

University of Windsor

Scholarship at UWindor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

10-19-2015

Predicting Composite Material Performance for LFT-D Using Processing Parameters

Xu Zhang

University of Windsor

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>

Recommended Citation

Zhang, Xu, "Predicting Composite Material Performance for LFT-D Using Processing Parameters" (2015). *Electronic Theses and Dissertations*. 5480.

<https://scholar.uwindsor.ca/etd/5480>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

Predicting Composite Material Performance for LFT-D Using Processing Parameters

By

Xu Zhang

A Thesis
Submitted to the Faculty of Graduate Studies
through the Department of Mechanical, Automotive, and Materials Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science
at the University of Windsor

Windsor, Ontario, Canada

2015

© 2015 Xu Zhang

**Predicting Composite Material Performance for LFT-D Using Processing
Parameters**

by

Xu Zhang

APPROVED BY:

Dr. Edrisy
Department of Mechanical, Automotive, and Materials Engineering

Dr. Minaker
Department of Mechanical, Automotive, and Materials Engineering

Dr. Johrendt, Advisor
Department of Mechanical, Automotive, and Materials Engineering

August 5th, 2015

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Carbon fiber-reinforced composite material properties can be directly related to the manufacturing process. No generally accepted model or system exists that can model the relationship between manufacturing process parameters and composite material properties. The purpose of this research is to develop an artificial neural network model to predict the manufacturing process parameters' influence on the properties of carbon fiber-reinforced composite material. Different types of artificial neural networks are compared in current research in order to obtain the best prediction results. In this research, the calculated sensitivities from the trained neural network are used to find the effect of processing on material properties. Finally, a complete artificial neural network model for predicting composite material performance manufactured using the LFT-D process was built.

*To my loving parents, Lirong and Junfeng,
And my loving wife, Rui,
For their unwavering support, unconditional love,
And continuous encouragement.*

ACKNOWLEDGEMENTS

There are a number of people without whom this journey would have not been possible. I am forever indebted to them. I would also like to thank the International Composites Research Centre at Western University and Fraunhofer Project Center at London, Ontario for their funding support and internship opportunity.

Foremost, I would like to express my deepest gratitude and appreciation for my supervisor, Dr. Jennifer Johrendt, for her patience, knowledge, motivation, technical guidance, and kindness. It is extremely fortunate for me as one of her Master students and I am very appreciative of her for believing in me and giving me lots of confidence. I would also like to thank my committee members, Dr. B. Minaker and Dr. A. Edrisky, for their support and comments. Thanks to the help and advice from PhD candidate Matthew Bondy and PhD candidate Zhe (Tom) Ma. They gave me lots of guidance on technique and gave me confidence to finish this research.

I would also like to thank PhD candidate Ying Fan in Western University who provided the test data applied in this research. Thanks to Kaptur Louis at FPC who patiently provided me all the LFT-D manufacturing processing data and technical guidance on the machine.

There are some very special people that I would like to acknowledge and to whom I express my sincerest gratitude. I would like to thank my parents, Lirong and Junfeng, for believing in me and their unlimited supporting for my twenty-year studying period. I would also like to thank my wife, Rui, for her patient encouragement.

CONTENT

DECLARATION OF ORIGINALITY	iii
ABSTRACT	iv
ACKNOWLEDGEMENTS	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS/NOMENCLATURE.....	xiii
LIST OF SYMBOLS	xiv
CHAPTER 1 INTRODUCTION	1
CHAPTER 2 THEORY	4
2.1 Composite Material.....	4
2.1.1 Composite Material Structures.....	5
2.1.2 Constituent Properties	7
2.2 Carbon Fiber Composite Materials	16
2.2.1 Fiber Composite Materials Classifications	16
2.2.2 Continuous Carbon Fiber Polymer-matrix Composite Properties	17
2.3 Composite Material Process Methods.....	18
2.3.1 Processing of Thermoset Matrix Composites	18
2.3.2 Processing of Thermoplastic Matrix Composites	20
2.3.3 Introduction to RTM, Injection Molding, SMC, and LFT-D-ILC	22
2.3.4 LFT-D Processes.....	26
2.4 Introduction to Neural Networks	29
2.4.1 Introduction to Neural Networks	29
2.4.2 Training of Neural Networks	34
2.4.3 Learning of Neural Networks.....	39
2.4.4 Training Methods Summary	43
2.5 Chapter Summary	45
CHAPTER 3 LITERATURE REVIEW	46
3.1 Predicting Fatigue Life Using ANNs.....	49

3.1.1 M. Al-Assadi et al.	49
3.1.2 Hany El Kadi et al.	50
3.1.3 J.A. Lee et al.	51
3.1.4 Yousef Al-Assaf and Hany El Kadi	53
3.1.5 Others	54
3.2 Predicting Composite Material Strength Using ANNs	54
3.2.1 Mohammed A. Mashrei et al.	54
3.2.2 Z. Zhang et al.	56
3.3 Predicting Other Properties of Composite Materials with ANNs	58
3.4 Predicting Material Performance using Neural Network Sensitivity	60
CHAPTER 4 COLLECTION AND SELECTION OF DATA	62
4.1 Data Resource	62
4.2 LFT-D Processing Parameters	63
4.2.1 Mean Data	63
4.2.2 Data Standard Deviation	64
4.2.3 Data Normalization	65
4.2.4 Reversing the Normalization	66
4.3 Material Test Data	66
4.3.1 Young’s Modulus and Tensile Strength	68
4.3.2 Test Data Organization	68
CHAPTER 5 NEURAL NETWORK PREDICTION	70
5.1 Function Selection	70
5.2 Training Method Selection	71
5.2.1 Network Types	72
5.2.2 Adaption Learning Functions	73
5.2.3 Comparison of Network Types	75
5.2.4 Section Summary	77
5.3 Hidden Layer Configuration	78
5.3.1 Number of Hidden Neurons	78
5.3.2 Number of Hidden Layers	80
5.3.3 Comparison of Different Network Architectures	83

5.4 Artificial Neural Network Training	85
5.4.1 Network Training using Mean Value of Processing Parameters	86
5.4.2 Network Training using Standard Deviation of Processing Parameters	87
5.4.3 ANN Development to Predict CFRP Properties Using LFT-D Processing Parameters	87
5.5 Sensitivity of Network Outputs to Network Inputs.....	91
5.5.1 Sensitivity Analysis.....	92
5.5.2 Neural Network Optimization.....	101
CHAPTER 6 RESULT ANALYSIS	105
CHAPTER 7 CONCLUSIONS AND RECOMMENDATIONS	106
7.1 Conclusions.....	106
7.2 Recommendations.....	108
REFERENCE.....	109
APPENDIX	111
Appendix A.....	111
Appendix B.....	118
Appendix C.....	122
Appendix D.....	126
VITA AUCTORIS	133

LIST OF TABLES

TABLE 2.1. 1 CLASSIFICATION OF COMPOSITE MATERIALS	6
TABLE 2.1. 2 GLASS FIBER COMPOSITIONS AND PROPERTIES.....	9
TABLE 2.1. 3 TYPICAL PROPERTIES OF E-GLASS FIBERS	10
TABLE 2.1. 4 TYPICAL STRUCTURAL MATRIX RESINS	15
TABLE 3. 1 MECHANICAL PROPERTIES	46
TABLE 3. 2 RMSE FUNCTION OF THE TYPE OF ARCHITECTURE AND TRAINING FUNCTION ..	50
TABLE 3. 3 COMPARISON OF NMSE AND R	51
TABLE 3. 4 MEAN ABSOLUTE ERROR BETWEEN EXPERIMENTAL AND PREDICTIONS RESULTS	53
TABLE 3. 5 COMPARISON OF BOND STRENGTH PREDICTION	55
TABLE 3. 6 COMPARISON OF CORRELATION COEFFICIENT, R.....	55
TABLE 4. 1 STANDARD DEVIATION AND MEAN OF INPUT DATA	69
TABLE 5. 1 THE ERROR OF DIFFERENT NETWORKS WITH DIFFERENT TRANSFER FUNCTIONS.	70
TABLE 5. 2 PERFORMANCE OF EACH LEARNING FUNCTION	74
TABLE 5. 3 ALTERNATIVE OPTIONS OF THE NETWORK TYPES	75
TABLE 5. 4 PERFORMANCE (MSE) OF DIFFERENT HIDDEN NEURONS	83
TABLE 5. 5 NETWORK TRAINING PARAMETERS	86
TABLE 5. 6 PERFORMANCE OF THE MEAN VALUE DATA TRAINED RESULT	87
TABLE 5. 7 PERFORMANCE OF STANDARD DEVIATION DATA RESULT	87
TABLE 5. 8 ARTIFICIAL NEURAL NETWORK TRAINING PARAMETERS.....	88
TABLE 5. 9 THE MEAN VALUE OF THE SUMMED ABSOLUTE SENSITIVITIES	94
TABLE 5. 10 BEST VALIDATION PERFORMANCE OF DIFFERENT TRAINING FOR NETWORK	98
TABLE 5. 11 REMAINED INPUTS FOR FOUR DIFFERENT OUTPUTS.....	98

LIST OF FIGURES

FIGURE 2.1. 1 TYPES OF FIBER-REINFORCED COMPOSITES	6
FIGURE 2.1. 2 SCHEMATIC OF GLASS FIBER MANUFACTURE.....	10
FIGURE 2.1. 3 (A) STRUCTURE OF GRAPHITIC LAYER. THE LAYERS ARE SHOWN NOT IN CONTACT FOR VISUAL EASE. (B) THE HEXAGONAL LATTICE STRUCTURE OF GRAPHITE.	11
FIGURE 2.1. 4 SCHEMATIC REPRESENTATION OF THE STRUCTURE OF CARBON FIBERS.....	12
FIGURE 2.1. 5 MATERIAL PROPERTIES OF CFRPs GOVERNING THE MACHINE ABILITY OF CFRPs	13
FIGURE 2.1. 6 POSSIBLE ARRANGEMENTS OF POLYMER MOLECULES: (A) AMORPHOUS, (B) SEMI-CRYSTALLINE	14
FIGURE 2.1. 7 DIFFERENT TYPES OF COPOLYMERS.....	15
FIGURE 2.3 1 (A) IN HAND LAYUP, FIBERS ARE LAID ONTO A MOLD BY HAND AND THE RESIN IS SPRAYED OR BRUSHED ON (B) IN SPRAY-UP, RESIN AND FIBERS (CHOPPED) ARE SPRAYED TOGETHER ONTO THE MOLD SURFACE.....	19
FIGURE 2.3 2 (A) SCHEMATIC OF FILAMENT WINDING PROCESS. (B) SCHEMATIC OF A FILAMENT WOUND PRESSURE VESSEL WITH A LINER; HELICAL AND HOOP WINDING ARE SHOWN	19
FIGURE 2.3 3 SCHEMATIC OF THE PULTRUSION PROCESS	20
FIGURE 2.3 4 VARIATION OF SOME MECHANICAL PROPERTIES OF A COMPOSITE AS A FUNCTION OF FIBER LENGTH	24
FIGURE 2.3 5 EXTRUSION/COMPRESSION MOLDING PROCESS FOR MAKING LONG FIBER REINFORCED THERMOPLASTIC (LFT) COMPOSITES.	25
FIGURE 2.3 6 PROCESS SCHEME OF LFT-D LINE.....	27
FIGURE 2.3 7 CARBON FIBER IN ROVINGS PREPARED FOR LFT-D FEEDING.....	27
FIGURE 2.3 8 CHARGE PRODUCED BY LFT-D PREPARED FOR PRESS	28
FIGURE 2.3 9 TEST PANELS PRODUCED BY LFT-D PROCESS.....	28
FIGURE 2.4 1 A 2-5-1 BACKPROPAGATION FEED-FORWARD NEURAL NETWORK CONFIGURATION	30
FIGURE 2.4 2 SOME NONLINEAR NEURON TRANSFER FUNCTIONS: (A) LOGISTIC, (B) HYPERBOLIC-TANGENT, (C) GAUSSIAN, (D) GAUSSIAN COMPLEMENT, (E) SINE FUNCTION	31
FIGURE 2.4 3 SIMPLE 1-1-1 NEURON NETWORK CONFIGURATION	35
FIGURE 2.4 4 M-N-L LOGSIG PURELIN NEURAL NETWORK CONFIGURATION.....	37
FIGURE 2.4 5 OPTIMAL LEARNING RATE FOR EFFICIENT ERROR MINIMIZATION.....	40
FIGURE 2.4 6 EFFECT OF A HIGH RATE ON LEARNING	40
FIGURE 2.4 7 EFFECT OF TOO HIGH A LEARNING RATE ON LEARNING	41
FIGURE 3. 1 OUTLINE OF THE NETWORK.....	48
FIGURE 3. 2 A TWO-LAYER RECURRENT ELMAN NEURAL NETWORK ARCHITECTURE	49
FIGURE 3. 3 RESULTS OF TRIALS TO ESTABLISH THE OPTIMUM ARCHITECTURE.....	52
FIGURE 3. 4 SCHEMATIC PRESENTATION OF HOW TO DESIGN THE COMPOSITION OF WEAR RESISTANT POLYMER COMPOSITES.....	56
FIGURE 3. 5 INPUT DATA, OUTPUT DATA AND SCHEMATIC CONSTRUCTION OF AN ANN	57
FIGURE 3. 6 THE FLOW CHART OF BPNN TRAINING AND OPTIMIZATION.....	59
FIGURE 4. 1 LFT-D PROCESSING PARAMETERS CLASSIFICATION.....	64
FIGURE 4. 2 CARBON FIBER-REINFORCED THERMOPLASTIC CHARGE AND PANEL BY LFT-D	67

FIGURE 4. 3 CARBON FIBER-REINFORCED THERMOPLASTIC PANELS PRODUCED BY LFT-D	.67
FIGURE 5. 1 HIDDEN NEURONS AND OUTPUT NEURONS SCHEMES	71
FIGURE 5. 2 MATLAB NEURAL NETWORK WORK SHOP WINDOW	72
FIGURE 5. 3 CONFIGURATION OF A 2-HIDDEN-LAYER CASCADE-FORWARD NETWORK	73
FIGURE 5. 4 THE CONFIGURATION OF A 2-HIDDEN-LAYER ELMAN NETWORK	73
FIGURE 5. 5 TRAINING PARAMETERS	79
FIGURE 5. 6 BEST RESULTS COMPARE WITH EXPECTED CURVE	79
FIGURE 5. 7 TRAINING PARAMETERS FOR DIFFERENT NUMBER OF HIDDEN LAYERS	80
FIGURE 5. 8 THE ARCHITECTURES OF TWO ANNS	81
FIGURE 5. 9 (A) MSE OF 1-HIDDEN-LAYER TRAINING (B) MSE OF 2-HIDDEN-LAYER TRAINING	82
FIGURE 5. 10 PERFORMANCE (MSE) OF DIFFERENT HIDDEN NEURONS IN A ONE HIDDEN LAYER LOGSIG-PURELIN NEURAL NETWORK	84
FIGURE 5. 11 PERFORMANCE OF NEURAL NETWORK WITH DIFFERENT NUMBER OF HIDDEN NEURONS	85
FIGURE 5. 12 NEURAL NETWORK CONFIGURATION FOR TRAINING	86
FIGURE 5. 13 TRAINING STATUS OF NEURAL NETWORK APPLIED FOR CURRENT RESEARCH	89
FIGURE 5. 14 VALIDATION, TRAINING, AND TEST MSE	90
FIGURE 5. 15 REGRESSION OF NETWORK TRAINING RESULTS: REGRESSION OF ORIGINAL DATA	91
FIGURE 5. 16 COMMON DESCRIPTIONS OF NEURONS FOR REGULAR NEURAL NETWORK	92
FIGURE 5. 17 (A) TO (D) THE MEAN OF THE SUMMED ABSOLUTE SENSITIVITIES FOR OUTPUTS TO INPUTS FROM OUTPUT Z1 TO Z4	96
FIGURE 5. 18 TRAINING RESULTS OF 39-INPUT NETWORK (A) BEST VALIDATION PERFORMANCE OF NETWORK AND THE MSE OF TRAINING, TEST, AND VALIDATION (B) ERROR COMPARISON ON TARGET AND OUTPUT (C) REGRESSION OF TARGET AND OUTPUT (D) REGRESSION OF NORMALIZED TARGET AND OUTPUT	104

LIST OF ABBREVIATIONS/NOMENCLATURE

ANNs	Artificial Neural Networks
BPNN	Backpropagation Neural Network
CC	Carbon-Carbon Composites
CFB	Cascade-Forward Backpropagation
CFRP	Carbon Fiber-Reinforced Plastic
CMC	Ceramic Matrix Composites
D-SMC	Direct Sheet Molding Compound
EB	Elman Propagation
FFB	Feed-Forward Backpropagation
FPC	Fraunhofer Project Centre
FRP	Fiber-Reinforced Polymer/Plastic
GFRP	Glass Fiber-Reinforced Plastic
HC	Hybrid Composites
LFT	Long Fiber Thermoplastic Compression Molding
LFT-D	Direct Long Fiber Thermoplastic Compression Molding
LM	Levenberg Marquardt Training Algorithm for ANN
MLP	Multi-Layer Perceptron
MMC	Metal Matrix Composites
MSE	Mean Square Error
NN	Neural Network
PC	Polynomial Classifier
PMC	Polymer Matrix Composites
RTM	Resin Transfer Molding
SMC	Sheet Molding Compound

LIST OF SYMBOLS

x_i	i^{th} neural network input
w_{ij}	neural network weight and bias from the i^{th} input to the j^{th} hidden neuron
b_j	neural network weight and bias from the j^{th} hidden neuron to the network output
μ_i	weighted sum of inputs in to the i^{th} hidden neuron
v_i	weighted sum of hidden-neuron output from the i^{th} hidden neuron
n	number of input neurons
l	number of hidden neurons
m	number of output neurons
y_j	calculated result of weighted sum of inputs in neural network
z_k	calculated result of weighted sum of hidden-neuron output
E	mean square error
ε	neural network learning rate
d_g	total gradient of network
g	neural network training epoch number
Δw	hidden weights and bias update
Δb	output weights and bias update

CHAPTER 1 INTRODUCTION

Composite materials are widely used in mechanical engineering to satisfy specific industrial requirements. Composites are a combination of two or more materials with different properties, whose combination produces a new material with beneficial properties for specific application. In the automotive and aviation fields, lighter, stronger and lower cost materials are always required, making composites an obvious choice.

One of the typical engineered composite materials used by these industries is reinforced plastic. Currently, increased research is being done in the automotive industry regarding the use of Fiber-Reinforced Plastics (FRP) in automotive applications in order to improve fuel economy by reducing the vehicle weight.

Many automotive companies have already applied lightweight materials to their vehicle designs like carbon fiber or glass fiber reinforced plastics, for example on Renault's new concept light-weight design car which only consumes 1 litre per 100km (Group 2014). In luxury cars and race cars, carbon fiber composites are generally used in inner trim and car body applications. Currently, additional applications are being investigated such as trunk lids, exhaust pipes, and the engine head.

Western University, in London, Ontario, Canada, and the Fraunhofer Institute of Chemical Technology (ICT) in Pfinztal, Germany, have launched long-term research collaboration on composite technologies for weight reduction. The University of Windsor, McMaster University, and the University of Toronto are Canadian members of this project as well through their membership in the International Composite Research Centre (ICRC). This comprehensive initiative, Fraunhofer Project Centre (FPC) for composite materials at Western, is working on design and development of composite technologies to achieve required material properties with industrial partners in the automotive industry.

One of the key research missions at FPC is to produce and analyze lightweight composite materials. FPC has four technologically advanced machine lines that produce lightweight composite materials. In order to address the interests of industrial

manufactures and academic research, FPC conducted trials to produce composite material panels with continuous and chopped glass fibers or carbon fibers in different various configurations.

In the automotive industry, applications of FRP can be classified into chassis, body, and engine components. For chassis components, many researchers have used fiber-reinforced composite in the manufacture of leaf spring and drive shafts. The high temperature requirement of the matrix (i.e. resin) component is a big challenge for composite engine components. Due to the complex structure and complicated process of manufacturing FRP materials, it cannot be analyzed with the same methods as singular component material. Therefore, there is a need for a simplified model that can accurately relate manufacturing properties and the performance of FRPs.

Artificial Neural Networks (ANNs) are such tool that can be used to predict mechanical properties of composite materials in this way. This black-box model has many advantages like global optimal searching and quick convergence speed (J. Xu 2007). Additionally, ANNs can be applied not only in the study of mechanical properties but wherever the complexity of a problem is overwhelming from a fundamental perspective and where over-simplification is unacceptable in the face of reduced accuracy (H. K. Bhadeshia 1999).

One goal of the current research is to use ANNs to determine the relationship between FRP performance and the manufacturing process parameters. Thus, there are three main topics relevant to the current research: composite materials, composite materials manufacturing processes, and neural network modeling.

This thesis focused on analyzing continuous carbon fiber-reinforced thermoplastic composite material (CFRP) which is produced by the LFT-D machine line (Direct Long Thermoplastic Molding) at FPC. As such, ANNs were developed relating the LFT-D processing parameters (model inputs) to mechanical properties of the material (model outputs). All the data used in this research were obtained from FPC and collaborating

ICRC researchers at the University of Western Ontario.

In this thesis, Chapter 2 introduces the basic theories of the research, including composite material classification and properties with respect to the geometry. Additionally, some traditional methods and advanced methods of composite materials manufacturing are introduced as well as a basic introduction to ANNs.

A review of relevant literature is included in Chapter 3, providing background concerning composite material analysis and neural network prediction. Specifically, some literature highlighting the use of ANNs in composite material modeling provide examples ANN applications in this field.

In Chapter 4, details of how to design and develop neural networks will be given. The prediction processing and analysis of the results are included in Chapter 5 and Chapter 6.

CHAPTER 2 THEORY

The aim of this research is to model the relationship between the material processing parameters and the material performance. As such, it is necessary to develop an understanding of the composite materials as well as the material manufacturing process. Additionally, a foundation in neural networks is also necessary. In this chapter, there are three sections: composite materials, composite material processing method, and neural networks.

2.1 Composite Material

In fact, composite materials have been used since ancient times to enhance the material properties. For instance, the civil engineers placed steel rebar in cement and aggregate to make a well-known composite material, i.e., reinforced concrete in 20th century (Jack R. Vinson and Robert L. Sierakowski, 2008). This research is focused on the advanced composite materials that are widely used in today's industrial applications.

To provide a more specific description of simple composite materials, there are two main categories of composite material components: the matrix and reinforcement (Chawla 2012). With a specific proportion of each component, the reinforcement is surrounded and supported by the matrix. The matrix performance is improved by mechanical and physical properties of the reinforcement. Additionally, composite materials are classified based on their material properties, another important aspect to study (Jack R. Vinson and Robert L. Sierakowski, 2008).

The properties of composite materials are dependent on the material components, the proportion and distribution of the components, the structure of the materials' crystallography, as well as the manufacturing process of the material. As this research is focused on the performance of one kind of composite material, the many factors' influence will be described specifically in a later section (Chawla 2012).

In many industrial applications, composite materials can be designed to satisfy

specific requirements by varying the amount and layout of composite components. One key advantage of composite materials used for in automotive applications is lightweight structure (Jack R. Vinson and Robert L. Sierakowski, 2008).

A common type composite material is fiber reinforced polymer (FRP) that is the focus of this research. The properties of FRPs can be highly influenced by the material characteristics of the fiber rather than that of the matrix. The arrangement, the shape, the specification, and the construction of the fibers are the primary factors that influence the performance of the FRP. In the automotive industry, there are two types of widely used FRPs: carbon fiber-reinforced polymer (CFRP) and glass fiber-reinforced polymer (GFRP).

After composite material structures and constituents are explained in general, CFRP will be described as the focus in this research.






2.1.1 Composite Material Structures

Composite materials are classified according to their structure. The three levels of structure for consideration with respect to the constituent elements of advanced composite materials are (Jack R. Vinson and Robert L. Sierakowski, 2008):

- (1) Basic/Elemental: single molecules, crystal cells;
- (2) Micro structural: crystals, phases, compounds;
- (3) Macro structural: matrices, particles, fibers.

In general, the macrostructure level of composite material is the focus of main research topics today. In order to obtain a clear understanding of composite materials, it is important to review classifications of composite materials as described by the forms of materials in Table 2.1.1.

Table 2.1. 1 Classification of composite materials (Chawla 2012)

Material forms	Description	Figures
Fiber	Either continuous (long) or chopped whiskers suspended in a matrix material	 Fiber Composite
Particulate	Particles suspended in a matrix material	 Particulate Composite
Flake	Flakes having large ratios of platform area to thickness suspended in a matrix material.	 Flake Composite
Filled/Skeletal	Continuous skeletal matrix filled by a second material	 Filled Composite
Laminar	Layers (lamina) bonded together by a matrix material	 Laminar Composite

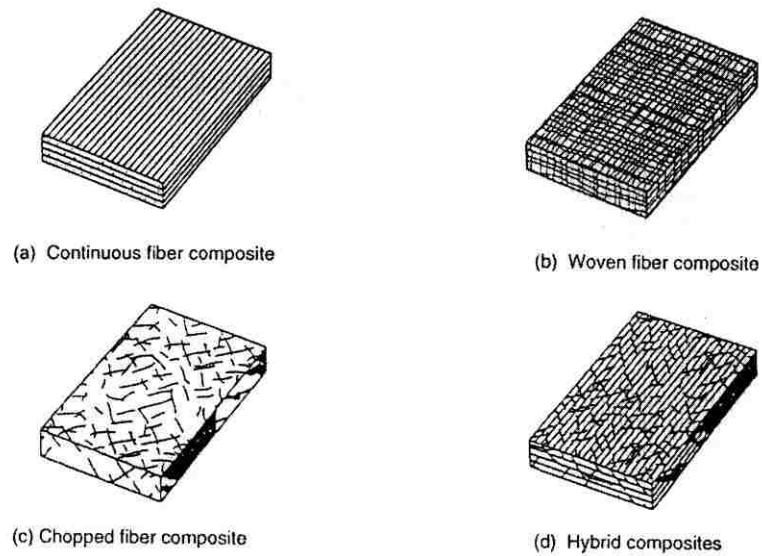


Figure 2.1. 1 Types of Fiber-Reinforced Composites (Chawla 2012)

Additionally, fiber composites can be further described with respect to direction and placement of the fibers, the types can be classified as described Figure 2.1.1.

Fiber composite materials have received more attention recently, especially with

respect to development of fibers such as glass and carbon. As seen in Figure 2.1.1, the four types of fiber composite materials can be recognized by the types of the fiber reinforcement. Fibers in Figure 2.1.1 (a) and (b) are ideally long continuous fibers which are able to contribute to directional stability in the material performance. Chopped fiber composites may not have superior performance in any one direction but instead has an average performance in any direction. When speaking of performance in this research, the research refers to the composite properties derived from the tension test results, such as Young's modulus and tensile strength of the material.

With regard to the matrix material, widely-used composites materials can be divided into five types that are distinguished by their matrix component: Polymer Matrix Composites (PMC), Metal Matrix Composites (MMC), Ceramic Matrix Composites (CMC), Carbon-Carbon (CC) and Hybrid Composites (HC).

Currently, the most commonly used fibers are glass, carbon, and aramid; polymer matrix composites are often used in industry due to their light weight and low cost. The constituent properties will be further discussed in Section 2.1.2.

2.1.2 Constituent Properties

The constituent materials work together to determine the mechanical properties of the composite material. The following subsection includes an introduction of constituent properties.

2.1.2.1 Reinforcement and fiber properties

Reinforcements can take the form of particles, flakes, whiskers, short fibers, continuous fibers, or sheets. Generally, most reinforcements used in composite materials are of the fibrous form due to their stronger and stiffer properties than the other forms. Their reinforcement provides the majority of the strength and stiffness to the composite material. Fibers can be (Jack R. Vinson and Robert L. Sierakowski, 2008):

(1) Organic

-
- (2) Metallic
 - (3) Synthetic
 - (4) Mineral

Naturally occurring fibers are already in use in industry. Although natural fibers cannot impart high strength, they have a great advantage in their low cost. For example, wood and straw fibers have been used in the paper industry (Chawla 2012). Other natural fibers, such as hair, wool, and silk, consist of different forms of protein. However, for the current research, popular fibers, such as glass and carbon fibers, are much more attractive because of their high strength and high stiffness coupled with a very low density.

Glass fiber, which is the most commonly used reinforcement in polymer matrices, can be easily made into different forms. Aramid fibers have much higher stiffness and lower weight than glass fibers. Additionally, fibers such as boron, silicon carbide, carbon, and alumina, combine high strength with high stiffness and have recently been developed in the second half of the twentieth century.

Three important characteristics are summarized by Drescher with regard to the use of fibers as high-performance engineering materials (Chawla 2012):

1. Small fiber diameter with respect to its grain size or other microstructure unit: There is a linear drop in strength with increasing fiber diameter; sometimes, a nonlinear relationship can exist; the probability of having imperfections in the material can be directly affected by the size of the grain.
2. High aspect ratio (length/diameter, l/d): a very large fraction of the applied load is able to be transferred from the matrix to the stiff, strong fiber.
3. Very high degree of flexibility: with low modulus or stiffness and small diameters, the fibers can be incorporated into composites using a variety of techniques.

2.1.2.1.1 Glass fibers

Glass fiber is a generic name used to distinguish a class of fibers. There are many different chemical compositions of glass fibers, which are all silicon based (~50-60%

SiO₂) and contain a host of other oxides of materials such as calcium, boron, sodium, aluminum, and iron (Chawla 2012).

SiO₂ is the largest population of glass fiber, and can be accompanied by the addition of oxides of calcium, sodium, iron and etc. Glass fiber composites can be divided into three popular types: E-glass, C-glass, and S-glass (Hull, D. & Cylne, T. W. 1996). Table 2.1.2 shows the basic properties and the component percentages of three different types of glass fiber.

Table 2.1. 2 Glass fiber compositions and properties (Hull, D. & Cylne, T. W. 1996)

	E-glass	C-glass	S-glass
<i>Composition (%)</i>			
SiO ₂	52.4	64.4	64.4
Al ₂ O ₃ + Fe ₂ O ₃	14.4	4.1	25.0
CaO	17.2	13.4	—
MgO	4.6	3.3	10.3
Na ₂ O + K ₂ O	0.8	9.6	0.3
B ₂ O ₃	10.6	4.7	—
BaO	—	0.9	—
<i>Properties</i>			
ρ (Mg m ⁻³)	2.60	2.49	2.48
K (W m ⁻¹ K ⁻¹)	13	13	13
α (10 ⁻⁶ K ⁻¹)	4.9	7.2	5.6
σ_* (GPa)	3.45	3.30	4.60
E (GPa)	76.0	69.0	85.5
T_{\max} (°C)	550	600	650

Among this three types of glass, E-glass (E for electrical) is most commonly used due to its good strength, stiffness, electrical and weathering properties. C-glass (C for corrosion) is used in some applications when it's necessary to prevent corrosion; however, it has lower strength than E-glass. S-glass (S for strength) has the highest strength, Young's modulus and temperature resistance among these types; it is the most expensive one as well.

