous experimentation as well. The experimentation in Section 4.3 demonstrated that an ANN could be trained to identify the polar coordinates of damage present. Since the "undamaged" scenario was filtered out before this system was introduced, the assumption that damage is always present, made in the previous experimentation, was applied again for this system. The flow chart is presented in Figure 5.8. Sub-agents in this system also used the damage size results obtained.

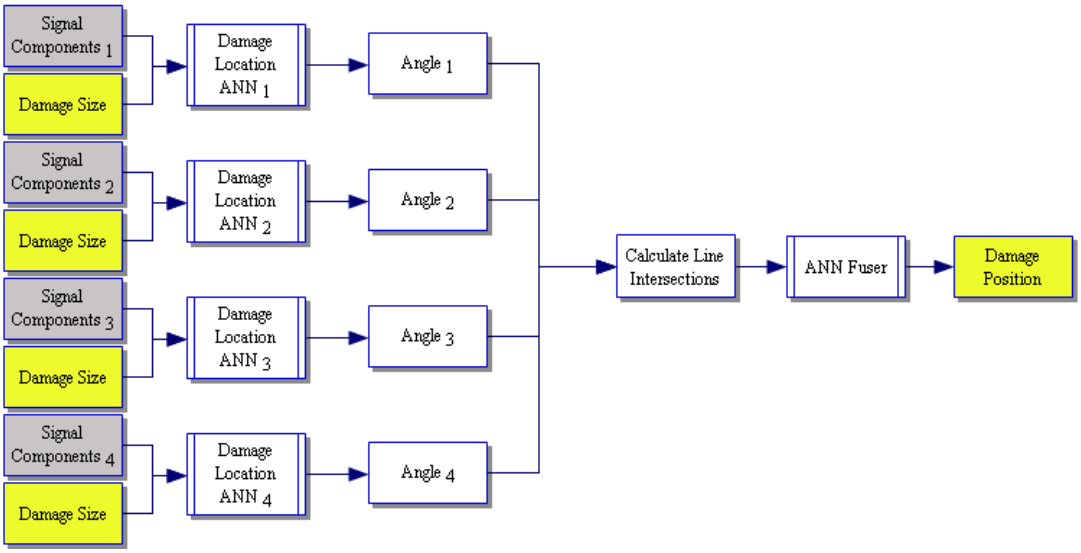


Figure 5.8 Flow chart of damage-positioning system.

The output for each sub-agent was damage location. However, based on previous research, the output was just the damage location angle in polar coordinates. An illustration of this angle for each actuator is shown in Figure 5.9. A scanning field of 0° to 90° around each actuator was presented for each network. This expanded on the previous experimentation (see Chapter 4). The angle of where damage was located formed a line coming from each actuator. These lines intersected each other, forming six intersection points. Ideally, the intersection points would be in the same location. However, due to variance in the artificial neural networks, the location was around a general

area instead. The intersection points, determined in Cartesian coordinates, were then used as inputs to a fuser network, which in turn determined the damage location.

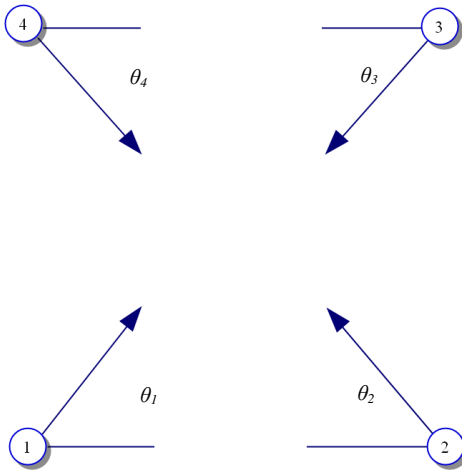


Figure 5.9 Angle orientation for artificial neural networks to locate positioning.

Thus, for damage detection and location, the system was reduced to two agents. The first agent was used to determine if damage was present in the scanning region. This agent would output the presence of damage as a number ranging from 0 to 1.5, based on the damage size. If a certain threshold was reached, then the set of signals was determined to be the results of a damage presence, and a locator agent would be activated. This damage sizing factor was used as an additional input parameter in the damage locator agent. A diagram of this two-agent system is illustrated in Figure 5.10. From this, a four-sensor, SHM system was developed, combining the abilities of artificial neural networks with the use of multi-agent system to obtain an accurate assessment of the change in an ultrasonic signal to detect damage.

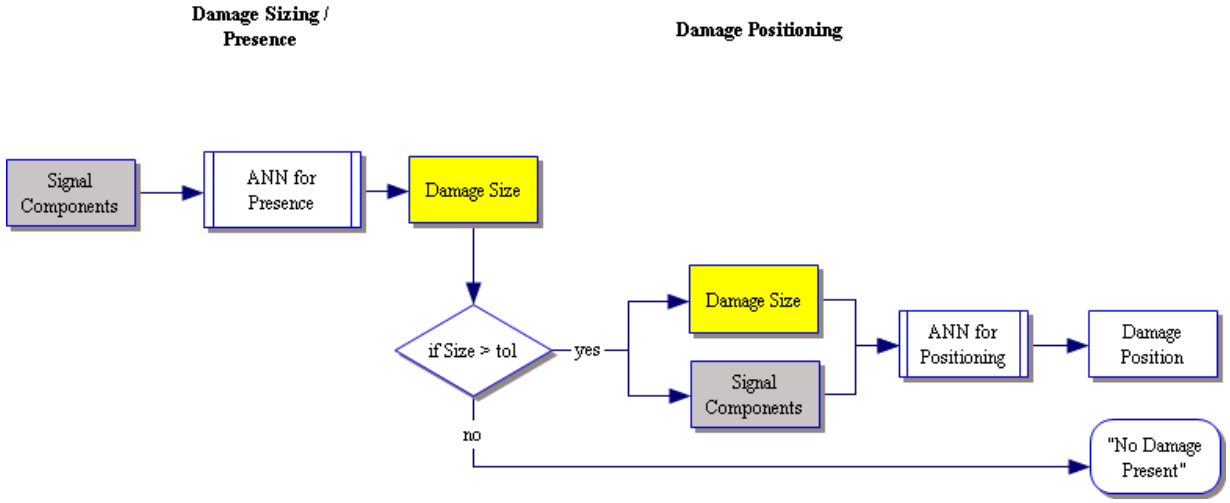


Figure 5.10 Flow chart of two-network system for four-sensor block.

CHAPTER 6

RESULTS AND DISCUSSION

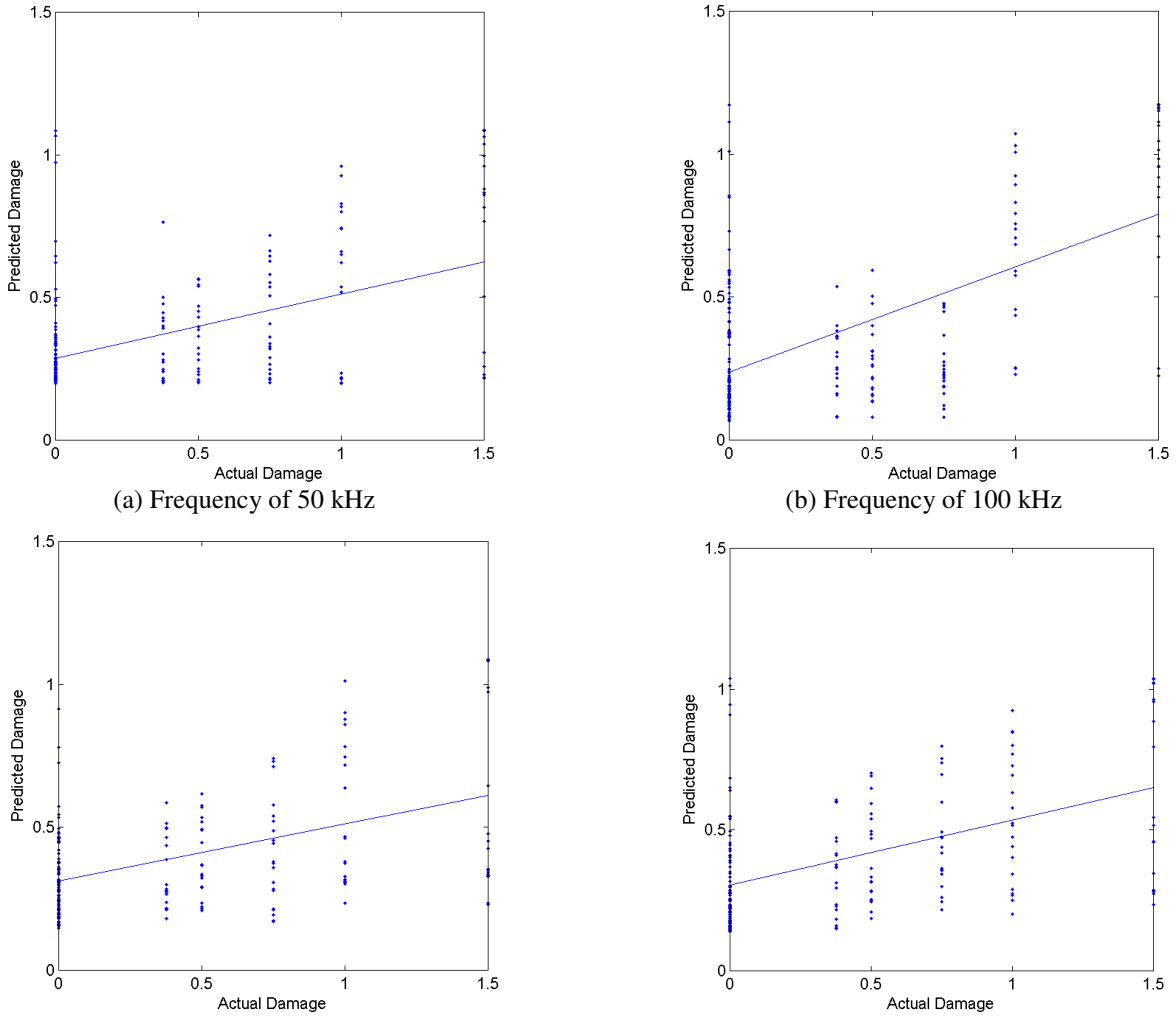
The experiments and methods described in Chapter 5 were conducted with an ultrasonic testing system to detect simulated damage on a laminate composite that was similar in configuration to aerospace industry structures. The experiments were separated into a damage-severity detection system and a damage-positioning system. This chapter presents a summary of and the results for all of these different experiments, including several theoretical configurations for a structural health monitoring system, which combines ultrasonic testing sensor systems and artificial neural network systems. Then this is compared to current inspection programs in the aerospace industry.

6.1 Damage Severity

The first agent of the SHM system was damage severity. As described in the previous chapter, damage wafers were placed on the surface of the composite laminate for experimentation. The signal with a damage presence was compared to an undamaged state. For initial training, all data points were used in a single dataset. The following measures were taken from each signal: amplitude, time of flight, rise time, and duration. A fifth measure, energy, was calculated using these traits. Energy consists of the area under the wave envelope, which can be approximated using the previous measures. For the ANNs, the energy trait was used as an input, since it incorporates most of the other wave properties. This is similar to the damage index approach of Rose [17]. The best trait for damage identification was determined to be the energy signal. From the set of experiments described in Section 4.4, different measures of a strain wave were compared to determine the effects of simulated and physical damage. From this, it was concluded that the duration of a wave and rise-time remained unchanged. The amplitude and time of flight had effects on an energy measure of the wave. Experimentation by Rose demonstrated that an integral factor could be used to relate a strain wave characteristic and could be altered significantly in the presence of damage. Following these conclusions, the energy measure of the strain wave was determined to be used for this experiment.

A feed-forward network was chosen for the damage-severity agent. Four sub-artificial neural networks were constructed, one for each actuator, resulting in three actuator-to-sensor paths. Input for the artificial neural network was the energy difference from the three paths, and the output was damage size. The ANNs had three input neurons, one hidden layer of 3 nodes with sigmoid activation functions, and one output neuron. Since damage size was always greater than zero, a sigmoid function was determined to be the most suitable for this application. Additional nodes in the hidden layer or additional hidden layers did not affect the results of learning datasets, so the

smallest network was chosen. Once all four networks were trained and generated an output, results were fed into a self-organizing map feed-forward hybrid network to function as a fuser for the data. The architecture here is similar to the favored ANN in experiments described in Chapter 4. This network had four results of the four networks as its input, and the damage severity for the scanning area as its output. The purpose of the ANN hybrid as a fuser was to allow any bias from a particular actuator to be accounted for and adjusted in a pattern-recognition network. The SOM feed-forward hybrid network allowed for any patterns present between the various actuator signals to be included in the combination results. Results for the individual frequency results are presented in Figure 6.1, which indicates a large spread in the results of all frequencies. The values for the "undamaged" state, or damage with a value of 0, are grouped tighter, but a plot of points can make this difficult to see.



(a) Frequency of 50 kHz

(b) Frequency of 100 kHz

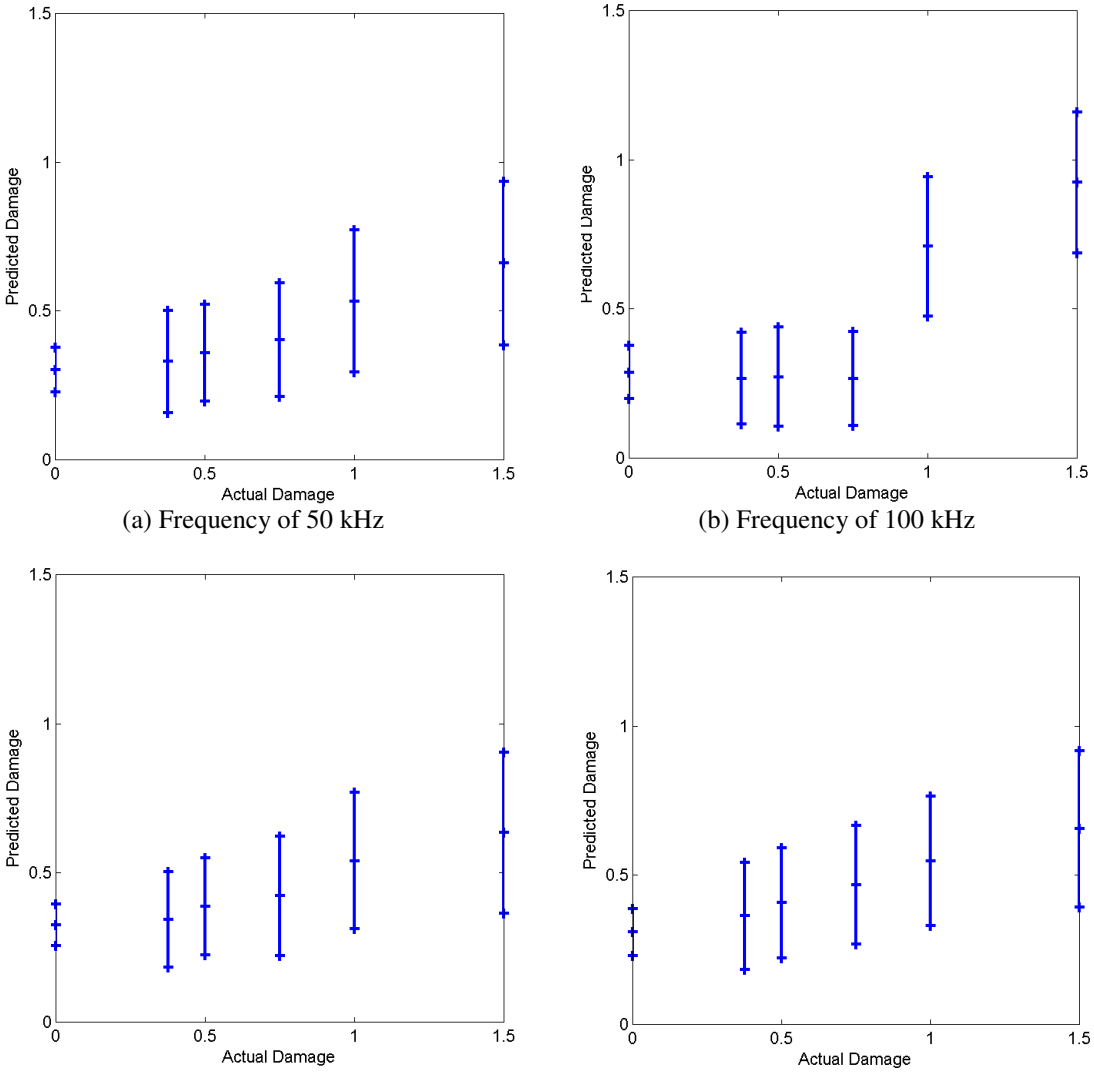
(c) Frequency of 150 kHz

(d) Frequency of 200 kHz

Figure 6.1 Training set results for damage severity, based on frequency of strain waves used (units in inches).