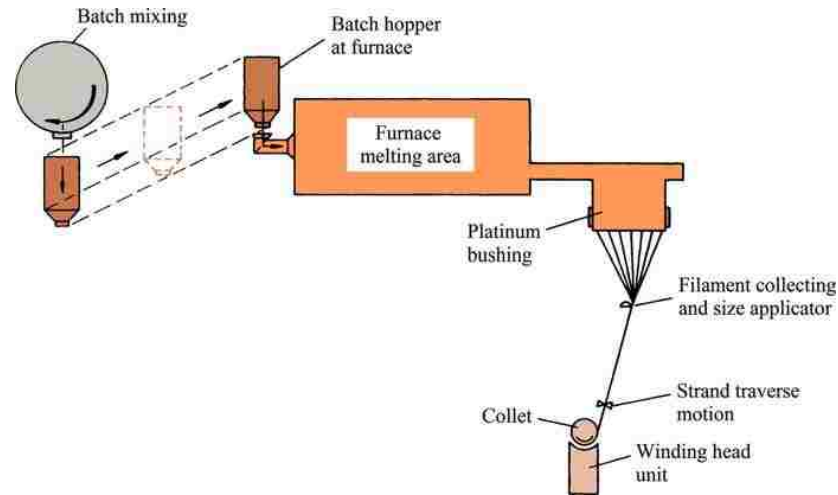


Figure 2.1. 2 Schematic of glass fiber manufacture (Chawla 2012)

Figure 2.1.2 shows a manufacturing schematic for the conventional fabrication of glass fiber. After melting the raw materials in a hopper, the molten glass is fed into the heated platinum bushing which contains about 200 holes at the base. The final fiber diameter can be determined as a function of the bushing orifice diameter. Then, the molten glass flows through these holes and forms fine continuous filaments. The viscosity is a function of composition and temperature. Lastly, the filaments are gathered together into a strand and a size is applied before being wound on a drum. As glass filaments are easily damaged by the introduction of surface defects, the size, or coating, is applied to protect and bind the filaments into a strand.

Table 2.1. 3 Typical properties of E-glass fibers (Chawla 2012)

Density (g/cm ³)	Tensile strength (MPa)	Young's modulus (GPa)	Coefficient of thermal expansion (K ⁻¹)
2.55	1,750	70	4.7×10^{-6}

Table 2.1.3 shows E-glass fibers have low density and high strength. The Young's modulus is not very high. Therefore, the strength-to-weight ratio of glass fiber is very high while the modulus-to-weight ratio is only average. Because of this property, glass fibers are not the primary choice for specialized applications in automotive racing or aerospace. As glass fiber costs are low and it is easily formed, it's often used as reinforcement in polyester, epoxy, and phenolic resins. This composite material is widely

found in the building and construction industry. Additionally, when subjected to a constant load for an extended time period, glass fibers can undergo subcritical crack growth.

As a matter of fact, the glass fiber reinforced resins, commonly called glass fiber reinforced plastics (GFRPs), are used in the form of cladding for other structural materials or as an integral part of a structural or non-load-bearing wall panel. Window frames, tanks, bathroom unit pipes, and ducts are examples of GFRP applications as well as boat hulls. Additionally, GFRPs are also widely used in the rail and road transportation industry.

2.1.2.1.2 Carbon fiber

The density of carbon is equal to 2.268 g/cm^3 , making it a very light element. In a single carbon crystal, the carbon atoms are arranged in a hexagonal array. This is a graphitic structure and will be further described here. The basal planes have strong covalent bonds in the crystal but weak Van Der Waals forces between them. For this reason, the in-plane Young's modulus (about 1,000 GPa) is much larger than that found in the perpendicular plane, so that the crystal units are highly anisotropic. In contrast to glass fiber which is isotropic, the properties of carbon fiber are directionally dependent. (Chawla 2012)

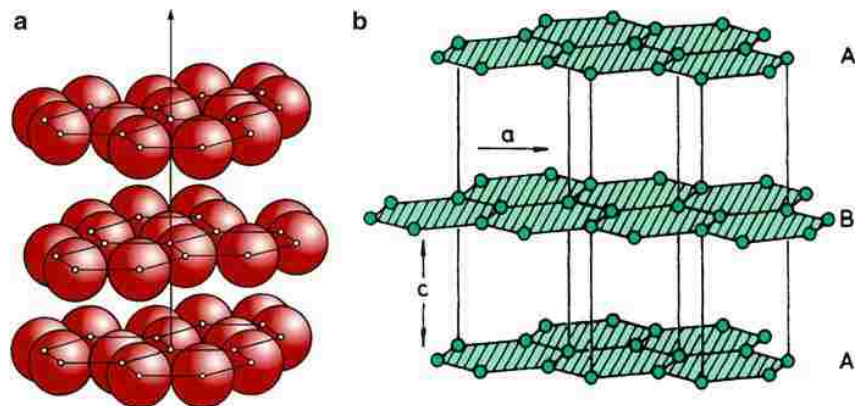


Figure 2.1. 3 (a) Structure of graphitic layer. The layers are shown not in contact for visual ease. (b) The Hexagonal lattice structure of graphite (Chawla 2012)

The graphite structure is very densely packed in each layer as Fig. 2.1.3 (a) shows. Fig. 2.1.3 (b) shows the structure of the lattice. As known, the modulus of a material is

determined by the bond strength. As the figure shows, there are high-strength bonds between carbon atoms in each layer, but, the weak Van Der Waals bonds between the layers result in a lower modulus in the direction perpendicular to the plane. Consequently, most of the processing techniques for carbon fiber aim to obtain a high degree of preferred orientation of hexagonal planes along the fiber axis to ensure high fiber strength.

To obtain good properties for carbon fiber-reinforced composites, the alignment of the basal planes requires attention. Better alignment of the basal planes parallel to the fiber axis can bring higher axial modulus and strength. Furthermore, the orientation of cross-section layer planes of the fiber can also affect the transverse and shear properties. Compared to glass fiber, carbon fiber is much more complex and costly. This is one of the reasons that glass fiber is commonly used in today's industry. As known, carbon fiber-reinforced plastic/polymer (CFRP) is an extremely strong and light-weight fiber-reinforced composite. However, machining of CFRPs is dramatically more difficult than any other composite material.

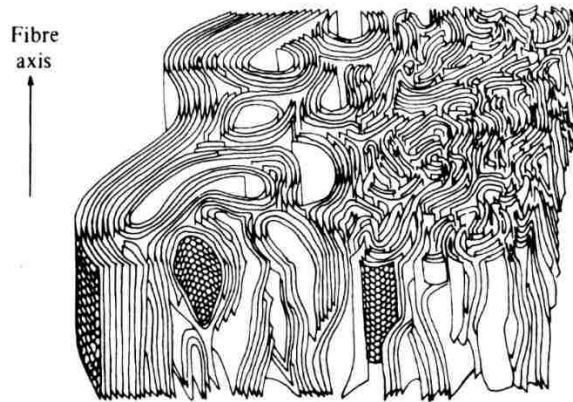


Figure 2.1. 4 Schematic representation of the structure of carbon fibers (Che, D., Saxena, I., Han, P., Guo, P., & Ehmann, K. F. 2014)

As Figure 2.1.4 shows, the structure of carbon fiber is anisotropic. The planes can be clearly seen on the schematic (Che, D., Saxena, I., Han, P., Guo, P., & Ehmann, K. F. 2014).

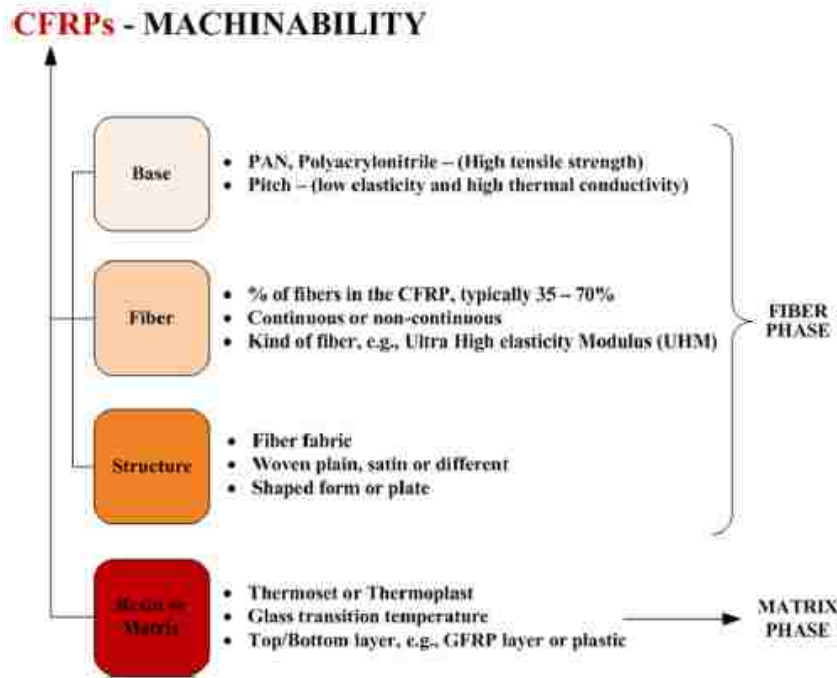


Figure 2.1. 5 Material properties of CFRPs governing the machine ability of CFRPs

Figure 2.1.5 shows the two different phases of material with different properties used to make CFRPs. Each machining method demonstrated different properties. There are three main methods to producing carbon fibers. The first method uses polyacrylonitrile fibers and produces the highest-modulus carbon fiber. The second method is from mesosphere pitch that produces a higher-thermal carbon fiber. The third method of producing carbon fiber uses a gas phase and pyrolytic deposition of hydrocarbons.

2.1.2.1 Matrix and Properties

Composite matrix materials are commonly polymers, metals, and ceramics. The most common of these are polymers; it's cheap and the structure is more complex than others. As polymers are the most often used resins in composite matrices while metal and ceramic are used for specialized applications, this research will focus on the use of polymers.

2.1.2.1.1 Polymers

Polymers possess an inexpensive complex structure that is easily manufactured at a relatively elevated temperature. Generally, polymers have lower strength and modulus

and cannot be used in elevated temperatures. Prolonged exposure to ultraviolet light and some solvents can degrade polymer properties. Generally, polymers are poor conductors of heat and electricity because of predominantly covalent bonding. However, they are more resistant to chemicals than metals are. Structurally, polymers are composed of giant chainlike molecules. Polymerization is the process of joining many monomers together to form large polymeric molecules.

There are two types of polymerization: condensation polymerization and addition polymerization (Chawla 2012). Condensation polymerization is a stepwise reaction of molecules. In each step, a molecule of a simple compound forms as a by-product. Addition polymerization is generally carried out in the presence of catalysts. In this process, the monomers do not produce any by-products.

There are also two major classes of polymer that can be produced by either the condensation or addition polymerization: thermoplastic and thermoset polymers.

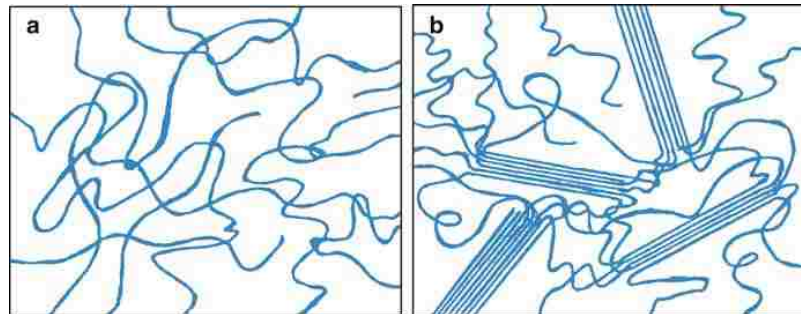


Figure 2.1. 6 Possible Arrangements of Polymer Molecules: (a) Amorphous, (b) Semi-crystalline (Chawla 2012)

Thermoplastic polymers soften or melt on heating and are suitable for liquid flow forming. Its polymer chains are not cross-linked. Thermoplastics can be either amorphous or semi-crystalline. When the structure is amorphous, there is no apparent order of the molecules as illustrated in Fig.2.1.6 (a). In fig. 2.1.6 (b), long molecular chains are folded in a regular manner; the small, plate-like single crystalline regions are lamellae or crystallites, the crystallites combine together and form spherulites. Some of the most commonly used thermoplastics are polyethylene, polystyrene, and polypropylene.

If the molecules in a polymer are cross-linked in the form of a network, the polymer does not soften or melt when heating. These polymers are called thermosets. As the cross-linked molecules cannot slide past one another, the polymer is strong and rigid. Thermoset composite materials are advantageous for high temperature applications. Epoxy, phenolic, polyester and vinylester are all thermoset polymers.

Table 2.1.4 shows some thermoplastic and thermoset materials and their properties.

Table 2.1. 4 Typical structural matrix resins (Chawla 2012)

Resin	Tensile Strength (MPa)	Tensile Modulus (GPa)	T _g (K)*
<i>Thermosets</i>			
Epoxy (TSMDA)	103.4	4.1	463
Bismaleimide	82.7	4.1	547
Polyimide	137.9	4.8	630
<i>Thermoplastics</i>			
Polyphenylene sulfide	65.5	4.3	366 (555 mp)
Polyetheretherketone	70.3	1.1	400

*Glass Transition Temperature

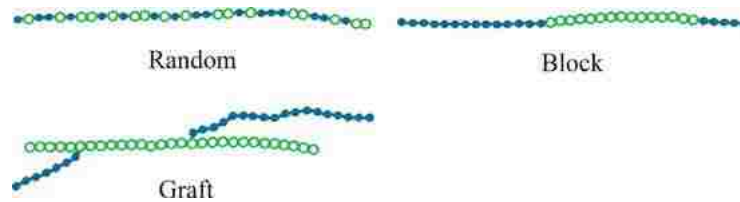


Figure 2.1. 7 Different types of copolymers (Chawla 2012)

Polymers also can be classified by the type of repeating unit. As Figure 2.1.7 shows, there are three types: random, block, and graft. Each repeating unit forming the polymer chain is called homopolymer. If polymer chains are composed of two different monomers, the constructed units are called copolymers.

The chemical resistance of the polymer matrix is better than only carbon fiber and the toughness and damage tolerance can be controlled by the design of the composite material. Carbon fiber composites also have dimensional stability which can be designed for zero coefficient of thermal expansion. In conclusion, carbon fiber composites, according to their superior performance, are the best choice for automotive and aerospace industries except their high cost and complex manufacturing processes.

2.2 Carbon Fiber Composite Materials

As discussed before, composite materials contain multiple materials blended together. Among the reinforcements and matrices, a carbon fiber composite has carbon fiber as at least one of the components. The fiber can be short or continuous, unidirectional or multidirectional. Commonly, the matrix is polymer, metal, carbon, ceramic, or hybrid combinations. Furthermore, the matrix is continuous in all directions. Compared with the matrix, fiber reinforcements are often discontinuous in all directions, which means most materials have different performance along the various fiber directions, unless the fibers are three-dimensionally interconnected (Chung 1994).

In section 2.1.2.1.2, carbon fiber was introduced. In carbon fiber itself, the chemical bonding and Van Der Waals bonding are strong enough to address most mechanical requirements. Moreover, for fiber composite materials, the high strength and modulus of carbon fibers makes them much more useful as the reinforcement for the matrix. In other words, the bonding between fiber reinforcement and matrix play an important role in material performance. In a unidirectional fiber composite, the longitudinal tensile strength is quite independent of the fiber-matrix bonding, but the transverse tensile strength and the flexural strength increases with the effect of fiber-matrix bonding. To further understand fiber composite materials, classifications of different types of combinations will be discussed.

As noted before, the matrix material described here only refers to polymer materials.

2.2.1 Fiber Composite Materials Classifications

Just as the polymers can be divided into thermoplastics and thermosets, so can carbon fiber composites be classified as carbon fiber-reinforced thermoplastics and carbon fiber-reinforced thermosets. The advantages of the carbon fiber-reinforced thermoplastics compared to thermoset-matrix composites were summarized by Deborah D.L. Chung as

follows (Chung 1994):

(1) Lower manufacturing cost:

No cure

Unlimited shelf-life

Reprocessing possible (for repair and recycling)

Less health risks due to chemicals during processing

Low moisture content

Thermal shaping possible

Weldability (fusion bonding possible)

(2) Better performance:

High toughness (damage tolerance)

Good hot/wet properties

High environmental tolerance

Of course, there are some disadvantages of thermoplastics compared to thermoset composite materials such as the limitations in processing methods, high processing temperatures, high viscosities, stiff and dry prepregs, and less developed fiber surface treatments.

In general, composite materials can be classified by the fibres' form. In this way, the carbon fiber polymer-matrix composites can be continuous fiber composites and short fiber composites. According to the fiber type, continuous fiber composites have significantly different properties from the short fiber composite mechanically, or even in their electrical resistivity, thermal conductivity, and other properties. Consequently, the material that is the focus of this research can be summarized as continuous carbon fiber-reinforced thermoplastic composite.

2.2.2 Continuous Carbon Fiber Polymer-matrix Composite Properties

Carbon fiber composites have lower density than aluminum and higher strength than

high-strength steels. They are all stiffer than titanium. Fatigue resistance and creep resistance of carbon fiber composites are ideal. Some carbon fiber composites have low friction coefficients resulting in good wear resistance (Chung 1994).

2.3 Composite Material Process Methods

In this section, some composites manufacturing processes will be introduced, including traditional and advanced methods. As described before, thermosets and thermoplastics are the two main classes of polymeric matrix materials. Thus, the process can be divided into two main categories even though some of the approaches can be used on both types of polymers.

2.3.1 Processing of Thermoset Matrix Composites

For composite materials having a thermoset matrix, there are many processing methods such as hand layup and spray techniques, filament winding, pultrusion and resin transfer molding.

2.3.1.1 Hand Layup and Spray Technique

Hand layup and spray techniques are simple manufacturing processes. As the name suggests, hand layup composites are created by manually placing the fibers in a mold and spraying or brushing the resin (commonly is the polyester) on fibers by hand as shown in Figure 2.3.1. Additionally, with spray-up techniques, resin and chopped fibers are sprayed together onto the mold surface. In both methods, the layers are formed by hand using rollers. Mixed fiber and matrix can be completely cured at room temperature or at a moderately high temperature in an oven (Chawla 2012).

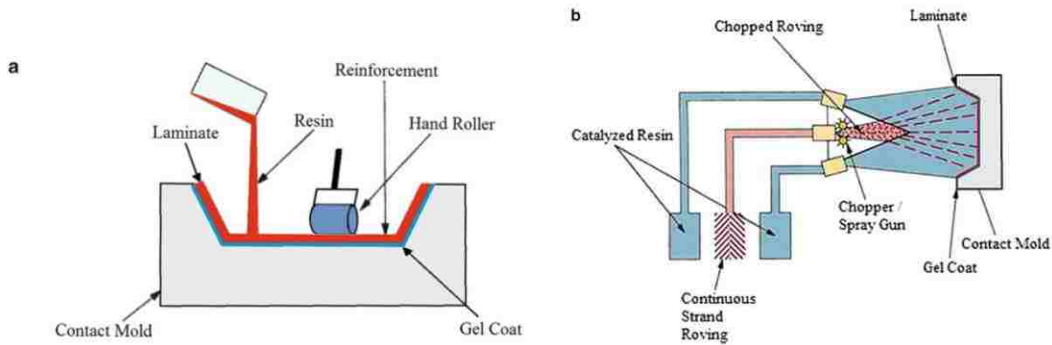


Figure 2.3 1 (a) in hand layup, fibers are laid onto a mold by hand and the resin is sprayed or brushed on (b) in spray-up, resin and fibers (chopped) are sprayed together onto the mold surface (Chawla 2012)

2.3.1.2 Filament Winding

Filament winding is a very versatile technique used for continuous fiber tows or rovings passing through a resin impregnation bath and wound over a rotating or stationary mandrel. This kind of approach can produce very large cylindrical like pipes and spherical vessels, as the winding of the roving can be hoop or helical. Curing of thermoset resin can be done at an elevated temperature, after which the mandrel is removed. (Figure 2.3.2) (Chawla 2012)

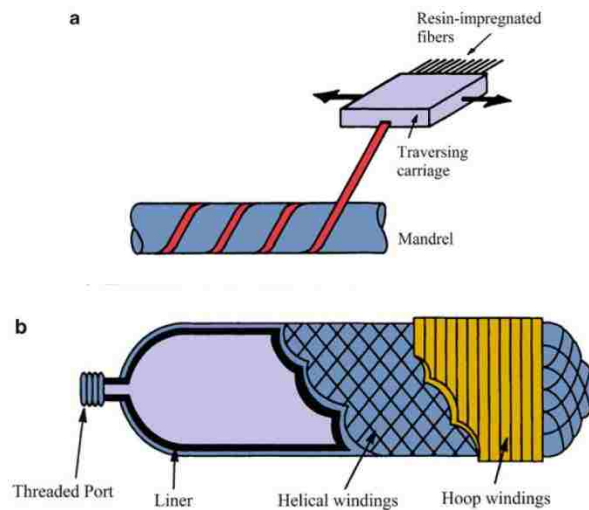


Figure 2.3 2 (a) schematic of filament winding process (b) Schematic of a filament wound pressure vessel with a liner; helical and hoop winding are shown (Chawla 2012)

There are two types of filament winding process. The first process is called wet winding. In this process, during the winding process, low-viscosity resin such as

polyesters and epoxies with viscosity less than $2 \text{ Pa} \cdot \text{s}$ (20 P) is applied to the filaments. Another process is named prepreg winding. The fiber, in this process, will be preimpregnated by a hot-melt or solvent-dip process. Unlike the wet winding process, the rigid amines, novolaces, polyimides and higher-viscosity epoxies are generally used. For the filament winding approach, the void sites are most likely roving crossovers and regions between layers with different fiber orientations.

2.3.1.3 Pultrusion

The pultrusion process has a continuous molding cycle which requires that the fiber distribution and the cross-sectional shapes are constant. Low labour cost and consistency of the product are advantages of the process. Additionally, products like rods, channels, angle, and flat stock are easily produced. The reinforcements used in this process can also be different forms such as roved continuous fibers and continuous strand fiber mats. If using the chopped fiber or chopped strand mat, carrier material is required (Figure 2.3.3) (Chawla 2012).

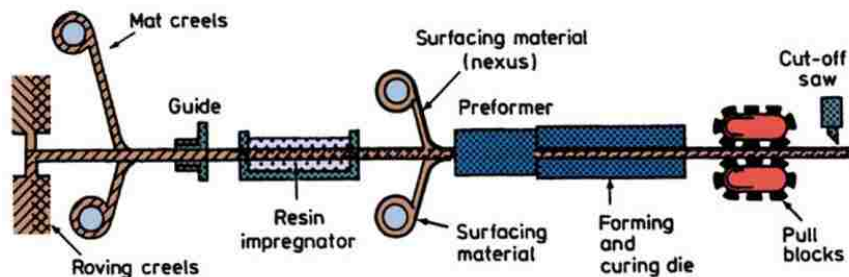


Figure 2.3 3 Schematic of the pultrusion process (courtesy of Morrison Molded Fiber Glass Co)

2.3.2 Processing of Thermoplastic Matrix Composites

There are several advantages and disadvantages of thermoplastic matrix composites compared to thermoset matrix composites. Chawla (Chawla 2012) summarized some advantages of thermoplastic matrix composites:

- (1) Refrigeration is not necessary with a thermoplastic matrix
- (2) Parts can be made and joined by heating

-
- (3) Parts can be remolded, and any scrap can be recycled
 - (4) Thermoplastic matrix composites have better toughness and impact resistance than thermoset matrix composites.

Disadvantages of thermoplastic matrix composites include:

- (1) Processing temperatures are generally higher than those with thermosets.
- (2) Thermoplastics are stiff and boardy, i.e., they lack the tackiness of the partially cured epoxies.
- (3) A wish quality composite laminate must have no void. For this reason, there must be sufficient flow of the thermoplastic matrix between layers and also within individual tows. The molding cycle time is shorter for thermoplastic matrices than that for thermoset matrices.

2.3.2.1 Film Stacking

The laminae are stacked alternately with thin films of pure polymer matrix material. The laminae consist of fibers impregnated with insufficient matrix and polymer films of complementary weight to give the desired fiber volume fraction of the finished product. The pressure and temperature must be sufficient to force the polymeric melt to flow into and through the reinforcement preform. As Darcy's Law (Chawla 2012) states, increasing the applied pressure and decreasing the viscosity of the molten polymer can obtain a better end product.

2.3.2.2 Diaphragm Forming

Diaphragm forming is the sandwiching of a freely floating thermoplastic prepreg layer between two diaphragms. The air in the gap between the diaphragms will be evacuated. The thermoplastic laminate is heated above the melting point of the matrix. The pressure, applied to one side, deforms the diaphragms and forces the material to take the shape of the mold. Because the laminate layers are freely floating and very flexible above the melting point of the matrix, the laminae can be conformed to the mold shape any time. The mold will cool down after the completion of the forming process and

the diaphragms stripped off to obtain the resulting composite part (Chawla 2012).

2.3.3 Introduction to RTM, Injection Molding, SMC, and LFT-D-ILC

Currently, there are several methods to produce FRP using different kinds of fiber-reinforced material. One of these processes is Direct Long Fiber Thermoplastic Molding (LFT-D). In this process, thermoplastic material is directly compounded with long fibers. There are two direct benefits to using LFT-D: precision of the fiber length, and determination of the approximate angle of the fiber. In addition, compound properties can be controlled. Compared to Sheet Molding Compound (SMC) processes for FRP, the LFT-D results in a lower quality surface finish. However, the processing time for LFT-D is much shorter and is easier to mold. Furthermore, it is obvious that long continuous fibers can bring better performance when fiber orientation of the material so greatly influences the material performance.

2.3.3.1 Resin Transfer Molding (RTM)

Resin transfer molding (RTM) is a closed-mold, low-pressure process for preformed fiber, such as glass or carbon. The fiber component is placed inside the mold and liquid resin, such as epoxy or polyester, is injected into the mold by means of a pump. Reinforcements can be stitched, but more commonly they are made into a preform sheet or desired shape. During injection of the polymer matrix the shape is already dictated by the shape of the mold. The completed composites are able to cure and form as a solid in the mold during the process.

For this process, the polymer viscosity should be low enough for the fibers to be wetted easily and the polymer itself to flow smoothly.

RTM has its own advantages as follows (Chawla 2012):

- (1) Easier to obtain large, complex shapes, and curvatures.
- (2) Processing has a higher level of automation than others.
- (3) Simpler layup and speedier process than manual operations.

Fiber volume fractions can be achieved as high as 65% by using woven, stitched, or braided performs.

The process uses a closed mold, resulting in reduction of styrene emissions. Generally, RTM produces much fewer emissions compared to hand layup or spray-up techniques.

Mold design is a critical element in the RTM process. Sometimes, the fibrous preform is preheated and the mold has built-in heating elements to accelerate the process of resin curing. Resin flow and heat transfer are analyzed numerically to obtain an optimal mold design. Usually, the automotive industry uses RTM because it is a cost-effective, high volume process.

2.3.3.2 Injection Molding

Because thermoplastics get soft when heated, melt flow techniques of forming can be used. Injection molding is one of these technologies. Generally, short fiber reinforced thermoplastic composites can be produced by reinforced reaction injection molding (RRIM) which is actually an extension of the reaction injection molding (RIM) of polymers. During RIM, two liquid components (resin and short fiber) are pumped into a mixing head at high speed and pressure and then they enter a mold together where the two components react to polymerize rapidly. During RRIM, short fibers (or fillers) are added to one or both of the components. The lengths of fiber, added into the mold, are generally short and the specifications are limited by the viscosity. Because a certain minimum length of fiber, the critical length, is required for effective fiber reinforcement, RRIM additives are fillers rather than reinforcements. Most RIM and RRIM applications are in the automotive industry as well (Chawla 2012).

2.3.3.3 Sheet Molding Compound (SMC)

The additives consist of fine calcium carbonate particles and mica flakes. Sometimes, for lower density, calcium carbonate powder is substituted with hollow glass microspheres. For further weight reduction, the polyester resin can be replaced by

vinylester. Both reduction techniques for density and weight are costly. Generally, in practice, some auto body parts, such as bumpers, beams, and radiator support panels, are produced by SMC. However, SMC has longer cure times of up to two days (Chawla 2012).

Currently, there is a process called Direct Sheet Molding Compound (D-SMC). The advantage of D-SMC is not only better material properties, but also reduced processing time, dramatically increasing efficiency and decreasing cost for industry.

2.3.3.4 Long Fiber Thermoplastic Compression Molding (LFT)

Long Fiber Thermoplastic Compression Molding, as the name implies, refers to processing composites using a thermoplastic matrix with fibers greater than 10 mm in length. As Figure 2.3.4 shows, the length of the fibers can determine the mechanical properties of composite materials.

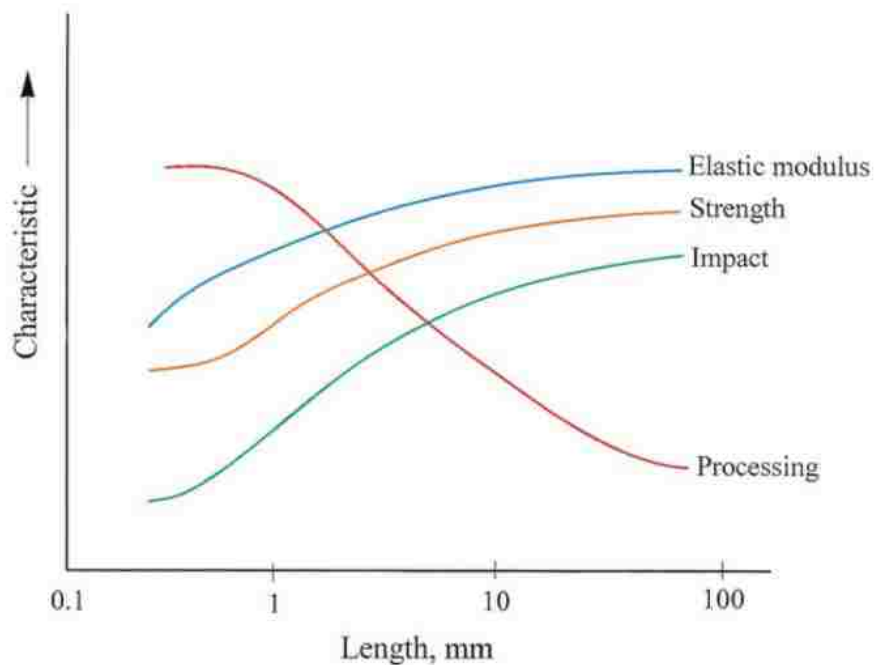


Figure 2.3.4 Variation of Some Mechanical Properties of a Composite as a Function of Fiber Length (Chawla 2012)

Figure 2.3.4 shows advantages of increased fiber length, illustrating why there is a focus on processing of long fiber thermoplastic matrix composites. The basic process of

LFT is shown in Fig. 2.3.5. The critical step in processing LFTs is the production of continuous fiber reinforced rods or tapes from which long fiber pellets are cut.

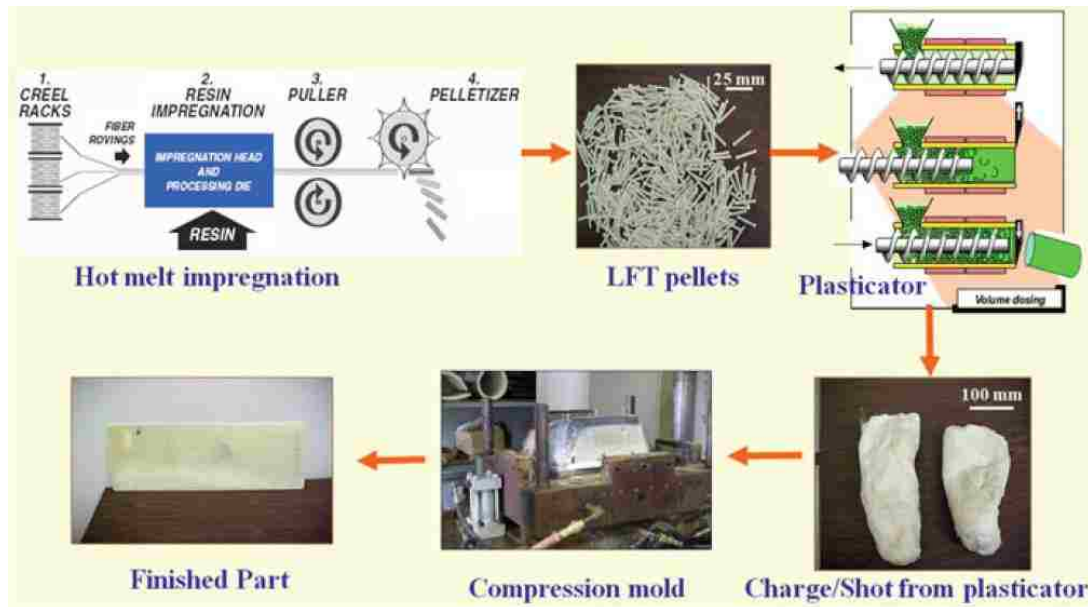


Figure 2.3 5 Extrusion/compression molding process for making long fiber reinforced thermoplastic (LFT) composites. Hot melt impregnation of fibers is used to produce tapes, rods or long pellets of LFT. Pelletized LFT material is fed into an extruder or plasticator. (Chawla 2012)

In this process, continuous fiber tows are first passed through a bath of molten matrix and then through a die for shaping into a rod or ribbon, followed by passage through a chiller for cooling. The final stage involves a puller/chopper; the puller pulls the tow at the desired speed as soon as the chopper cuts the continuous, impregnated tow to desired length of pellets suitable for use in an extruder and compression mold. The long fiber pellets are suitable for the conventional injection molding process, injection compression molding, as well as the extrusion compression molding processes. The LFT pellets made by hot melt impregnation are fed into a plasticator where they are metered into a barrel, heated above the melting point of the thermoplastic resin, and the mixture of polymer plus fiber flows through a low shear plasticator to form a molten charge. The molten charge coming out of the plasticator resembles cotton candy and is quickly transferred to a heated mold where it is compressed in a closed tool (generally, a high tonnage press). The part is then removed after sufficient cooling (Chawla 2012).

2.3.4 LFT-D Processes

There is another commercial process called, LFT-D-ILC (direct long-fiber thermoplastic in-line compounding) or LFT-D, which has been used to make LFT composites consisting of styrene copolymers/glass fibers. Polypropylene has also been used as a thermoplastic matrix in this process. Essentially, it is an extrusion/ compression molding process. The distinctive feature of the LFT-D-ILC process is that the long fiber composite is produced directly from the basic materials. The polymer matrix material and any modifiers/additives are mixed and melted in a compounding extruder. This mixture is combined with the reinforcing fibers in a twin screw extruder (Krause et al. 2003). The special screw-design disperses the fibers in the matrix and further fiber breakage is avoided. The extruder machines work continuously and produce a continuous long fiber reinforced extrudate, which is cut into pieces of the desired length and is then directly compression molded.

LFT-D-ILC is a technology useful in establishing long-fiber thermoplastic material processing in the automotive industry. The aim is to produce lighter and stronger composite material as compared to metal. The LFT-D-ILC technology produced by Dieffenbacher consists of two technologies: the hydraulic press and LFT-D line.

The LFT-D line produces charges for pressing. These charges consist of mixed melted polymer and preheated fibers. The hydraulic press is a high-speed compression mold in which the charge is placed. In practice, the hydraulic press can fabricate a carbon-fiber composite panel in as little as 30 seconds.

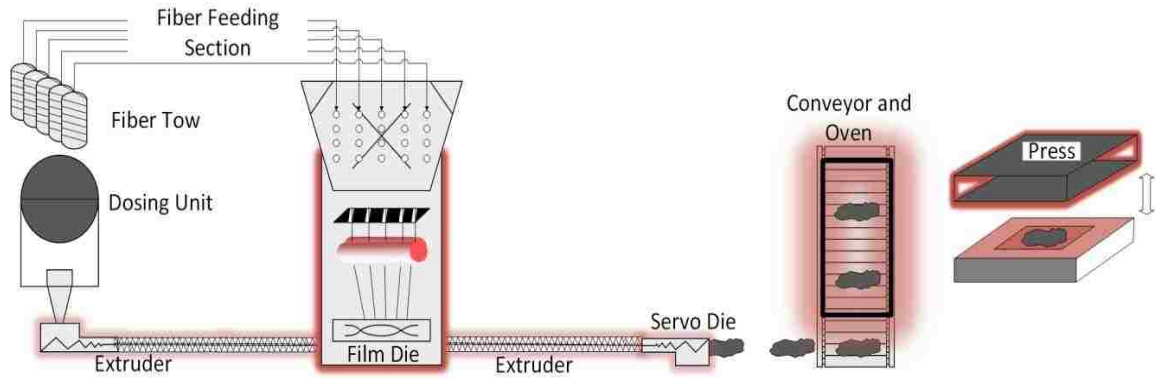


Figure 2.3 6 Process scheme of LFT-D line

As Figure 2.3.6 shows, throughout the fiber feeding section, the continuous fiber is fed into the extruder from the fiber tow, where it is mixed with the polyamide. The polyamide, coming from the dosing unit, is melted as it passes through the heated extruder. At the second extruder, the fiber is mixed in a set direction with the melted polyamide and the charge is produced for the press. The charge goes in the oven on a conveyor, and then charge put into the press machine where the panel is produced.



Figure 2.3 7 carbon fiber in rovings prepared for LFT-D feeding



Figure 2.3 8 Charge produced by LFT-D prepared for press

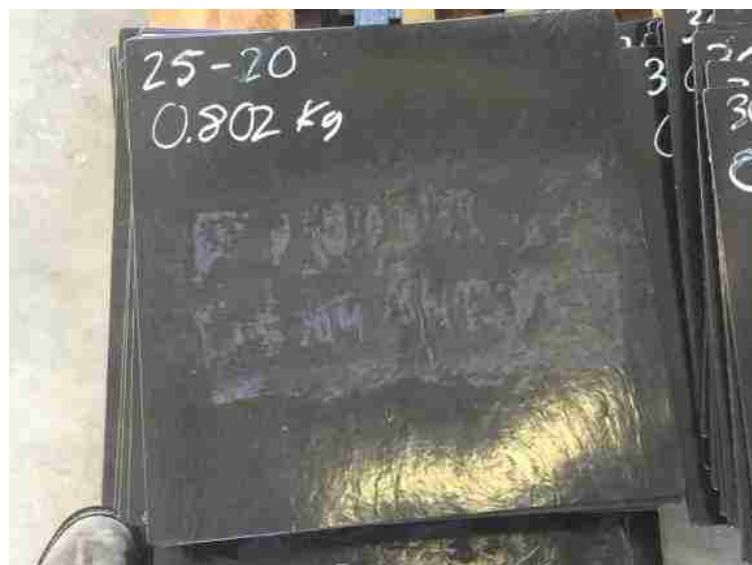


Figure 2.3 9 Test panels produced by LFT-D process

Figure 2.2.7 shows fiber tow on rovings as it is fed into the LFT-D line. Generally, continuous long fiber (fiber tow) is easier for feeding. Unless the fiber is fed smoothly into the machine, the mechanical properties cannot be guaranteed. Figure 2.3.8 shows the presscharge which is produced by the LFT-D line. The weight and the fiber volume of the charge can be set and determined by the machine. Commonly, using an oven to maintain the temperature of the charge can ensure the quality when it enters into the press. In

Figure 2.3.9, the black panel is the material produced by the LFT-D trial. These panels are also the test panels from which data was measured for the current research.

2.4 Introduction to Neural Networks

2.4.1 Introduction to Neural Networks

Artificial neural networks (ANNs), in machine learning and cognitive science, evolved from the basic information processing methods found in the brain. The ANN is a model of statistical learning algorithms to estimate or approximate a related function by examining patterns in a large amount of data. Generally, it is a black-box tool used to describe the relationship between input and output data sets.

Derived from the biological central nervous system in the brain, the basic units are called neurons and connect to each other to form a network. Generally, an artificial neural network has three types of layers to imitate animals' feedback processes: an input layer, hidden layers, and an output layer. The input and output data are held in the input and output layers respectively. The hidden layers use transfer functions to connect the input and output layers (Samarasinghe 2006), with relative importance achieved by weighting the various paths from input through to the output layers with weights and bias values that will be described (see Figure 2.4.1).

In the general case, the full process of neural network development includes training, validation, and testing. Before training a neural network, the first key step is to select a proper network configuration. Figure 2.4.1 shows a 2-5-1 feed-forward backpropagation neural network, as an example, to introduce the neural network development process. A 2-5-1 network is named according to the number of neurons in the input, hidden, and output layers, respectively.

So far, the ANN has a complete basic structure: the neurons in different layers are connected by weights and biases as well as different transfer functions. In this way, it is

obvious that the ANN can be represented by different numbers of neurons, layers, and combinations of transfer functions. Figure 2.4.1 is the structure of a one hidden-layer feed-forward backpropagation neural network (section 5.2.1 describes the feed-forward backpropagation neural network).

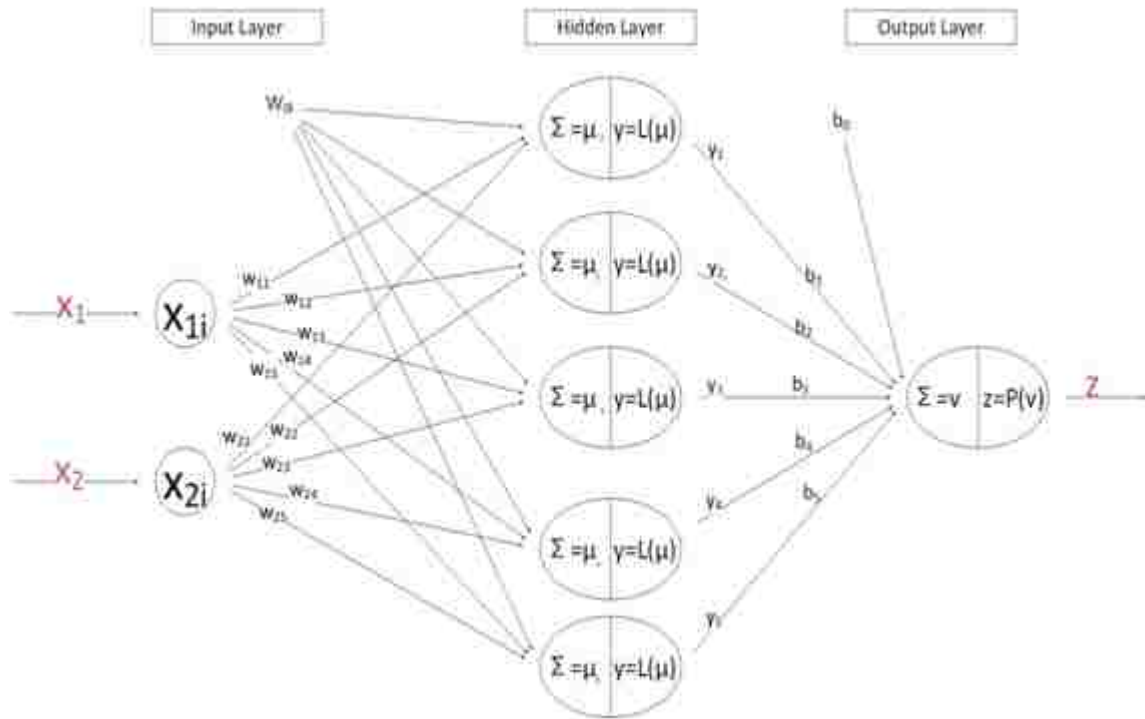


Figure 2.4 1 A 2-5-1 backpropagation feed-forward neural network configuration

Between the input and hidden layers, and between the hidden and output layers, weights connect each pair of neurons (w_{11} , w_{21} , b_1 , b_2 , etc.). Bias values at each neuron act to shift the summed weighted inputs before being passed to the transfer function (w_{0i} and b_0). Both weights and biases are key parameters of neural network that are adjusted during network training.

In the first layer (input layer): x_1 and x_2 represent the input data (i.e. two input variables). As soon as the inputs are transferred to the hidden layer, they are multiplied by the connecting weights and a bias added to form variable μ_i , that are input to the hidden layer neurons:

$$\mu_i = w_{1i}x_{1i} + w_{2i}x_{2i} + w_{0i} \quad (i = 1,2,3 \dots, n) \quad (2-1)$$

In this equation, μ_i represents the weighted sum of inputs to the i^{th} hidden layer neuron. (n is the number of neurons in hidden layer). Then, the μ_i pass through the i^{th} neuron transfer function (also referred to as activation function) denoted $y_i=L_i(\mu_i)$. Various transfer functions can be used for nonlinear neurons in a neural network, as shown in Figure 2.4.2 (Samarasinghe 2006).

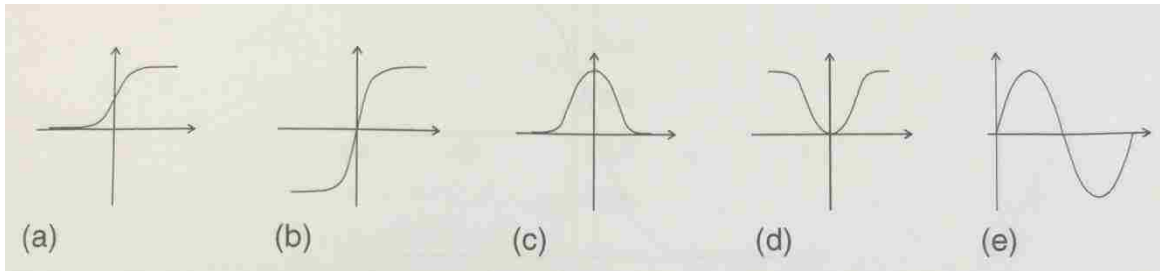


Figure 2.4.2 Some Nonlinear Neuron Transfer Functions: (a) Logistic, (b) Hyperbolic-Tangent, (c) Gaussian, (d) Gaussian Complement, (e) Sine Function (Samarasinghe 2006)

The functions shown in Figure 2.4.2 are vital to neural information processing with their special characteristics. These functions are nonlinear, continuous, bounded functions. Their nonlinear character enables the neural network to model nonlinear relationships between inputs and outputs. Continuity of the function makes it possible to adjust the weights during the error backpropagation process.

There are three main classifications of nonlinear transfer functions: sigmoid functions, Gaussian functions, and sine functions. Sigmoid functions are a family of S-shaped functions. The sigmoid functions include the logistic function and hyperbolic tangent function.

The logistic function has a wider range (i.e. [-1, 1]) than that of the hyperbolic tangent (i.e. [0,-1]). The formulation of the logistic function is

$$y = L(\mu) = \frac{1}{1+e^{-\mu}} \quad (2-2)$$

Where e is the base of natural logarithm, equal to 2.71828. Another commonly used sigmoid function is the hyperbolic tangent function,

$$y = \tanh(\mu) = \frac{1+e^{-\mu}}{1-e^{-\mu}} \quad (2-3)$$

Gaussian functions also include two commonly used functions: standard normal curve and Gaussian complement.

$$y = e^{-\mu^2} \text{(Standard normal curve)} \quad (2-4)$$

and

$$y = 1 - e^{-\mu^2} \text{(Gaussian complement)} \quad (2-5)$$

Both of these functions have range [0, 1]. The standard normal curve peaks at $\mu=0$ and Gaussian complement assumes a value of zero when $\mu=0$.

In general, the sigmoid function, Gaussian functions and sine functions have additional variations. In the current research, sigmoid functions are applied due to their popularity. In MATLAB's neural network toolbox, there are three commonly used transfer functions for feed-forward networks (introduced in section 5.2.1):

Logsig function

$$y = 1/[1 + \exp(-x)] \quad (2-6)$$

Tansig function

$$y = 2/[1 + \exp(-2x)] - 1 \quad (2-7)$$

Purelin function

$$y = \mu \quad (2-8)$$

It is obvious that in the input-hidden layer, the logsig function and tansig function are the same as function (2-2) and function (2-3) when $x=\mu$. These two functions are used to implement in these research as the neural networks are developed and evaluated.

The purelin function is often used in the hidden-output layer to calculate the output. Actually, tansig and logsig functions can be applied in the output neurons as well. However, these two functions make the calculations more complicated without increasing the accuracy (J. Xu 2007). If it becomes necessary to improve the network accuracy, it's more common to increase the number of hidden layers instead of using nonlinear functions in the output neurons.

In the output neuron, the formed variable is v is the sum of the weighted

hidden-neuron output y_i . The function can be written as

$$v = b_0 + b_1y_1 + b_2y_2 + b_3y_3 + b_4y_4 + b_5y_5 \quad (2-9)$$

Where b_0 is the bias weight for output neuron and b_j are the weights for the hidden-output neuron weight. If the purelin function is used in the output layer,

$$z = v \quad (2-10)$$

For the general cases, the weighted sum v comes from the hidden neurons and can be represented by y and the output z as:

$$v_k = b_{0k} + \sum_{j=1}^{\ell} b_{jk}y_{jk} \quad (2-11)$$

$$z_k = v_k \quad (2-12)$$

Where b_{jk} is the weight linking hidden neuron output y_{jk} and the k^{th} output neuron. ℓ is the number of hidden neurons and m is the number of network outputs.

For the example 2-5-1 backpropagation feed-forward neural network, the neural network variables can be written as shown in equations (2-13) to (2-17).

Input-hidden layers:

$$\mu_i = w_{1i}x_1 + w_{2i}x_2 + w_0, i = 1 \text{ to } 5 \quad (2-13)$$

$$y_j = \frac{1}{1+e^{-\mu}} \quad , j = 1 \text{ to } 5 \quad (2-14)$$

or

$$y_j = \frac{1+e^{-\mu}}{1-e^{-\mu}} \quad , j = 1 \text{ to } 5 \quad (2-15)$$

Hidden-output layers:

$$v = b_0 + b_1y_1 + b_2y_2 + b_3y_3 + b_4y_4 + b_5y_5 \quad (2-16)$$

$$z = v \quad (2-17)$$

Where z is the output of the neural network. During training, z will be compared with the target data t to get the error in network prediction.

Multilayer networks, such as feed-forward neural networks, can perform complex

prediction or classification tasks. In general, neural network development has three steps: training, validation and testing.

2.4.2 Training of Neural Networks

When training the network, the weights and biases will have random initial values, although they may be restricted to a predetermined range. Commonly, the training process is not complete with only one iteration. When training a network, the aim is to get the best training performance. This means that the trained network should have a minimum mean square error (MSE) after completing a number of epochs, or training iterations. MSE is the most commonly used error indicator; it measures the error of the prediction over all the samples in each input that are called training patterns. For a one output neural network, the MSE can be written as

$$E = \frac{1}{2N} \sum_i^N (t_i - z_i)^2 \quad (2-18)$$

Where E is MSE, t_i and z_i are the target data and the predicted output for the i^{th} training pattern. The total number of training patterns is N .

Single epoch training can never satisfy the expected MSE. The output z depends on the transfer function and the inputs. As the function is determined by the network configuration and the input cannot be changed, the only changeable conditions are the values of the weights and biases.

To update the weights and biases, the neural network undergoes a learning process. Commonly, the MSE is minimized as a function of the weights and biases. As for linear data fitting, linear regression is used to minimize the error by using the least squares fit on a set of data, for highly nonlinear problems the gradient descent approach is very efficient to find the global minimum of the error between network output and target data during network training.

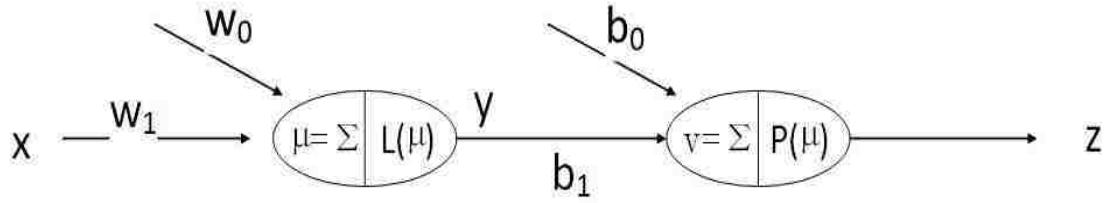


Figure 2.4.3 Simple 1-1-1 neuron network configuration

The backpropagation training method will be illustrated with a simple 1-1-1 network as seen in figure 2.4.3. During training, it is necessary to find the error derivative with respect to all weights, including the ones in the hidden and output layers, so that the weights can be adjusted simultaneously.

Backpropagation is a concept of the chain rule that links the MSE to the hidden-neuron output y , the network output z , as well as the weighted input μ . Denote the derivative with respect to output neuron weight as $\partial E/\partial b$, and the derivative with respect to the hidden neuron weights as $\partial E/\partial w$. The backpropagation chain rule, for a simple 1-1-1 neural network (with logsig and purlin transfer functions), the error gradient with respect to output neuron weight is:

$$\mu = w_0 + w_1 x \quad (2-19)$$

$$y = \frac{1}{1+e^{-\mu}} = \frac{1}{1+e^{-(w_0+w_1x)}} \quad (2-20)$$

$$v = b_0 + b_1 y \quad (2-21)$$

$$z = v = b_0 + b_1 y = b_0 + \frac{b_1}{1+e^{-(w_0+w_1x)}} \quad (2-22)$$

The predicted output and expected target have a prediction error measure defined as, E

$$E = \frac{1}{2}(z - t)^2 = \frac{1}{2}\left\{b_0 + \frac{b_1}{1+e^{-(w_0+w_1x)}} - t\right\}^2 \quad (2-23)$$

According to the backpropagation rule, the error derivative for any hidden-output weight b is calculated with the chain rule, as

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial z} \frac{\partial z}{\partial v} \frac{\partial v}{\partial b} \quad (2-24)$$

where

$$\frac{\partial E}{\partial z} = (z - t) \quad (2-25)$$

$$\frac{\partial z}{\partial v} = 1 \quad (2-26)$$

$$\begin{cases} \frac{\partial v}{\partial b} = 1, b = b_0 \\ \frac{\partial v}{\partial b} = y, b = b_1 \end{cases} \quad (2-27)$$

The error derivative for hidden-output weight b_1 and b_0 in the 1-1-1 neural network mentioned before is

$$\frac{\partial E}{\partial b_0} = \frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial b_0} = (z - t) \cdot 1 \cdot 1 = z - t \quad (2-28)$$

$$\frac{\partial E}{\partial b_1} = \frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial b_1} = (z - t) \cdot 1 \cdot y = (z - t)y \quad (2-29)$$

The error derivative for any input-hidden weight w is

$$\frac{\partial E}{\partial w} = \left(\frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial y} \right) \cdot \frac{\partial y}{\partial \mu} \cdot \frac{\partial \mu}{\partial w} \quad (2-30)$$

where

$$\frac{\partial v}{\partial y} = b_1 \quad (2-31)$$

$$\frac{\partial y}{\partial \mu} = y(1 - y) \quad (2-32)$$

$$\begin{cases} \frac{\partial \mu}{\partial w} = 1, w = w_0 \\ \frac{\partial \mu}{\partial w} = x, w = w_1 \end{cases} \quad (2-33)$$

The error derivative for input-hidden weight w_1 and w_0 in the 1-1-1 neural network mentioned before is

$$\begin{aligned} \frac{\partial E}{\partial w_0} &= \left(\frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial y} \right) \cdot \frac{\partial y}{\partial \mu} \cdot \frac{\partial \mu}{\partial w_0} \\ &= (z - t) \cdot 1 \cdot b_1 \cdot y(1 - y) \cdot 1 \\ &= (z - t) \cdot b_1 \cdot y(1 - y) \end{aligned} \quad (2-34)$$

$$\begin{aligned} \frac{\partial E}{\partial w_1} &= \left(\frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial y} \right) \cdot \frac{\partial y}{\partial \mu} \cdot \frac{\partial \mu}{\partial w_1} \\ &= (z - t) \cdot 1 \cdot b_1 \cdot y(1 - y) \cdot x \\ &= (z - t) \cdot b_1 \cdot y(1 - y)x \end{aligned} \quad (2-35)$$

Denote $\partial E/\partial b_0 = z - t = p$, then $\partial E/\partial b_1 = py$. And denote $\partial E/\partial w_0 = q$. The error gradient with

respect to all the weights and bias for the 1-1-1 network is

$$\frac{\partial E}{\partial b_0} = \frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial b_0} = z - t = p \quad (2-36)$$

$$\frac{\partial E}{\partial b_1} = \frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial b_1} = (z - t)y = py \quad (2-37)$$

$$\frac{\partial E}{\partial w_0} = \left(\frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial y} \right) \cdot \frac{\partial y}{\partial \mu} \cdot \frac{\partial \mu}{\partial w_0} = (z - t) \cdot b_1 \cdot y(1 - y) = pb_1 \cdot y(1 - y) = q \quad (2-38)$$

$$\frac{\partial E}{\partial w_1} = \left(\frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial y} \right) \cdot \frac{\partial y}{\partial \mu} \cdot \frac{\partial \mu}{\partial w_1} = (z - t) \cdot b_1 \cdot y(1 - y)x = qx \quad (2-39)$$

Comparing the error gradient result of the 1-1-1 network with the 2-5-1 backpropagation feed-forward neural network, the error gradient for the simple network is still valid.

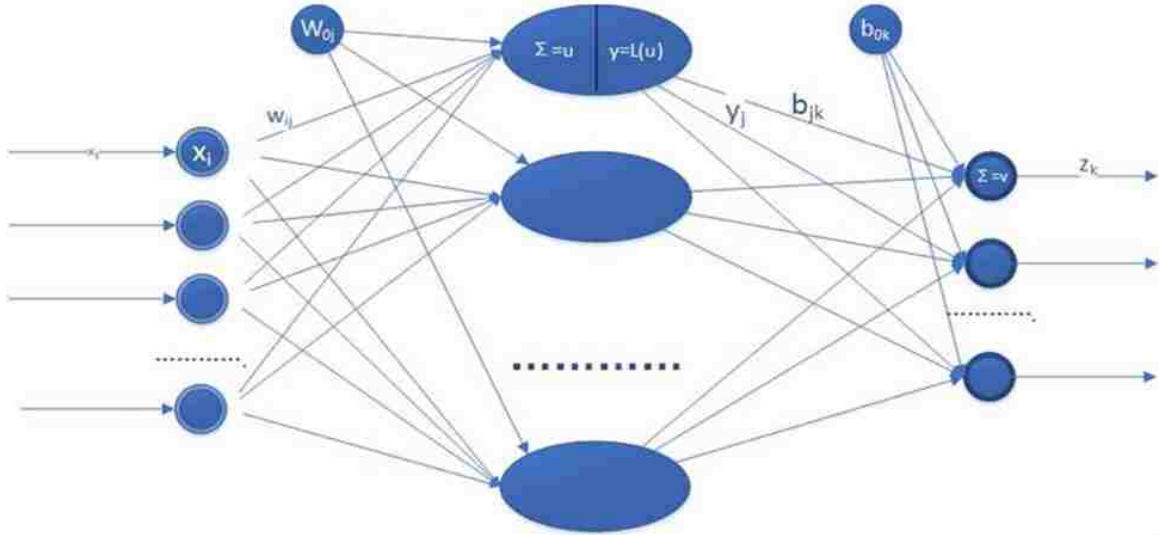


Figure 2.4 4 m-n-l logsig purelin neural network configuration

Generally, for a complex multiple-layer neural network ($m-n-l$ logsig purelin neural network shown in figure 2.4.4), the error gradient can be deduced as the following function:

$$\frac{\partial E}{\partial b_{0k}} = \frac{\partial E}{\partial z_k} \cdot \frac{\partial z_k}{\partial v_k} \cdot \frac{\partial v_k}{\partial b_{0k}} = \frac{1}{l} (z_k - t_k) \quad (2-40)$$

$$\frac{\partial E}{\partial b_{jk}} = \frac{\partial E}{\partial z_k} \cdot \frac{\partial z_k}{\partial v_k} \cdot \frac{\partial v_k}{\partial b_{jk}} = \frac{1}{l} (z_k - t_k) y_k \quad (2-41)$$

$$\frac{\partial E}{\partial w_{0j}} = \left(\frac{\partial E}{\partial z_k} \cdot \frac{\partial z_k}{\partial v_k} \cdot \frac{\partial v_k}{\partial y_j} \right) \frac{\partial y_j}{\partial \mu_j} \cdot \frac{\partial \mu_j}{\partial w_{0j}} = \frac{1}{l} (z_k - t_k) b_{jk} y_j (1 - y_j) \quad (2-42)$$

$$\frac{\partial E}{\partial w_{ij}} = \left(\frac{\partial E}{\partial z_k} \cdot \frac{\partial z_k}{\partial v_k} \cdot \frac{\partial v_k}{\partial y_j} \right) \frac{\partial y_j}{\partial \mu_j} \cdot \frac{\partial \mu_j}{\partial w_{ij}} = \frac{1}{l} (z_k - t_k) b_{jk} y_j (1 - y_j) x_i \quad (2-43)$$

In these functions, i , j , and k the i^{th} , j^{th} , and k^{th} neurons in the m - n - l logsig purelin neural network which mean that: $i=[1,m]$, $j=[1,n]$ and $k=[1,l]$ in integer.