Figure 6.2 shows 95% confidence intervals for the same data. In these plots, each target value has a spread of predictions from the multi-agent system. Each bar for a target value contains a range of 95% of the data, assuming a normal distribution. The middle mark in each bar is the mean. If the bars do not overlap in prediction values of two target values, then they are considered to be statistically significantly different. This comparison method was employed to validate the multi-agent systems developed in this dissertation.

In these plots, significant difference occurs for the 100 kHz frequency, where any damage greater than 1 in. diameter was detectable, thus demonstrating that using one frequency is not a good solution for this sensor configuration. Previous experimentation (see Chapter 4) used multiple frequency inputs for ANN sets. This led to a hypothesis that information between frequencies could provide more information to artificial neural networks.



(a) Frequency of 50 kHz

(b) Frequency of 100 kHz

(c) Frequency of 150 kHz

(d) Frequency of 200 kHz

Figure 6.2 Training set results for damage severity as 95% confidence intervals (units in inches).

Four systems were created again, but the data from all four frequencies was put into each network. The artificial network used a set of 12 inputs, consisting of the energy differences for three paths with four different frequencies. The output consisted of the simulated damage size with diameters of 0 in., 0.375 in., 0.5 in., 0.75 in., 1 in., and 1.5 in. Three hidden layers had hidden nodes of 10 nodes, 7 nodes, and 3 nodes. Notation for the network setup was [12 / 10 / 7 / 3 / 1]. This architecture had a decrease in nodes per hidden layer, compressing the data slowly to a single output value. Architectures with only one or two hidden layers seemed to truncate too much information too quickly, resulting in poor training. The learning was delta rule learning, where weight updates between each layer of the network were updated after each input to the output dataset. The activation functions were sigmoid. Since all values were above zero, this was determined appropriate. The results of these four systems were collected together into a fuser network composed of an SOM feed-forward hybrid network configuration. The map was a Kohonen layer of 10 x 10 nodes fed into a 5-node hidden layer and then a final output. The hidden layer was trained via backwards propagation. The SOM feed-forward hybrid network was determined to help with any bias that may be present in a single actuator set. If a sensor went out, then theoretically the SOM could adjust itself accordingly to account for a malfunctioned sensor through pattern recognition. Results of this training are displayed in Figure 6.3. There was some spread in the resulting values, so as was done previously, the data is displayed as 95% confidence intervals in Figure 6.4. There were some outliers in the results of this training; however, for statistical verification, the comparison of data as confidence intervals is used to determine the significance of the results.

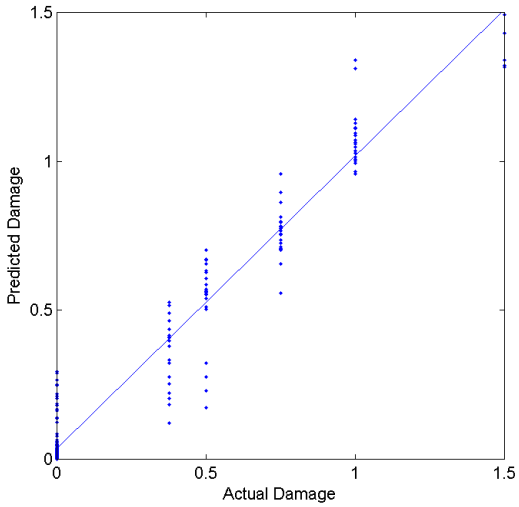


Figure 6.3 Training set results for damage severity (units in inches).

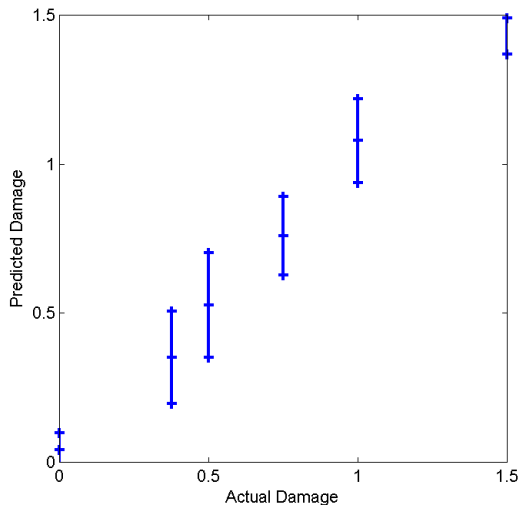


Figure 6.4 Training set results for damage severity as 95% confidence intervals (units in inches).

The 95% confidence intervals for the six damage sizes trained had a much smaller spread, compared to the individual frequencies. This demonstrated that there is some link between the information held in different frequencies, and this can be "learned" by an artificial neural network system. The smaller the size of damage resulted in less confidence in damage sizing. However, there was significant difference between no damage and the 0.375 in.-diameter damage with 95% confidence. This shows that with 95% confidence, the system is capable of detecting damage that is 0.375 in. in diameter.

The system was verified through a bootstrap method. The data was trained on 80% of the datasets and tested on the remaining 20%. Results were similar to those of the training set. There was a larger range in the 95% interval. However, the results still indicated a significant difference between no damage presence and at least 0.375 in.-damage presence. Figure 6.5 shows the results of two samples of the testing sets from a bootstrap method. The smaller damage sizes had a larger spread, indicating less accuracy or confidence in results. But the "undamaged" prediction was always significantly different. This aids in validating artificial neural networks used to determine the presence and severity of damage for the scanning region of the composite laminate in experimentation.

From Figures 6.4 and 6.5, 95% confidence was shown in determining whether damage is of size 0.375 in. or larger. In both figures, this is due to the significant difference in the 95% confidence between damage of size 0 in. and size 0.375 in. In the flow chart shown previously shown in Figure 5.10, the size-prediction network was used to also determine if damage is present in the structure. A filter with a hard threshold was used to differentiate the results from the size-prediction network to either "damaged" or "undamaged." A threshold of 0.2 in. was used for

this research. This is a conservative number because it is on the lower end of the range of detection for the 0.375-in. damage. The determining factor for this would be to have the system respond with a small damage point or false reading, rather than not detect any possible damage. The resulting values with predictions above the 0.2 threshold line were assumed to be the result of damage presence and carried over into the next agent, damage positioning.

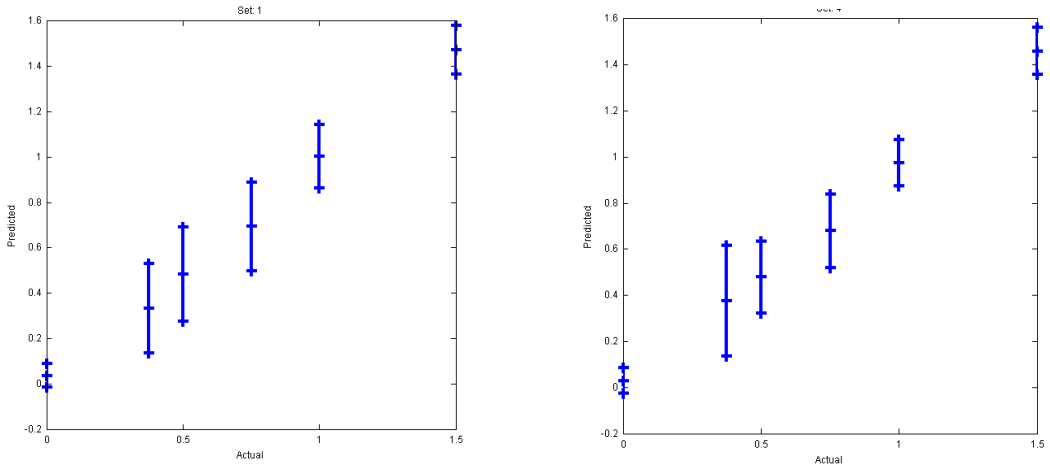


Figure 6.5 Bootstrap testing results for two different samplings (units in inches).

6.2 Damage Location

The most promising results of the size-prediction network were from the setup of all four frequencies simultaneously. The flowchart shown previously in Figure 5.8 describes the size-prediction network. Here, each sub-network used 12 inputs, four frequencies over three paths. The output was a damage-severity factor. The damage-positioning network (see previous Figure 5.9) operates in a similar manner to the size-prediction network. The damage-severity factor, or output of the sizing network, was added as an additional input to the damage-positioning network. Since the size of the damage affects the signals obtained for ultrasonic scanning, this prediction was assumed to be useful information in training a damage-positioning network. Therefore, each of the four sub-agent networks in the damage-positioning network had 13 inputs. These were fed into an SOM feed-forward hybrid network, which had an output of the angle of damage positioning. Each artificial neural network consisted of a Kohonen level of 20 x 20 nodes and one hidden layer of 5 nodes. Figure 5.9 includes an illustration of the angle orientation used for the output set. The results from the individual training sets are shown in Figure 6.6. The correlation between the target and predicted angles is close to a trend line, but some scatter is present. Therefore, another fuser network was implemented.

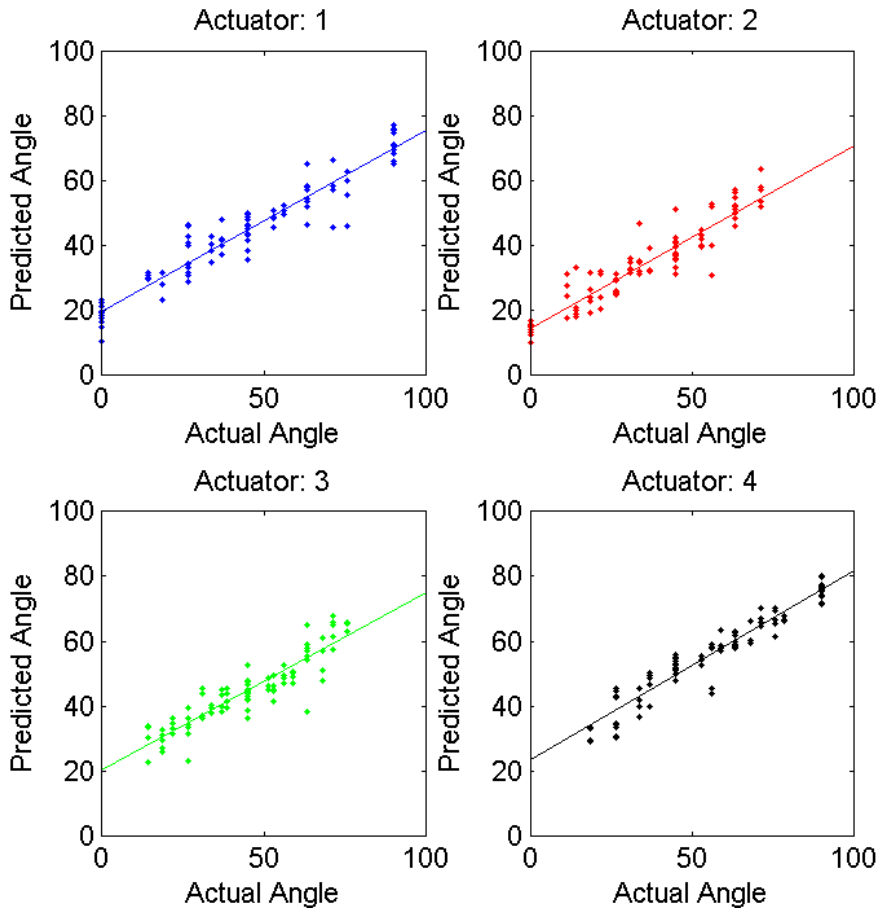


Figure 6.6 Angle positions of artificial neural networks for individual actuator sets (units in degrees).

The output of these sub-agent networks was to be an angle value. From the set of experiments, described in Section 4.3, it was concluded that a self-organizing map can identify the angle of damage within a domain. Since the radius was never shown to train well in that experiment set, an alternative method was developed. Because the outputs of these four sub-agents are an angle and no radius, the result would be four lines of possible damage locations. The four lines will intersect one another, forming six intersection points. These intersection points ideally would be the same and at the damage position. However, due to some scatter in the data, these will cluster around the damage location. A final self-organizing map was developed to input the six intersection points and predict the position of the damage. The six intersection points were provided as Cartesian points, resulting in twelve inputs. The origin was placed at Actuator 1, and the units used for the points were in inches. Actuator 1 was considered point (0,0), and Actuator 3 was considered point (4,4), or 4 in. away from the origin in both the x- and y-directions. There were two outputs. Figure 6.7 shows the predicted point results, indicating a spread of data, and Figure 6.8 contains the data, reconfigured to be 95% confidence intervals.

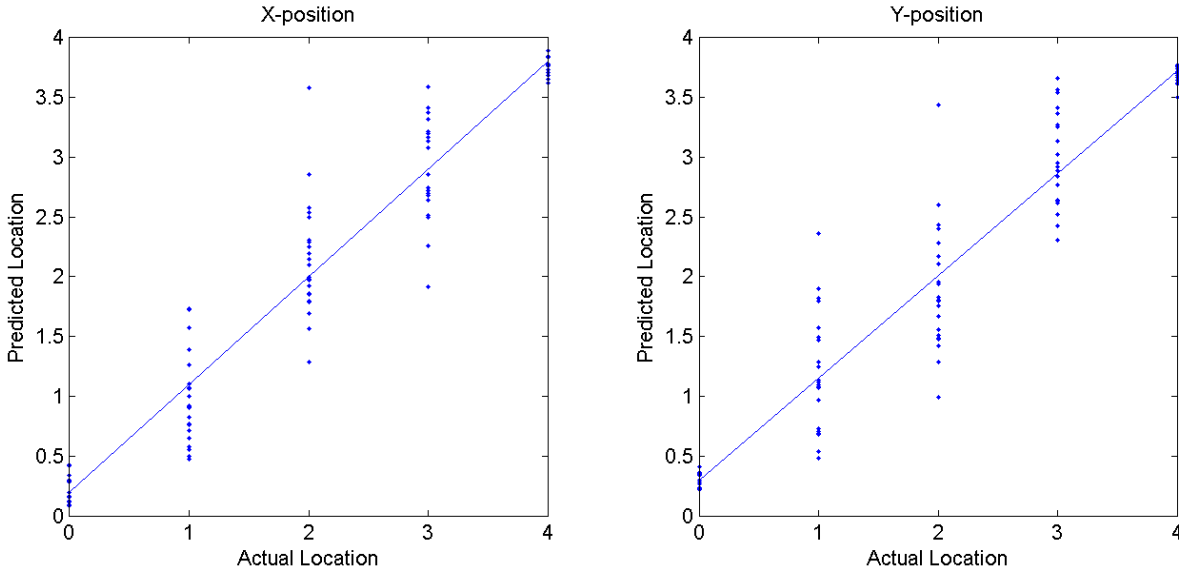


Figure 6.7 Positioning results for training dataset (units in inches).