In this way, the mean square error of a general neural network can be described as the function below.

$$E = \frac{1}{2l} \sum_k^l (t_k - z_k)^2 \quad (2-44)$$

And then, the total gradient d_g for epoch g can be expressed as

$$d_g = \sum_{h=1}^N \left[\frac{\partial E}{\partial w_g} \right]_h \quad (2-45)$$

Basically, the sum of the gradients for each of the h training patterns of the m^{th} epoch is the gradient for the m^{th} epoch.

The gradient of the error and the mean square error are used to update the weights and biases. Generally, the network cannot accurately predict the target with initially chosen parameters in only one epoch. Multiple epochs are required to optimize the network weights and biases. If the training performance cannot achieve the desired accuracy in predicting a specified target in g^{th} epoch, the network will update the weights and the biases to new values in the $(g+1)^{th}$ epoch. The total gradient d_g and the learning rate ε determine the efficiency and the quality of the network learning. As shown below, the updated weights and biases are the previous values plus the negative learning rate multiplied by the total gradient. (Samarasinghe 2006)

$$w_{g+1} = w_g + \Delta w_g \quad (2-46)$$

$$\Delta w_g = -\varepsilon d_g \quad (2-47)$$

The learning rate controls the speed of network leaning. If the learning rate is too small, adjustments are smaller and too many epochs are required. Otherwise, if the learning rate is too large, the weights may not converge to an optimal value.

2.4.3 Learning of Neural Networks

As discussed before, neural network training cannot be completed in a single epoch. The method described in Section 2.4.2 is the standard backpropagation learning. The error derivative with respect to all weights is important for network learning. The chain rule is used to derive $\partial E/\partial w$ and $\partial E/\partial b$, and then the gradient of E with respect to all weights and biases is known.

2.4.3.1 Backpropagation Learning

The method of linking E, z, v, y, and u to obtain a relationship for $\partial E/\partial w$ and $\partial E/\partial b$ is required for the backpropagation learning. After calculating the error gradient for all weights and biases as previously discussed, the backpropagation has several uses.

At the ends of every epoch, the weights will be updated using the gradient descent as part of backpropagation. The gradient descent method refers to the process of minimizing the error between network output and target in the direction of the steepest descent of the error when viewed as a function of the weight (or bias), which is the negative of the previously calculated gradients (as equation 2-45). Batch learning is one of the most widely used methods for training such that overall error with respect to the training set decreases incrementally in an average sense.

2.4.3.2 Batch Learning

As described previously, the gradients for all of the patterns are stored after the whole dataset has been trained. After the whole set has been processed during each epoch, the average or the resultant gradient can be found. The total gradient as shown in equation (2-45) is used for the weight update. In batch learning, when total gradient is calculated, the variable equals to the negative learning rate multiplied by the total gradient (the distance of the descent as a portion of the gradient) as in equation (2-47). Then the new weight and bias for next epoch training of batch learning can be written as:

$$w_{g+1} = w_g + (-\varepsilon \cdot \sum_{h=1}^N \left[\frac{\partial E}{\partial w_g} \right]_h) \quad (2-48)$$

As discussed before, the learning rate is also a factor that affects the learning performance. The optimum learning rate can result in a highly efficient learning process to achieve the minimum MSE. Like Figure 2.4.5 shows, the optimum learning rate can lead a quick and accurate way to achieve the minimization. Generally, the learning process shown in Figure 2.4.5 is idealized, but, if the learning rate is too high, the result will be such as shown figure 2.4.6 and Figure 2.4.7 which may never reach the minimum MSE.

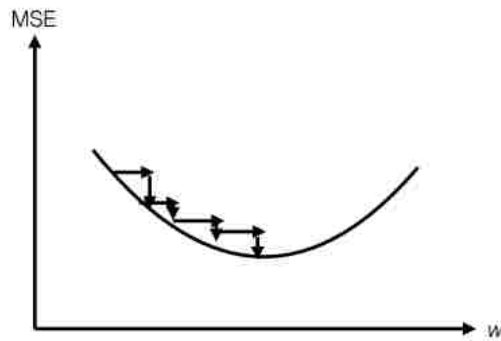


Figure 2.4 5 Optimal learning rate for efficient error minimization (Samarasinghe 2006)

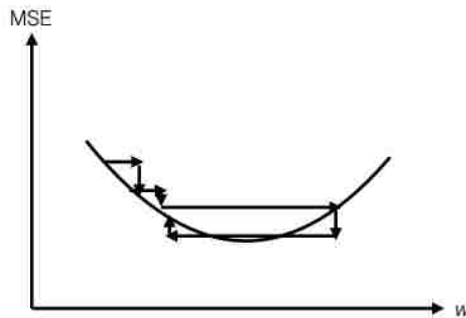


Figure 2.4 6 Effect of a high rate on learning (Samarasinghe 2006)

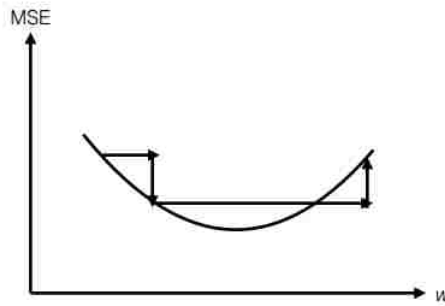


Figure 2.4 7 Effect of too high a learning rate on learning (Samarasinghe 2006)

After every epoch, the performance of the network (as indicated by MSE) will determine whether the training is complete. Often, achieving the minimum MSE is one trigger that will halt training, as well as reaching a gradient limit or the maximum number of epochs.

2.4.3.3 Example-by-Example Learning

Different from the previously described method, the method of individual weight adjustment prior to each input in every epoch is called example-by-example learning. This method takes more computation time than batch learning due to the number of calculations required per epoch.

For some problems, example-by-example learning will yield effective results, especially for online learning for which the training data arrives in real time. Otherwise, this method may bring instability to the problem. Compared to batch learning, the weight and bias adjustments cannot follow the total error derivative over the whole training set.

2.4.3.4 Momentum

There is another approach for network learning used to optimize the weights and biases which incorporates a momentum term. This is an approach that can use the exponential average of all of the previous weight changes to guide the current change (Samarasinghe 2006). Momentum links the average changes of past weights to the new weight increment at each epoch. Compared with batch learning, momentum can be presented mathematically as follows:

$$\Delta w_g = \mu \Delta w_{g-1} - (1 - \mu) \varepsilon d_g^w \quad (2-49)$$

The w_g and w_{g-1} are the weights of current and previous epoch, respectively. The ε is learning rate as per the batch learning method. μ represents a momentum parameter that is between 0 and 1. d_g^w is the g^{th} epoch's total gradient (the same as d_g in equation 2-45). The momentum parameter indicates a relevance between the past weight change on the new weight increment. The current total derivative alone for the weight is now weighted by $(1-\mu)$ which is related to the delta weight by μ . Therefore, if μ is equal to 0, the momentum is not a factor and the equation is the same as equation become to equation (2-47). When the momentum parameter equal 1, it means the momentum change now is totally depended on the past changes. If the value of momentum parameter is between 0 and 1, there is a combined contribution to the weight change. At this time, the above equation can be extended for first to last epochs as below:

$$\begin{aligned} \Delta w_g &= \mu \Delta w_{g-1} - (1 - \mu) \varepsilon d_g^w \\ \Delta w_{g-1} &= \mu \Delta w_{g-2} - (1 - \mu) \varepsilon d_{g-1}^w \\ \Delta w_{g-2} &= \mu \Delta w_{g-3} - (1 - \mu) \varepsilon d_{g-2}^w \\ &\dots\dots \\ \Delta w_2 &= \mu \Delta w_1 - (1 - \mu) \varepsilon d_2^w \\ \Delta w_1 &= (1 - \mu) \varepsilon d_1^w \end{aligned}$$

Combining these equations together, one obtains the Δw_g with respect to Δw_{g-1} which can be expressed as:

$$\begin{aligned} \Delta w_g &= \mu^g (1 - \mu) \varepsilon d_1^w - \mu^{g-1} (1 - \mu) \varepsilon d_2^w - \mu^{g-2} (1 - \mu) \varepsilon d_3^w - \dots \\ &\dots\dots - \mu (1 - \mu) \varepsilon d_g^w \end{aligned} \quad (2-50)$$

Therefore, all the past weight changes are exponentially averaged with momentum. The previous weight changes will affect the current one. This method can be used in both batch and example-by-example learning. The momentum can offer stability to the

gradient descent for the former learning method and minimize the oscillations in the latter ones.

2.4.3.5 The Levenberg-Marquardt (LM) Method

The Levenberg-Marquardt (LM) Method, a second-order method of error minimization and weight optimization, improves the solution to problems that are much harder to solve by adjusting the learning rate for each epoch if required (Samarasinghe 2006). The LM method incorporates a new term e^λ added to the second derivative term instead of learning rate ϵ . (where e is the natural logarithm.)

Firstly, within the general concept of second-order methods, updating the weights in the negative direction of the gradient can be expressed as:

$$\Delta w_g = -\frac{\partial E/\partial w}{\partial^2 E/\partial w^2} \quad (2-51)$$

$$w_g = w_{g-1} + \epsilon \Delta w_g = w_{g-1} - \epsilon \mathbf{R} d_g \quad (2-52)$$

Where all the variables are the same as before except \mathbf{R} . \mathbf{R} is introduced here as the inverse of the second derivative. If \mathbf{R} is not equal to 1 in this equation, more advanced second-order learning methods result, including the LM method. Derived from the second order concept, the \mathbf{R} now becomes $[1/(d_g^s + e^\lambda)]$ and the new weight update for epoch g for is expressed as (Samarasinghe 2006):

$$\Delta w = -\frac{d_g}{d_g^s + e^\lambda} \quad (2-53)$$

In the LM method, the λ is random generated variables until a downhill step is produced for each epoch. If the MSE during training does not decrease, λ will repeatedly increase until a decrease is produced.

2.4.4 Training Methods Summary

In practice, there are more training methods than discussed here. The aim of all the methods discussed is to minimize the error between network output and target. For both the batch learning and the example-by-example learning methods, the LM or the

momentum method are effective for network training.

Backpropagation using the LM method is commonly used according to research. Samarasinghe introduced six training methods in detail (Samarasinghe 2006), which included backpropagation, adaptive learning rate, the steepest descent, Quick Prop, the Gauss-Newton method, and the LM method. The former three methods use the first derivative of error (slope) or are first-order methods. The latter three methods use second derivative of error and on first-order methods (slope and curvature) in order to optimize weights. In general, the second-order methods are based on the first-order ones and provide faster solutions because of the incorporation of additional information in the form of the second derivative of the error. The second-order method for adjusting weights can be more efficient. Otherwise, for some problems, the first-order methods give the researcher the flexibility and direction to improve the analysis when higher order methods may fail to converge.

2.5 Chapter Summary

This chapter discussed the main theories associated with this research: composite materials, material manufacture processing, and artificial neural networks. As previously described, the advanced composite materials currently used in industry are FRPs which have many advantages over traditional materials. CFRP is the focus of this research. To understand the performance characteristics of CFRP, knowledge of the manufacturing process is required in addition to basic composite materials knowledge. In this chapter, many general processing methods for composite materials are introduced especially for the direct long fiber thermoplastic molding (LFT-D). As discussed before, the manufacturing process can be a main factor of influence on the performance of composite materials. The LFT-D processing method is very widely used in mechanical industries. For automotive and aerospace industries, the use of CFRP made using LFT-D has been the practice of many companies. However, even though CFRP is thought of as an ideal material for mechanical engineering applications, there are still many improvements that can be explored. For this reason, many researchers have recently put a lot of effort into increasing CFRP performance.

ANNs are one of the most accessible mathematical methods. However, they also have a potentially complex internal structure that requires understanding. As there are many training and learning methods, it is difficult to introduce neural networks in a comprehensive way. This chapter introduced some basic information about neural network applications and expressed some equations that are commonly used in neural networks. As neural networks research is varied on its use and intent, the exact network configuration of this research will be described in chapter 5 with the network development process specific to the current research.

CHAPTER 3 LITERATURE REVIEW

Neural network models applied to composite materials are most common for predicting mechanical properties of materials in the available literature. There are many advantages of this global optimal search technique including its quick convergence speed (J. Xu 2007). Because the prediction of mechanical properties is generally based on large data sets, the neural network model is a very useful tool in this respect because it benefits from large sets of available data.

Table 3. 1 Mechanical Properties need to be expressed in quantitative models as a function of large numbers of variables (H. K. Bhadeshia 1999)

Property	Relevance
Yield strength	All structural applications
Ultimate tensile strength	All structural applications
YS/UTS ratio	Tolerance to plastic overload
Elongation	Resistance to brittle fracture
Uniform elongation	Related to YS and UTS
Non-uniform elongation	Related to inclusions
Toughness	Tolerance to defects
Fatigue	Cyclic loading, life assessments
Stress corrosion	Slow corrosion & cracking
Creep strength	High temperature service
Creep ductility	Safe design
Creep-fatigue	Fatigue at creep temperatures
Elastic modulus	Deflection, stored energy
Thermal expansivity	Thermal fatigue/stress/shock
Hardness	Tribological properties

The list of properties in Table 3.1 indicates that the development and processing of materials can be very complex. Neural networks can predict the relationships between these properties may not have obvious relevance to each other (H. K. Bhadeshia 1999).

The multilayer feed-forward backpropagation neural network is a widely used ANN. In general, when large amounts of experimental data are to be simulated, other ANNs can be used for prediction. They conveniently relate less obvious properties as a black-box tool.

For many researchers, the choice to use ANNs to model mechanical properties for

composite materials is a popular one. S. Jayabal and S. Rajamuneeswaran (2013) recently published their research on calcium carbonate impregnated coir-polyester composites using ANNs to model the mechanical properties (Jayabal, S., Rajamuneeswaran, Ramprasath, R., & Balaji, N. S. 2013). Natural fiber, like coir-polyester, is low cost but can provide higher acoustic damping than glass/carbon fiber composite. It's a good choice for automotive interior applications to reduce noise and vibration. There are three different types of coir fiber: white, brown, and green husk. The green husk coir fiber has the highest strength of the three. After collecting data experimentally, ANNs were used to form three types of multiple layer propagation network (MLP) to model three output variables: tensile, flexural, and impact behavior. As the coir fiber is low cost and is easily incorporated into composites, low load automotive components can be fabricated with this fiber. Jaybal et al. also listed applications of coir fiber composite in their paper.

There are some researchers using ANNs to predict the failure life of composite materials. P. Labossiere and N. Turkkan (1993) used multilayer feed-forward backpropagation network to predict the failure of anisotropic fiber-reinforced materials under various loading conditions (P. Labossiere and N. Turkkan 1993). They were one of the first to use ANNs as is a method for predicting failure. They used the logistic function in two hidden layers and updated the weights using momentum. They utilized three inputs as shown in Figure 3.1.

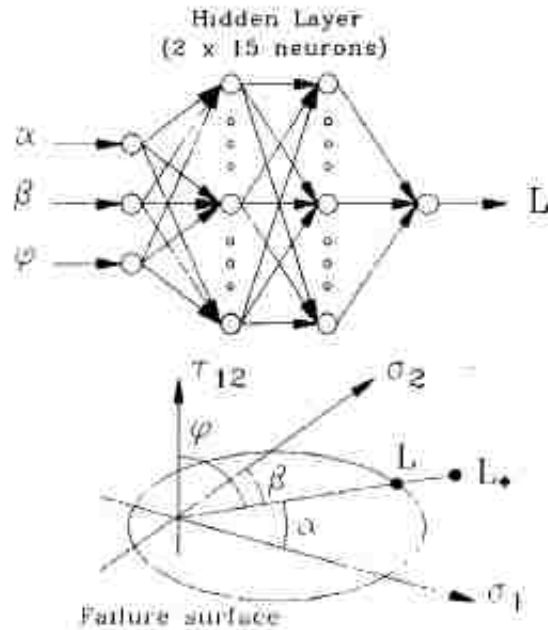


Figure 3. 1 Outline of the network (P. Labossiere and N. Turkkari 1993)

The inputs α , β , and φ are special coordinates describing L in σ_1 - σ_2 - τ_{12} . L is defined of the error to evaluate the strength properties of the particular lamina. In this paper, experimental results were used to train the network and the ANN successfully predicted failure of fiber-reinforced laminate. They concluded that with enough experimental data neural networks can replace analytical failure criteria for a given material.

The isotropic characteristic of many fibers are not ideal for all uses. Therefore, random chopped fibers have become a more popular research topic. Victor A. Gotlib (2001) used neural networks to compute the effective properties of random composite materials (Victor A Gotlib, Tadanobu Sato, Abraham I Beltzer 2001). Gotlib et al. compared the self-consistent scheme and neural computations in the case of composites with particularly large gaps between the Hashin-Shtrikman bounds. They showed the neural networks provided better and more relevant prediction capabilities and revealed useful insight into the effective behavior of disordered heterogeneous materials.

3.1 Predicting Fatigue Life Using ANNs

3.1.1 M. Al-Assadi et al.

The fatigue life of materials is a critical factor to be evaluated. Neural networks can successfully predict the fatigue life for different composite materials. M. Al-Assadi et al. (2010) used ANNs to model the fatigue behavior of fiber-reinforced composite materials. They indicated that ANNs can be good at dealing with multivariable non-linear behavior for which analytical solutions are difficult to obtain. The benefit of applying ANNs is that they are able to predict the cyclic behavior of material for which no fatigue data is available by using the known characteristics of other composites. In their work, the cascade-forward neural networks (CFFN) Elman networks (ELM), and layer recurrent network (LRN) were used (Al-Assadi, M., Kadi, H. E., & Deiab, I. M. 2010).

The CFFN is similar to feed-forward networks. The only difference is there is a weight connection between input-hidden and hidden-output layers. These additional connections might improve the network training speed. ELMs are multi-layer back-propagation networks. There are feedback connections from the output of the hidden layer to the hidden layer input. The structure is shown in Figure 3.2.

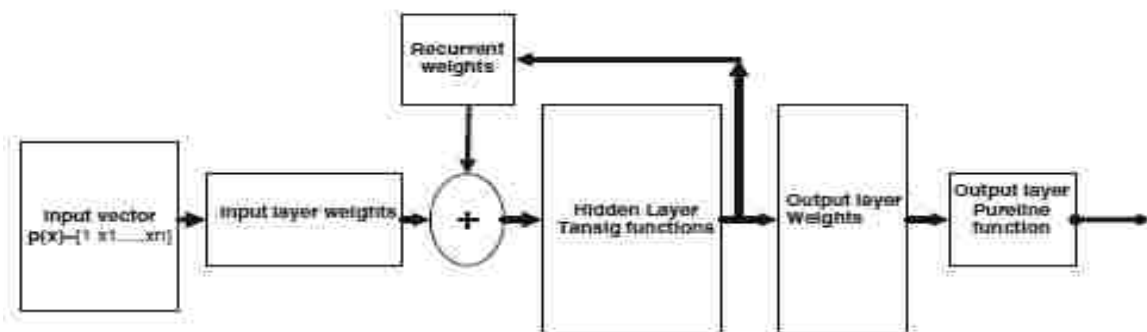


Figure 3. 2 A two-layer recurrent Elman neural network architecture (Al-Assadi, M., Kadi, H. E., & Deiab, I. M. 2010)

In this way, the recurrent connection forces the network to recognize the temporal and spatial patterns in the data. LRNs differences lie in the fact that they are a type of

feed-forward neural network with at least one feedback loop. The feedback loops can involve the use of branches composed of unit-delay elements.

The ANN training algorithms used momentum weight updates as well. They also utilized variable training functions to train the ANN. Table 3.2 shows network performance as measured by root means squared error (RMSE) employing various learning algorithms and architectures.

Table 3. 2 RMSE obtained as a function of the type of architecture and training function (Al-Assadi, M., Kadi, H. E., & Deiab, I. M. 2010)

Training Function	Neural Network Architecture					
	FEN		CFEN		ELM	
	16 neurons	20 neurons	16 neurons	20 neurons	16 neurons	20 neurons
Resilient Back-propagation (RP)	15.60%	9.70%	16.80%	21.10%	13.60%	15.60%
Gradient descent (GD)	53.50%	23.10%	34.50%	40%	17.62%	35.10%
Gradient descent with momentum (GDM)	15.70%	17.80%	35.40%	26.80%	19.50%	25.20%
Variable Learning Rate (GDA)	33.10%	19.10%	22.20%	23.10%	34.60%	17.02%
Variable Learning Rate with momentum (GDx)	17.60%	24.30%	23.90%	30.60%	30.40%	18.70%

The training algorithms, number of hidden neurons, and network architectures all influence the result of network prediction. However, the ANNs can still accurately predict the fatigue life of a composite material within a specified error.

In the following year, M. Al-Assadi and his group (2011) published another paper to predict the fatigue life of composite materials that included the stress ratio (Al-Assadi, M., Kadi, H. E., & Deiab, I. M. 2011). With optimized ANNs and stress ratios, they achieved higher accuracy in prediction of the fatigue life of composites materials.

3.1.2 Hany El Kadi et al.

Hany El Kadi et al. (2002) published a method using neural networks to predict the fatigue life of FRPs. Besides Feed-forward Neural Networks (FNN), they also employed

Modular Networks (MN) and Radial Basis Function networks (RBF) to predict the fatigue life of GFRP. In their study, they published a comparison of network performance normalized mean square error, NMSE, for different structures (Kadi, H. E., & Al-Assaf, Y. 2002).

Table 3.3 Comparison of NMSE and r for the various neural network structures (Kadi, H. E., & Al-Assaf, Y. 2002)

	NMSE	r
FNN	0.14273	0.9260
MN	0.05668	0.9726
RBF	0.6600	0.4703
SOFM	0.0743	0.9624
PCAN	0.1402	0.9280

Table 3.3 shows the NMSE and r the correlation coefficient, for various networks. Compared to the classical FFN, MN improves the ability to decompose the modeling task among the other three kinds of models.

3.1.3 J.A. Lee et al.

In a study by J.A. Lee et al. (1999), the constant-stress fatigue life of five CFRPs and one GFRP were predicted (J.A Lee, D.P Almond, B Harris 1999). It is known that neural network can have any number of neurons in a hidden layer, affecting complexity of the ANN. Sometimes, over-sized ANNs can not only have excessive training times but may also negatively affect their prediction accuracy.

Lee's study shows the relationship between root-mean-square error (RMS error) and the number of hidden layer neurons (Figure 3.3).

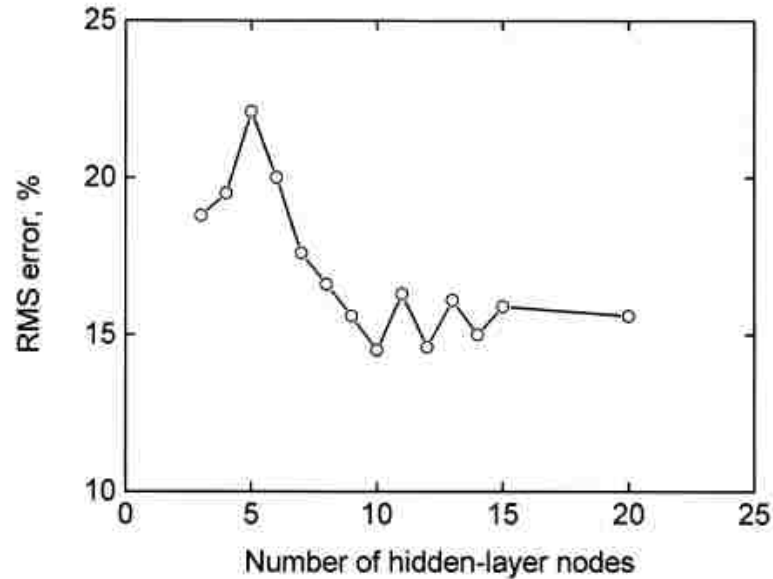


Figure 3. 3 Results of Trials to Establish the Optimum Architecture of The ANN (Nodes (neurons) are in a single hidden layer) (J.A Lee, D.P Almond, B Harris 1999)

In Figure 3.3, the RMS error changes with the number of hidden-layer neurons. More complex ANN structures do not always improve prediction results. To optimize an ANN, like Figure 3.3 shows, the number of hidden-layer neurons should be optimized.

The study incorporated the following parameters as inputs:

- The peak stress in the fatigue cycle, σ_{\max} ;
- The minimum stress in the cycle, σ_{\min} ;
- The stress ratio, $R = \sigma_{\min}/\sigma_{\max}$;
- The stress range, $\Delta\sigma = (\sigma_{\min} - \sigma_{\max})$;
- The mean stress, $\sigma_m = (\sigma_{\min} + \sigma_{\max})/2$;
- The failure probability level or quantiles.

Another important parameter in their work is the inverse relationship of stress ratio, R (the ratio of minimum stress to maximum stress in the cycle, $\sigma_{\min}/\sigma_{\max}$) with RMS error; they show that when R increases, the RMS decreases.

3.1.4 Yousef Al-Assaf and Hany El Kadi

Yousef Al-Assaf, Hany El Kadi (2007) used polynomial classifiers (PC) and recurrent neural networks to predict the fatigue life of composite materials. In their study, they tested five different specimens with different fiber orientation angles (Yousef Al-Assaf, Hany El Kadi 2007). An MTS testing machine was used to measure the experimental data for the two different methods.

PCs use classification, regression, and recognition with remarkable properties as a learning algorithm. It is a method used to predict fatigue life of composites. Its performance is also evaluated by the mean squared error between the output and target values. PCs will be used as ANNs in this study

Elman Neural Networks (ENN) were used, which are basically ELM (as noted previously) with the addition of a feed-back connection from the output to the hidden layer.

Table 3. 4 Mean absolute error between experimental and predictions results using ENN and PC (Yousef Al-Assaf, Hany El Kadi 2007)¹

	First order input, $p_1(x)$		Second order input, $p_2(x)$	
	ENN	PC	ENN	PC
19°	1.4436	1.0224	1.1778	1.5322
45°	0.7913	0.8451	0.8073	0.7985
71°	0.9275	1.1078	0.8829	0.6374

As the authors noted, the polynomial classifier is a valuable tool useful for predicting the fatigue life of composite materials but it cannot satisfy any a priori decision. PCs can lead to a repeatable result with less computational requirements but ANNs are generally suitable for all kinds of prediction tasks.

3.1.5 Others

Recently, many researchers have used neural network models to predict fatigue life of composite materials. Ke-lu Xiang et al. (2014) used ANNs to predict the fatigue life of natural rubber composite (Ke-Lu Xiang, Pu-Yu Xiang, You-Ping Wu 2014). Additionally, Abderrezak Bezazi et al. (2007) used Bayesian trained ANNs to predict the fatigue life of sandwich composite materials under flexural tests (Abderrezak Bezazi, S. Gareth Pierce, Keith Worden, El Hadi Harkati 2007). Anastasios P. Vassilopoulos et al. (2007) also published their method to predict the fatigue life of composite materials using ANNs (Anastasios P. Vassilopoulos, Efstratios F. Georgopoulos, Vasileios Dionysopoulos 2007).

3.2 Predicting Composite Material Strength Using ANNs

3.2.1 Mohammed A. Mashrei et al.

In addition to fatigue life, strength is another important material property for composite materials required by mechanical engineers. Mohammed A. Mashrei et al. (2013) used ANNs to predict the bond strength of FRP-to-concrete joints (Mohammed A. Mashrei, R. Seracino, M.S. Rahman 2013). They proposed that back-propagation neural networks are a reliable means of predicting the bond strength of FRP-to-concrete joints. Before ANNs, many models had been used to predict the strength of this type of joint with poor results. In their study, they used 150 experimental CFRP-to-concrete joint test specimens to train and test neural networks. The backpropagation neural network (BPNN) was established in MATLAB with a sigmoid transfer function in the hidden layer and linear function in the output layer.

Table 3. 5 Comparison of bond strength prediction (Mohammed A. Mashrei, R. Seracino, M.S. Rahman 2013)

Model	Average P_e/P_t		STDEV P_e/P_t	
	Training	Testing	Training	Testing
BPNN	1.00	0.99	0.10	0.09
Lu et al. [13]	0.98	1.04	0.21	0.20
Dai et al. [34]	0.85	0.90	0.18	0.17
Wu et al. [7]	1.21	1.30	0.41	0.42
<i>fib</i> [35]	0.83	0.87	0.17	0.13
CNR-DT2000/2004 [37]	1.40	1.48	0.29	0.26

Table 3. 6 Comparison of correlation coefficient, R (Mohammed A. Mashrei, R. Seracino, M.S. Rahman 2013)

Model	R	
	Training	Testing
BPNN	0.99	0.99
Lu et. al. [12]	0.87	0.89
Dai et al. [34]	0.90	0.92
Wu et al. [7]	0.77	0.80
<i>fib</i> [35]	0.93	0.94
CNR-DT2000/2004 [37]	0.91	0.92

This paper presented a comparison with existing models that predicted the strength of FRP-to-concrete joints. From Table 3.5, the average and standard deviation of the ratio of the experimental bond strength to the predicted can be seen. The BPNN provided 1.0 and 0.99 average ratio for training and testing data sets, with a standard deviation of 0.1 and 0.09, respectively. These results indicated that the prediction of BPNN is much more accurate and reliable than the other five models. In table 3.6, the coefficient, R , is 0.99 for the BPNN training and testing data. This result is better than those of the other five models which indicate the benefit of the BPNN.

3.2.2 Z. Zhang et al.

Z. Zhang et al. (2002) used ANNs to predict the tribological properties of short fiber reinforced thermoplastic (SFRT) materials. Their research is a typical example for people who use ANNs as a research tool to model the relationship between multiple inputs and outputs (Z. Zhang, K. Friedrich, K. Velten 2002).

More and more SFRT materials are being used as structural materials because of their ability to be transformed into complex shapes. However, their characteristic mechanical properties cannot match that of continuous fiber reinforced composites. Mathematical models derived from experimental data have generally been a useful method to model composite material behaviors. But, for SFRT, micromechanical theory, the relevant mathematical model, can be used only in very limited cases. For predicting SFRT behaviour, the artificial neural network is an ideal tool to use for modeling complex conditions occurring during tribological test situations.

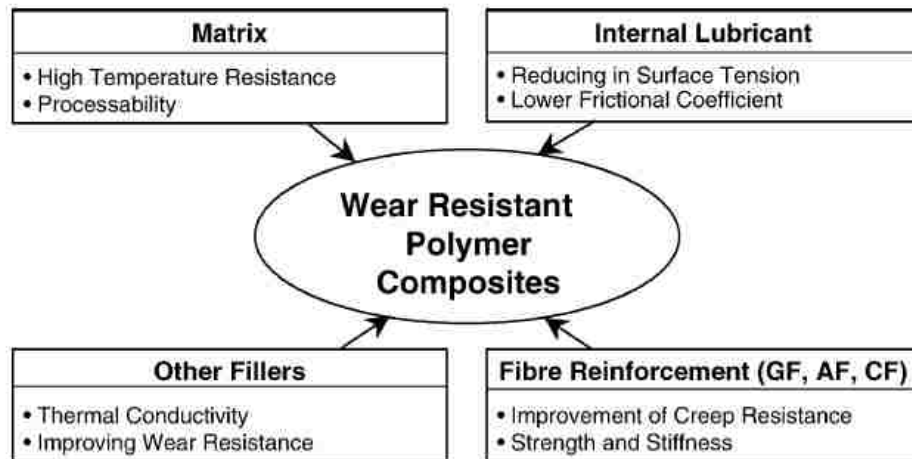


Figure 3. 4 Schematic presentation of how to design the composition of wear resistant polymer composites (Z. Zhang, K. Friedrich, K. Velten 2002)

Figure 3.4 is the schematic presentation of the design of wear resistant polymer composites. It is a complex task to simulate the mechanical properties related to each parameter. From Fig. 3.4, the complexity to produce a wear resistant polymer composite is apparent. Enhancing the hardness, stiffness and compressive strength and reducing the

adhesion are traditional methods used to improve the friction and wear behaviors of polymeric materials. To increase the creep resistance and the compressive strength, short aramid fibers, glass fibers or carbon fibers play an important role. In Zhang's research, there are three categories totaling ten inputs and two outputs: specific wear rate/wear volume and frictional coefficient (Figure 3.5).

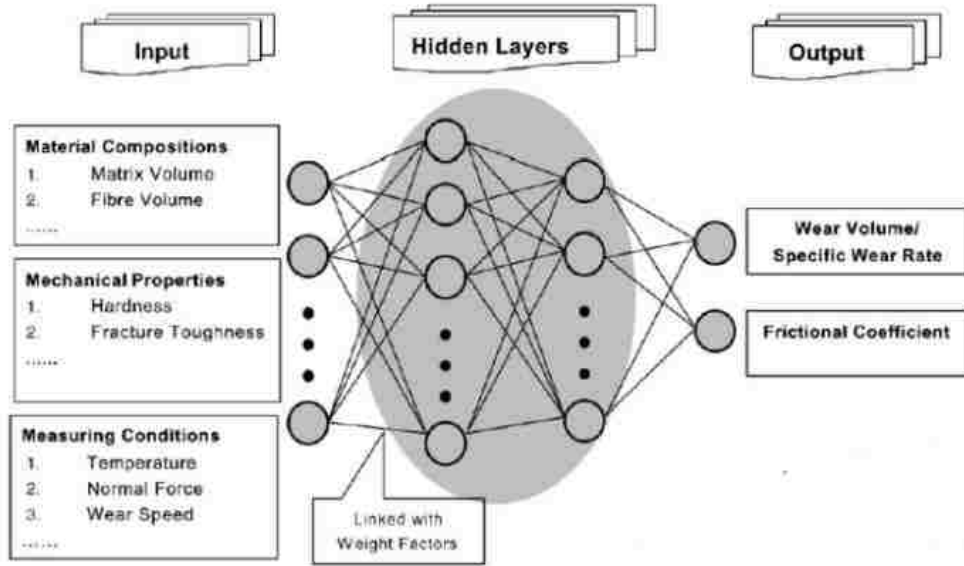


Figure 3.5 Input Data, Output Data and Schematic Construction of an ANN for Correlating Tribological Properties with Testing and Material Parameters (Z. Zhang, K. Friedrich, K. Velten 2002)

In their paper, they compared the influence of changing different training parameters on neural network prediction capabilities. To evaluate and optimize the results of the ANN, the researchers used the coefficient of determination B as a quality measure. B is defined by:

$$B = 1 - \frac{\sum_1^M [O(p^{(i)}) - O^{(i)}]^2}{\sum_1^M [O^{(i)} - O]^2} \quad (3-1)$$

Where $O(p^{(i)})$ is the i^{th} predicted wear characteristic; $O^{(i)}$ is i^{th} measured value; O is the mean value of $O^{(i)}$; M is the number of test data points.

The higher the coefficients were, the better the output approximation capabilities of the ANN. In their paper, they first compared the influence of the learning rules of ANNs. Five training algorithms are used; the BR (Bayesian Regularization) and CGB

(Powell-Beale Conjugate Gradient algorithm) both show a high predictive quality. Secondly, they tried various ANN structures to compare the results based on the CGB algorithm. In their test, the $9 - [15 - 10 - 5]_3 - 1$ ANN structure provided excellent results for the coefficient B . In this way, the very complex relationship of wear performance of the short fiber composites becomes much clearer. In addition, they also tried to train a $9 - [4]_1 - 1$ ANN configuration for evaluation. The results of this configuration were not stable. Although the simple ANN configuration had poorer accuracy (and instability), it could be used for fast estimations with small training datasets.

In their research, the number of the datasets, the structure of the configuration, and the rules of the learning processes influenced the results of the prediction. For an excellent prediction in capabilities of this black-box tool, the network design and training processes require attention.

3.3 Predicting Other Properties of Composite Materials with ANNs

Glass fiber-reinforced plastic composite material (GFRP), as introduced before, is one of the most widely used composite materials due to its high mechanical properties with low cost and light weight. Ömer Erkan et al. (2013) used artificial neural networks to predict the damage factor in the end milling process of GFRP (Erkan, Ömer, Işık, B., Çiçek, A., & Kara, F 2013). In their research, they used the basic logistic sigmoid transfer function. The inputs of their experiment are the number of flutes, the cutting speed, and the depth of cut and feed rate. The output parameter is the damage factor which can be calculated by the nominal slot width and the maximum damage slot width of the specimens. They also tried five different algorithms to find a most suitable training method for this prediction. Their study is a good example for predicting the results of machining composite materials. Instead of high the cost of long-term experiments, the ANN prediction is a better choice for a mechanical engineer to analyze the properties of machining composite materials.

Yung-Kuang Yang et al. (2012) used ANNs to optimized some approaches that integrate the D-optimal Mixture Design (DMD) method, Genetic Algorithm (GA), and Simulated Annealing Algorithms (SAA). Using this concept, they successfully determined the optimal mixture ratios for short glass fiber (SGF) and polytetrafluoroethylene (PTFE) reinforced polycarbonate (PC) composites (Yung-Kuang Yang, Rong-Tai Yang, Chorng-Jyh Tzeng 2012).

DMD is a technique of DOE for optimizing complex processes. GAs are usually used for parameter optimization. In addition, SAA is another algorithm for optimization problems. As ANNs are powerful and practical models for complex non-linear systems, Yung-Kuang Yang et al. combined the BPNN with SAA and GA in their study.

In their study, they used DMD to plan the experiments and implemented both GA and SAA approaches combined with BNPP to optimize the simulation.

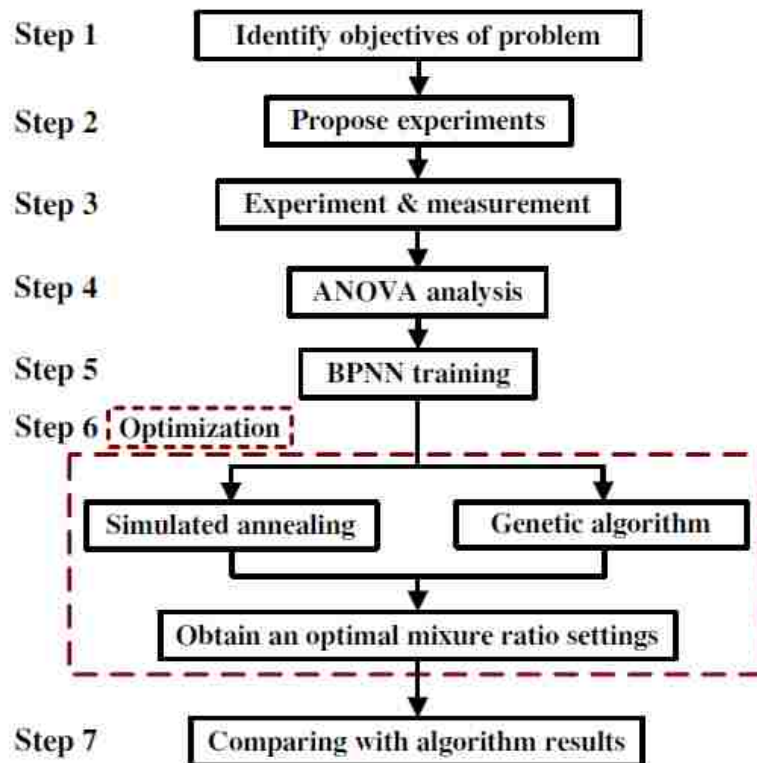


Figure 3. 6 The flow chart of BPNN training and optimization (Yung-Kuang Yang, Rong-Tai Yang, Chorng-Jyh Tzeng 2012)

As Figure 3.6 shows, the optimization process has seven steps. In step 2, a DMD method is used to design and conduct the experiments. And in step 6, the GA and SAA was use to obtain an optimal mixture ratio setting by combine the BPNN separately. After that, the results by each prediction method were compared.

In this paper, the BPNN successfully predicted the mechanical properties of ultimate strength, flexural strength and impact resistance. At the same time, the combination of BPNN/ GA and BPNN/SAA methods can obtain very good results for prediction. This is a good example for combining the use of ANNs with other methods in practice.

3.4 Predicting Material Performance using Neural Network Sensitivity

A commonly used method to predict the relevance of artificial neural network inputs on outputs is sensitivity analysis. The sensitivity of the output to changes in the inputs is a function of the trained network's weights and biases. (Goldman 2001)

RW. Goldman, in his PhD dissertation gives the function of the sensitivity analysis. The sensitivity analysis of the ANN would provide the starting point for designing the experiments. In his research, a two hidden-layer feed-forward backpropagation network structure was chosen to train the data using tan-sigmoid activation functions in the hidden layer and linear function in the output layer. The research provided the equation of the sensitivity for a tansig-output three layer network as the following:

$$\frac{\partial y_i}{\partial p_j} = 4 \sum_{k=1}^h w_{i,k}^2 \cdot w_{k,j}^1 \cdot e^{-2M_k} (1 + e^{-2M_k})^{-2} \quad (3-2)$$

$$M_k = \sum_{c=1}^n w_{k,c}^1 \cdot p_c + b_k \quad (3-3)$$

Where M_k is the sum of the weighted input, y_i is i^{th} output and p_j is j^{th} input of the network. This result is independent of time and it computes the effect on the i^{th} output y_i due to changes in j^{th} input p_j at a discrete point in time.

w^1 and w^2 are the weights of the first and second layer of the network. The concept to

obtain this equation is from the derivation of the mean square error to all weight of the network as section 2.4.2 introduced.

The sensitivity of outputs to inputs can be a direct indicator of the neural network-based relationship between outputs and inputs. The larger the sensitivity is, the more influence that the input has on the output.

This concept is used in the current research and the final result of the sensitivity study will be introduced in Chapter 5.

CHAPTER 4 COLLECTION AND SELECTION OF DATA

ANNs, used for prediction, are often based on large amounts of data, especially when nonlinear pattern recognition is involved. Preparing the data is the first step in neural network design and development. In order to achieve optimal results, data should be properly organized. This chapter will introduce how the data was managed for the purposes of the current research.

4.1 Data Resource

As introduced in Chapter 1, FPC is a research center where composite materials are designed and manufactured. In this advanced workshop, there are four complete machine lines for producing composite materials specifically using carbon and glass fibers. These lines include SMC-D, LFT-D, injection, and RTM. As discussed, these lines are all the state of the art in composite material manufacturing.

This research focuses on the processing parameters of the LFT-D line. On January 12, 2015, FPC ran a group of trials for CFRP. The matrix used in this trial was polyamide (Ultramid 8020HS) and the fiber was carbon fiber tow (PANEX 35). The properties of these two materials are in Appendix A.1 Material Properties.

The LFT-D trials were used to produce fiber reinforced thermoplastic plaques for an automotive company. The test plaques were made to determine if the performance of the material would satisfy the requirements for the vehicle parts.

The data for this research is organized into two sets: inputs and targets. The processing parameters were identified as the inputs and the performance testing raw data as the targets.

4.2 LFT-D Processing Parameters

The LFT-D process, as mentioned before, is a highly automated process for thermoplastic-matrix composite material production. The LFT-D line at FPC is instrumented so that processing parameters can be measured throughout the process. There are three important sections (ZSG, ZSE, and PAZ) that make up the LFT-D line. The processing parameters are set before each run.

There are 54 LFT-D processing parameters used in current research, including temperature, speed, torque, and flow rate. These 54 parameters can be divided into 3 categories according to their location in the system: ZSE, ZSG, and PAZ.

Additionally, the 54 input parameters are set for six different trials classified by the fiber volume weight for each trial. Thus, the data is organized as shown in Figure 4.1. The first step to pre-processing the data was to remove data that is not available for all trials or is not relevant. The machine run time recorded was removed at first because there was no need to consider it. As some parameters showed a zero reading from the beginning to the end, they were removed as well.

After pre-processing, there were 52 different processing parameters that were available as inputs for the network. However, to accelerate the speed and improve the accuracy of the network results, the data often needs to be normalized before network training and simulation.

4.2.1 Mean Data

As the LFT-D line runs, parameters are recorded once by its own computer every second. For each parameter in a trail, there are at least 1,700 values in the data set.

To consolidate data for this problem in this research, the parameters in each fiber volume weight trial are organized by mean value. In this way, the processing parameters become a 6 x 52 matrix. Six rows of data are the mean values of each parameter for each

of the different fiber weights, and 52 columns of data correspond to the 52 processing parameters shown in Appendix A.3

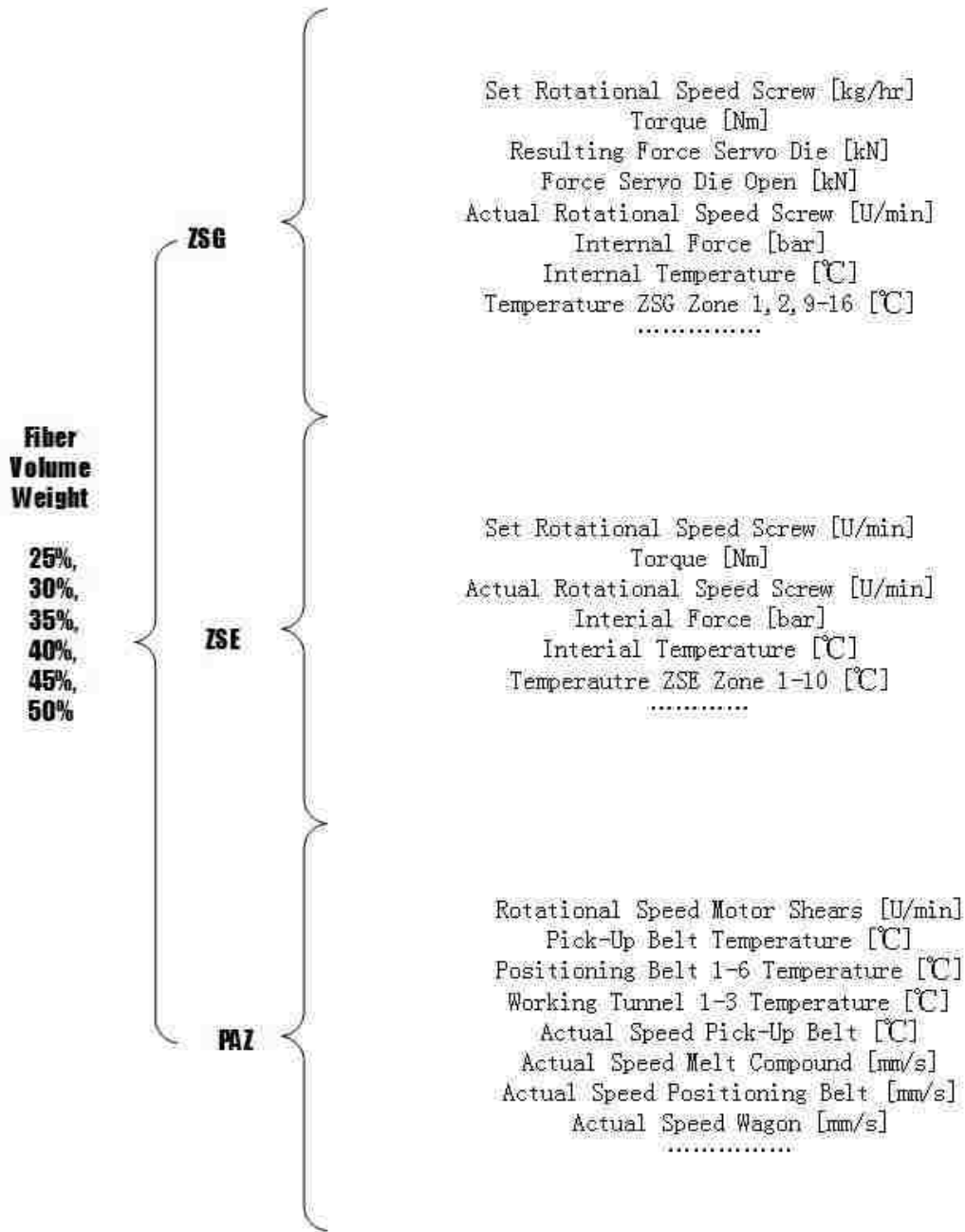


Figure 4. 1 LFT-D processing parameters classification

4.2.2 Data Standard Deviation

Standard deviation is a statistic that describes variation, or scatter, in the data. Similar

to the mean values, the standard deviation values are organized into a 6 x 52 matrix that describes the 52 processing parameters for six different fiber volume weights. The standard deviation of the 52 processing parameters can be found in Appendix A.2

4.2.3 Data Normalization

Data normalization communizes the range of all the data channels. This practice reduces network sensitivity to errors in order for the neural network to provide more stable results. Often, data is normalized before data enters the network and is reverted to its original scale after the network training.

There are several kinds of statistical normalization. Network activation functions such as *logsig* and *tansig* in MATLAB have ranges 0 to 1 and -1 to 1, respectively. To normalize data to the range [0, 1], equation (4-1) is used:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (4-1)$$

Where X' is the normalized data, X is the initial data, and X_{min} and X_{max} are the minimum and maximum values respectively of the dataset.

As previously mentioned, the *tansig* function in network training processes data to the range [-1, 1]. MATLAB offers a function *mapminmax* to normalize the data to [-1, 1] before training. This function in MATLAB is frequently used and its algorithm is shown as the following equation (4-2):

$$X' = \frac{(Y_{max} - Y_{min}) * (X - X_{min})}{(X_{max} - X_{min}) + Y_{min}} \quad (4-2)$$

This function in MATLAB is a more flexible version of the former equation. Y_{max} and Y_{min} are the set maximum and minimum values for normalization. For example, in $X' = \text{mapmaxmin}(X, 0, 1)$, the $Y_{max} = 1$ and $Y_{min} = 0$ respectively. In the other word, the function can normalize the data in range $[Y_{min}, Y_{max}]$. In this way, the data can be normalized to whatever ranges that the research requires.

4.2.4 Reversing the Normalization

Opposite to the data normalization, reversing the process will restore the normalized data to its initial scale.

When data are normalized by the *mapminmax* MATLAB function, the maximum and minimum values can be stored by the function as $[X_n, X_s] = \text{mapminmax}(X, Y_{min}, Y_{max})$. X_n is normalized data and X_s holds the values of the extrema.

The organized data of processing parameters are in the table in Appendix A.2. The 6 x 52 dataset shows the normalized processing parameters which are also the input data of the neural network.

After training the network, the network outputs are reverted back to the initial scale of the target data.

4.3 Material Test Data

As mentioned section 4.2, there are six groups of LFT-D process parameters for producing composite materials having six different fiber volume weights. Therefore, there are six different groups of FRPs used in this research.

Figure 4.2 (a) shows the charges which are put into the press machine to form the panels for testing as shown in Figure 4.2 (b). The charge is the material produced by the LFT-D. In this research, the panels underwent tension testing to obtain the Young's modulus and tensile strength of the material.



(a) (b)
 Figure 4.2 Carbon fiber-reinforced thermoplastic charge and panel by LFT-D

As the photos show, the materials are labeled and there are six different fiber volume weights from 25% to 50% in 5% increments.



Figure 4.3 Carbon fiber-reinforced thermoplastic panels produced by LFT-D

Figure 4.3 shows all six fiber weight panels produced by LFT-D. As the photos shown, the visual quality of the panels' surface finish of the panels is an indicator of the panel quality. Researchers at the University of Western Ontario conducted the experimental testing for these materials. The test data was provided to all researchers in the ICRC. Appendix B.1 includes the specifications of the specimens for the tension test.

4.3.1 Young's Modulus and Tensile Strength

The tension test data recorded included test time, extension, strain, load, tensile stress, tensile extension, tensile strain, displacement, position, corrected position, and tenacity. Among this data, for Young's modulus and tensile strength calculations, the load, tensile stress, and tensile strain are needed. The Young's modulus and tensile strength can be calculated using the following the equation (Engineering 2015):

$$(Young's) \text{ Modulus of Elasticity} = \frac{\text{stress}}{\text{strain}} \quad (4-3)$$

$$\text{Ultimate Stress (tensile strength)} = \frac{\text{maximum load}}{\text{original cross-sectional area}} \quad (4-4)$$

In practice, the Young's modulus is often obtained from the slope of the stress-strain curve below the proportional limit. Even though the cross-sectional areas of the specimens after test are not known in the recorded test data, from the relation between load and stress, the original cross – sectional area can be calculated by equation (4-5):

$$\text{stress} = \frac{\text{load}}{\text{original cross-sectional area}} \quad (4-5)$$

Thus, the unknown variable in Equation (4-4) can be calculated using equation (4-5). In this way, the Young's modulus and tensile strength of the materials can be obtained from load, tensile strain, and tensile stress in by the test data.

4.3.2 Test Data Organization

As discussed before, the Young' modulus and tensile strength of the material can be calculated by using the experimental test data from the three equations. Appendix B.2 gives details of the data and these values are used as the target data of neural network as well. There are 54 groups of values in the table including six different fiber volume and two different fiber directions. Each fiber direction has nine test specimens including five 0° and four 90° fiber orientation specimens.

Similar to the processing parameters, the target data needs to be normalized as well.

In this research, to compare the neural network prediction ability, the target data was organized using two approaches.

The first approach to organize the data was to normalize Young's modulus and the tensile strength as discussed before. Then a 54 x 2 dataset is obtained as shown in Appendix B.2. This dataset included the entire target data directly derived from the tension test data. However, using this data to train the neural network produced unsatisfactory results. The details of this problem will be discussed in Chapter 5.

The second approach taken was to classify the data into different groups according to the fiber volume weight and the fiber directions. As there were two different fiber directions and two target variables, Young's modulus and tensile strength, the target data was divided into four groups as shown in Appendix B.3. Using the same normalization method as before, the target data was organized into ranges of [0, 1] or [-1, 1] as shown in Appendix B.4. Finally, mean and standard deviation values for the target data were calculated.

Table 4. 1 Standard deviation and mean of Young's modulus and tensile strength for target data

	MEAN				STANDARD DEVIATION			
	Young's Modulus		Tensile Strength		Young's Modulus		Tensile Strength	
	0°	90°	0°	90°	0°	90°	0°	90°
Trial 1/25%	202.091	109.753	192.125	109.131	35.342	14.555	11.656	7.865
Trial 2/30%	242.125	115.399	212.372	114.690	40.292	21.057	10.323	23.218
Trial 3/35%	244.804	130.118	201.163	128.642	29.907	58.435	15.942	41.083
Trial 4/40%	273.735	140.423	208.122	89.731	44.878	29.197	32.387	13.453
Trial 5/45%	299.430	121.722	201.427	94.109	60.802	22.072	36.523	8.1616
Trial 6/50%	304.886	135.554	227.467	76.464	47.364	20.966	28.004	6.314

Table 4.1 shows the target data after pre-processing in preparation for network training. Organized this way, the six groups of targets (material performance) correspond to the six groups of input data (processing parameters) for material made having six different fiber volume weights.

CHAPTER 5 NEURAL NETWORK PREDICTION

ANNs, as discussed before, are black-box models that can simulate and predict large nonlinear data sets. After pre-processing the input and target data, the network configuration is selected in order to obtain the best performance of network prediction. This chapter describes how to select the neural network configuration. The configuration process includes selection of the transfer functions, training algorithm, number of hidden layers, and number of hidden neurons.

5.1 Function Selection

As discussed in Chapter 2, there are three activation functions often used in neural networks: tansig, logsig, and purelin. Tansig or logsig functions are generally used in the hidden layers as transfer (activation) functions. Linear functions are generally used in the output layer neurons.

As in Xiaochuan Wang's research, when the weights, biases, and network structure are held constant, substitution of each of the three transfer functions will result in nine different performance values for the network. Table 5.1 shows the performance with the error percentage and mean square error of nine different single hidden layer networks (Wang 2.13).

Table 5.1 The error of different networks with different transfer functions (Wang 2.13)

Hidden Layer Transfer Function	Output Layer Transfer Function	Error Percentage	MSE
logsig	tansig	40.63%	0.9025
logsig	purelin	0.08%	0.0001
logsig	logsig	352.65%	181.2511
tansig	tansig	31.90%	1.1733
tansig	logsig	340.90%	162.9698
tansig	purelin	1.70%	0.0107
purelin	logsig	343.36%	143.76334
purelin	tansig	120.08%	113.0281
purelin	purelin	196.49%	99.0121

According to Table 5.1 taken from Wang, it is apparent that when tansig or logsig functions are used in the hidden layer with the linear function in the output layer, the errors are minimized. The highlighted values show that the combination of logsig and purelin activation functions in the hidden and output layers, respectively, result in the lowest MSE. Thus, in this research, the logsig activation function will be used in the hidden layer neurons and the purelin function in the output layer.

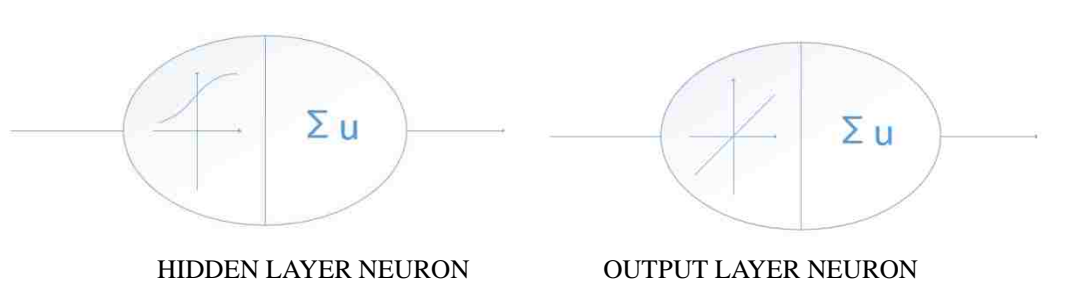


Figure 5.1 Hidden neurons and output neurons schemes

According to the logsig function's characteristic, the input data and target data should be normalized to the range of [0, 1]. This positive range also corresponds to the fact that the data does not exhibit any negative polarity, with all values being greater than zero. Then the hidden neurons and the output neurons in the network are shown in Figure 5.1.

5.2 Training Method Selection

As discussed in the previous chapter, neural networks can use many kinds of training methods and even use some in combination with each other. Many of the training methods are defined already. However, for different input and output data, to find a suitable method is an important step of the research. According to the training method offered by MATLAB, there are at least 19 types of networks and 18 types of learning functions. Not all of the network types provided is valid in this research. Most of them are used in special situations. This section will compare some selected network types and learning functions, then the training method for the ANN development will be determined.

5.2.1 Network Types

MATLAB offers a toolbox for neural network simulation with all kinds of network types as well as training functions and learning functions. Figure 5.2 is the screenshot of network toolbox in MATLAB; the drop-down list is used for network selection. Once the type, the functions, and the hidden layer are determined, the network configurations can be viewed as shown.

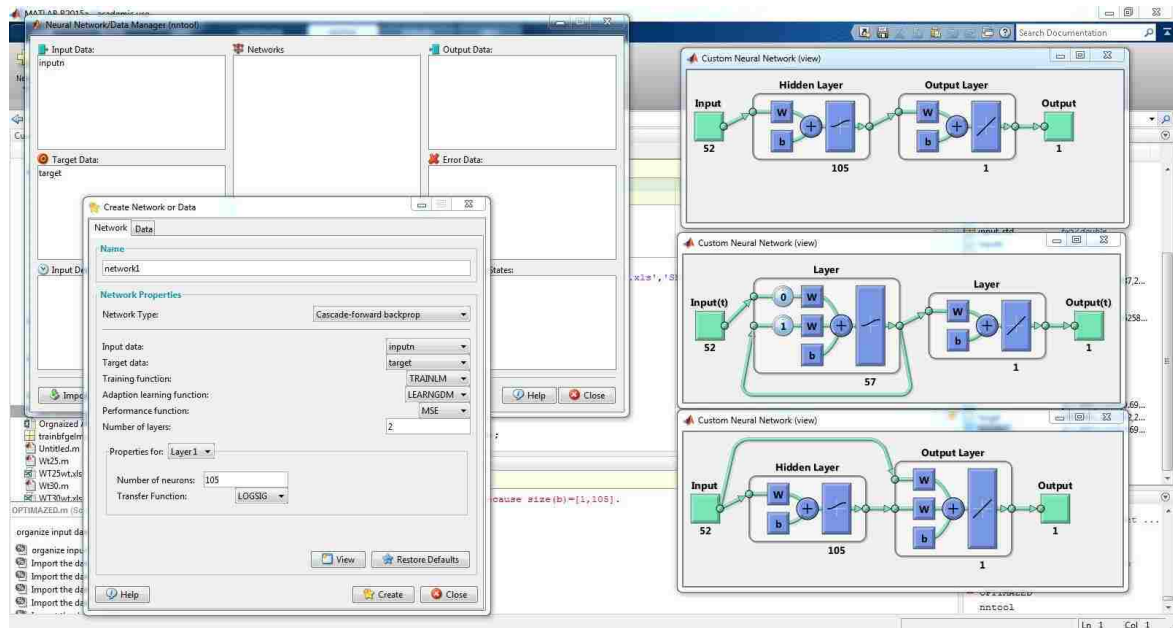


Figure 5.2 MATLAB neural network programming window

There are three common types of networks used for nonlinear data prediction: feed-forward backpropagation (FFB), cascade-forward backpropagation (CFB), and Elman backpropagation (EB). As introduced before, backpropagation is a method typically used to adjust the weights and biases of the neural network based on the error between output and target. These three types of neural network all use the backpropagation learning method; the difference lies in the configuration of the weights. The three network configurations shown in Figure 5.2 are FFB, CFB, and EB, from top to bottom.