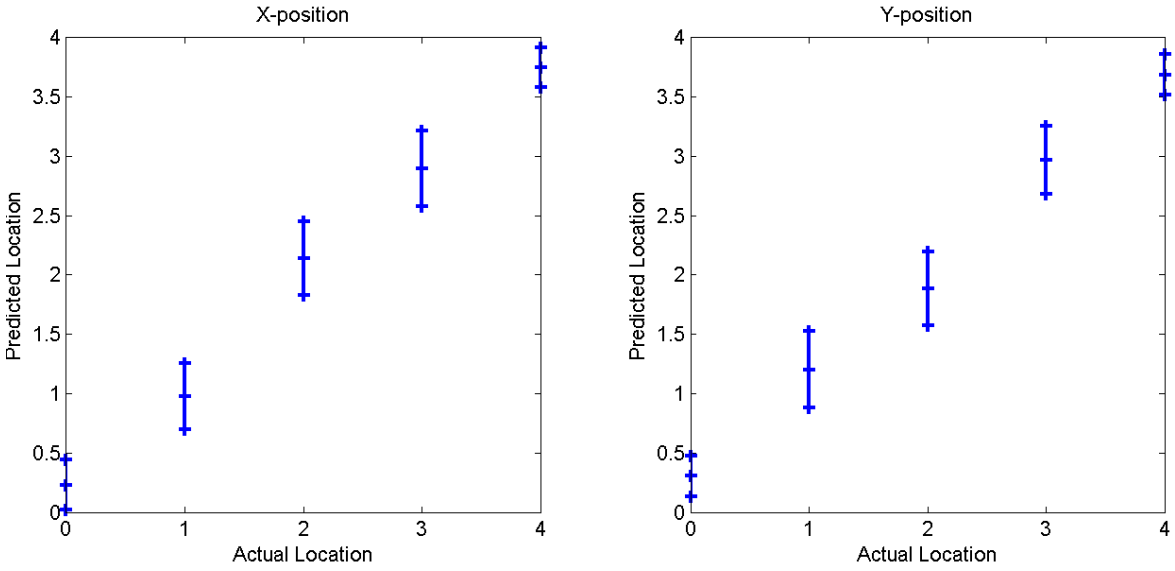


Figure 6.8 Positioning results for training dataset as 95% confidence intervals (units in inches).

From these plots, it can be seen that positioning makes a significant difference in predicting damage 1 in. apart in the x- and y-directions throughout the scanning domain. Results were further examined by determining the distance between the predicted location and the actual damage location. The histogram shown in Figure 6.9 provides the frequencies of predicted and actual distances in nine evenly spaced bins. This follows a Weibull distribution, since there cannot be a negative distance, with an average of 0.522-in. difference. A majority of the distances are below the 1-in. mark, which verifies the result of the 95% confidence intervals shown in Figure 6.8.

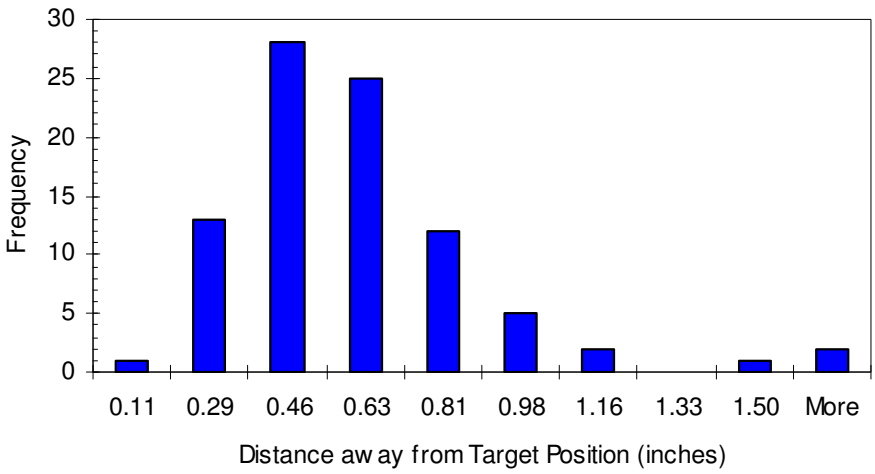
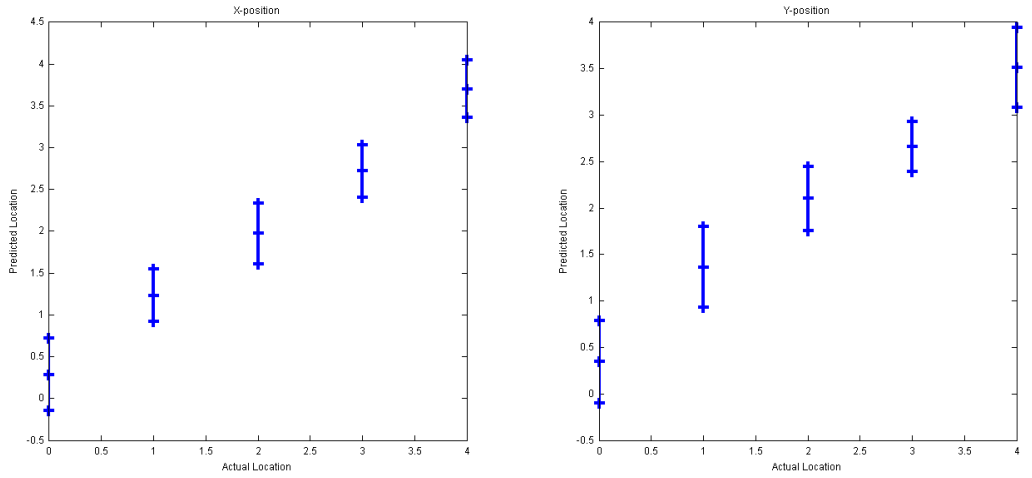


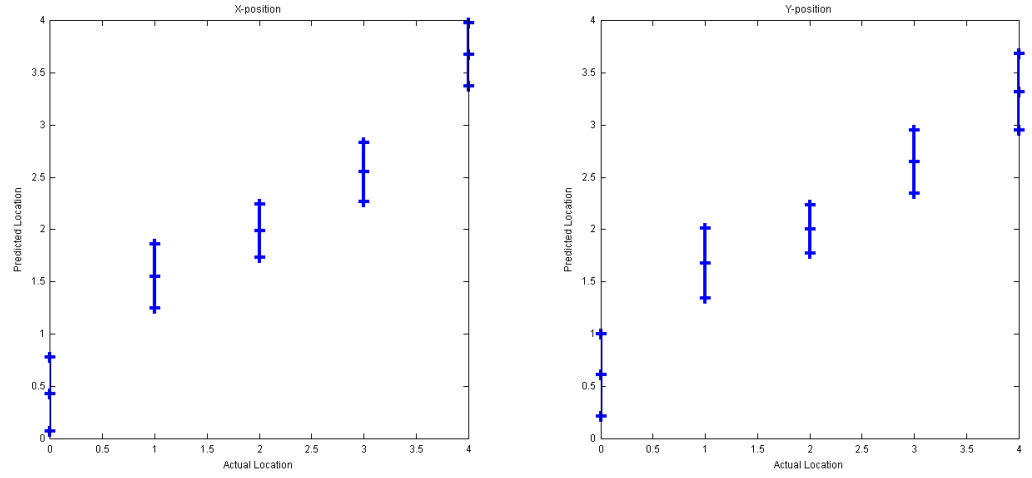
Figure 6.9 Histogram of distance between predicted and actual damage locations.

Similar to the damage-severity ANN, the bootstrap datasets were carried through the positioning artificial neural network system as well. A sampling of the results from testing sets is presented in Figure 6.10. Results show that there is a significant difference among damage located on the boarder of the scanning region, while damage inside the region is not as significantly different. Since damage on the boarders is on a direct actuator-to-sensor path, this type of result is expected. Although some of the internal scanning locations overlap others in predictions, as evidenced in Set 4 of Figure 6.10(b), if there is enough data for the system to train on, then the distinctions in location can be made. This is shown in the initial training results shown previously in Figure 6.8, where the system had the entire domain to train.

This experimentation research has demonstrated that an artificial neural network system could be implemented with an ultrasonic sensor system to detect and locate damage. More frequencies scanned by the system can result in more information. This research has determined that with more scanning frequencies, artificial neural network systems can be trained to have higher accuracy in damage severity and location.



(a) Set 1 of Bootstrap Data



(b) Set 4 of Bootstrap Data

Figure 6.10 Sample of results from testing dataset of bootstrap datasets for damage location (units in inches).

6.3 Expanded Research for Structural Health Monitoring System

The four-sensor system was verified to be capable of detecting and locating damage within a small square area. The next layer in this multi-agent system would be to examine the effects of having two adjacent four sensor agents. For this layer, an agent is defined as a four sensor system. This system could contain two four sensor systems, which share two sensors at a connected boundary. These two agents could then scan for damage within their square, but could collaborate on any damage results close to their shared boundary. Figure 6.11 contains an illustration of two agents sharing a boundary. The darken region is predicted to be detectable by both agents.

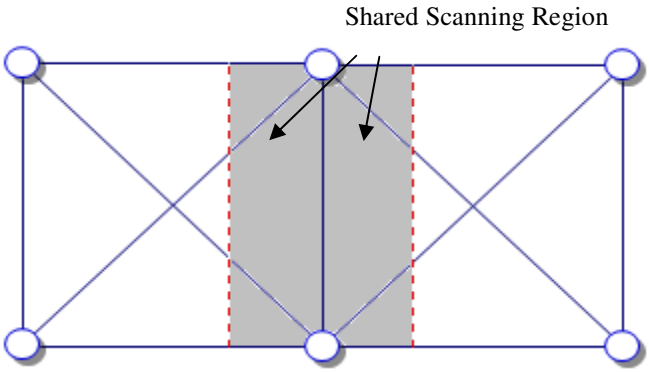


Figure 6.11 Six-sensor system forming two areas of detection with neighboring boundary.

Forms of communication, either at the sub-agent layers of damage size and location or at the agent layer could be shared. The objective would be to provide a more accurate damage assessment on the shared boundary between the sensor agents. Once a suitable method is established, a second shared boundary could be introduced. This would form a three-agent system with one agent sharing two boundaries. Figure 6.12 illustrates this setup.

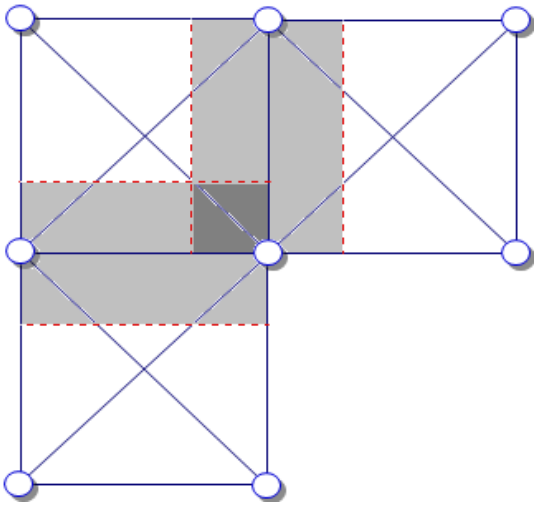
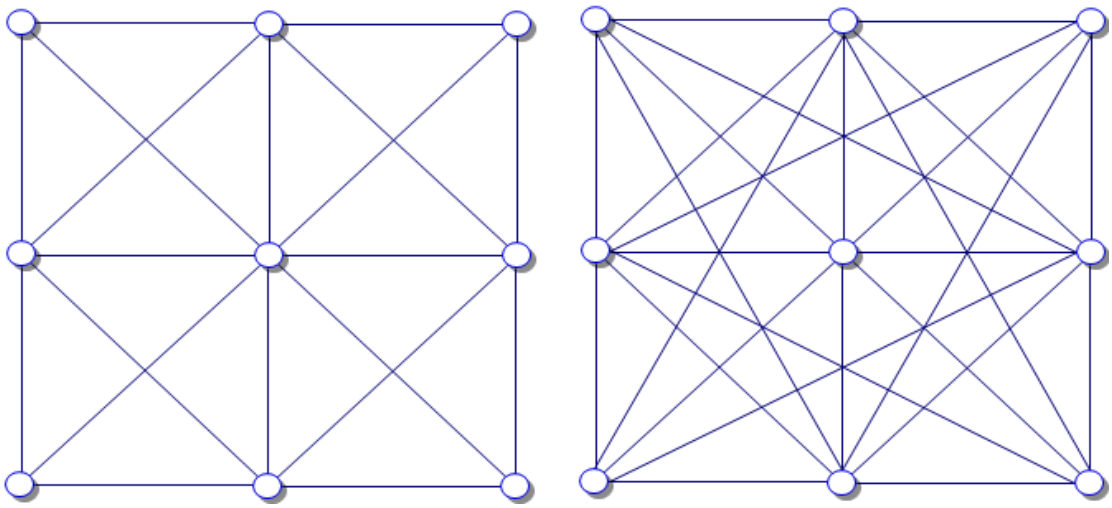


Figure 6.12 Eight-sensor system forming three areas of detection and two neighboring boundaries.

This process could be extended to four agents with four shared boundaries, as shown in Figure 6.13(a). There could be potential for the overlapping of multiple detectable regions. The scope of this research is for a four-sensor agent and the possibility of sharing one boundary. However, this setup results in a modular system, which could be expanded indefinitely to cover a composite laminate of infinite size.

Current ultrasonic testing approaches include scanning all possibilities of paths. For example, consider a nine-sensor system. This can be broken down into a four-agent system as shown in Figure 6.13(b) and could consist of 48 different paths. Removing redundant actuator-to-sensor paths, the system could be reduced to 40 paths. All

possible actuator-to-sensor paths involve seventy-two paths. The SHM system uses about 44% less scanning paths, which would decrease scanning time. This idea could be extended to larger sizes of square sensor systems.



(a) Paths Required by System (b) All Possible Paths to Scan

Figure 6.13 Nine-sensor system forming three areas of detection and four neighboring boundaries.

Table 6.1 contains a comparison of the number of paths required for scanning in both the SHM system and for every possible scanning path. As more paths are included in scanning, more comparisons would have to be made by any post-processing system. For the SHM system, the detecting method would remain on a localized scale. This would result in a decentralized damage-detection system with fewer computations to perform while maintaining the accuracy of systems with many paths to compare. This could reduce scanning time as well as processing time to obtain a close to real-time damage assessment of the structural component.

TABLE 6.1

NUMBER OF SCANNING PATHS WITH SQUARE SCANNING AREA

Sensor Count on Edge	Total Sensors	Actuator-to-Sensor Paths	
		Multi-Agent System	All Possible
2	4	12	12
3	9	40	72
4	16	84	240
5	25	144	600
6	36	220	1260

6.4 Comparison to Current Inspection Methods

The current methods of inspection involve different forms of nondestructive inspection methods at different intervals in a structure's life. Each item in a structure could have differing inspections, which can result in more downtime of an aircraft, based on the structure to be analyzed. Thousands of structural details on an aircraft are analyzed individually, and each has an inspection program in place.

These programs have been developed to determine a scanning interval, based on the probability to detect damage and, at an early enough stage, to prevent any catastrophic failure. However, these inspection intervals are measured in years and thousands of flights for commercial airlines. In comparison, with a structural health monitoring system in place, the probability to detect damage continuously during each flight would drastically increase. An SHM system would allow for a sensor network to be in place, constantly scanning for damage severity and location and possibly predicting a part's remaining life. The researched method of combining artificial neural networks to ultrasonic sensors is a novel approach to the development of SHM systems. The ability of ANNs allows them to adapt to new structures and be trained to identify damage. Their implementation could reduce the number of details, which need individual inspection programs, since the neural network system could adapt to new structures. Only a single four-sensor agent would need to be trained per region or structural detail, which could then be repeated over similar details or structural areas. The result could be a faster response time to the onset of damage, while minimizing downtime of an aircraft for inspections. The same conclusion could be drawn for wind turbines at a wind farm as well as spacecraft or any other aerospace structure. The results from this research demonstrate a novel method for data interpretation of an ultrasonic sensor structural health monitoring system. An SHM system could reduce not only downtime of aerospace structures in a fleet but also analysis required for new aerospace systems.

CHAPTER 7

CONCLUSION AND FUTURE WORK

This objective of this dissertation research was to develop the basic structure of a structural health monitoring system. It utilized bonded ultrasonic sensors to detect the internal damage of a composite laminate, as used in common aerospace structures. Ultrasonic sensors are capable of sending strain waves through a structural component by means of an actuator and sensor. Strain waves will distort due to travel distance and a structure's geometric boundaries. From a known state, an ultrasonic sensor system will have known signals produced. If damage is introduced to the structure, then the signals will change. The goal of this research was to obtain information about damage size and location based on differences in ultrasonic signals between a known "undamaged" state and an unknown state. Artificial neural networks were implemented as a post-processing of wave signals. These have the ability to "learn" damage signal changes and predict future damage sightings in the structure. The system developed in this dissertation demonstrates that ANNs can be used as a post-processor for ultrasonic sensor signals. The system presented used a combination of four sensors, acting in sub-agents to assess the damage state of a laminate. The results were combined by a fuser to increase the accuracy of predicting damage size and location. It was shown that the multi-agent system, consisting of smaller artificial neural networks was capable of detecting damage of size 0.375 in. or larger with 95% confidence. For damage larger than 0.75 in., there was a 95% confidence in the size prediction. The damage location could be predicted to within 1 in. with 95% confidence. It was demonstrated that different frequencies contain more information, which can increase the accuracy of results. The research in this dissertation validated the use of ANNs and multi-agents systems for use in a structural health monitoring system with ultrasonic sensors.