The FFB network has weights applied to inputs as they pass to the hidden layer. Each

subsequent layer has weights and a bias applied to data passing from the previous layer. The CFB network also consists of hidden layer and output layer. Actually, for more than one hidden layer in CFB, the weights are increased by the hidden layer number. If there are S layers including hidden layers and output layer, then the number of weights in S^{th} layer is $(I+2+\dots+S)$. Figure 5.3 is the two hidden layer CFB network.

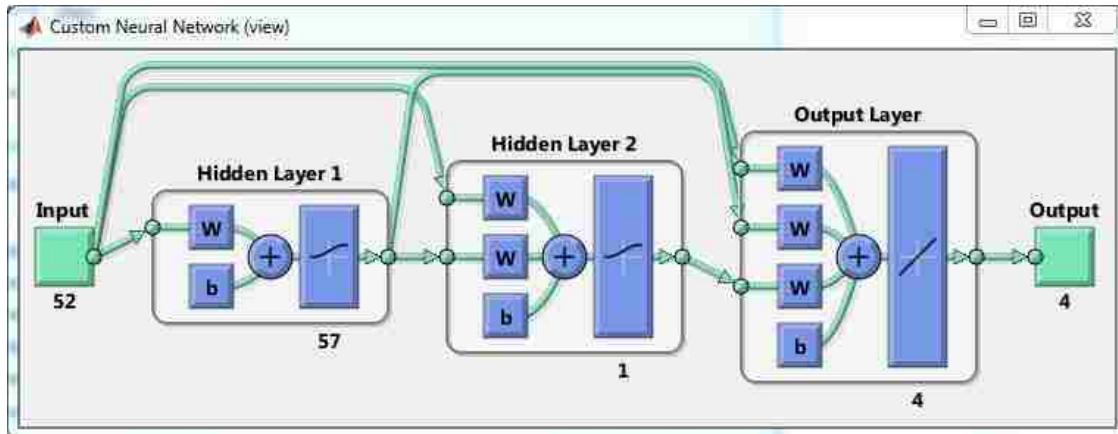


Figure 5.3 Configuration of a 2-hidden-layer Cascade-forward network

The Elman network is another commonly used neural network type. The first layer has weights coming from the input. And each subsequent layer has weights coming from the previous layer. The difference is that all hidden layers have a recurrent weight as Figure 5.4 shows.

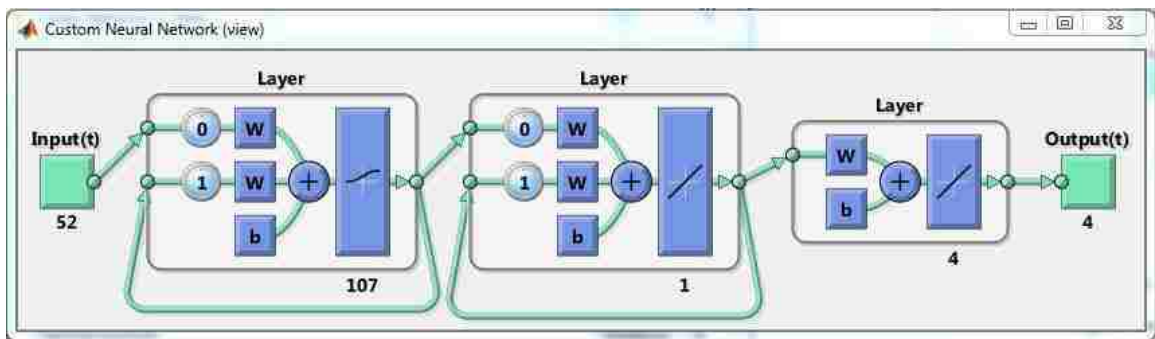


Figure 5.4 The configuration of a 2-hidden-layer Elman network

5.2.2 Adaption Learning Functions

Adaption Learning Functions, in MATLAB, refer to the training method discussed

previously. The learning functions can also be selected when using the MATLAB toolbox in the drop-down menu. MATLAB provides 18 different learning functions. The default and most commonly used one is Levenberg-Marquardt. As introduced before, this second-order training methods is the most efficient algorithm used in neural network training. In order to find the best learning function for this research, Table 5.2 shows the performance (MSE) of all 18 learning functions for each of the three mentioned network types in MATLAB.

Table 5. 2 Performance of each learning function with respect to the training functions

Learning Function \ network type	Feed-Forward Backprop.	Cascade-Forward Backprop.	Elman Backprop.
Learn_b	0.35	1.66	0.36
Learn_bu			
Learn_buwb			
Learn_bfg	0.7271	2.7	0.62145
Learn_br	0.11818	0.045	
Learn_c	0.666	4.09	1.92
Learn_cgf	2.2	0.54	0.218
Learn_cgp	0.14	2.1719	0.03
Learn_gd	1.293	2.1196	0.16
Learn_gdm	0.34026	20.8	0.199
Learn_gda	0.32053	2.1	0.126
Learn_gdx	0.47	0.38	0.46
Learn_lm	0.75	1.15	
Learn_oss	0.002	0.7	0.15
Learn_r	0.7	0.8	0.8
Learn_rp	0.9	2.3	2.36
Learn_ru			
Learn_scg	0.5	0.9	0.35

The first column in Table 5.2 is the selected learning function. The blank entries indicate that the function is not a valid option for the network type for the research data being used. From Table 5.2, most of these learning functions give good performance as indicated by the very small MSE values.

Considering the network performance and the error between the network output and target data, there are nine combinations of networks that can be further trained in this

research. Table 5.3 shows the available learning functions for each of the three network types.

Table 5.3 Alternative options of the network types

Learning Functions	Feed-Forward Backprop.	Cascade-Forward Backprop.	Elman Backprop.
Learn_cgf			o
Learn_cgp	o		
Learn_gd			
Learn_gdm	o		o
Learn_gda	o		
Learn_gdx	o		
Learn_lm	o	o	
Learn_scg			o

5.2.3 Comparison of Network Types

As Table 5.3 indicated, there are five different learning functions used with feed-forward backpropagation networks: three used with Elman backpropagation, and only one used with cascade-forward backpropagation.

5.2.3.1 Learning Function Algorithms

5.2.3.1.1 Traincgf and Traincgp

The traincgf learning function is the conjugated gradient backpropagation with Fletcher-Reeves updates (MATLAB n.d.). Traincgf can be used to train any network as long as its weight, network input, and activation functions have derivatives. The backpropagation in traincgf is used to calculate derivatives of performance with respect to the weight and bias variables X . X is then adjusted according to the equation below;

$$X = X + a * dX \quad (5-1)$$

Where dX is the search direction and can be represented as

$$dX = -gX + dX_{old} * Z \quad (5-2)$$

The parameter is used to minimize the performance along the search direction. gX is

the gradient. The parameter Z , for Fletcher-Reeves variation of conjugate gradient, can be computed as:

$$Z = \text{normnew}_{sqr} / \text{norm}_{sqr} \quad (5-3)$$

Where, the norm_{sqr} is the norm of the square of the previous gradient and normnew_{sqr} is the norm of the square of the current gradient. `Traincgf` is a very fast way to train the network with large a dataset. However, this function only performs very well when using Elman backpropagation.

The `traincgp` learning function is a method that updates weights and biases according to the conjugate gradient backpropagation with Polak-Rbiere updates. The difference between `traincgf` and `traincgp` is the formula to compute the parameter Z . In `traincgp` function, Z is computed as:

$$Z = ((gX - gX_{old})' * gX) / \text{norm}_{sqr} \quad (5-4)$$

Where gX_{old} is the gradient from the previous iteration. The `traincgp` learning function is also a very fast training method and can get very good performance, achieving 0.03 by using Elman backpropagation; however, the error between network outputs and target values is still very large.

5.2.3.1.2 `Traingdm`, `Traingda`, and `Traingdx`

These three learning functions are all gradient descent methods for training neural networks. `Traingdm` adjusts the weights and biases according to gradient descent with momentum. `Traingda` is used to adjust the weights and biases according to the gradient descent with an adaptive learning rate. These two methods are the very basic ways to update the weights and biases for network training. These algorithms were previously introduced in Chapter 2.

`Traingdx` is a network training function that updates weights and bias values according to gradient descent with momentum and an adaptive learning rate (MATLAB n.d.). This means that `traingdx` combines these two methods together as a network learning process. According to the function given by MATLAB, the formula is expressed

as follows:

$$dX = mc * dX_{prev} + lr * mc * dperf/dX \quad (5-5)$$

Where dX_{prev} is the previous change to the weight or bias, mc is the momentum constant, lr is the learning rate, and $dperf/dx$ is the gradient of MSE over inputs.

For each epoch, if performance decreases toward the goal then the learning rate is increased by the factor lr_inc in MATLAB. If the performance increases by the factor max_perf_inc (maximum performance increase), the learning rate is adjusted by the factor lr_dec .

These three learning functions all provide good performance and small errors between targets and outputs. However, compared to the results of training, the gradient descent only performed well using the feed-forward backpropagation network.

5.2.3.1.3 Trainlm

Trainlm is a network training function in MATLAB that updates weights and biases according to the Levenberg-Marquardt optimization method. The LM method is not the fastest and best performing, but after several trials of training, LM is the most stable method. As introduced before, LM algorithm is designed to approach second-order training speed.

5.2.3.1.4 Trainscg

The trainscg learning function performed well on MSE and gave small errors by using it with Elman backpropagation network. Trainscg is a network training function that updates weights and biases according to the scaled conjugate gradient method. The algorithm of trainscg is similar to traincgp and traincgf. The difference is trainscg does not perform a line search at each iteration.

5.2.4 Section Summary

Due to a large number of network types and learning functions, determining the best network is a challenge. According to the comparisons discussed, FFB networks are

simplest but can provide very good performance. Additionally, in order to calculate the sensitivity of the neural network after training, the FFB is the best choice for this research.

In the Section 5.4, a comparison will be made among several learning functions using a FFB network in order to examine the results of network prediction. The selected learning functions are `traincgp`, `traingdx` and `trainlm`.

5.3 Hidden Layer Configuration

For ANNs, the configuration is a key factor affecting the training results. The better the training result is, the better the capabilities of the network. Except for the activation functions of the neural network, when the input and output are selected, the most variable factor is the hidden layer; factors requiring determination include the number of hidden layers and the number of hidden neurons per layer.

5.3.1 Number of Hidden Neurons

The main difference comparing ANNs in this research is the number of the neurons and the number of layers. To compare the effect of the number of hidden neurons, six different architectures of MLPs were used; all were 1- x -1 neural networks. The x parameter was set to equal 1, 2, 3, 10, 20 and 90. The parameters for all six ANNs were the same. (Figure 5.5)

The results are plotted in Appendix C.1. The plots show that the number of hidden layer neurons can make very significant difference in network performance. When x equal 1, 2, 3 and 10, the errors were similar and close to zero. Among the six plots, the 1-3-1 result is the most ideal one.

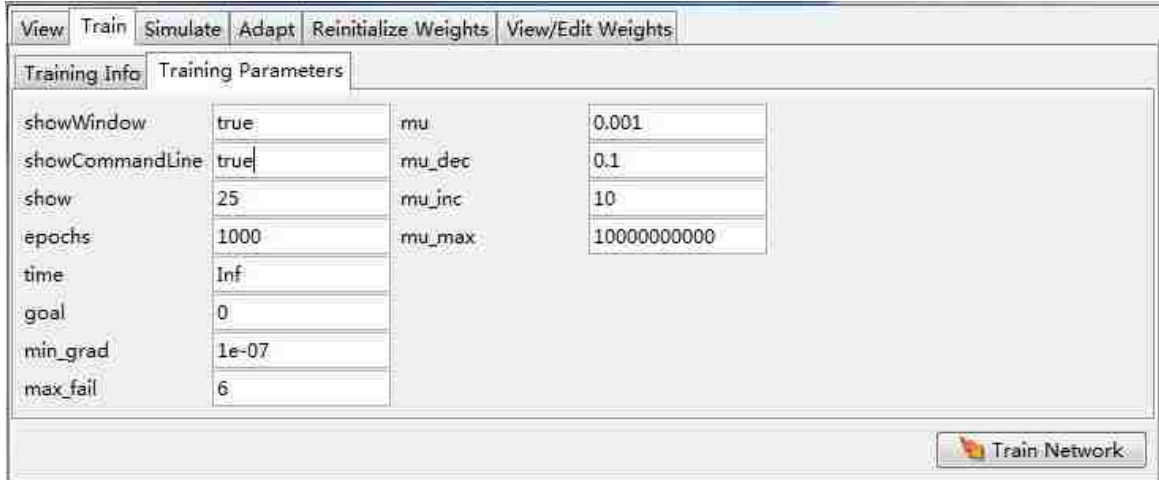


Figure 5. 5 Training parameters

In neural network architecture, the weights provide flexibility by introducing degrees of freedom. More hidden neurons bring more weights to the neural network, thus increasing a model's degrees of freedom. As a result, the 1-90-1 and 1-20-1 networks closely fit the data. However, with regard to predicting trends, the larger network over-fits the data. As shown in Figure 5.6, the 1-3-1 network provides the best result with regard to low training error and good ability to generalize the behaviour of the data.

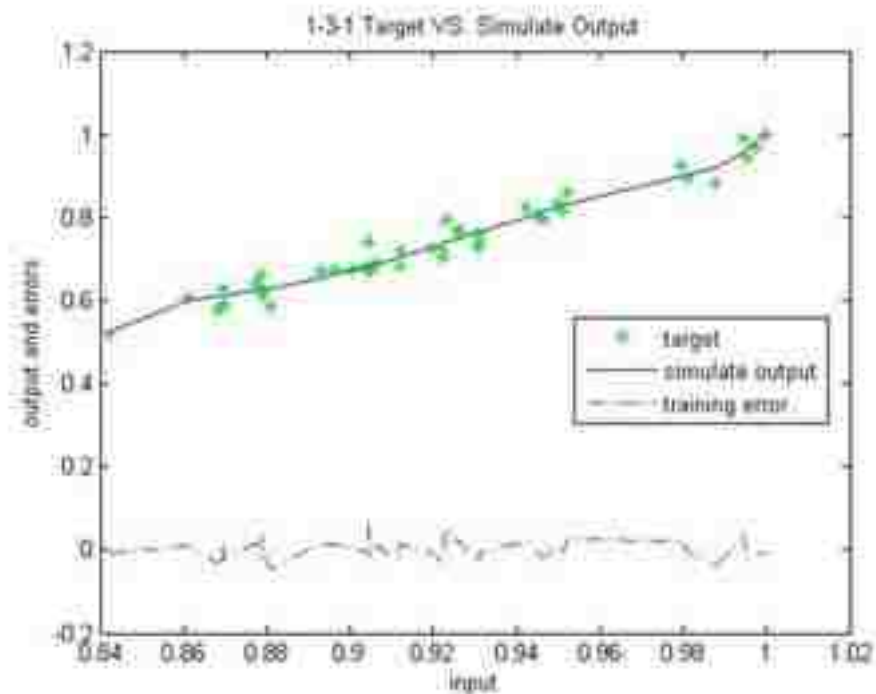


Figure 5. 6 Best results compare with expected curve

There are several methods to deciding the number of hidden neurons. Most commonly, Hecht-Nielsen's $2n+1$ method is a popular way to determine the number of hidden neurons, where n is the number of input neurons (Hecht-Nielsen 1987). They report that it is a good method to determine the number of hidden neurons in a single hidden layer network, but not necessarily for multiple hidden-layer neural networks.

5.3.2 Number of Hidden Layers

To compare results for different numbers of hidden layers, two different ANN architectures were used to train and simulate the concrete data included in Appendix C.2 (Concrete Data.xls 2008). Smaller data sets do not generally require multiple hidden layers; there are eight inputs in this data set. Before training, all inputs and targets were normalized by the same method introduced in Chapter 4. The range of input and target are all [0, 1]. The training parameters were set as in Figure 5.7:

Training Info		Training Parameters	
showWindow	true	mu	0.001
showCommandLine	false	mu_dec	0.1
show	25	mu_inc	10
epochs	1000	mu_max	10000000000
time	Inf		
goal	0		
min_grad	1e-07		
max_fail	10		

Figure 5. 7 Training parameters for different number of hidden layers

For one hidden layer, the number of neurons was set using the $2n+1$ rule (for n = the number of input neurons). In this case, there will be $2 \times 8 + 1 = 17$ hidden neurons. For two hidden layers, the numbers of neurons were determined by the method proposed by Stathakis (Stathakis 2009). The function to calculate the number of hidden neurons in the

different layers is shown below:

$$\text{First layer: } \sqrt{(m + 2)N} + 2\sqrt{N/(m + 2)} \quad (5-6)$$

$$\text{Second layer: } m\sqrt{N/(m + 2)} \quad (5-7)$$

Where m is the number of output neurons and N is the number of test samples used in training. Thus, for two hidden layer network, the first layer neuron should be 47 and the second layer neuron should be 31. The architectures of these two different NNs are shown below:

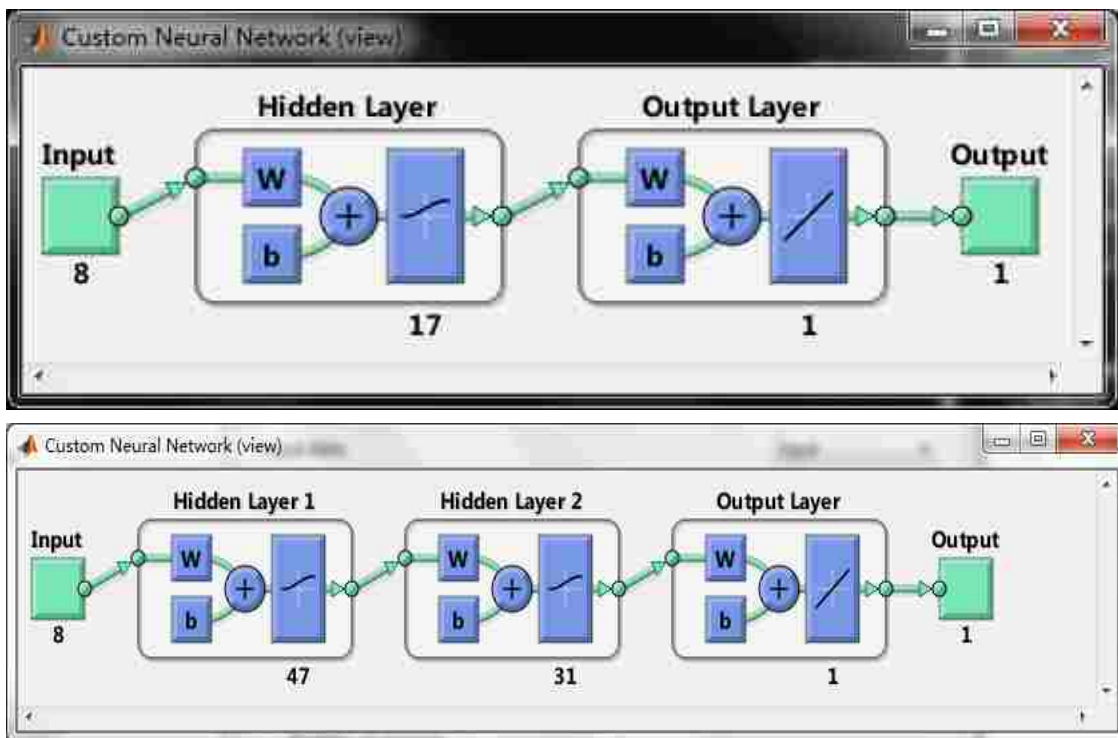
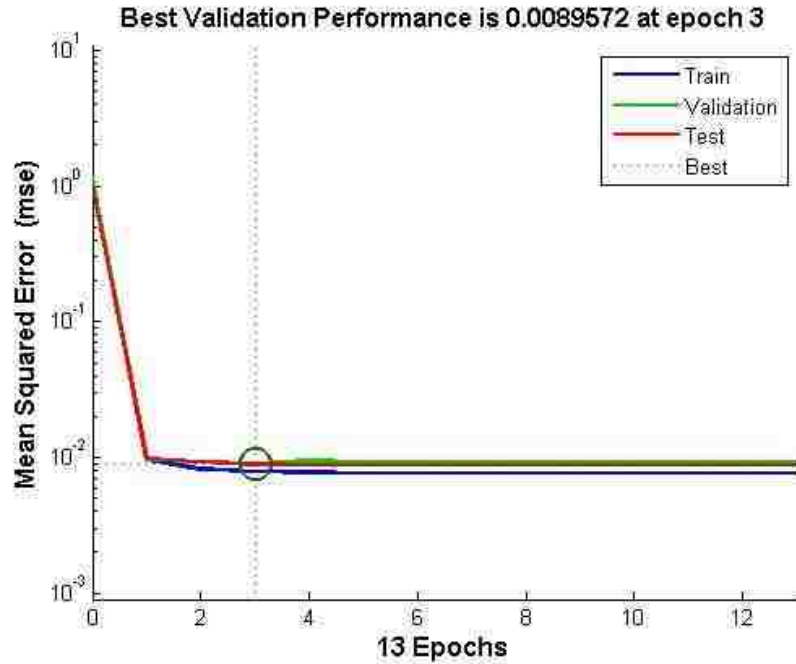
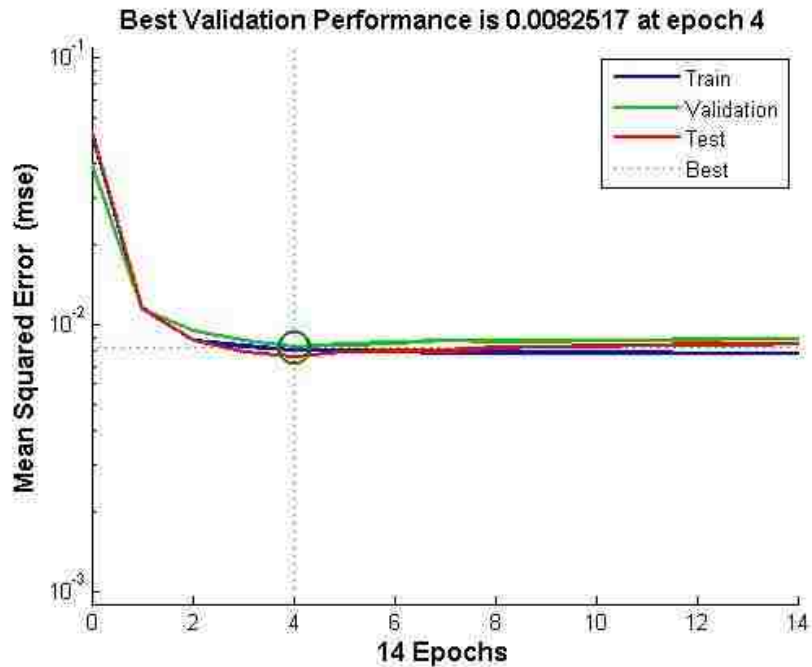


Figure 5. 8 The architectures of two ANNs

When training these two networks, one-hidden-layer training was faster than another one. The difference between these two training processes is the slope of the gradient. Both used ten validation checks, and the two networks almost reached their limit at the same epoch. However, the two-hidden-layer network took more time. The most significant difference is that the two-hidden-layer network provided a slower approach to the minimum MSE (Figure 5.9).



(a)



(b)

Figure 5.9 (a) MSE of 1-hidden-layer training (b) MSE of 2-hidden-layer training

The second hidden layer adds 61 additional weights and biases, increasing the complexity of the network with more degrees of freedom. The results of the multiple hidden layer networks may not be very different from those of the simpler network, but the process of training and simulation is more accurate. More detail of this comparison

can be found in Appendix C.1.

5.3.3 Comparison of Different Network Architectures

Neural network architecture is important to the quality of the simulation results. For a single-hidden-layer neural network, the $2n+1$ method is a commonly used way to estimate an appropriate number of hidden neurons. More neurons can reduce the error between the network outputs and targets, but it can also cause overfitting of the data. However, to date, there is not a generally accepted method to calculate the number of hidden neurons. To get a proper result, people often use trial and error to set the network architectures.

The number of hidden layers is one of the things to be considered before training and simulation. However, the number of hidden layer is not as important as the number of neurons. In today’s research, many researchers are focusing more attention on how many neurons are efficient to solve the problem. On the other hand, changing the number of hidden layers is another way of changing the number of hidden neurons.

In this research, there are 52 inputs and four targets in the data. Thus, according to section 5.3.1, there should be $52 \times 2 + 1$ hidden neurons if a single hidden layer were chosen. However, as a neural network doesn’t have a set rule for setting the number of hidden neuron, an experimental comparison is needed.

Table 5.4 shows the comparison network performance for different architectures with respect to the number of hidden neurons.

Table 5. 4 Performance (MSE) of different hidden neurons in a one hidden layer logsig-purelin neural network

52-N-4 LOGSIG-PURELIN NETWORK BY FEED-FORWARD BACKPROPAGATION WITH TRAINLM												
Neuron NO. N=	10	21	32	43	54	65	76	87	98	105	112	121
Best Val. Perform (MSE)	0.24288	0.2678	0.59049	0.34306	0.76306	0.16458	0.40162	0.79686	0.7254	0.11389	0.34659	0.25684
Performance (MSE)	0.1321	0.1666	0.179	0.4227	0.2138	0.2487	0.1953	0.2326	0.1907	0.0885	0.186	0.0887
Epoch (At Best Perform)	1	1	1	6	1	1	1	2	4	3	1	1

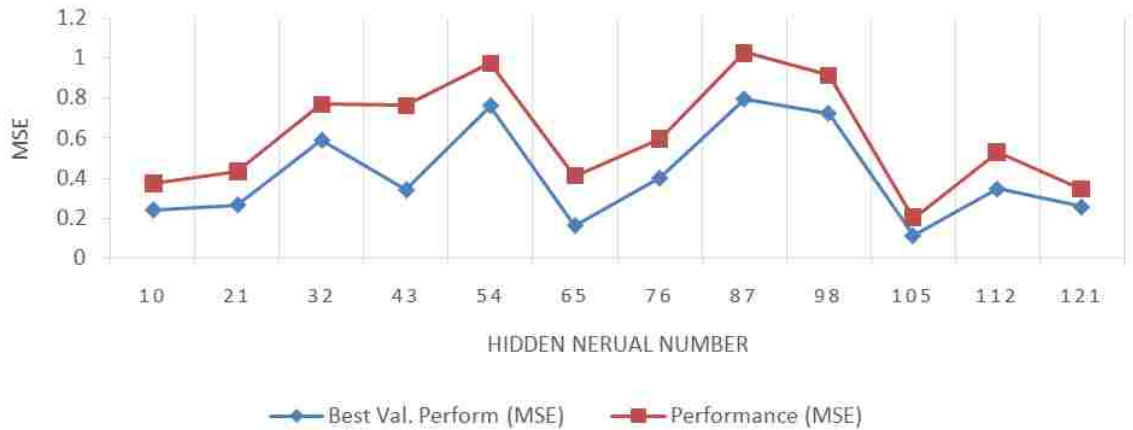


Figure 5.10 Performance (MSE) of different hidden neurons in a one hidden layer logsig-purelin neural network

In Table 5.4 and Figure 5.10, the performance of the network and the best validated performance occur together when the number of hidden neurons is 105. Because the performance of the neural network changes each time it's trained, further examination is necessary during training. However, from this preliminary comparison, the $2n+1$ method to determine the number of hidden layers provides a better result.

In order to obtain a more comprehensive understanding of hidden neuron influence on network performance, Figure 5.11 gives a visual plot of MSE with the respect to the number of hidden neurons. It must be mentioned that with the increase of hidden neurons, the training time increases significantly. When there are less than 90 hidden neurons, the training process finishes within one second. The largest networks trained take as much as three minutes to complete. However, as the number of hidden neurons increases, the regression of normalized output to normalized target is much improved.

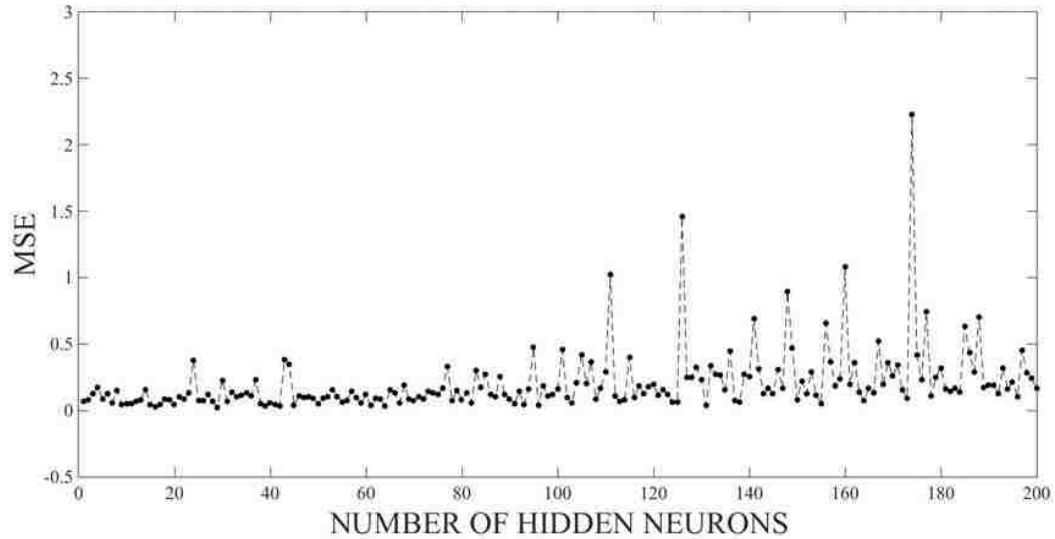


Figure 5.11 Performance of neural network with different number of hidden neurons

In Figure 5.11, the black points indicate performance (MSE) of neural networks with one to two hundred single hidden layer neurons. The MSE in Figure 5.11 seems stable when there are less than 130 neurons. As discussed previously, there is no standard rule for determining the number of hidden neurons. For this research, the number will be determined by starting with the $2n+1$ empirical method.

5.4 Artificial Neural Network Training

The architecture of the ANN has been selected to include the logsig transfer function in the hidden layer, the FFB network structure and the 52-105-4 input-hidden-output configuration.

Thus, for this research, the network specifications can be described by Table 5.5 below; the training parameters are also included.

Recall that the input data has 52 normalized samples described by mean and standard deviation. The network outputs are Young's modulus and tensile strength at different fiber directions derived from test raw data.

Table 5. 5 Network training parameters

Artificial Neural Network Detail	
Target Data	Young's modulus & tensile strength at 0° and 90° fiber orientation
Input Data	LFT-D process parameters for fiber volume weights 25% 30% 35% 40% 45% 50%
Network Type	Feed-forward backpropagation
Training Function	Traincgp/Traingdx/Trainlm
Performance Function	Mean Square Error (MSE)
Number of Hidden Layer	1
Number of Layers	3
Number of Input Neurons	52
Number of Hidden Neurons	105
Number of Output Neurons	4

With the aid of MATLAB, the work of network training can be simplified. The code used for MATLAB training is included in Appendix D. The code includes the data loading, data organization, neural network set up, parameter loading, and sensitivity calculation. To ensure a good result, as mentioned before, there will be a comparison between the different training methods.

5.4.1 Network Training using Mean Value of Processing Parameters

The dataset was organized to 52 inputs and 4 targets. The configuration can be viewed in Figure 5.12 as MATLAB would show it.

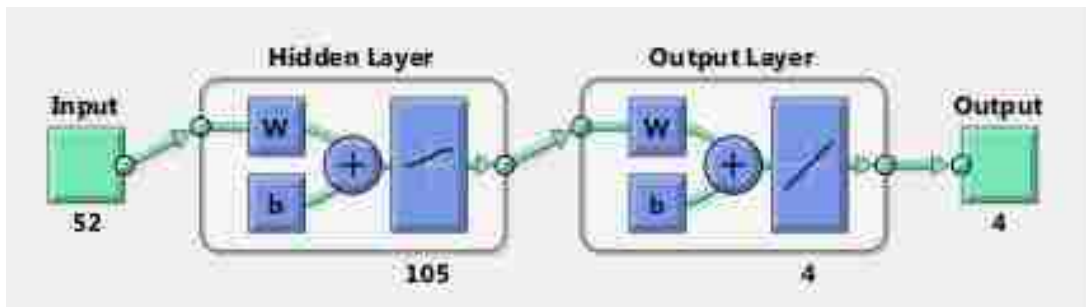


Figure 5. 12 Neural network configuration for training

The 52 inputs are normalized data of the mean value of each processing parameter. The target in this configuration is the mean value of Young's modulus and tensile strength

in two directions (0°, 90°). For accurate prediction, the three training functions were applied here for comparison. The performance values are listed in Table 5.6.

Table 5. 6 Performance of the mean value data trained result

Training Function	LM	CGP	GDX
Best Performance (MSE)	0.18559	0.094646	0.056176
Epoch at Best Performance	1	4	47
Performance (MSE)	0.0478	0.0787	0.1246

In Table 5.6, the best performance is the MSE calculated at the epoch when the validation check stops the iterations. The performance is the final MSE calculated after the network training is completed. If the performance curve after best performance of the validation check still decreases sharply, it may mean that the network is overfitting the target data.

5.4.2 Network Training using Standard Deviation of Processing Parameters

Similar to network training using mean values of processing parameters, the training in this section uses the standard deviation instead of the average value for the input and target data. The configuration and the training method are the same as the approach in Section 5.4.1. The results are listed in Table 5.7.

Table 5. 7 Performance of standard deviation data result

Training Function	LM	CGP	GDX
Best Performance (MSE)	0.90202	0.21579	0.91888
Epoch at Best Performance	1	1	25
Performance (MSE)	0.2327	0.6573	0.5082

5.4.3 ANN Development to Predict CFRP Properties Using LFT-D Processing Parameters

In the previous two sections, basic ANNs have been introduced. To refine the training results for the selected network structure, a randomly generated input is added to the input data. In this way, the relevance of each input parameter relative to a random value can be

evaluated and the network structure optimized (see Table 5.8).

Table 5. 8 Artificial neural network training parameters

Artificial Neural Network	
<i>Network Type</i>	Feed-forward Backpropagation Network
<i>Network Training Function</i>	Levenberg-Marquardt Method
<i>Performance Function</i>	Mean Square Error (MSE)
<i>Transfer Function (Hidden Layer/Output Layer)</i>	logsig/purelin
<i>Number of Hidden Layer</i>	1
<i>Number of Hidden Neurons</i>	107 (2n+1 Method)
<i>Number of Layers</i>	3
<i>Number of Input Neurons</i>	53
<i>Number of Output Neurons</i>	4
<i>Input Data</i>	mean value of Young's modulus, tensile strength at 0° and 90° fiber orientation, randomly generated value
<i>Target Data</i>	mean value of LFT-D process parameters for fiber volume weights of 25%, 30%, 35%, 40%, 45%, 50%

As discussed before, the FFB network is one of the simplest but is very effective when compared with other, more complex, types of networks. Training time and performance associated with the Levenberg-Marquardt training function make it a popular choice of researchers. The use of mean value for both the input and the output datasets gives a better performance results than the use of the standard derivation.

The process of neural network training uses MATLAB and code included in Appendix D. Figure 5.13 gives the training results of the network.

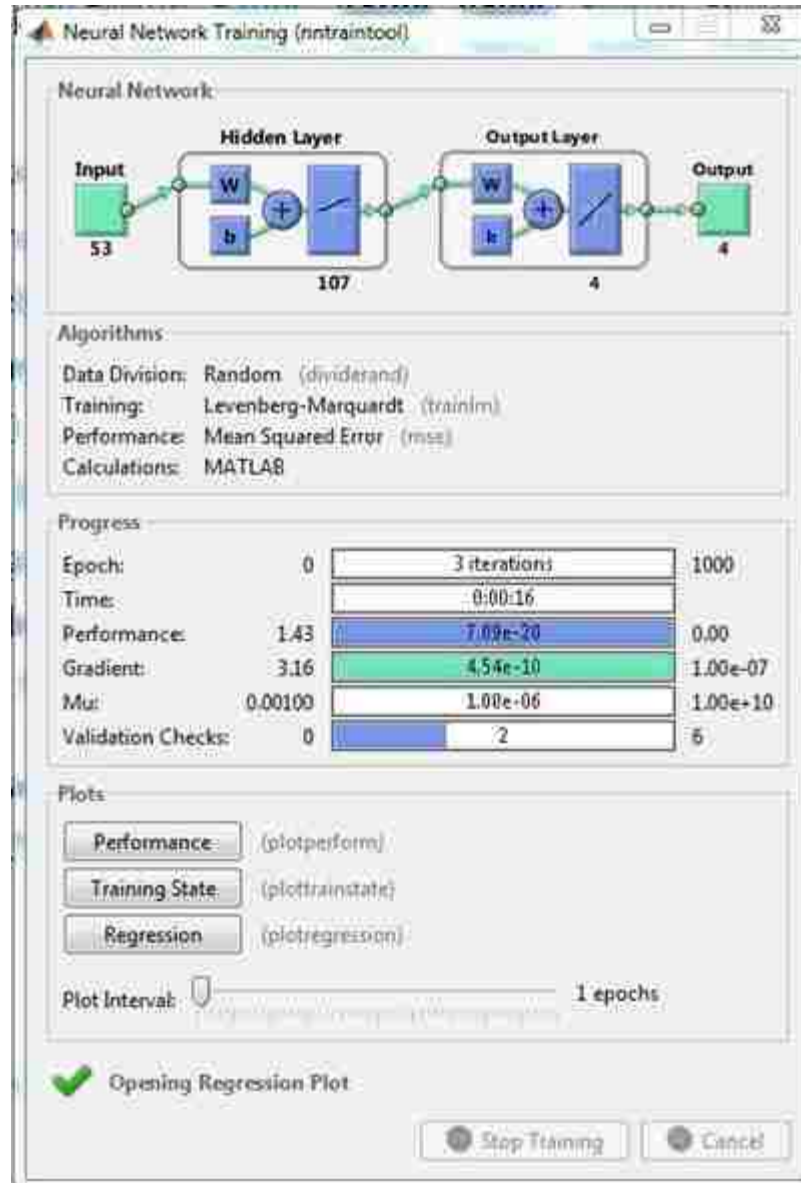


Figure 5.13 Training status of neural network applied for current research

This network training used a 53-107-4 logsig-purelin configuration with the LM training method. The process was 16 seconds until the gradient achieve the default setting value. At this time there were two validations. The final performance was 0.0617 at the last epoch with the final weights and biases. The best validation performance of this training is 0.1036 at epoch 1, as Figure 5.14 shows.

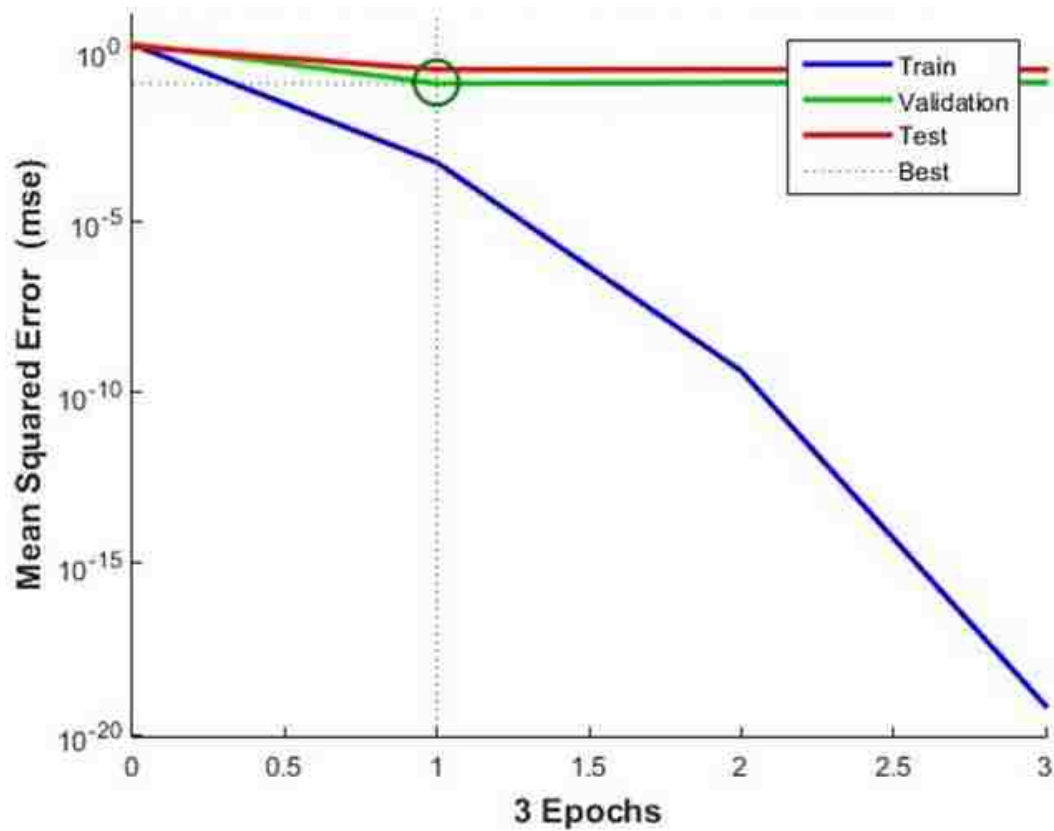


Figure 5. 14 Validation, training, and test MSE

As Figure 5.14 shows, there were three epochs during training. Due to reaching the minimum gradient of the MSE, the network training stopped with results in an acceptable range.

Figure 5.15 gives the results of the network training, with the regression between the target and network output illustrating how well the network trained.

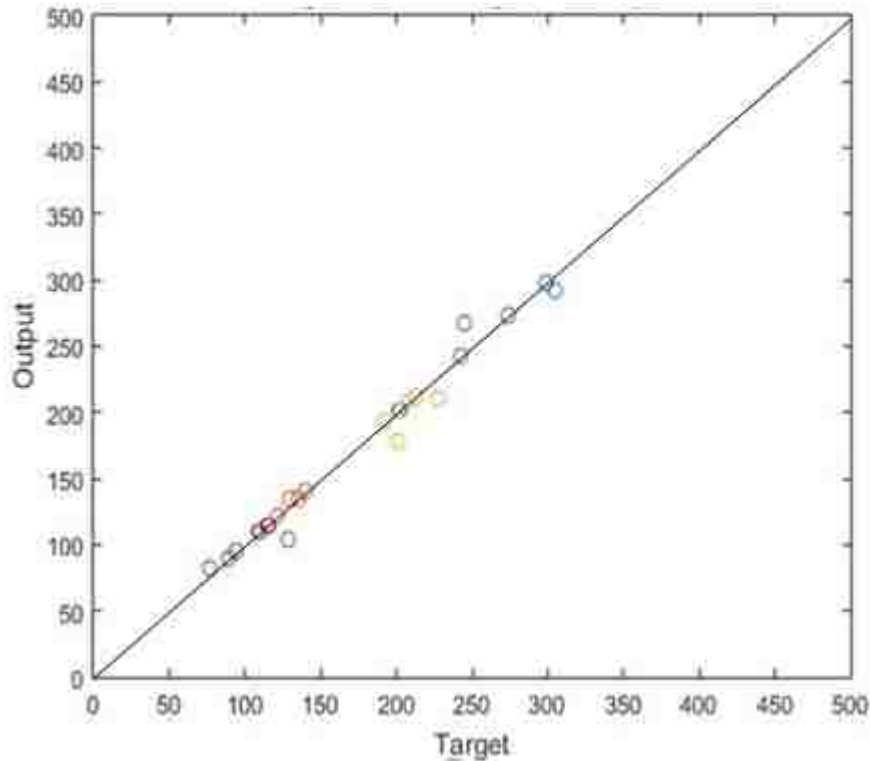


Figure 5. 15 Regression of network training results: regression of original data

The regression between target and output shows a good result in Figure 5.15. Thus, the network developed during training can be used for prediction as the performance of the network is acceptable.

5.5 Sensitivity of Network Outputs to Network Inputs

The aim of this research is to find the relationship between the LFT-D processing parameters and the CFRP properties. When developing the neural network, all inputs are used to predict the outputs. To simplify the network, the sensitivity of each output to each input can be calculated to determine if any of the inputs can be eliminated from the model due to a lack of influence on any of the output variables. The sensitivity is expressed as the partial derivative of each network output with respect to each of the inputs (Samarasinghe 2006).

Using feed-forward neural networks, the approach to determine the relative influence

of inputs on outputs can be performed using a sensitivity analysis.

5.5.1 Sensitivity Analysis

Sensitivity analysis, as mentioned before, uses the expression of the partial derivative of the network output with respect to each input. Sensitivity analysis is based on the influence observed in an output z_i due to a small change in an input x_i . The sensitivity can provide information as to the relative influence of each network input on each network output. The greater the sensitivity is, the greater the effect of the input.

As the sensitivity of the input pattern can provide different values, the mean value is used to represent the overall sensitivity of an output to an input. The method to obtain the partial derivatives was introduced in Section 2.4. The gradient of mean square error with respect to any weight has been calculated. Similar to the concept to calculate the error derivative for any hidden-output weight, the change in output due to changes in input can be expressed, in general, as follows:

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial y} \cdot \frac{\partial y}{\partial \mu} \cdot \frac{\partial \mu}{\partial x} \quad (5-8)$$

Where, x , z , v , y , μ , are as shown in Figure 5.16.

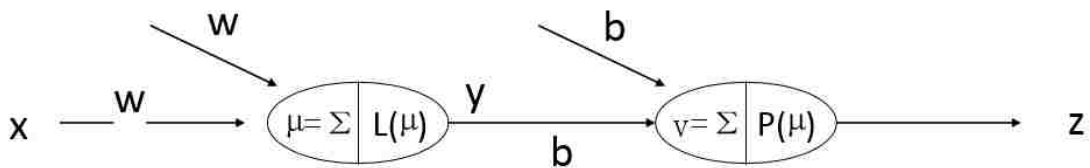


Figure 5. 16 Common descriptions of neurons for regular neural network

As mentioned previously, the logsig transfer function is used in the hidden layer neurons and the purelin transfer function in the output layer. Equation 5-8 then becomes:

$$\left\{ \begin{array}{l} z_k = v_k \text{ (purelin transfer function)} \\ v_k = b_0 + \sum_{j=1}^l b_{jk} y_k \text{ (sum of weighted hidden layer output)} \\ y_k = \frac{1}{1 + e^{-u_j}} \text{ (logsig transfer function)} \\ u_j = w_0 + \sum_{i=1}^n w_{ij} x_i \text{ (sum of weighted input)} \end{array} \right.$$

As a result, the sensitivity of output z_k with respect to the input x_i is as follows:

$$\frac{\partial z_k}{\partial x_i} = \frac{1}{l} \sum_{j=1}^l w_{ij} b_{jk} e^{-u_j} (1 + e^{-u_j})^2 \quad (5-9)$$

For an m - n - l FFB network with logsig-purelin transfer functions, the above equation can be used to calculate the sensitivity of outputs with respect to each of the inputs.

As discussed before, the sensitivity of a network output to an input can imply a degree of influence of the processing parameter on the material property. However, as the weight of the trained network can vary from training session to training session, the calculated sensitivity cannot represent an absolute value. For this reason, the way to obtain a reliable result from the sensitivity study is to calculate the average value of multiple sensitivities of several good performing networks. In this way, if the input generally has less influence on the output, the mean value of the sensitivities will remain low. The average value of sensitivities can be indicative of the trends apparent in the relationships between network inputs and outputs.

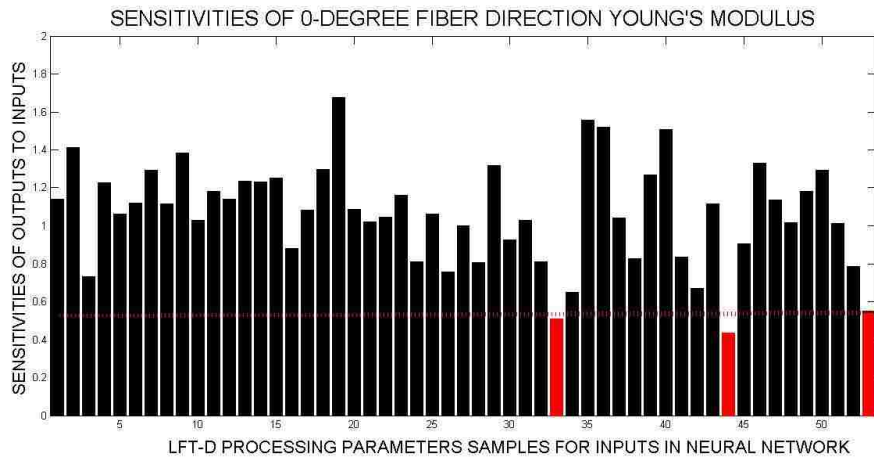
For this research, acceptable performance is taken to be no more than 0.3 for the best MSE and having no more than two outliers in the regression plot. Training was conducted ten times with the exact same training parameters as the detailed in section 5.4. Table 5.11 shows the mean value of the sensitivities' absolute values for ten training sessions. Absolute values are used to restrict our examination to the magnitude of the sensitivities. The sum of negative value and positive values will not provide an accurate representation of the overall affect.

Table 5.9 gives the results of the sensitivity of each output to every input for the data.

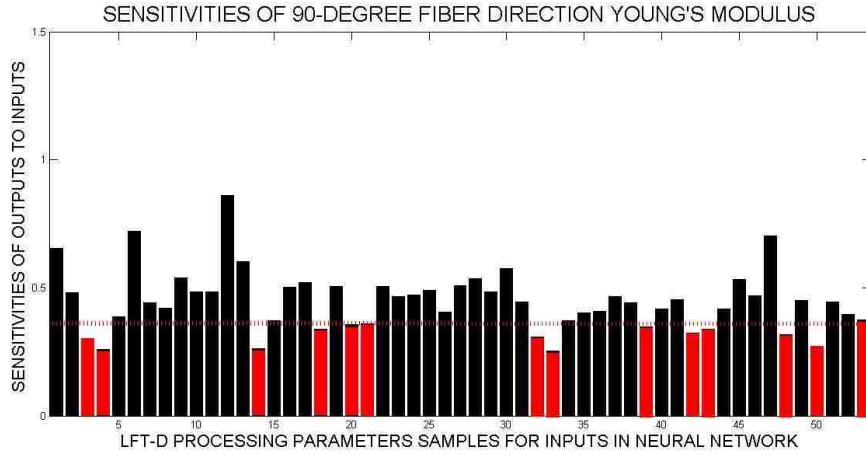
Table 5. 9 The mean value of the summed absolute sensitivities for ten network training sessions

Material Properties	Young's Modulus		Tensile Strength	
	0°	90°	0°	90°
<i>Fiber Direction</i>				
<i>Throughput Set Value Dosing 5</i>	1.142	0.653	1.010	0.701
<i>Throughput Actual Value Dosing 5</i>	1.411	0.481	1.054	0.952
<i>Set Rotational Speed Screw ZSG</i>	0.732	0.300	0.681	0.911
<i>Torque ZSG</i>	1.227	0.259	0.842	1.103
<i>Resulting Force Servo Die</i>	1.061	0.388	1.092	1.323
<i>Force Servo Die Open</i>	1.120	0.720	0.986	0.915
<i>Force Servo Die Close</i>	1.293	0.443	1.378	1.599
<i>Actual Rotational Speed Screw ZSG</i>	1.116	0.420	1.322	1.549
<i>Internal Force ZSG</i>	1.385	0.538	1.338	1.421
<i>Internal Temperature ZSG</i>	1.028	0.483	0.816	0.907
<i>Temperature ZSG Zone 1</i>	1.180	0.485	1.096	1.495
<i>Temperature ZSG Zone 2</i>	1.142	0.862	0.858	1.017
<i>Temperature ZSG Zone 9</i>	1.233	0.602	1.101	1.032
<i>Temperature ZSG Zone 10</i>	1.229	0.263	0.872	1.170
<i>Temperature ZSG Zone 11</i>	1.253	0.373	1.365	1.415
<i>Temperature ZSG Zone 12</i>	0.880	0.504	0.910	0.846
<i>Temperature ZSG Zone 13</i>	1.083	0.521	1.185	1.551
<i>Temperature ZSG Zone 14</i>	1.295	0.338	0.918	1.341
<i>Temperature ZSG Zone 15</i>	1.675	0.505	1.349	1.523
<i>Temperature ZSG Zone 16</i>	1.085	0.357	1.015	1.144
<i>Rotational Speed Motor Shears</i>	1.021	0.361	0.831	1.104
<i>Pick-Up Belt Temperature</i>	1.044	0.507	0.879	0.663
<i>Positioning Belt 1 Temperature</i>	1.161	0.467	1.054	1.115
<i>Positioning Belt 2 Temperature</i>	0.809	0.471	0.984	1.104
<i>Positioning Belt 3 Temperature</i>	1.061	0.491	1.131	0.948
<i>Positioning Belt 4 Temperature</i>	0.755	0.406	0.825	0.853
<i>Positioning Belt 5 Temperature</i>	1.000	0.510	0.678	0.645
<i>Positioning Belt 6 Temperature</i>	0.805	0.535	1.024	1.028
<i>Safety Tunnel Temperature</i>	1.319	0.485	1.018	0.823
<i>Working Tunnel 1 Temperature</i>	0.926	0.576	0.464	0.474
<i>Working Tunnel 2 Temperature</i>	1.027	0.445	1.049	1.256
<i>Working Tunnel 3 Temperature</i>	0.809	0.309	0.808	0.828
<i>Actual Speed Pick-Up Belt</i>	0.508	0.255	0.662	0.620
<i>Actual Speed Melt Compound</i>	0.650	0.371	0.957	1.235
<i>Actual Speed Positioning Belt 1</i>	1.556	0.401	1.185	1.251

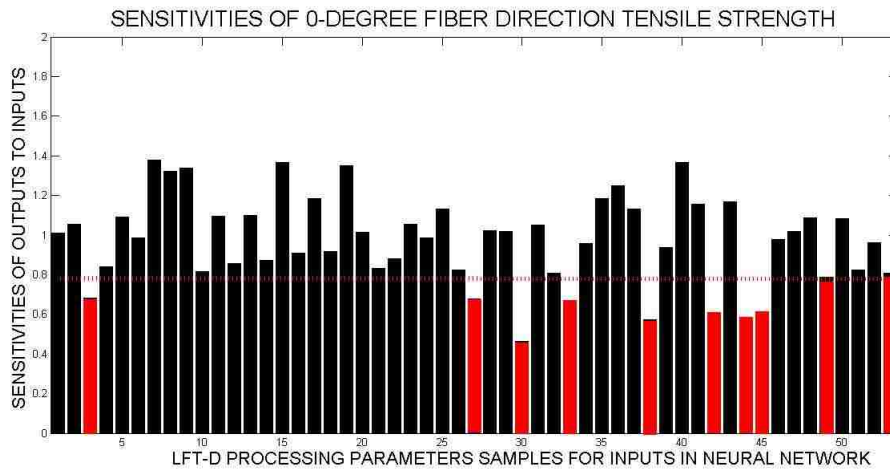
<i>Actual Speed Wagon</i>	1.520	0.407	1.250	1.533
<i>Set Rotational Speed Screw ZSE</i>	1.042	0.467	1.133	1.423
<i>Torque ZSE</i>	0.825	0.441	0.571	0.713
<i>Actual Rotational Speed Screw ZSE</i>	1.266	0.347	0.938	1.312
<i>Internal Force ZSE</i>	1.508	0.417	1.368	1.407
<i>Internal Temperature ZSE</i>	0.837	0.453	1.154	1.130
<i>Temperature ZSE Zone 1</i>	0.672	0.324	0.611	0.486
<i>Temperature ZSE Zone 2</i>	1.114	0.339	1.166	1.198
<i>Temperature ZSE Zone 2</i>	0.432	0.417	0.582	0.710
<i>Temperature ZSE Zone 3</i>	0.905	0.532	0.611	0.836
<i>Temperature ZSE Zone 4</i>	1.328	0.468	0.976	1.105
<i>Temperature ZSE Zone 5</i>	1.137	0.704	1.020	1.399
<i>Temperature ZSE Zone 6</i>	1.017	0.318	1.088	0.931
<i>Temperature ZSE Zone 7</i>	1.183	0.452	0.786	1.246
<i>Temperature ZSE Zone 8</i>	1.294	0.266	1.082	1.292
<i>Temperature ZSE Zone 9</i>	1.012	0.444	0.825	1.086
<i>Temperature ZSE Zone 10</i>	0.786	0.395	0.961	1.124
<i>Randomly Generated Input</i>	0.552	0.374	0.807	0.905



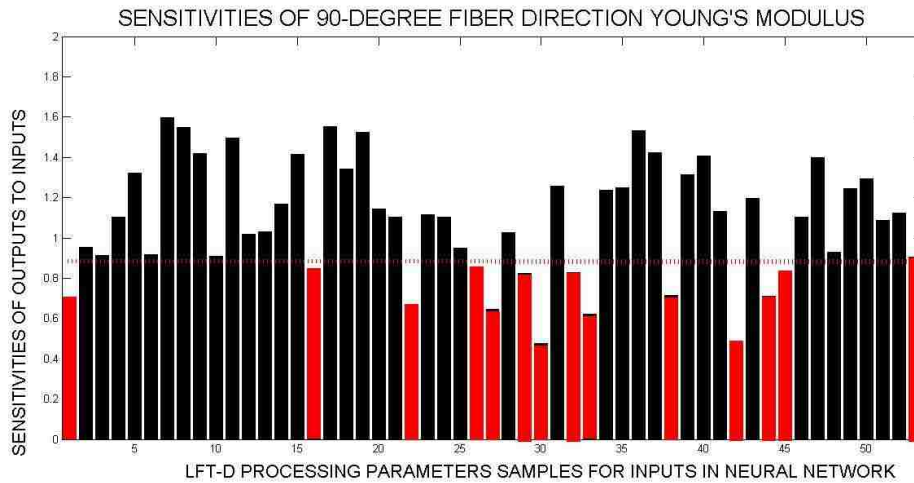
(A)



(B)



(C)



(D)

Figure 5. 17 (a) to (d) the Mean of the Summed Absolute Sensivities for outputs to inputs from output z_1 to

z_4

Figures 5.17 (a) to (d) show the sensitivities of four outputs with respect to all inputs.

The more influence the input has, the larger the sensitivity is.

By calculating sensitivity of output to input, the influence of the LFT-D processing parameters on CFRP properties can be described by characteristics of the trained neural network.

In Figure 5.17, the red dash lines in each of the four plots correspond to the sensitivity of each network output to the randomly generated input variable. Those network inputs with the same or lower sensitivity can be removed from the network structure as their influence is no more than that of the randomly generated network input.

With respect to Young's modulus for 0° fiber orientation, the processing parameters shown as black bars have significant influence, specifically, the throughput set value of the dosing unit, the set rotational speed of the ZSG's speed, the force of the blade of the servo die, the internal force of ZSG, the rotational speed of the motor, etc. As the temperature of the same machine parts in LFT-D often consistent at steady state operation, one might assume that the influence of the temperature channels on the material properties would also be similar. Interestingly, in the case of the current research, the sensitivities of the properties to temperatures in a similar region in the machine sometimes have very different values.

Comparing the four targets, the influence of processing parameters to the tensile strength with 0° fiber orientations is less than the other three properties. The result from both the optimized network and the network with randomly generated input shows the same result. Therefore, the 0° tensile strength of CFRP in this research is not affected as much as the 90° tensile strength and Young's modulus. The fiber direction is an important factor that influences the tensile strength as discussed previously in the review of continuous CFRP properties.

After calculating the mean value of summed absolute sensitivities for ten training sessions, the result shows all inputs have more or less influence on the four outputs. Comparing the four outputs, tensile strength for 0° fiber orientations has the least effect

from inputs as shown by the relative position of the trace in Figure 5.17. Additionally, the sensitivities of 0° fiber orientation tensile strength are the most stable ones among the four outputs. Table 5.10 gives the best validation performance of these ten training results.

Table 5. 10 Best validation performance of different training sessions for the network

	1	2	3	4	5	6	7	8	9	10
Best validation Performance	0.13	0.09	0.3	0.14	0.01	0.03	0.08	0.15	0.28	0.10

Compared with the other three sensitivity curves, the Figure 5.17 (b) has smaller range of sensitivity. It means that the influence of the processing parameters for the 0° fiber direction on Young's modulus is less than the other three outputs.

According to these results, the number of network inputs can be reduced by eliminating those inputs with mean sensitivities that are smaller than the sensitivity of the randomly generated input. Thus, the reduced set of inputs for the neural network training is listed in Table 5.11. Note that the reduced input sets are not the same for each of the four outputs, requiring four different network structures.