However, the experiments performed in this dissertation were the initial stages in the development of an SHM system. This project could be expanded into future research. For a four-sensor network, other artificial neural network architectures could be examined. Different learning rates, numbers of nodes, activation functions, or other concepts in the formation of feed-forward and self-organizing map networks could be examined. The structure and material could be altered as well. In this dissertation, a composite laminate was chosen, due to its complexity, but metal structures are still being used in the aerospace industry. Structures with metals can become complex, so studies into the analysis of other materials or complex geometries that are found in aircraft, wind turbines, or other aerospace structures could be examined. The number of sensors could be altered as well. An additional layer of

agent post-processing could be added to the system, combining results from neighboring four-sensor groups. This would lead to a decentralized damage-detection sensor system, which could detect damage at multiple locations in a structure while operating in real-time throughout the structure. After a system is deemed sufficient at detecting and locating damage, effects in damage over time could be studied. Changes in size, based on loading cycles could be performed with such a detection system. Studies into the damage growth rate and lifetime of a structure could potentially be assessed by such a real-time system. The results of such studies could decrease the amount of ground inspections of aircraft, for example. Aircraft could be repaired after a known number of flight hours, based on crack-growth detections.

Many researchers are investigating the possibilities of a structural health monitoring system for aerospace systems. For those investigating ultrasonic testing methods, complex analytical methods are being studied. The use of artificial intelligence methods in this research allowed for some complexities of aerospace structures to exist, while maintaining an accurate detection of damage. The research presented in this dissertation was the initial investigation on the basic structure of a modular sensor system. Through four piezoelectric disks, a basic SHM system is expected to be capable of scanning for small damage on any thin-walled structure. This research could be expanded to develop structural health monitoring the size of an entire structure to scan for damage in real-time and assess the remaining life and performance in the near future.

REFERENCES

REFERENCES

[1] Hawaiian Steam Engineering, "Aircraft Accident Report: Aloha Airlines, Flight 243, Near Maui Hawaii, April 28, 1998," URL: www.aloha.net/~icarus/index.htm [retrieved January 2009].

[2] NOVA, "Crash of Flight 111," Public Broadcasting Service, February 2004, URL: www.pbs.org/wgbh/nova/aircrash/, [retrieved May 2012].

[3] Federal Aviation Administration, "Airworthiness Compliance Check Sheet Handbook," Department of Transportation, AFS-350.

[4] Falk, J., "Nondestructive Testing Technique for Composite Panels Using Tap Test Acoustic Signals," M.S. Thesis, Wichita State University, Wichita, KS, 2002.

[5] Vestas, *Vestas | Life Cycle Assessment (LCA)*, 2007, URL: www.vestas.com/en/about-vestas/sustainability/wind-turbines-and-the-environment/life-cycle-assessment-(lca).aspx, [retrieved May 2010].

[6] Veers P., "Blade Reliability Initiative," 2008 Wind Turbine Blade Workshop, Albuquerque, NM, 2008

[7] Barthelmie, R. J., Hansen, K., Frandsen, S. T., Rathmann, O., Schepers, J. G., Schlez, W., Phillips, J., Rados, K., Zervos, A., Politis, E. S., and Chaviaropoulos, P. K., "Modeling and Measuring Flow and Wind Turbine Wakes in Large Wind Farms Offshore," *Wind Energy*, Vol. 12, No. 5, p. 431, 2010.

[8] Mickey, J., "ERCOT Wind Experience," International Workshop on Wind Energy Develpoment, Cairo Egypt, 2010.

[9] Hyer, M. W., *Stress Analysis of Fiber-Reinforced Composite Materials*, DEStech Publications, Lancaster PA, 2009.

[10] Herakovich, C. T., *Mechanics of Fibrous Composites*, Wiley, New York, 1998.

[11] Istvan, T., *Failure Modes in Composites*, American Institute of Mining, Metallurgical, and Petroleum Engineers, New York, 1973.

[12] Kessler, S., "Piezoelectric Based In-Situ Damage Detection of Composite Materials for Structural Health Monitoring Systems," Ph.D. Dissertation, Massachusetts Institute of Technology, MA, 2002.

[13] Lee, J., Takatsubo, J., and Tsuda, H., "Simultaneous Multipoint Acousto-Ultrasonic Sensing Based on Fiber Bragg Grating Sensors," National Institute of Advanced Industrial Science and Technology, Japan, *2007 SPIE Proceedings*, Vol. 6532, Sect. 3.

[14] Nothnagel, M. J., Park, G., and Farrar, C. R., "Wireless Energy Transmission for Structural Health Monitoring Embedded Sensor Nodes," The Engineering Institute, Los Alamos National Laboratory, Los Alamos, NM, *2007 SPIE Proceedings*, Vol. 6532, Sect. 9.

[15] Gao, H., "Ultrasonic Sensor Placement Optimization in Structural Health Monitoring Using CMA Evolutionary Strategy," CE 563 Project report, Pennsylvania State University, PA.

[16] Krautkramer, J., and Krautkramer, H., *Ultrasonic Testing of Materials*, 4th ed., Springer-Verlag, New York City, NY, 1990.

[17] Rose, J. L., *Ultrasonic Waves in Solid Media*, Cambridge University Press, New York, NY, 1999.

[18] Kundu, T., *Ultrasonic Nondestructive Evaluation*, CRC Press Publishers, Boca Raton, FL, 2004.

[19] Fausett, L., *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*, Prentice Hall, Upper Saddle River, NJ, 1994.

[20] Kennedy, J., and Eberhart ,R.C., *Swarm Intelligence*, Morgan Kaufmann Publishers, San Diego, CA, 2001.

[21] Kral, Z., Development of an Artificila Neural Network Damage Detection Module for a Structural Health Monitoring System, M.S. Thesis, Wichita State University, Wichita, KS 2009.

[22] Kral, Z., Horn, W., and Steck, J., "Composite Panel Damage Detection Using Ultrasonic Testing and Neural Networks," 2009 SEM Annual Conference & Exposition on Experimental and Applied Mechanics, Albuquerque, NM, June 2009

[23] Smith, B. L., *Mechanics of Damage Tolerance*, Class note packet, Wichita State University, Wichita, KS, 2008.

[24] Barsom, J.M., and Rolfe, S.T., *Applications of Fracture Mechanics*, 3rd ed., Prentice Hall, West Conshohocken, PA, 1999.

[25] Saxena, A., *Nonlinear Fracture Mechanics for Engineers*, CRC Press, Boca Raton, FL, 1998.

[26] Strong, A. B., *Fundamentals of Composite Manufacturing: Materials, Methods, and Applications*, Society of Manufacturing Engineers, Dearborn, MI, 2008.

[27] "Market Research Report: Strategic Business Expansion of Carbon Fiber, Torayca" *Toray* Industries press release, Toray Industries, 2005, retrieved July 2007.

[28] Grande, J. A., "Wind Power Blades Energize Composites Manufacturing," Plastics Technology, Gardner Publications, Inc., 2009, URL: www.ptonline.com/articles/200810fa2.html, retrieved October 2010.

[29] Hedgepath, J. M., and Van Dyke, P., "Stress Concentrations in Filamentary Structures," NASA Langley Research Center, 1961, NASA TN D-882.

[30] Batdorf, S.B., "Tensile Strength of Unidirectionally Reinforced Composites – I," *Journal of Reinforced Plastics and Composites*, Vol. 1, pp153, 1982

[31] Batdorf, S.B., "Tensile Strength of Unidirectionally Reinforced Composites – II," *Journal of Reinforced Plastics and Composites*, Vol. 1, pp. 165, 1982.

[32] Fukuda, H., and Chou, T. W., "An Advanced Shear-Lag Model Applicable to Discontinuous Fiber Composites," *Journal of Composite Materials*, Vol. 15, pp. 79, 1981.

[33] Overbey, L. A., and Todd, M. D., "Damage Identification through Generalized Correlations between Measurements," UCSD, La Jolla, CA, *2008 SPIE Proceedings*, Vol. 6532, Sect. 7.

[34] Tikka, J., Hedman, R., and Silijander, A., "Strain Gauge Capabilities in Crack Detection," 4th International Workshop on Structural Health Monitoring, Stanford, CA, 2003.

[35] Hill, K.O., "Photosensitivity in Optical Fiber Waveguides: Application to Reflection Fiber Fabrication," *Appl. Phys. Lett.*, 1978.

[36] Liu, C., Chou, G.W., Liang, X., Reinhall, P., and Wang, W., "Design of a Multi-Layered Optical Bend Loss Sensor for Pressure and Shear Sensing," National Defense University, Taiwan, *2008 SPIE Proceedings*, Vol. 6532, Sect. 13.

[37] Edwards, A.T., "Comparison of Strain Gage and Fiber Optic Sensors on a String Balance in a Supersonic Wind Tunnel, M.S. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2000.

[38] Ren, L., Song, G., Conditt, M., Noble, P., and Li, H., "Design and Performance of a Fiber Bragg Grating Displacement Sensor for Movement Measurement of Tendon and Ligament," Lianoning, China, *2008 SPIE Proceedings*, Vol. 6532, Sect. 13.

[39] Bhargava, A., Shivakumar, K., and Emmanwori, L., "Stress Concentration and Failure in Composite Laminates Embedded with Fiber Optic Sensor," North Carolina A&T State University, Greensboro, NC, 2003, AIAA 2003-1848.

[40] Voigt, W., *Lehrbuch der Kristallphysik (mit Ausschluss der Kristalloptik)*, B. G. Teubner, Leipzig and Berlin, Germany, 1910.

[41] Physical Acoustics Corporation, "PCI-2 Based AE System User's Manual, Rev. 3," Equipment and Software Package, Princeton Junction, NJ, 2007.

[42] Acellent Technologies, Inc., "ACESS 2.1 User's Manual," Equipment and Software Package, Sunnyvale, CA, 2007.

[43] Cowley County Community College, Iowa State University, Northeast Iowa Community College, Ridgewater College, Salt Lake Community College, and Southeast Community College, NDT Resource Center, 2008, URL: www.ndt-ed.org, retrieved January 2009.

[44] Kral, Z., Horn, W., and Steck, J., "Composite Panel Damage Detection using Ultrasonic Testing and Neural Networks," 2009 SEM Annual Conference & Exposition on Experimental and Applied Mechanics, Albuquerque, NM, 2009.

[45] Datta, S. K., and Shah, A. H., *Elastic Waves in Composite Media and Structures: With Applications to Ultrasonic Nondestructive Evaluation*, CRC Press, Baton Rouge, LA, 2009.

[46] Chen B., "Artificial Immune Pattern Recognition for Damage Detection in Structural Health Monitoring Sensor Networks," 2009 SPIE Smart Structures and Materials/Nondestructive Evaluation for Health Monitoring and Diagnostics, San Diego, CA 2009

[47] Weiss, G. (editor), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, Cambridge, MA, 1999.

[48] Norbert, G., *Conceptual Modeling of Multi-Agent Systems: The CoMoMAS Engineering Environment*, Kluwer Academic Publishers, Boston, MA, 2002.

[49] Demazeau, Y., and Müller ,J. P., *Decentralized A.I.*: *Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Cambridge*, England, August 16–18, 1989.

[50] Sichman, J. S., Demazeau, Y., and Boissier, O., "When Can Knowledge-Based Systems be Called Agents?" *Proceedings of Simpósio Brasileiro De Inteligência Artificial*, Rio, Vol. 9, 1992.

[51] Chaudron, L., Erceau, J., and Trousse, B., "Cooperative Decisions and Actions in Multi-Agent Worlds," International Conference on Systems, Man and Cybernetics 1993, Le Touquet, France, October 1993.

[52] Engelbrecht, A.P., *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons Inc., Sussex, England, 2005.

[53] Boeing Company, *Boeing: Commercial Airplanes – Commercial Aviation Services – Maintenance Services – Airplane Health Management*, Seattle, WA, 2008, URL: www.boeing.com/commercial/ams/mss/brochures/airplane_health.html, retrieved September 2008.

[54] Murri, G., "Effect of Embedded Piezoelectric Sensors on Fracture Toughness and Fatigue Resistance of Composite Laminates under Mode I Loading," American Society of Composites 21st Annual Technical Conference, NASA Langley Research Center, Hampton, VA, 2006.

[55] Kirikera, G. R., Lee, J. W., Ghoshal, A., et al., "Initial Evaluation of an Active/Passive Structural Neural System for Health Monitoring of Composite Materials," *Smart Materials and Structures*, Vol. 15, 2006, pp. 1275–1286.

[56] Lee, J. W., Kirikera, G. R., Kang, I., Schulz, M., and Shanov, V., "Structural Health Monitoring Using Continuous Sensors and Neural Network Analysis," *Smart Materials and Structures*, Vol. 15, 2006.

[57] Crupi, V., Guglielmino, E., and Milazzo, G., "Neural-Network-Based System for Novel Fault Detection in Rotating Machinery," *Journal of Vibration and Control*, Vol. 10, No. 8, pp.1137–1150, August 2004.

[58] Zapico, J. L., Worden, K., and Molina, F. J., "Vibration-Based Damage Assessment in Steel Frames Using Neural Networks," *Smart Materials and Structures*, Vol. 10, 2001, pp. 553–559.

[59] Reismann ,H., and Pawlik, P.S., *Elasticity: Theory and Applications*, Krieger Publishing Company, Malabar, FL, 1991.

[60] Wijngaards, N., Overeinder, B., van Steen, M., and Brazier, F., "Supporting Internet-Scale Multi-Agent Systems," *Data & Knowledge Engineering*, Vol. 41, No. 2-3, June 2002, pp. 229–245.

[61] Shukri, S. R., and Mohd Shaukhi, M. K., "A Cooperative Multi-Agent System and Its Real Time Application to Robot Soccer," *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, NM, 1997.

[62] The MathWorks, Inc., "MATLAB v6.5.0.180913a Release 13 Manual," Software Package, 2002.

[63] NeuralWare, Inc., "NeuralWorks Professional II/PLUS ™ User's Manual," Software Package, 1997.

[64] Viscovery Software GmbH, "Viscovery SOMine 5.0 Manual," Software Package, Vienna, Austria, 2007.

APPENDICES

APPENDIX A

FURTHER READINGS

[65] Al-Enezi, J. R., Abbod, M. F., and Alsharhan, S., "Artificial Immune Systems – Models, Algorithms and Applications," *International Journal of Research and Reviews in Applied Sciences*, Vol. 3, No. 2, 2010.

[66] Automation Creations Inc., "Online Materials Information Resource – MatWeb," www.matweb.com, 2009.

[67] Beer, F. P., Johnson, E. R., and DeWolf, J. T., *Mechanics of Materials, 3rd edition*, McGraw-Hill Companies Inc., New York, 2001.

[68] Blackstock, D., *Fundamentals of Physical Acoustics*, Wiley-Interscience Publication, Austin, TX, 2000.

[69] Callister, W. D., *Materials Science and Engineering as Introduction*, 6th ed., Wiley-Interscience, New York City, NY, 2003.

[70] Chapman, S. J., *MATLAB® Programming for Engineer, 2nd edition*, Brooks/Cole, Pacific Grove, CA, 2002.

[71] Chang, F. H., Couchman, J. C., and Yee, B. G. W., "Ultrasonic Resonance Measurements of Sound Velocity in Thin Composite Laminates," *Journal of Composite Materials*, Vol. 8, No. 4, pp.356–363, 1974.

[72] Chen, S., Razzaqi, S., and Lupien, V., "Towards the Automated Design of Phased Array Ultrasonic Transducers – Using Particle Swarms to Find 'Smart' Start Points," *New Trends in Applied Artificial Intelligence*, Springer/Berlin/Heidelberg, 2007.

[73] Craig, R. R., and Kurdila, A. J., *Fundamentals of Structural Dynamics, 2nd edition*, John Wiley & Sons Inc., Hoboken, NJ, 2006.

[74] Devenport, W. J., *AOE 3054: Experiment 5 – Fracture Toughness Testing*, class notes, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2004.

[75] Elgun, S. Z., "Cold Work," Farmingdale State College, Farmingdale, NY, 1999 URL: info.lu.farmingdale.edu/depts/met/met205/coldwork.html [Retrieved: Feb 2009].

[76] Farmer, J. D., "Rosetta Stone for Connectionism," Ninth annual international conference of the Center for Nonlinear Studies on Self-organizing, Collective, and Cooperative Phenomena in Natural and Artificial Computing Networks on Emergent computation, Los Alamos, NM, 1990.

[77] Feres, P., "The Effectiveness of Flaw Detection Caused by Cracking Using Acoustic Emission Technique," European Conference on Nondestructive Testing, Berlin, Germany, 2006.

[78] Hallauer, W. L., *AOE 3034 Course Book: Vehicle Vibration and Control – Introduction to Linear, Time-Invariant, Dynamic Systems*, A-1 Copies, Blacksburg, VA, 2004.

[79] Hansen, N., "Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation," 0-7803-2902-3/96 IEEE, 1996.

[80] Harris, J. G., *Linear Elastic Waves*, Cambridge University Press, New York, NY, 2001.

[81] Haykin S., and Kosko B., *Intelligent Signal Processing*, IEEE Press, New York, 2001.

[82] Jourdan, D. B., and Roy, N., "Optimal Sensor Placement for Agent Localization," IEEE/ION Position, Location and Navigation Symposium (PLANS 2006), San Diego, CA, April 2006.

[83] Junjie, P., and Dingwei, W., "An Ant Colony Optimization Algorithm for Multiple Travelling Salesman Problem," First International Conference on Innovative Computing, Information and Control, Beijing, China, 2006.

[84] Kampe, S. L., *Materials in Aerospace and Oceanic Systems*, class notes, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2004.

[85] Konovalov, N. N., "Estimation of the Reliability Coefficients of Ultrasonic Testing of Welded Joints," *Russian Journal of Nondestructive Testing*, Vol. 39, No. 9, p.649, 2003.

[86] Lee, E. H., and Yang, W. H., "On Waves in Composite Materials with Periodic Structure," *SIAM Journal on Applied Mathematics*, Vol. 25, No. 3, p.492, 1973.

[87] Lissenden, C. J., Blackshire, J. L., and Puthillath, P.K., "Structural Health Monitoring of Composite Laminates with Embedded Piezoelectric Fibers," *Review of Quantitative Nondestructive Evaluation*, Vol. 28, pp.974–981, 2009.

[88] Liu, W., "Multiple Wave Scattering and Calculated Effective Stiffness and Wave Properties in Unidirectional Fiber-Reinforced Composites," Ph.D. Dissertation, Virginia Polytechnic Institute and Stae University, Blacksburg, VA, 1997.

[89] Mendenhall, W., and Sincich, T., *Statistics for Engineering and the Sciences, 5th edition*, Prentice Hall Inc., Upper Saddle River, NJ, 2007.

[90] Montgomery, D. C., *Design and Analysis of Experiments, 7th edition*, John Wiley & Sons Inc., Hoboken, NJ, 2009.

[91] Nisan, N., Roughgarden, T., Tardos, É., and Vazirani, V. V., *Algorithmic Game Theory*, Cambridge University Press, Cambridge, MA, 2007.

[92] Miller, D., "Corrosion Control on Aging Aircraft: What is being done?" *Materials Performance*, pp.10–11, October 1990.

[93] Placko, D., and Kundu, T., *DPSM for Modeling Engineering Problems*, Wiley-Interscience, Hoboken, New Jersey, 2007.

[94] Reddy, J. N., *Theory and Analysis of Elastic Plates and Shells, 2nd edition*, CRC Press, Baton Rouge, LA, 2007.

[95] Reddy, J. N., An *Introduction to the Finite Element Method, 3rd edition*, McGraw-Hill Companies Inc., New York, NY, 2004.

[96] Rhian, G. E., "Acoustic Emission in Composite Laminates," *Journal of Nondestructive Evaluation*, Vol. 17 No. 3, 1998.

[97] Rose, J. L., Wang, A. S. D., and Deska, E. W., "Wave Profile Analysis in a Unidirectional Graphite-Epoxy Plate," *Journal of Composite Materials*, Vol. 8, p.419, 1974.

[98] Rummel, W. D., "Probability of Detection as a Quantitative Measure of Nondestructive Testing End-To-End Process Capabilities," American Society for Nondestructive Testing, 1998, URL: www.asnt.org/publications/Materialseval/basics/jan98basics/jan98basics.htm [Retrieved Mar 2010]

[99] Rummel, W. D., Todd, P., Rathke, R. A., and Castner, W. L., "The Detection of Fatigue Cracks by Nondestructive Testing Methods," *Materials Evaluation*, Vol. 32, No. 10, p.205, 1974.

[100] Schmerr, L. W., *Fundamentals of Ultrasonic Nondestructive Evaluation*, Plenum Press, New York, NY, 1998.

[101] Schmid, J., *Performance of European Wind Turbines: A Statistical Evaluation from the European Wind Turbine Database EUROWIN*, Elsevier Applied Science, New York, 1991.

[102] Smith, B. L., and Keshavanarayana, S. R., *Mechanics of Laminated Composites*, Class note packet, Wichita State University, Wichita, KS, 2009.

[103] Spera, D. A., *Wind Turbine Technology: Fundamental Concepts of Wind Turbine Engineering*, ASME Press, New York, 2009.

[104] Stat-Ease, Inc., "Design Expert, Version 8.0.2," Software Package, 2010.

[105] Sumathi, S., and Paneerselvam, S., *Computational Intelligence Paradigms: Theory and Applications Using MATLAB*, CRC Press, Baton Rouge, LA, 2010.

[106] Thomson, W. T., and Dahleh, M. D., *Theory of Vibration with Applications*, Prentice Hall Inc., Upper Saddle River, NJ, 1998.

[107] Vallen Systeme GmbH, "AGU-Vallen Wavelet R2008.0915," Software Package, Tokyo, Japan, 2009 URL: www.vallen.de/wavelet/ [Retrieved: Mar 2009].

[108] Von Neumann, J., and Morgenstern, O., *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ, 1953.

[109] Winter, M., "WebElements: the periodic table on the WWW, " University of Sheffield and WebElements Ltd, UK, 1998-2008 URL: www.webelements.com [Retrieved: Feb 2009].

APPENDIX B

MATLAB CODES

B.1 Laminate Composites Properties

The following code utilizes classical laminar plate theory to obtain the general stresses and strains within a laminate composite. The properties must be adjusted within the first section of the code to satisfy the condition requirements.

```
function composites_main_2

% composites_main
%
% Written by: Zachary Kral
% Date: Spring 2010
%
% Combines sub-functions together to form composites analysis program. Go
% to setup section below to provide inputs for data analysis.
%
% Files to be read: Properties text file - matrix with each column as
%                                          column as a material I.D.
%
%
% Functions used: ply_format.m
%                 prop_to_Q.m
%                 trans.m
%                 uf_solve.m
%
%

%--------------------------------------------------------------------------
%--------------------------------------------------------------------------
%
% - SETUP COMPOSITE -
%

% Path to write output files and plots
cd C:\'Documents and Settings'\user06\Desktop\'WSU Spring 10'\'AE 853 - Advanced Composites'\'Final Exam'

% Location of subfunctions
addpath C:\'Documents and Settings'\user06\Desktop\'MATLAB Common Programs'\Composites

%---------
% - Tolerance -
% Anything less than this number will be considered a zero value in any
% calculations.
tol = 1e-10;

%---------
% - Material Properties and output file -
% Specify text file name for material lists. Put materials into column form
% for material.
% NOTE: '.txt' is optional to include
filename_prop = 'comp_prop';
filename_out = 'p3 laminate';

strain_strain_plot = 'Example stress-strain';
                                % Do NOT include a suffix. This name is
                                % for plots to be saved. Two plots are
                                % created and have GLOBAL or LOCAL added
                                % to the end to distinguish between the
                                % two co-ordinate systems.
interlaminar_plot = 'Example interlam';
                                % Do NOT include a suffix.

%---------
% - Ply configuration -
% Use compressed version of ply layout and use ply_format to extend to long
% format.
ply = [45,-45,0,90,45,-45,0,45,-45];     % (degrees) in Interger values
rep = [1,1,3,1,1,1,2,1,1];          % Number of times the plies repeat
iteration   =   3;                              %   Number   of   times   the   stacking   sequence   repeats
```

```
ply_mat = [1,1,1,1,1,1,1,1,1];     % Material I.D. number. Must be same length as ply
                                       % (or adjusted below)
ply_t = 0.01*[1,1,1,1,1,1,1,1,1];     % Ply thickness. Must be the same length as ply
                                       % (or adjusted below)
ply = ply_format(ply,'s',rep,iteration);
ply_mat = ply_format(ply_mat,'s',rep,iteration);
ply_t = ply_format(ply_t,'s',rep,iteration);
      % INPUT: ply count (condensed)
      %        's' for symmetric/'so' for odd symmetric/'a' for asymmetric
      %        [i,i,i,...] for repition of plies
      %        j for repitition of entire set
      %
      % NOTE: Keep the three above the same for symmetric laminates.

points_per_ply = 4; % Number of points to be calculated for stress and
                    % strain values



%---------
% - Applied loads -
% Use given values of load or displacemnt to determine the remainder.
% NOTE: Be careful of UNITS. Make sure they match units of material
%       properties.
NM = [0, 0, 0, 20, 0, 0]';
strain = zeros(1,6);
NM_binom = ones(1,6);
strain_binom = strain;
      % binom parts are used to specify known and unknown variables
      % 0 = unknown
      % 1 = known

%
% - Free-edge parameters -

width = 2;     % width of plate. To be used if only normal force is applied
               %  (to determine the free edge stresses)
free_edge_dist = 0; % specific distance from free edge to calculate the six
                    % components of stress in GLOBAL system
                    % NOTE: needs to be less than width/2
%
% - Moment applied for beam problems -

Qx = -6.5;    % Force in Qx direction
dQx_dx = 1; % Derivative of Qx term

%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%
% - SETUP FILES AND PATHS -
%

% Used for sub-functions
addpath C:\'Documents and Settings'\user06\Desktop\'MATLAB Common Programs'

% Setup file to have output data
try
    file = fopen([filename_out,'.txt'],'wt');
catch                                                                %#ok<CTCH>
    file = fopen(filename_out,'wt');
end
fprintf(file,'Analysis and Output Verification of Steps\n\n');

%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%
% - Q(Local) AND LAYUP -
%

try
    prop = load([filename_prop,'.txt']);
catch                                                                %#ok<CTCH>
    prop = load(filename_prop);
end

%---------
% Assemble local Q matrices based on material
Q = zeros(3,3,length(prop(1,:)));  % [ Q ,Material I.D. ]
for i = 1:length(prop(1,:))
```

```
    Q_temp = prop_to_Q(prop(:,i));
    Q(:,:,i) = Q_temp;
end

%---------
% Print out to verify Q(local) and ply layout
fprintf(file,'\nMATID\t   Q11    \t   Q12    \t   Q22    \t   Q66    \n');
for i = 1:length(Q(1,1,:))
    fprintf(file,'%3d  \t%10.3e\t%10.3e\t%10.3e\t%10.3e\n\n\n',i,...
        Q(1,1,i),Q(1,2,i),Q(2,2,i),Q(3,3,i));
end

fprintf(file,'LAYER No.\tANGLE\tMATERIAL I.D.\tTHICKNESS\n');
for i = 1:length(ply);
    fprintf(file,'%6d   \t%4d \t%8d     \t%6.3f\n',i,ply(i),ply_mat(i),...
        ply_t(i));
end
fprintf(file,'\n\n');

fprintf(file,'\n-------------------------------------------------------');
fprintf(file,'-------------------------------------------------------\n\n');
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%
% - Q_BAR AND ABD MATRIX -
%

%---------
% Assemble Q_bars for each layer
Q_bar = zeros(3,3,length(ply)); % [ Q_bar , ply ]
for i = 1:length(ply)
    [T,T_hat] = trans(ply(i));
    Q_bar(:,:,i) = T^(-1)*Q(:,:,ply_mat(i))*T_hat;
end

% Print out to verify Q_bar's
fprintf(file,'\nQBAR-MATRIX ELEMENTS FOR EACH LAYER\n');
fprintf(file,'LAYER No.\t  QBAR(1,1)\t  QBAR(1,2)\t  QBAR(1,6)\t  ');
fprintf(file,'QBAR(2,2)\t  QBAR(2,6)\t  QBAR(6,6)\n');
for i = 1:length(ply)
    fprintf(file,'%6d   \t%+10.3e\t%+10.3e\t%+10.3e\t%+10.3e\t',i,...
        Q_bar(1,1,i),Q_bar(1,2,i),Q_bar(1,3,i),Q_bar(2,2,i));
    fprintf(file,'%+10.3e\t%+10.3e\n',Q_bar(2,3,i),Q_bar(3,3,i));
end
fprintf(file,'\n\n');

%---------
% Assemble A, B, and D matrices
A = zeros(3);
B = zeros(3);
D = zeros(3);
h_global = zeros(length(ply)+1,1);

h(1) = -sum(ply_t)/2;   % Find h0
h(2) = h(1) + ply_t(1); % Find h1
h_global(1) = h(1);     % Compile set of h0 to hN
p = 2;
% Determine summations for A B & D matrices
for k = 1:length(ply)
    for i = 1:3
        for j = 1:3
            A(i,j) = A(i,j) + Q_bar(i,j,k)*(h(2)-h(1));
            B(i,j) = B(i,j) + Q_bar(i,j,k)*(h(2)^2-h(1)^2);
            D(i,j) = D(i,j) + Q_bar(i,j,k)*(h(2)^3-h(1)^3);
        end
    end
    if k < length(ply)
        h(1) = h(2);
        h(2) = h(1) + ply_t(k+1);
        h_global(p) = h(1);
        p = p+1;
    else
        h_global(p) = h(2);
    end
end
B = B./2;
D = D./3;

% Reduce for computer error
```

```
for i = 1:3
    for j = 1:3
        if A(i,j) < tol
            A(i,j) = 0;
        end
        if B(i,j) < tol
            B(i,j) = 0;
        end
        if D(i,j) < tol
            D(i,j) = 0;
        end
    end
end

% Assemble larger matrix
ABD = [A,B;
       B,D];

% Print out to verify A, B, and D matrices
for k = 1:3
    switch k
        case 1
            fprintf(file,'\nA-MATRIX\n');
            ABD_temp = A;
        case 2
            fprintf(file,'\nB-MATRIX\n');
            ABD_temp = B;
        case 3
            fprintf(file,'\nD-MATRIX\n');
            ABD_temp = D;
    end
    for i = 1:3
        fprintf(file,'\t[\t');
        for j = 1:2
            fprintf(file,'%+12.4e\t',ABD_temp(i,j));
        end
        fprintf(file,'%+12.4e\t]\n',ABD_temp(i,j+1));
    end
    fprintf(file,'\n\n');
end

fprintf(file,'\n------------------------------------------------------');
fprintf(file,'------------------------------------------------------\n\n');
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%
% - SOLVE FOR LOADS AND MID-PLANE STRAINS -
%

[strain_0,NM] = uf_solve(ABD,strain,NM,strain_binom,NM_binom);

% Print out to verify Load and mid-plane strains
fprintf(file,'\nLOAD VECTOR\n');
fprintf(file,'\t    Nxx \t    Nyy \t    Nxy \t    Mxx \t    Myy \t    ');
fprintf(file,'Mxy\n\t');
for i = 1:(length(NM)-1)
    fprintf(file,'%+8.2f\t',NM(i));
end
fprintf(file,'%+8.2f\n\n\n',NM(i));

fprintf(file,'\nMID-PLANE STRAINS & CURVATURES\n');
fprintf(file,'\t   exxo \t   eyyo \t   exyo \t   kxxo \t   kyyo \t   ');
fprintf(file,'kxyo\n\t');
for i = 1:(length(strain_0)-1)
    fprintf(file,'%+11.3e\t',strain_0(i));
end
fprintf(file,'%+11.3e\n\n\n',strain_0(i));
fprintf(file,'\n------------------------------------------------------');
fprintf(file,'------------------------------------------------------\n\n');
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%
% - SOLVE FOR GLOBAL STRESS AND STRAIN -
%

% Setup output array
clear h
p = 1;
for i = 1:(length(h_global)-1)
```

```
    h(p:(p+points_per_ply-1)) = h_global(i):ply_t(i)/(points_per_ply-1):h_global(i+1);
    p = p+4;
end

%---------
% Mid-plane strain to GLOBAL strain
strain_xy = zeros(4,length(h));
p = 1;
for i = 1:length(ply)
    for j = 0:(points_per_ply-1)
        strain_xy(1:3,p+j) = strain_0(1:3) + h(p+j)*strain_0(4:6);
        strain_xy(4,p+j) = h(p+j); % Keep track of distance, z
    end
    p = p+points_per_ply;
end

%---------
% GLOBAL strain to GLOBAL stress
stress_xy = zeros(size(strain_xy));
p = 1;
stress_xy(1:3,1) = Q_bar(:,:,p)*strain_xy(1:3,1);
stress_xy(4,:) = strain_xy(4,:); % Keep track of distance, z
for i = 2:length(strain_xy(1,:))
    if strain_xy(4,i) == strain_xy(4,i-1);
        p = p+1;
    end
    stress_xy(1:3,i) = Q_bar(:,:,p)*strain_xy(1:3,i);
end

%---------
% Print out to verify GLOBAL strains and stresses
fprintf(file,'\nGLOBAL STRAIN & STRESS DISTRIBUTIONS ACROSS LAMINATE ');
fprintf(file,'THICKNESS\n');
fprintf(file,'Numer of points per ply = %d\n',points_per_ply);
fprintf(file,'Units - Strain:in/in  Stress:psi\n\n');

fprintf(file,'     z  \t   exx(z)  \t   eyy(z)  \t   exy(z)  \t   ');
fprintf(file,'Sxx(z)  \t   Syy(z)  \t   Sxy(z)\n');

fprintf(file,'%+8.4f\t',strain_xy(4,1));
% Strains
for j = 1:3
    fprintf(file,'%+8.3e\t',strain_xy(j,1));
end
% Stresses
for j = 1:3
    fprintf(file,'%+8.3e\t',stress_xy(j,1));
end
fprintf(file,'\n');
for i = 2:length(strain_xy(1,:))
    if strain_xy(4,i) == strain_xy(4,i-1)
        fprintf(file,'- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - ');
        fprintf(file,'- - - - - - - - - - - - - - - - - - - - - - - - - - - - - \n');
    end
    fprintf(file,'%+8.4f\t',strain_xy(4,i));
    % Strains
    for j = 1:3
        fprintf(file,'%+8.3e\t',strain_xy(j,i));
    end
    % Stresses
    for j = 1:3
        fprintf(file,'%+8.3e\t',stress_xy(j,i));
    end
    fprintf(file,'\n');
end
fprintf(file,'\n\n');


fprintf(file,'\n-------------------------------------------------------');
fprintf(file,'-------------------------------------------------------\n\n');
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%
% - SOLVE FOR LOCAL STRESS AND STRAIN -
%

%---------
% GLOBAL strain to LOCAL strain
strain_12 = zeros(size(strain_xy));
```

```
p = 1;
[T,T_hat] = trans(ply(1));                                                  %#ok<*NASGU>
strain_12(1:3,1) = T*strain_xy(1:3,1);
strain_12(4,:) = strain_xy(4,:); % Keep track of distance, z
for i = 2:length(strain_xy(1,:))
    if strain_xy(4,i) == strain_xy(4,i-1);
        p = p+1;
    end
    [T,T_hat] = trans(ply(p));
     strain_12(1:3,i) = T*strain_xy(1:3,i);
end

%---------
% LOCAL strain to LOCAL stress
stress_12 = zeros(size(strain_xy));
p = 1;
stress_12(1:3,1) = Q(:,:,ply_mat(1))*strain_12(1:3,1);
stress_12(4,:) = strain_12(4,:); % Keep track of distance, z
for i = 2:length(strain_xy(1,:))
    if strain_xy(4,i) == strain_xy(4,i-1);
        p = p+1;
    end
    stress_12(1:3,i) = Q(:,:,ply_mat(p))*strain_12(1:3,i);
end

%---------
% Print out to verify LOCAL strains and stresses
fprintf(file,'\nSTRAIN & STRESS DISTRIBUTIONS ACROSS LAMINATE THICKNESS');
fprintf(file,'\nIN LOCAL CO-ORDINATES\n');
fprintf(file,'Numer of points per ply = %d\n',points_per_ply);
fprintf(file,'Units - Strain:in/in  Stress:psi\n\n');

fprintf(file,'     z  \t   e11(z)  \t   e22(z)  \t   e12(z)  \t   ');
fprintf(file,'S11(z)  \t   S22(z)  \t   S12(z)\n');

fprintf(file,'%+8.4f\t',strain_12(4,1));
% Strains
for j = 1:3
    fprintf(file,'%+8.3e\t',strain_12(j,1));
end
% Stresses
for j = 1:3
    fprintf(file,'%+8.3e\t',stress_12(j,1));
end
fprintf(file,'\n');
for i = 2:length(strain_12(1,:))
    if strain_12(4,i) == strain_12(4,i-1)
        fprintf(file,'- - - - - - - - - - - - - - - - - - - - - - - - - - - - - ');
        fprintf(file,'- - - - - - - - - - - - - - - - - - - - - - - - - - - - \n');
    end
    fprintf(file,'%+8.4f\t',strain_12(4,i));
    % Strains
    for j = 1:3
        fprintf(file,'%+8.3e\t',strain_12(j,i));
    end
    % Stresses
    for j = 1:3
        fprintf(file,'%+8.3e\t',stress_12(j,i));
    end
    fprintf(file,'\n');
end
fprintf(file,'\n\n');
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%
% - CREATE PLOTS OF GLOBAL AND LOCAL STRESS AND STRAIN DISTRIBUTIONS -
%

for i = 1:4
    switch i
        case 1
            temp = strain_xy;
            name = 'Global Strain_{xy}';
            xname = 'Global Strain \epsilon';
            count = 4:6;
            p = 1;
        case 2
            temp = stress_xy;
            name = 'Global Stress_{xy}';
```

```
            xname = 'Global Stress \sigma';
            count = 1:3;
        case 3
            temp = strain_12;
            name = 'Local Strain_{12}';
            xname = 'Local Strain \epsilon';
            count = 4:6;
            p = 2;
        case 4
            temp = stress_12;
            name = 'Local Stress_{12}';
            xname = 'Local Stress \sigma';
            count = 1:3;
    end
    figure (p)
    for j = 1:3
        subplot(2,3,count(j));
        plot(temp(j,:),temp(4,:),'bx-');

        % Label plots correctly
        k = count(j);
        switch k
            case 1
                if p == 1
                    h1 = xlabel([xname,'_{xx} (psi)']);
                else
                    h1 = xlabel([xname,'_{11} (psi)']);
                end
            case 2
                if p == 1
                    h1 = xlabel([xname,'_{yy} (psi)']);
                else
                    h1 = xlabel([xname,'_{22} (psi)']);
                end
            case 3
                if p == 1
                    h1 = xlabel([xname,'_{xy} (psi)']);
                else
                    h1 = xlabel([xname,'_{12} (psi)']);
                end
            case 4
                if p == 1
                    h1 = xlabel([xname,'_{xx} (in/in)']);
                else
                    h1 = xlabel([xname,'_{11} (in/in)']);
                end
            case 5
                if p == 1
                    h1 = xlabel([xname,'_{yy} (in/in)']);
                else
                    h1 = xlabel([xname,'_{22} (in/in)']);
                end
            case 6
                if p == 1
                    h1 = xlabel([xname,'_{xy} (in/in)']);
                else
                    h1 = xlabel([xname,'_{12} (in/in)']);
                end
        end
        if count(j) == 2 || count(j) == 5
            h3 = title(name);
        end
        h2 = ylabel('Distance from mid-surface (in)');

        % Format plots
        set(gca,'Fontsize',6);
        for j2 = 1:2
            eval(['set(h',num2str(j2),','''Fontsize''',6);']);
        end
    end
end

% Save plots as *.png images
for i = 1:2
    figure (i)
    set(gcf,'PaperUnits','inches','PaperPosition',[0 0 5 5]);
    if i == 1
        print('-dpng', [strain_strain_plot,'_GLOBAL.png'], '-zbuffer');
    else
```

```
        print('-dpng', [strain_strain_plot,'_LOCAL.png'], '-zbuffer');
    end
    close
end

fprintf(file,'\n-------------------------------------------------------');
fprintf(file,'-------------------------------------------------------\n\n');
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%
% - FREE EDGE STRESSES (IN-PLANE LOADS APPLIED)-
%

% Check for possible free edge conditions
if NM(1) > tol
    free_edge_perform = 'y';
    for i = 2:3 % Just look at in-plane running loads
        if NM(i) > tol
            free_edge_perform = 'n';
            break
        end
    end
else
    free_edge_perform = 'n';
end

h_thickness = h_global(length(h_global)) - h_global(1);

if free_edge_perform == 'y';

    sigma_0 = NM(1)/h_thickness;
    ply_count = length(ply);

    % Average global stress in each ply
    stress_xy_avg = zeros(4,length(ply));
    p = 1;
    for i = 1:ply_count
        for j = 1:3
            stress_xy_avg(j,i) = mean(stress_xy(j,p:(p+points_per_ply-1)));
%             strain_xy_avg(4,i) = h(p);
        end
        p = p+points_per_ply;
    end

    % q0 and p0 values
    q0 = zeros(ply_count,1);
    q1 = q0;
    p0 = zeros(ply_count,1);
    p1 = p0;
    p2 = p0;
    for i = 1:ply_count
        q0(i) = stress_xy_avg(3,i)/sigma_0;
        if abs(q0(i)) < tol
            q0(i) = 0;
        end
        p0(i) = stress_xy_avg(2,i)/sigma_0;
        if abs(p0(i)) < tol
            p0(i) = 0;
        end
    end

    % Setup for first ply
    q1(1) = -h_global(1)/h_thickness*q0(1);
    p1(1) = -h_global(1)/h_thickness*p0(1);
    p2(1) = -h_global(1)/h_thickness*(p1(1) + h_global(1)/...
        (2*h_thickness)*p0(1));

    % Remaining plies
    for k = 2:ply_count
        q1(k) = q1(k-1) + h_global(k-1)/h_thickness*(q0(k-1) - q0(k));
        p1(k) = (1/ply_count)*sum(p0(1:k-1)) - h_global(k)/...
            h_thickness*p0(k);
        p2(k) = p2(k-1) + (p1(k-1) - p1(k))*h_global(k)/h_thickness + ...
            (p0(k-1) - p0(k))*(h_global(k))^2/(2*h_thickness^2);
    end

    %---------
    % Print out to verify q0, q1, p0, p1, p2
    free_edge_constants = [q0, q1, p0, p1, p2];
```

```
fprintf(file,'\n\nPLY-LEVEL CONSTANTS\n');
fprintf(file,'PLY#\t    Q0\t    Q1\t    P0\t    P1\t    P2\n');
for i = 1:ply_count
    fprintf(file,'  %d\t',i);
    for j = 1:5
        fprintf(file,'%+8.5f\t',free_edge_constants(i,j));
    end
    fprintf(file,'\n');
end
fprintf(file,'\n\n');
fprintf(file,'\n-------------------------------------------------------');
fprintf(file,'-------------------------------------------------\n');
fprintf(file,'\n');

%---------
% Find stresses at SPECIFIED DISTANCE
R = 90/7;
C = 1/5;
b = width/2;

% Double the count of points for plotting purposes
z(1) = h(1);
p = 2;
for i = 2:length(h)
    if h(i) ~= h(i-1)
        z(p) = (h(i)+h(i-1))/2;
        p = p+1;
    end
    z(p) = h(i);
    p = p+1;
end
z = z';

p = 1;
for i = 1:length(z(:,1))
    if i > 1.5
        if z(i,1) == z(i-1)
            p = p+1;
        end
    end
    % Sigma_xx

    z(i,2) = stress_xy_avg(1,p);

    % Sigma_yy
    if free_edge_dist < h_thickness/3
        z(i,3) = 9*sigma_0*p0(p)/7*(free_edge_dist/h_thickness)^2* ...
            (5-6*free_edge_dist/h_thickness);
    elseif free_edge_dist < h_thickness
        z(i,3) = -sigma_0*p0(p)/7*(2-18*free_edge_dist/h_thickness ...
            + 9*(free_edge_dist/h_thickness)^2);
    else
        z(i,3) = sigma_0*p0(p);
    end

    % Sigma_xy
    if free_edge_dist < h_thickness
        z(i,4) = free_edge_dist/h_thickness*(2 - ...
            free_edge_dist/h_thickness)*sigma_0*q0(p);
    else
        z(i,4) = sigma_0*q0(p);
    end

    % Sigma_xz
    if free_edge_dist < h_thickness
        z(i,5) = 2*(free_edge_dist/h_thickness - 1)*(z(i,1)*q0(p) + ...
            q1(p)*h_thickness)*sigma_0;
    else
        z(i,5) = 0;
    end

    % Sigma_yz
    sig_yzM = -25/14*(p1(p) + z(i,1)/h_thickness*p0(p))*sigma_0;
    if free_edge_dist < h_thickness/3
        z(i,6) = 36/25*(free_edge_dist/h_thickness)*(5 - 9* ...
            free_edge_dist/h_thickness)*sig_yzM;
    elseif free_edge_dist < h_thickness
        z(i,6) = 36/25*(1 - free_edge_dist/h_thickness)*sig_yzM;
    else
```

```
        z(i,6) = 0;
    end

    % Sigma_zz
    sig_zzM = (p0(p)*(1/2)*(z(i,1)/h_thickness)^2 + p1(p)*z(i,1)/ ...
        h_thickness + p2(p))*R*sigma_0;
    if free_edge_dist < h_thickness/3
        z(i,7) = (5 - 18*free_edge_dist/h_thickness)*sig_zzM/5;
    elseif free_edge_dist < h_thickness
        z(i,7) = -sig_zzM/5;
    else
        z(i,7) = 0;
    end

end

%---------
% Print out to verify stresses
fprintf(file,'\n\nFREE EDGE STRESSES\n\n');
fprintf(file,'STRESSES AT Y =\t%8.2f ',free_edge_dist);
fprintf(file,'FROM THE FREE EDGE\n\n');
fprintf(file,'   z    \t  Sxx    \t  Syy     \t  Sxy     \t  Sxz    ');
fprintf(file,' \t  Syz     \t  Szz\n');
for i = 1:length(z(:,1))
    fprintf(file,'%8.5f\t',z(i,1));
    for j = 2:(length(z(1,:))-1)
        fprintf(file,'%+9.2f\t',z(i,j));
    end
    fprintf(file,'%+9.2f\n',z(i,j+1));
end
fprintf(file,'\n\n');
fprintf(file,'\n-------------------------------------------------------');
fprintf(file,'-------------------------------------------------------\n');
fprintf(file,'\n');

% Plot out results to verify stresses
color = 'brg';
shape = '+-x';
figure (3)
for i = 1:2
    subplot(1,2,i)
    for j = 2:4
        eval(sprintf('plot(z(:,j+(i-1)*3),z(:,1),''%s%s-'');',...
            color(j-1),shape(j-1)));
        hold on
    end
    h1 = xlabel(['Stress (psi) at point ',num2str(free_edge_dist)]);
    h2 = ylabel('Distance from mid-point, ''z'' (in)');
    switch i
        case 1
            h3 = legend('\sigma_{xx}','\sigma_{yy}','\sigma_{xy}');
            h4 = title('In-plane Stresses');
        case 2
            h3 = legend('\sigma_{xz}','\sigma_{yz}','\sigma_{zz}');
            h4 = title('Interlaminar Stresses');
    end
    % Format plots
    set(gca,'Fontsize',6);
    for j2 = 1:3
        eval(['set(h',num2str(j2),',''Fontsize'',6);']);
    end
    set(h4,'Fontsize',8);
end
set(gcf,'PaperUnits','inches','PaperPosition',[0 0 10 5]);
print('-dpng', [interlaminar_plot,'_FORCE-SPECIFIC_POINT-', ...
    num2str(free_edge_dist),'.png'], '-zbuffer');
close

%---------
% Find interlaminar stresses BETWEEN PLIES

y(:,1) = 0:b/1000:(h_thickness*(5/4));
for i2 = 2:(length(h_global)-1)
    h_local = h_global(i2);

    p = (i2-1);
    for i = 1:length(y(:,1))
        % Sigma_xx
```

```
    y(i,2) = stress_xy_avg(1,p);

    % Sigma_yy
    if y(i,1) < h_thickness/3
        y(i,3) = 9*sigma_0*p0(p)/7*(y(i,1)/h_thickness)^2* ...
            (5-6*y(i,1)/h_thickness);
    elseif y(i,1) < h_thickness
        y(i,3) = -sigma_0*p0(p)/7*(2-18*y(i,1)/h_thickness ...
            + 9*(y(i,1)/h_thickness)^2);
    else
        y(i,3) = sigma_0*p0(p);
    end

    % Sigma_xy
    if y(i,1) < h_thickness
        y(i,4) = y(i,1)/h_thickness*(2 - ...
            y(i,1)/h_thickness)*sigma_0*q0(p);
    else
        y(i,4) = sigma_0*q0(p);
    end

    % Sigma_xz
    if y(i,1) < h_thickness
        y(i,5) = 2*(y(i,1)/h_thickness - 1)*(h_local*q0(p) + ...
            q1(p)*h_thickness)*sigma_0;
    else
        y(i,5) = 0;
    end

    % Sigma_yz
    sig_yzM = -25/14*(p1(p) + h_local/h_thickness*p0(p))*sigma_0;
    if y(i,1) < h_thickness/3
        y(i,6) = 36/25*(y(i,1)/h_thickness)*(5 - 9* ...
            y(i,1)/h_thickness)*sig_yzM;
    elseif y(i,1) < h_thickness
        y(i,6) = 36/25*(1 - y(i,1)/h_thickness)*sig_yzM;
    else
        y(i,6) = 0;
    end

    % Sigma_zz
    sig_zzM = (p0(p)*(1/2)*(h_local/h_thickness)^2 + ...
        p1(p)*h_local/h_thickness + p2(p))*R*sigma_0;
    if y(i,1) < h_thickness/3
        y(i,7) = (5 - 18*y(i,1)/h_thickness)*sig_zzM/5;
    elseif y(i,1) < h_thickness
        y(i,7) = -sig_zzM/5;
    else
        y(i,7) = 0;
    end
end

%---------
% Print out to verify interlaminate stresses
fprintf(file,'\nINTERLAMINATE STRESSES AT INTERFACE OF PLIES');
fprintf(file,'\t%d\t&\t%d\n\n',i2-1,i2);
fprintf(file,'   y    \t  Sxx    \t  Syy     \t  Sxy     \t  Sxz');
fprintf(file,'     \t  Syz     \t  Szz\n');
for i = 1:length(y(:,1))
    fprintf(file,'%8.5f\t',y(i,1));
    for j = 2:(length(y(1,:))-1)
        fprintf(file,'%+9.2f\t',y(i,j));
    end
    fprintf(file,'%+9.2f\n',y(i,j+1));
end
fprintf(file,'\n');
if i2 < (length(h_global)-1)
    fprintf(file,'\n- - - - - - - - - - - - - - - - - - - - - - - - - ');
    fprintf(file,'- - - - - - - - - - - - - - - - - - - - - - - \n');
else
    fprintf(file,'\n\n');
    fprintf(file,'\n-------------------------------------------');
    fprintf(file,'-------------------------------------------\n');
    fprintf(file,'\n');
end

%---------
% Plot results to verify interlaminate stresses
figure (1)
```

```
        for j = 2:7
            subplot(2,3,(j-1))
            plot(y(:,1),y(:,j));
            h1 = xlabel('Distance from free edge (in)');
            switch j
                case 2
                    h2 = ylabel('\sigma_{xx} (psi)');
                case 3
                    h2 = ylabel('\sigma_{yy} (psi)');
                case 4
                    h2 = ylabel('\sigma_{xy} (psi)');
                case 5
                    h2 = ylabel('\sigma_{xz} (psi)');
                case 6
                    h2 = ylabel('\sigma_{yz} (psi)');
                case 7
                    h2 = ylabel('\sigma_{zz} (psi)');
            end
            if j == 3
                h3 = title(['h=h_{',num2str(i2-1),'} Interface of ', ...
                    num2str(ply(i2-1)),' and ',num2str(ply(i2)),...
                    ' degree plies']);
            end
            set(gca,'Fontsize',6);
            for j2 = 1:2
                eval(['set(h',num2str(j2),','''Fontsize'',6);']);
            end
        end
        set(h3,'Fontsize',8);
        set(gcf,'PaperUnits','inches','PaperPosition',[0 0 5 5]);
        print('-dpng', [interlaminar_plot,'_between_plies_', ...
            num2str(i2-1),'_',num2str(i2),'.png'], '-zbuffer');
        close
    end

end
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
%
% - INTERLAMINAR STRESSES (MOMENTS APPLIED)-
%

% Check for possible free edge conditions
if NM(4) > tol
    free_edge_perform = 'y';
    for i = 5:6 % Just look at in-plane running loads
        if NM(i) > tol
            free_edge_perform = 'n';
            break
        end
    end
else
    free_edge_perform = 'n';
end

if free_edge_perform == 'y'
    b = width/2;
    color = 'brg';
    shape = '+-x';

    D_star = D^(-1);
    M = NM(4);

    stress_xy_fixed = zeros(5,length(h));
    p = 1;
    for i = 1:length(ply)
        for j = 0:(points_per_ply-1)
            stress_xy_fixed(1:3,p+j) = h(p+j)/b*Q_bar(:,:,i)*D_star*...
                [M;0;0];
            stress_xy_fixed(4,p+j) = h(p+j); % Keep track of distance, z
            stress_xy_fixed(5,p+j) = i;      % Save ply number used
        end
        p = p+points_per_ply;
    end

    % Initialize
    F = zeros(1,length(stress_xy_fixed(1,:))+1);
    G = F;
    H = F;
```

```
sigma_xz_fixed = zeros(1,length(stress_xy_fixed(1,:)));
sigma_yz_fixed = sigma_xz_fixed;
sigma_zz_fixed = sigma_xz_fixed;
for i = 1:length(stress_xy_fixed(1,:))
    sigma_xz_fixed(i) = -Qx*(Q_bar(1,:,stress_xy_fixed(5,i))* ...
        D_star(:,1))*((stress_xy_fixed(4,i))^2-(h_global(...
        stress_xy_fixed(5,i)))^2)/2 + F(i);

    sigma_yz_fixed(i) = (Q_bar(1,:,stress_xy_fixed(5,i))* ...
        D_star(:,1))*((stress_xy_fixed(4,i))^2-(h_global(...
        stress_xy_fixed(5,i)))^2)/2 + G(i);

    sigma_zz_fixed(i) = -dQx_dx*(Q_bar(1,:,stress_xy_fixed(5,i))* ...
        D_star(:,1))*((stress_xy_fixed(4,i))^3-(h_global(...
        stress_xy_fixed(5,i)))^3)/3 + H(i);

    % Update F, G, and H if ply changes
    if i > 1
        if stress_xy_fixed(5,i) == stress_xy_fixed(5,i-1)
            F(i+1) = sigma_xz_fixed(i);

            G(i+1) = sigma_yz_fixed(i);

            H(i+1) = sigma_zz_fixed(i);
        else
            F(i+1) = F(i);

            G(i+1) = G(i);

            H(i+1) = H(i);
        end
    else
        F(i+1) = F(i);

        G(i+1) = G(i);

        H(i+1) = H(i);
    end
end


fixed_end_stress = [stress_xy_fixed(4,:)',stress_xy_fixed(5,:)',...
    stress_xy_fixed(1:3,:)',sigma_xz_fixed',sigma_yz_fixed',...
    sigma_zz_fixed'];
z = [stress_xy_fixed(4,:)',stress_xy_fixed(1:3,:)',sigma_xz_fixed',...
    sigma_yz_fixed',sigma_zz_fixed'];

%----------
% Print out results to verify

fprintf(file,'\nMOMENT INTERLAMINATE STRESSES AT FIXED END\n');
fprintf(file,'   z  \t     ply    \t  Sxx    \t  Syy     \t  Sxy');
fprintf(file,'     \t  Sxz     \t  Syz     \t  Szz\n');

for i = 1:length(fixed_end_stress(:,1))
    fprintf(file,'%8.5f\t',fixed_end_stress(i,1));
    fprintf(file,'%d\t',fixed_end_stress(i,2));
    for j = 3:(length(fixed_end_stress(1,:))-1)
        fprintf(file,'%+9.2f\t',fixed_end_stress(i,j));
    end
    fprintf(file,'%+9.2f\n',fixed_end_stress(i,j+1));
end
fprintf(file,'\n\n');
fprintf(file,'\n-------------------------------------------------------');
fprintf(file,'-------------------------------------------------------\n');
fprintf(file,'\n');

%----------
% Plot out results to verify
figure (1)
for i = 1:2
    subplot(1,2,i)
    for j = 2:4
        eval(sprintf('plot(z(:,j+(i-1)*3),z(:,1),''%s-'');',...
            color(j-1))); %,shape(j-1)
        hold on
    end
    h1 = xlabel(['Stress (psi) at point ',num2str(free_edge_dist)]);
    h2 = ylabel('Distance from mid-point, ''z'' (in)');
```

```
        switch i
            case 1
                h3 = legend('\sigma_{xx}','\sigma_{yy}','\sigma_{xy}',2);
                h4 = title('In-plane Stresses');
            case 2
                h3 = legend('\sigma_{xz}','\sigma_{yz}','\sigma_{zz}',2);
                h4 = title('Interlaminar Stresses');
        end
        % Format plots
        set(gca,'Fontsize',6);
        for j2 = 1:3
            eval(['set(h',num2str(j2),','''Fontsize''',6);']);
        end
        set(h4,'Fontsize',8);
    end
    set(gcf,'PaperUnits','inches','PaperPosition',[0 0 10 5]);
    print('-dpng', [interlaminar_plot,'_MOMENT.png'], '-zbuffer');
    close
end



%-------------------------------------------------------------------------
%-------------------------------------------------------------------------
fclose(file);
cd C:\'Documents and Settings'\user06\Desktop\'MATLAB Common Programs'\Composites
fprintf('\n\n Program is done running!!!\n\n\n');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function ply_long = ply_format(ply,symm,iteration,repitition)

% function ply_long = ply_format(ply,iteration, repitition,symm)
%
% Convert shorthand ply configuration into long format for calculations.
%
% Inputs - ply(matrix) - ply configuration in shorthand form
%                        (NOTE: All angles are in DEGREES)
%          symm (string) - Put string 's' for symmetric configuration, with
%                          an even count. Put 'so' for an odd number of
%                          plies. Put 'a' for asymmetric.
%                          (NOTE: Assumed assymmetric if not present.)
%          iteration(matrix) - matrix vector of same length as ply input to
%                              specify individual repititions
%                          (NOTE: Assumed values of 1 for not present)
%          repitition(interger) - Number of times the configuration itself
%                                 repeats.
%                          (NOTE: Assumed values of 1 for not present)
%
% Outpus - ply_long(matrix) - longhand form of ply configuration
%
%


%----------------------
% Setup for various input cases
x = nargin;

switch x
    case 1
        symm = 'a';
        iteration = ones(length(ply),1);
        repitition = 1;
    case 2
        iteration = ones(length(ply),1);
        repitition = 1;
    case 3
        repitition = 1;
end

%----------------------
% Iteration within set
if length(iteration) ~= length(ply)
    error('Error in ply_format.m  Iteration value is not correct length.')
end
p = 1;
for i = 1:length(ply)
```

```
    for j = 1:iteration(i)
        ply_long(p) = ply(i);                                                %#ok<*AGROW>
        p = p+1;
    end
end
ply = ply_long;

%----------------------
% Repitition of set
p = 1;
for i = 1:repitition
    for j = 1:length(ply)
        ply_long(p) = ply(j);
        p = p+1;
    end
end
ply = ply_long;

%----------------------
% Symmetry Issue
if symm(1) == 's' || symm(1) == 'S'
    if length(symm) > 1.5 && (symm(2) == 'o' || symm == 'O')
        p = length(ply_long);
        for i = p+1:2*p-1
            ply_long(i) = ply(p-1);
            p = p-1;
        end
    else
        p = length(ply_long);
        for i = p+1:2*p
            ply_long(i) = ply(p);
            p = p-1;
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Q = prop_to_Q(prop)

% function Q = prop_to_Q(prop)
%
% Takes property values and creates Q matrix.
%
% Inputs - prop(matrix) - column vector, which mimics the load file in
%                         load_prop.m
%                         [E11,E22,nu12,nu21,G12,CTE1,CTE2,
%                          S1T,S1C,S2T,S2C,S12S,rho]
%
% Outputs - Q(matrix) - 3x3 matrisx to relate stress_12 and strain_12
%
%

if prop(4) == 0
    prop(4) = prop(2)*prop(3)/prop(1);
end

Q = zeros(3);

Q(1,1) = prop(1)/(1-prop(3)*prop(4));
Q(1,2) = (prop(4)*prop(1))/(1-prop(3)*prop(4));
Q(2,1) = Q(1,2); % Symmetry
Q(2,2) = prop(2)/(1-prop(3)*prop(4));
Q(3,3) = prop(5);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [T,T_hat] = trans(theta)

% function [T,T_hat] = trans(theta)
%
% Creates transformation matrices based on fiber angle, theta
%
% Inputs - theta(double) - angle of fiber direction
%                          (NOTE: Angle is in DEGREES)
```

```
%
% Outputs - T(matrix) - transformation matrix
%                       stress_12 = T*stress_xy
%           T_hat(matrix) - transformation matrix
%                           strain_12 = T_hat*strain_xy
%
%

m = cos(theta*pi/180);
n = sin(theta*pi/180);

%-----------------------
T = [m^2, n^2, 2*m*n;
    n^2, m^2, -2*m*n;
    -m*n, m*n, m^2-n^2];

%-----------------------
T_hat = T;
T_hat(1:2,3) = T_hat(1:2,3)/2;
T_hat(3,1:2) = T_hat(3,1:2)*2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [u2,f2] = uf_solve(k,u,f,u_binom,f_binom)

% function [u,f] = uf_solve(k,u,f,u_binom,f_binom)
%
% Solves the system of k*u = f, unknowns within u and r. Returns u and f
% vectors with all quantities to form a complete system.
%
% Inputs - k(matrix) - Stiffness matrix of system.
%                      (NOTE: Square matrix)
%          u(matrix) - Displacement vector. Can contain all known or some
%                      unknown variables. Insert 'dummy' number for unknown
%                      variables. (i.e. u(2) = unknown = 0)
%                      (NOTE: Column vector)
%          f(matrix) - Force vector. Can contain all known or some
%                      unknown variables. Insert 'dummy' number for unknown
%                      variables. (i.e. f(2) = unknown = 0)
%                      (NOTE: Column vector)
%          u_binom(matrix) - Vector of length, u. Each place contains 1 for
%                            known value of u, and 0 for unknown value.
%          f_binom(matrix) - Vector of length, f. Each place contains 1 for
%                            known value of f, and 0 for unknown value.
%
% Outputs - u(matrix) - Displacement vector with all known values.
%                       (NOTE: Column vector)
%           f(matrix) - Force vector with all known values.
%                       (NOTE: Column vector)
%
%

%---------------------%---------------------%---------------------
% Check inputs
if nargin < 5
    error('Error in uf_solve.m    Not enough inputs present.')
end

%---------------------
% Check vector lengths
x = length(u);
y(1) = length(k(:,1));
y(2) = length(k(1,:));
y(3) = length(f);
y(4) = length(u_binom);
y(5) = length(f_binom);

for i = 1:5
    switch y(i)
        case x
            continue
        otherwise
            error('Error in uf_solve.m    Input dimensions are not correct.')
    end
end

%---------------------
```

```
% Check binomial vectors
for i = 1:length(u_binom)
    if (abs(u_binom(i)) + abs(f_binom(i))) ~= 1
        error('Error in uf_solve.m    Binomial vectors not setup appropriately.')
    end
end


%---------------------%---------------------%---------------------
%---------------------%---------------------%---------------------

%---------------------
% - Special cases -
if sum(u_binom) == length(u_binom)                % All u known
    try
        u2 = u;
        f2 = k*u;
    catch                                                           %#ok<CTCH>
        u2 = u;
        f2 = k*u';
    end
elseif sum(f_binom) == length(f_binom)            % All f known
    try
        u2 = k\f;
        f2 = f;
    catch                                                           %#ok<CTCH>
        u2 = k\f';
        f2 = f;
    end
else                                              % Mixed u and f known
    %---------------------
    % - Separate into known and unknown values -
    p0 = 1;
    p1 = 1;
    B_1 = zeros(1,sum(u_binom));
    B_0 = zeros(1,(length(u)-length(B_1)));
    for i = 1:length(u)
        if u_binom(i) == 0
            B_0(p0) = i;
            p0 = p0 + 1;
        else
            B_1(p1) = i;
            p1 = p1 + 1;
        end
    end

    %---------------------
    % - Break down into sub-matrices -
    K_11 = k(B_0,B_0);
    K_12 = k(B_0,B_1);
    K_21 = k(B_1,B_0);
    K_22 = k(B_1,B_1);

    U_2 = u(B_1);  % known values
    F_1 = f(B_0);  % known values

    %---------------------
    % - Solve for unknowns -
    U_1 = K_11\(F_1 - K_12*U_2);
    F_2 = K_21*U_1 + K_22*U_2;

    %---------------------
    % - Reassemble u and f -
    u2 = zeros(1,length(u));
    f2 = zeros(1,length(f));
    for i = 1:length(F_1)
        u2(B_0(i)) = U_1(i);
        f2(B_0(i)) = F_1(i);
    end
    for i = 1:length(F_2)
        u2(B_1(i)) = U_2(i);
        f2(B_1(i)) = F_2(i);
    end

    u2 = u2';
    f2 = f2';
end
```

B.2 Signal Comparison Properties

The following function code was used as a means of identifying the major differences between a baseline signal wave and a current signal wave by comparing the first wave modes.

```
function [time_shift,amp_change] = signal_comparison(baseline,current)

% FUNCTION  [time_shift,amp_change] = signal_comparison(baseline,current)
%
% Determine the differences in amplitude and time of flight for the first
% mode of an ultrasonic wave. Comparison is made between a BASELINE signal
% and CURRENT signal. This is meant for use in tandem with the ACESS
% software by Acellent Technologies. All the actuator-to-sensor paths are
% compared.
%
%
% Functions called upon: initial_time.m (Lines 161-256)
%                        max_envelope.m (Lines 261-307)
%                        wave_beats.m   (Lines 312-395)
%           (NOTE: These are written below as sub-functions.)
%
% Input: baseline (string) - name of *.mat file to be loaded and used as
%                            the baseline signal for comparison
%                            (OPTIONAL: Can incude '.mat' in string)
%        current  (string) - name of *.mat file to be loaded and used as
%                            the current signal for comparison
%                            (OPTIONAL: Can incude '.mat' in string)
%
% Output: time_shift (vector) - Difference in time-of-flight between the
%                               two signals' first wave mode. Presented in
%                               microseconds.
%         amp_change (vector) - Difference in amplitudes between the two
%                               signals' first wave mode. Presented in
%                               volts.
%
%

%---------------------------------------------------------------------
% Default values for breakdown of signals
timing_ratio = 4; % The first fraction of the wave to be scanned to reduce
                  % computation time. (i.e. timing_ratio = n, scans first
                  % 1/n fraction of signal.
threshold = 10;   % Defines the tolerance above and below the zero value to
                  % be considered a notable change in signal.
beat_count = 5;   % Number of oscillation groups to be considered.

%---------------------------------------------------------------------
% Use baseline to measure path length
try
    load(baseline);
catch                                                          %#ok<*CTCH>
    load([baseline,'.mat']);
end
path_count = 0;
while exist(['a',num2str(path_count)])                         %#ok<EXIST>
    path_count = path_count+1;
end
path_count = path_count-1;

%---------------------------------------------------------------------
% Go through an convert all paths definitions
p = 1;
time_shift = zeros(1,path_count);
amp_change = zeros(1,path_count);
for j = 0:path_count
    try
        load(baseline);
    catch
        load([baseline,'.mat']);
    end
    signal_length = length(a0);
    eval(['act_base = double(a',num2str(j),');']);
    eval(['sen_base = double(s',num2str(j),');']);

    try
        load(current);
```

```
catch
    load([current,'.mat']);
end
eval(['act_sig = double(a',num2str(j),');']);
eval(['sen_sig = double(s',num2str(j),');']);

%--------------------------------------------------------------------
% Shift raw signals to oscillate around Volts = 0
t = 1:length(act_base);

p1 = polyfit(t,act_base',1); act_base = act_base - p1(2);
p2 = polyfit(t,sen_base',1); sen_base = sen_base - p2(2);
p3 = polyfit(t,act_sig',1);  act_sig = act_sig - p3(2);
p4 = polyfit(t,sen_sig',1);  sen_sig = sen_sig - p4(2);

%--------------------------------------------------------------------
% Remove initial reverb wave

t0_base = initial_time(act_base,sen_base,threshold,timing_ratio);
t0_sig = initial_time(act_sig,sen_sig,threshold,timing_ratio);

if t0_sig > t0_base
    t0 = t0_sig;
else
    t0 = t0_base;
end

% Reduce feedback
sen_base = [t(t0:length(t))',sen_base(t0:signal_length)];
sen_sig  = [t(t0:length(t))',sen_sig(t0:signal_length)];

%--------------------------------------------------------------------
% Create wave envelope of max numbers
base_env = max_envelope(sen_base(:,1),sen_base(:,2));
sig_env  = max_envelope(sen_sig(:,1), sen_sig(:,2));

%--------------------------------------------------------------------
% Find max point amplitude and time for both base and sig
% (Using threshold_max to start wave beat)

base_beats = wave_beats(base_env(:,1),base_env(:,2),beat_count);
sig_beats = wave_beats(sig_env(:,1),sig_env(:,2),beat_count);

base_wave = [0,0];
sig_wave = [0,0];
for i2 = 1:length(base_beats(:,1))
    if base_beats(i2,3) > base_wave(2)
        base_wave = base_beats(i2,2:3);
    end
    if sig_beats(i2,3) > sig_wave(2)
        sig_wave = sig_beats(i2,2:3);
    end
end

% Ensure the peaks are together
if base_wave(1) < sig_wave(1)
    sig_diff = 100;
    for i2 = 1:length(sig_beats)
        if abs(base_wave(1)-sig_beats(i2,2)) < sig_diff
            sig_wave = sig_beats(i2,2:3);
            sig_diff = abs(base_wave(1)-sig_beats(i2,2));
        end
    end
else
    base_diff = 100;
    for i2 = 1:length(base_beats)
        if abs(sig_wave(1)-base_beats(i2,2)) < base_diff
            base_wave = base_beats(i2,2:3);
            base_diff = abs(sig_wave(1)-base_beats(i2,2));
        end
    end
end

%--------------------------------------------------------------------
% Find differences

time_shift(p) = sig_wave(1) - base_wave(1);
amp_change(p)  = sig_wave(2) - base_wave(2);
p = p+1;
```

```
end

%=========================================================================
%=========================================================================
%
% SUB-FUNCTIONS USED
%
%=========================================================================
%=========================================================================

function y = initial_time(time,act,sen,threshold,timing_ratio)

% FUNCTION  y = initial_time(time,act,sen,threshold,timing_ratio)
%
% Determine the ending time of residual signal from actuator
% Find the start of response wave for detection in surface attached.
% Similar process to functions of ACESS software by Acellent.
% NOTE: Timing for this is based of sample number, NOT the elapsed time for
%       when the sample of recorded.
%
% Input: time (vector) - (OPTIONAL) Time corresponding to both actuator and
%                         sensor signals. If it is not present, than sample
%                         number is usde for time.
%                        (matrix of one column)
%        act (vector) - Actuator signal from ACESS
%                        (matrix of one column)
%        sen (vector) - Sensor signal from ACESS
%                        (matrix of one column)
%        threshold (double) - Signal value for determining end of reverb
%                             beat pulse wave
%        timing_ratio (double) - Fraction to scan for reverb pulse wave
%                                (i.e. use only 1/timing_ratio of signal)
%
% Output: y (double) - Time when cutoff begins, based on input times, t
%
%

%-------------------------------------------------------------------------
% If time variable is not present, shift remaining variables
if nargin < 4.5
    timing_ratio = threshold;
    threshold = sen;
    sen = act;
    act = time;
end

%-------------------------------------------------------------------------
% Check inputs for consistent length
if length(act) ~= length(sen)
    error('Input arrays are not of the same length.')
end

t = length(act);

%-------------------------------------------------------------------------
% Zero actuator signal
act_zero = mean(act(t/2:t));
act = act - act_zero;

%-------------------------------------------------------------------------
% Find max/mins of actuator signal
if nargin < 4.5
    t = 1:length(act);
else
    t = time;
end
u = act;

%-------------------------------------------------------------------------
% Find maximum/minimum values
u_max = zeros(1,2);    % [ u, t ]
peak = [0,0];
k = 1;
for i = 2:(length(t)/timing_ratio) % <-------- Timing_ratio so that not all
                                   %           actuator signal is scanned
    if (abs(u(i)) > abs(u(i-1))) && (abs(u(i)) > abs(u(i+1)))
        u_max(k,:) = [u(i), t(i)];

        %-----------------------------------------------------------
```

```
        % Find maximum peak of pulse wave
        if u(i) > peak(1)
            peak = [u(i), t(i)];
        end
        k = k+1;
    end
end

%-------------------------------------------------------------------------
% Find maximum values of reverb (after peak)
p = 1;
for i = 1:length(u_max(:,1))
    if u_max(i,2) >= peak(2)
        u_max_reverb(p,:) = u_max(i,:);                                    %#ok<AGROW>
        p = p+1;
    end
end

%-------------------------------------------------------------------------
% Determine end of beat pulse based on threshold value
for i = 1:length(u_max_reverb(:,1))
    if abs(u_max_reverb(i,1)) <= threshold
        t_initial = u_max_reverb(i,2);
        break
    end
end

y = t_initial;

%=========================================================================
%=========================================================================

function u_max = max_envelope(t,u,t_step)

% FUNCTION  u_max = max_envelope(t,u,t_step)
%
% Take signal and convert to wave envelope, based on local max and min
% values.
%
% Input: t (vector) - Time values
%        u (vector)- Signal value at each time, t
%        t_step (vector) - (OPTIONAL) Time step for uniform output points
%                          on wave envelope
%
% Output: u_max (matrix) - Matrix of time and signal value on envelope
%                          [ time , value on envelope ]
%
% WARNING: Needs to oscillate around 0 value for maximum efficiency
%
%

%-------------------------------------------------------------------------
% Check inputs for consistent length
if length(t) ~= length(u)
    error('Input arrays are not of the same length.')
end

%-------------------------------------------------------------------------
% Find maximum/minimum values
u_max = zeros(1,2);    % [ t, u ]
k = 1;
for i = 2:(length(t)-1) % <-------- Timing_ratio so that not all actuator
                        %           signal is scanned
    if (abs(u(i)) > abs(u(i-1))) && (abs(u(i)) > abs(u(i+1)))
        u_max(k,:) = [t(i), abs(u(i))]; % NOTE: abs(max/min) to obtain only
                                        %       upper envelope
                                        % (no oscillations in signal)
        k = k+1;
    end
end

%-------------------------------------------------------------------------
% Interpolate function along t_step (if input is present)
if nargin > 2.5
    t2 = t(1):t_step:t(length(t));
    u_max2 = interp1(u_max(:,1),u_max(:,2),t2);

    for i = 1:length(t2)
        u_max(i,:) = [t2(i), u_max2(i)];
```

```
    end
end

%=========================================================================
%=========================================================================

function NN_in = wave_beats(t,u,beat_count,threshold)

% FUNCTION [NN_in, threshold_new] = wave_beats(t,u,threshold,beat_count)
%
% Recreate the functions of PAC-NDT software for PCI-2 board.
% Input wave signals (t, u) and breaks down into sub detections.
%
% Input: t (vector) - Time of wave points
%        u (vector) - Signal of wave --> used with raw data from ACESS
%                     software
%        beat_count (double) - Number of detections to record before
%                              stopping. Output will fill with zeros if
%                              insufficient detections
%        threshold (double) - (OPTIONAL) Detection amplitude (in volts) for
%                             each wave. Default value is 8.
%                            (see PAC-NDT software for more details)
%
% Output: NN_in (matrix) - properties (columns) for set number of
%                          detections (rows)
%                      [ time , peak time, amplitude , duration, rise time ]
%
%


if length(t) ~= length(u)
    error('Input arrays are not of the same length.')
end

%-------------------------------------------------------------------------
% Relative height if threshold not specified
if nargin < 3.5
    threshold = max(u)/8; % <------------------Uncomment for ratio
                          %                       threshold value
end
%-------------------------------------------------------------------------

beat = 0;
beat_set = zeros(1,2);  % [ t , u ]
k = 1;
NN_in = zeros(beat_count,5);     % [ time , amplitude(time,value) ,
                                 %        duration, rise time ];
p = 1;
i2 = 1;
peak = zeros(1,2);

while p-1 < beat_count

    switch beat
        %-----------------------------------------------------------------
        case 0   % Initial detection
            if u(i2) > threshold
                beat_set(k,:) = [t(i2), u(i2)];                              %#ok<*NASGU>
                k = k+1;
                beat = 1;
            end
        %-----------------------------------------------------------------
        case 1   % During detection
            if u(i2) < threshold
                beat = 2;
            else
                beat_set(k,:) = [t(i2), u(i2)];
                k = k+1;
                if u(i2) > peak(2)
                    peak = [t(i2), u(i2)];
                end
            end
        %-----------------------------------------------------------------
        case 2   % End of detection
            beat = 0;
            %-------------------------------------------------------------
            % ADJUST for parameters desired:
            NN_in(p,:) = [beat_set(1,1), peak(1), peak(2), ...
                (beat_set(k-1,1)-beat_set(1,1)), (peak(1)-beat_set(1,1))];
```

```
            p = p+1;

            % Reset parameters for next detection
            beat_set = zeros(1,2);  % [ t , u ]
            k = 1;
            peak = zeros(1,2);
    end
    i2 = i2+1;
    %-------------------------------------------------------------------
    % Incase max number is not reached in detections
    if i2 > length(u)
        break
    end
end
```

B.3 Creating a Self-Organizing Map

The following function was used to create a self-organizing map, using the artificial neural network toolbox in MATLAB. This code was used for the initial setup of the ANNs to be used in agents.

```
function SOM_system_create(net_name,training_set,range,SOM_rows,SOM_cols,epoch_count)

% FUNCTION SOM_system_create(net_name,training_set,range,SOM_rows,SOM_col,epoch_count)
%
% Creates and trains a self-organizing map (SOM), using Neural Networks
% Toolbox. The network is saved to a *.mat file of the desired name of the
% network. Mapping setup is a hexgrid configuration.
% NOTE: Program requires extended time for training.
%
% Input: net_name (string) - The name of the desired network. Name of *.mat
%                            file created. Network name in MATLAB as a
%                            'network' variable.
%        training_set (matrix) - Dataset used to train the SOM network. A
%                                single input set is taken as each column
%                                in the matrix.
%        range (matrix) - Each row represents the minimum and maximum
%                         values for each input variable.
%                         i.e.    input variable 1  [ min_1 , max_1 ;
%                                 input variable 2    min_2 , max_2 ;
%                                        ...              ...
%                                 input variable n   min_n  , max_n ];
%        SOM_row (scalar) - The number of rows in the Kohonen layer of the
%                           SOM neural network.
%        SOM_col (scalar) - The number of columns in the Kohonen layer of
%                           the SOM neural network.
%        epoch_count (scalar) - (OPTIONAL) Number of epochs to use for
%                               training. Default value is 1000.
%
% Files created: 'net_name'.mat - The trained SOM network, ready to be
%                                 loaded into other programs or functions.
%                           (NOTE: net_name (string) is also saved due to
%                                  limitations of save command)

%-------------------------------------------------------------------
% Network setup
net = newsom(range,[SOM_rows,SOM_cols]);
if nargin < 5.5
    net.trainParam.epochs = 1000;
else
    net.trainParam.epochs = epoch_count;
end

%-------------------------------------------------------------------
% Network training
net = train(net,training_set);                                    %#ok<NASGU>


%-------------------------------------------------------------------
% Remove and only save network into *.mat file

clearvars -except net_name net
eval([net_name,' = net;']);
clear net
eval(['save ',net_name]);
```