Table 5. 11 Remained inputs for four different outputs

Original Inputs	Remained inputs			
	Young's Modulus		Tensile Strength	
	0°	90°	0°	90°
Throughput Set Value Dosing 5	Throughput Set Value Dosing 5	Throughput Set Value Dosing 5	Throughput Set Value Dosing 5	
Throughput Actual Value Dosing 5	Throughput Actual Value Dosing 5	Throughput Actual Value Dosing 5	Throughput Actual Value Dosing 5	Throughput Actual Value Dosing 5
Set Rotational Speed Screw ZSG	Set Rotational Speed Screw ZSG			Set Rotational Speed Screw ZSG
Torque ZSG	Torque ZSG		Torque ZSG	Torque ZSG
Resulting Force Servo Die	Resulting Force Servo Die	Resulting Force Servo Die	Resulting Force Servo Die	Resulting Force Servo Die
Force Servo Die Open	Force Servo Die Open	Force Servo Die Open	Force Servo Die Open	Force Servo Die Open
Force Servo Die Close	Force Servo Die Close	Force Servo Die Close	Force Servo Die Close	Force Servo Die Close
Actual Rotational Speed Screw ZSG	Actual Rotational Speed Screw ZSG	Actual Rotational Speed Screw ZSG	Actual Rotational Speed Screw ZSG	Actual Rotational Speed Screw ZSG

<i>Safety Tunnel Temperature</i>	<i>Safety Tunnel Temperature</i>	<i>Safety Tunnel Temperature</i>	<i>Safety Tunnel Temperature</i>	
<i>Working Tunnel 1 Temperature</i>	<i>Working Tunnel 1 Temperature</i>	<i>Working Tunnel 1 Temperature</i>		
<i>Working Tunnel 2 Temperature</i>	<i>Working Tunnel 2 Temperature</i>	<i>Working Tunnel 2 Temperature</i>	<i>Working Tunnel 2 Temperature</i>	<i>Working Tunnel 2 Temperature</i>
<i>Working Tunnel 3 Temperature</i>	<i>Working Tunnel 3 Temperature</i>		<i>Working Tunnel 3 Temperature</i>	<i>Working Tunnel 3 Temperature</i>
<i>Actual Speed Pick-Up Belt</i>			<i>Actual Speed Pick-Up Belt</i>	
<i>Actual Speed Melt Compound</i>	<i>Actual Speed Melt Compound</i>	<i>Actual Speed Melt Compound</i>	<i>Actual Speed Melt Compound</i>	<i>Actual Speed Melt Compound</i>
<i>Actual Speed Positioning Belt 1</i>	<i>Actual Speed Positioning Belt 1</i>	<i>Actual Speed Positioning Belt 1</i>	<i>Actual Speed Positioning Belt 1</i>	<i>Actual Speed Positioning Belt 1</i>
<i>Actual Speed Wagon</i>	<i>Actual Speed Wagon</i>	<i>Actual Speed Wagon</i>	<i>Actual Speed Wagon</i>	<i>Actual Speed Wagon</i>
<i>Set Rotational Speed Screw ZSE</i>	<i>Set Rotational Speed Screw ZSE</i>	<i>Set Rotational Speed Screw ZSE</i>	<i>Set Rotational Speed Screw ZSE</i>	<i>Set Rotational Speed Screw ZSE</i>
<i>Torque ZSE</i>	<i>Torque ZSE</i>	<i>Torque ZSE</i>	<i>Torque ZSE</i>	<i>Torque ZSE</i>
<i>Actual Rotational Speed Screw ZSE</i>	<i>Actual Rotational Speed Screw ZSE</i>			<i>Actual Rotational Speed Screw ZSE</i>
<i>Internal Force ZSE</i>	<i>Internal Force ZSE</i>	<i>Internal Force ZSE</i>	<i>Internal Force ZSE</i>	<i>Internal Force ZSE</i>
<i>Internal Temperature ZSE</i>	<i>Internal Temperature ZSE</i>	<i>Internal Temperature ZSE</i>	<i>Internal Temperature ZSE</i>	<i>Internal Temperature ZSE</i>
<i>Temperature ZSE Zone 1</i>	<i>Temperature ZSE Zone 1</i>			
<i>Temperature ZSE Zone 2</i>	<i>Temperature ZSE Zone 2</i>	<i>Temperature ZSE Zone 2</i>	<i>Temperature ZSE Zone 2</i>	<i>Temperature ZSE Zone 2</i>
<i>Temperature ZSE Zone 2</i>				
<i>Temperature ZSE Zone 3</i>	<i>Temperature ZSE Zone 3</i>	<i>Temperature ZSE Zone 3</i>		
<i>Temperature ZSE Zone 4</i>	<i>Temperature ZSE Zone 4</i>	<i>Temperature ZSE Zone 4</i>	<i>Temperature ZSE Zone 4</i>	<i>Temperature ZSE Zone 4</i>
<i>Temperature ZSE Zone 5</i>	<i>Temperature ZSE Zone 5</i>	<i>Temperature ZSE Zone 5</i>	<i>Temperature ZSE Zone 5</i>	<i>Temperature ZSE Zone 5</i>
<i>Temperature ZSE Zone 6</i>	<i>Temperature ZSE Zone 6</i>	<i>Temperature ZSE Zone 6</i>	<i>Temperature ZSE Zone 6</i>	<i>Temperature ZSE Zone 6</i>
<i>Temperature ZSE Zone 7</i>	<i>Temperature ZSE Zone 7</i>			<i>Temperature ZSE Zone 7</i>

<i>Temperature ZSE Zone 8</i>	<i>Temperature ZSE Zone 8</i>	<i>Temperature ZSE Zone 8</i>	<i>Temperature ZSE Zone 8</i>	<i>Temperature ZSE Zone 8</i>
<i>Temperature ZSE Zone 9</i>	<i>Temperature ZSE Zone 9</i>		<i>Temperature ZSE Zone 9</i>	<i>Temperature ZSE Zone 9</i>
<i>Temperature ZSE Zone 10</i>	<i>Temperature ZSE Zone 10</i>	<i>Temperature ZSE Zone 10</i>	<i>Temperature ZSE Zone 10</i>	<i>Temperature ZSE Zone 10</i>

5.5.2 Neural Network Optimization

Optimizing the neural network depends on several aspects. During the training, if the results cannot accurately generalize the target data, the configuration and training methods may need to be adjusted. But, as mentioned before, the approaches used to modify the configuration and adjust the training methods are done by trial and error. Additionally, a method to optimize the neural network is to combine another modeling method with neural network.

There is some research illustrating the use of genetic algorithms to optimize neural networks (Koehn 1994). The genetic algorithm can be used as an initial weight selection tool so that the neural network can provide a more accurate result in predicting the target (Wang 2.13).

Another method is to reduce the structural complexity of the neural network (Samarasinghe 2006). The purpose of reducing the complexity of the neural network is to retain the essential weights and neurons in the model, reduce computations, but maintain accuracy. This commonly used method is referred to as network pruning. Pruning reduces the structural complexity of a neural network and the method can be based on variance of network sensitivity. The pruning of the structural complexity can include elimination of inputs, weights, and/or hidden neurons.

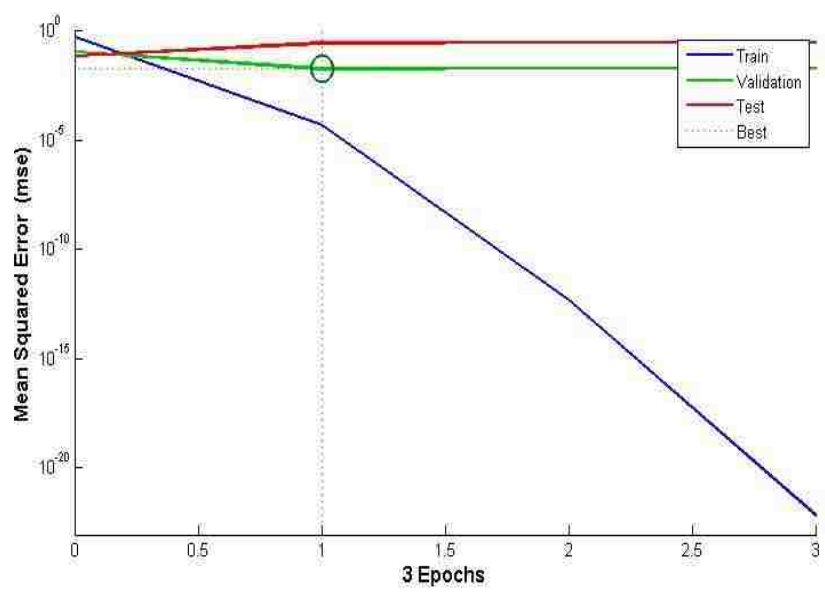
For the current research, in Section 5.5.1, the sensitivities of outputs to the inputs have been analyzed. Figure 5.17 shows the sensitivities of the 53 inputs (including the randomly generated input) on the 4 outputs. According to this result, the complexity of

the network can be reduced by eliminating those inputs with the same or less influence on the outputs as the random input (see Figure 5.17). Table 5.10 lists the final remaining inputs.

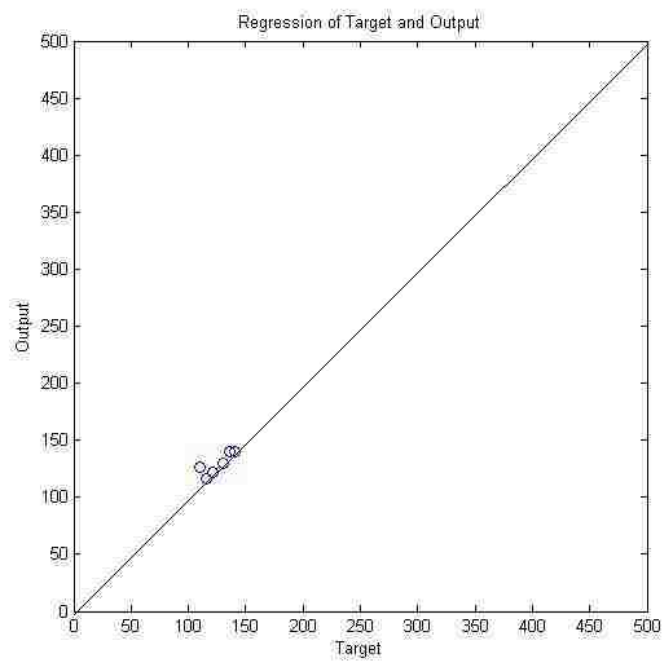
Recall Figure 5.11, the performance of the logsig-purelin feed-forward backpropagation network with different hidden neurons with one-hidden-layer. In order to obtain the best validation performance (best MSE before the validation check expired) of the network, the network was trained again and the best performance for a different number of hidden neurons is shown in Figure 5.18 (a). In a neural network, the final MSE cannot really indicate the performance on its own. The best validation performance of the network is the MSE value of the network training before over-fitting occurs. To evaluate a good performing network, there are three basic rules to follow:

- (1) The final mean-square error is small.
- (2) The test set error and the validation set error has similar characteristics.
- (3) No significant over-fitting has occurred where the best validation performance occurs.

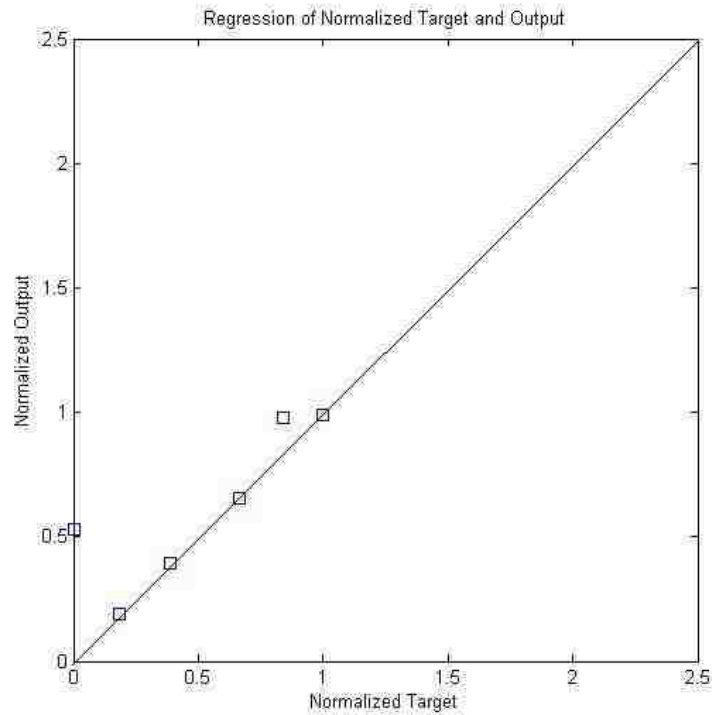
In Table 5.11, the remaining inputs for the four neural networks predicting each of the four outputs are different. For the tensile strength in the 0° fiber orientation, the sensitivities of the original inputs were not larger than the randomly generated input which means all inputs were retained. The numbers of inputs for the other three networks (and three outputs) are all reduced according to the sensitivities. In this section, the Young's modulus in the 90° fiber direction will be trained as an output with the remaining inputs as an example. All the training parameters are the same as the parameters described in Section 5.5.1. The only difference is there are now only 39 inputs instead of 53.



(a)



(b)



(c)

Figure 5.18 Training results of 39-input network: (a) best validation performance of network and the MSE of training, test, and validation; (b) regression of target and output; (c) regression of normalized target and output.

Figure 5.18 shows the results of the optimized neural network with 39 inputs. The trained network with a reduced number of inputs provides a MSE equal to 0.05 at epoch 1. From Figure 5.18, the result of this training is acceptable because of the regression plot have all points on the 45° line. Figure 5.18 (b) and (c) shows the points lie on or close to a 45° line, and the error between the target and output is very small.

CHAPTER 6 RESULT ANALYSIS

As discussed in Chapter 5, there are at least five different styles of neural network that are used for prediction. At very beginning of Chapter 5, the neural network was trained for original inputs and outputs with 105 hidden neurons. Using this configuration, the network was too complex and the training time was very long. To optimize the network, the number of hidden neurons was reduced. With the hidden neuron reduction, the result of the network training was better than the original but further optimization was sought to reduce the number in network inputs using the results of the sensitivity study. In order to make any conclusions about the relative sensitivity of each output to each input, a randomly generated input was added to the network. Comparing the sensitivity of each output to each input relative to the random variable, the inputs with less influence than the randomly generated input were eliminated.

Using this optimization scheme, a final neural network was trained with only 39 inputs for the Young's modulus at 90° fiber orientation, for example. The training results can be seen in Figure 5.18 The regression between target and output lies mostly on a 45-degree line which means the network gives an acceptable training result. With the value of 0.05 for the MSE, the network for 39 inputs trained to Young's modulus at 90° fiber orientation can be further analyzed by the sensitivity of the output to the inputs.

For all intents and purposes, the neural network development is a somewhat random procedure. A better training result only can indicate that the neural network can be used for predicting the relationship between the CFRP material properties and the processing parameters set for material production. As the input data are the mean value of the processing parameters, the sensitivity analysis cannot describe their influence on properties extremely accurately, but instead provides a generalized relative comparison between parameters. In Figure 5.17, the largest values highlight which inputs has more influence on the output. The ZSE extruder screw speed has a large influence on material properties, as does the temperature of the fiber re-heating section.

CHAPTER 7 CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

CFRP is an advanced composite material. As discussed in Chapter 2, carbon fiber is a material showing it can benefit all areas of mechanical engineering with a possible exception with regard to its high cost. However, with improvements in future technology, carbon fiber will become more available and more affordable. Thermoplastic is an excellent material for engineering applications due to its low cost, ease of manufacturability, and high strength. Thus, CFRP combines the advantages of these two materials as well as producing a material that is much more than the sum of its parts. Currently, automotive industries are very interested in CFRP products because of its high performance. Thus, the manufacturing process of CFRP by LFT-D is widely used because it is a very efficient manufacturing process for composite materials.

In this research, the model's input data come from the processing parameters of LFT-D. LFT-D is already widely used for composite materials, especially for the production of continuous FRP. The charge produced by LFT-D can be easily formed to any shape that is required. With high temperatures, the process of LFT-D mixes the continuous reinforcement and the matrix materials efficiently. CFRP material produced by LFT-D has already been tested or used by many companies in the automotive industry. As there are a large number of processing parameters involved, sometimes not all will significantly affect the material properties, so optimization of the production process is still needed. Therefore, examining the influence of processing parameters on material properties is necessary in efforts to optimize manufacturing.

The ANN is an excellent tool for predicting the material performance as introduced before, especially applied to large data sets. In this research, the processing parameters of

the LFT-D process serve as the inputs to the neural network. The target and output data sets are organized into four groups based on the test data of the CFRP and the fiber directions of the test specimens. After several networks were trained, the results of the influence of processing parameters on material properties are calculated as the sensitivities of the network outputs to the inputs. From the results of the network training, the 52 processing parameters all have an effect on the material properties that were examined, but for different network training results the degree of their influence varies. When considering each of the outputs individually, all but one of the networks was optimized using a reduced set of input channels.

According to sections 5.5.1 and 5.5.2, the sensitivity analysis directly indicates the influence of the processing parameters on the Young's modulus and tensile strength of the material with 0° and 90° fiber orientation. In Figure 5.17, after training the network ten times, the Young's modulus of 0° fiber orientation shows sensitivity to almost all processing parameters. The 33nd and 44th processing parameters of the 52 processing parameter input data (one tunnel temperature and one ZSE temperature) have lower sensitivity than that due to the randomly generated value. Young's modulus for 90° fiber orientation shows less sensitivity, in general, to the processing parameters. Although there are 13 processing parameters having less influence on the 90° Young's modulus compared to the randomly generated values, the range of all calculated sensitivities is less than that of other three properties. For the tensile strength of CFRP with 0° and 90° fiber orientation (Figure 5.17 (C) and (D)), in general, the sensitivities between about 20th and 50th processing parameters have less effect than the other processing parameters. According to Table 5.11, these processing parameters describe operational settings in the ZSE region of the LFT-D machine which is the section of the extruder to mix the fiber and the melted polymer together in a specific direction.

Comparing the four model outputs, the latter thirty processing parameters have less influence than that of the former ones, in general. The influences of the 20th to 52th

processing parameters (Figure 5.17 (A) to (D)) are smaller than the influence of the 1st to 19th processing parameters. This means that, according to the calculated sensitivities, the ZSG region of the LFT-D machine has more effect on the property of Young's modulus and tensile strength of the material in both 0° and 90° fiber orientation. ZSG mainly consists of the fiber and polymer feeding sections and the pre-heated sections. According to this result, the properties of the final CFRP composite produced by LFT-D are most influenced before the fiber and the polymer are mixed together. As a result, consistency in parameter control during the preparation of the fiber and polymer can provide better stability in material performance (Young's modulus and tensile strength in 0° and 90° fiber orientation) of the CFRP when produced using LFT-D.

7.2 Recommendations

There are still areas for improvement for this research. The first is in regard to the accuracy of the data obtained from FPC. The data used for this research are the average values of each processing parameter. If the test specimens are matched to their own recorded data from the LFT-D trial, the network training result will be more robust. Another issue concerns the lack of measured deformation of the test specimens. If more tests were included in the data, the network's predictive capabilities would likely be more comprehensive.

A result of this research is the creation of an ANN model for performance prediction of CFRP produced by LFT-D. In the future, after the processing parameters and material properties are related by neural network model development, the relationship can be described by the function as dictated by the neural network structure. Furthermore, in the future, as more data becomes available, an inverse model can be generated, allowing for the calculation of processing parameters necessary for a desired set of material properties. A design tool of this type will prove valuable in increasing the efficiency of CFRP design and development tasks.

REFERENCE

- Abderrezak Bezazi, S. Gareth Pierce, Keith Worden, El Hadi Harkati. 2007. "Fatigue life prediction of sandwich composite materials under flexural tests using a Bayesian trained artificial neural network." *International Journal of Fatigue* 728-747.
- Al-Assadi, M., Kadi, H. E., & Deiab, I. M. 2010. "Predicting the fatigue life of different composite materials using artificial neural networks ." *Appl Compos Mater* 1-14.
- Al-Assadi, M., Kadi, H. E., & Deiab, I. M. 2011. "Using artificial neural networks to predict the fatigue life of different composite materials including the stress ratio effect." *Apple Compos Mater* 297-309.
- Anastasios P. Vassilopoulos, Efstratios F. Georgopoulos, Vasileios Dionysopoulos. 2007. "Artificial neural networks in spectrum fatigue life prediction of composite materials." *International Journal of Fatigue* 20-29.
- Bhadeshia, H. K. D. H. 1999. "Neural networks in materials science." *ISIJ international*.
- Bhadeshia, H. K. D. H. 1999. *Neural networks in materials science*. ISIJ International.
- Chawla, Krishan K. 2012. *Composite Materials Science and Engineering*. New York: Springer.
- Che, D., Saxena, I., Han, P., Guo, P., & Ehmann, K. F. . 2014. "Machining of carbon fiber reinforced plastics/polymers: A literature review." *Journal of Manufacturing Science and Engineering*.
- Chung, Deborah D. L. 1994. *Carbon Fiber Composite*. Newton, Ma: Butterworth-Heinemann.
2008. *Concrete Data.xls*. 03 24.
<http://archive.ics.uci.edu/ml/machine-learning-databases/concrete/compressive/>.
- Engineering, University Of Windsor Faculty of. 2015. *Experiment 1- Tension Test*. Windsor: Mechanics of Deformable Bodies.
- Erkan, Ömer, Işık, B., Çiçek, A., & Kara, F. 2013. "Prediction of damage factor in end milling of glass fiber reinforced plastic composites using artificial neural network." *Appl Comps Mater* 517-536.
- Goldman, Robert W. 2001. *Development of a Rollover-Warning Device for Road Vehicles*. Department of Mechanical and Nuclear Engineering: The pennsylvanis State University.
- Group, Renault. 2014. *Renault presents EOLAB*. september 16.
<http://group.renault.com/en/news/blog-renault/renault-presents-eolab-a-prototype-consuming-1-litre-per-100-km/>.
- Hecht-Nielsen, R. 1987. "Kolmogorov's mapping neural network existence theorem." *IEEE First Annual International Conference on Neural Networks* 11-13.
- Hull, D. & Cylne, T. W. 1996. *An introduction to composite materials*. Cambridge: Great Britain University Press.
- J.A Lee, D.P Almond, B Harris. 1999. "the use of neural networks for the prediction of fatigue lives of composite materials." *compoistes part* 1159-1169.
- Jack R. Vinson and Robert L. Sierakowski, . 2008. *The Behavior of Structures Composed of Composite Materials*. Springer.
- Jayabal, S., Rajamuneeswaran, Ramprasath, R., & Balaji, N. S. . 2013. "Artificial neural network

-
- modeling of mechanical properties of calcium carbonate impregnated coir-polyester composites." *Trans Indian Inst Met* 247-255.
- Kadi, H. E., & Al-Assaf, Y. . 2002. "Prediction of the fatigue life of unidirectional glass fiber/epoxy composite laminae using different neural network paradigams." *Copposite Structures* 239-246.
- Ke-Lu Xiang, Pu-Yu Xiang, You-Ping Wu. 2014. "prediction of the fatigue life of natural rubber composites by artificial neural network approaches." *Materials & Design* 180-185.
- Koehn, Philipp. 1994. *Combining Genetic Algorithm and Neural Networks*. Knoxville: The university of Tennessee.
- MATLAB. n.d.
- Mohammed A. Mashrei, R. Seracino, M.S. Rahman. 2013. "Application of artificial neural networks to predict the bond strength of FRP-to-concrete joints." *Construction and Building Materials* 812-821.
- P. Labossiere and N. Turkkkan. 1993. "Failure Prediction of Fiber-Reinforced materials with Neural Networks." *journal of Reinforced Plastics and Composite* 1270-1280.
- Samarasinghe, Sandhya. 2006. *Neural Networks for Applied Sciences and Engineering*. New York: Auerbach Publications.
- Stathakis, D. 2009. "How many hidden layers and nodes?" *International Journal of Remote Sensing* 2133-2147.
- Victor A Gotlib, Tadanobu Sato, Abraham I Beltzer. 2001. "Neural computing of effective properties of random composite materials." *Computers & Structures* 1-6.
- Wang, Xiaochuan. 2013. *43 Cases analysis for neural network by MATLAB*. Beijing: Publisher of Beihang University.
- Xu, J. 2007. "A genetic neural network for predicting materials mechanical properties." *Natural Computation* 710-714.
- Xu, Jianlin. 2007. "A Genetic Neural Network for Predicting Materials Mechanical Properties." *Natural Computation* 710-714.
- Yousef Al-Assaf, Hany El Kadi. 2007. "fatigue life prediction of composite materials using polynomial classifiers and recurrent neural networks." *Composite Structures* 561-569.
- Yung-Kuang Yang, Rong-Tai Yang, Chorng-Jyh Tzeng. 2012. "optimization of mechanical characteristics of short glass fiber and polytetrafluoroethylene reinforced polycarbonate composites using the neural network approach." *Expert Systems with Applications* 378302792.
- Z. Zhang, K. Friedrich, K. Velten. 2002. "Prediction on tribological properties of short fibre composites using artificial neural networks." *Wear* 668-675.

APPENDIX

Appendix A

Appendix A.1 Material Properties

Technical Datasheet

Panex® 35 Continuous Tow

ZOLTEK

Commercial Carbon Fiber for Industrial Applications

Description

Panex® 35 continuous carbon fiber is manufactured from polyacrylonitrile (PAN) precursor. The consistency in yield and mechanical properties that are provided by large filament count strands gives the user the ability to design and manufacture composite materials with greater confidence and allows for efficient and fast buildup of carbon fiber reinforced composite structures.

Panex® 35 50K fibers are available with a variety of sizing formats for different composite processing methods and for compatibility with a wide range of standard resin systems. Zoltek quality focuses on spool to spool consistency yielding low coefficients of variation.

Recommended Use

Any article or component benefiting from the unique properties of carbon fiber composites.

Applications

Wind energy, automotive, petroleum production, aviation, marine, industrial, other transportation, medical (including X-ray), sports and recreation, etc.



Material Overview		SI	US
Tensile Strength		4,137 MPa	600 ksi
Tensile Modulus		242 GPa	35 msi
Electrical Conductivity		0.00155 ohm-cm	0.00061 ohm-in
Density		1.81 g/cc	0.065 lb/in ³
Fiber Diameter		7.2 microns	0.283 mils
Carbon Content		95%	95%
Yield		270 m/kg	400 ft/lb
Spool Weight		5.5 kg, 11 kg	12 lb, 24 lb
Spool Length		1,500 m, 3,000 m	1,640 yd, 3,280 yd

Zoltek Corporation | 3101 McKelvey Rd., St. Louis, MO 63044
 P: 314-291-5110 | F: 314-291-8636 | E: sales@zoltek.com | W: www.zoltek.com

Panex® 35 Continuous Tow



Typical Packaging

Wound on a 3' x 11' (7.6 cm x 28 cm) cardboard spool, sealed in heat shrink covering, and placed in cardboard box; two splices are permitted per every 1,500m spool and five per every 3,000m spool.

- Spool outside diameter:
 - 1,500 m: 180-185 mm x 255-260 mm
 - 3,000 m: 235-245 mm x 255-260 mm

Approval

Geminalscher Lloyd (GL) has granted GL approval to Panex® 35 Continuous Tow for use in wind energy applications.

Geminalscher Lloyd Approval Approval No. MP 1080012/HF



Safety

Obtain, read, and understand the Material Safety Data Sheet (MSDS) before use of this or any other Zoltek product.

Certification

Panex® 35 Continuous Tow is manufactured in accordance with Zoltek's written and published data. A Certificate of Conformance is provided with each shipment.



Zoltek Corporation, 3101 Midway Rd., St. Louis, MO 63144
P: 314-291-5110 | F: 314-291-8536 | E: sales@zoltek.com | www.zoltek.com



401

Product Information

Oct 2013

Ultramid® 8202 HS Polyamide 6



The Chemical Company

Product Description

Ultramid 8202 HS is a heat stabilized, low viscosity, general purpose PA6 injection molding. It possesses the combination of strength and toughness and has excellent chemical and abrasion resistance. The heat stabilizer system extends the retention of properties at the more elevated temperatures. Excellent in filing thin walls or areas.

Applications

Ultramid 8202 HS is generally recommended for drapery hardware, gears, fittings, furniture casters, bearings, handles, clips, zippers and trim parts.

PROPERTY	ASTM Test Method	Purity Value	Conditioned
Specific Gravity	D-792	1.13	
Mold Shrinkage (10° C/h, min)	D-570	0.012	
Moisture, %		1.8	
(24 Hour)		2.7	
(20% RH)		5.5	
(Saturation)			
Mechanical	ASTM Test Method	Dry	Conditioned
Tensile Strength, Yield, MPa (psi)	D-638	126 (18,300)	110 (16,000)
-40C (-40F)			
23C (73F)		79 (11,500)	58 (8,220)
80C (176F)		36 (5,070)	30 (4,350)
121C (250F)		25 (3,620)	20 (2,900)
Tensile Strength, Break, MPa (psi)	D-638		
23C (73F)		79 (10,900)	60 (8,700)
Elongation, Yield, %	D-638		
23C (73F)		4	15
80C (176F)		42	36
121C (250F)		36	40
Elongation, Break, %	D-638		
23C (73F)		56	>100
Flexural Modulus, MPa (psi)	D-790		
-40C (-40F)		3,010 (436,000)	3,060 (437,000)
23C (73F)		2,830 (410,000)	2,480 (357,000)
60C (149F)		500 (72,500)	-
90C (194F)		360 (50,700)	-
121C (250F)		305 (44,200)	-
Flexural Strength, MPa (psi)	D-790		
-40C (-40F)		170 (24,600)	154 (22,300)
23C (73F)		109 (16,700)	96 (13,700)
60C (149F)		30 (4,300)	-
90C (194F)		20 (2,900)	-
121C (250F)		17 (2,400)	-

BASF Corporation
Engineering Plastics
1009 Bidde Avenue
Wyandotte, MI 48182

General Information: 800-86-CRESIN
Technical Assistance: 800-527-TECH (734-324-5150)
Web address: <http://www.plasticsportal.com>

Ultramid® 8202 HS



The Chemical Company

Ultramid® 8202 HS



The Chemical Company

Property	D-785	112	Conditioned
Mechanical	ASTM Test Method		
Modulus (at 23°C) (49F)	D-336	46 (0.8)	43 (0.8)
23°C (73F)		58 (1.1)	NB
Drop Weight Impact (40s, 23°C)	BASF Drop Weight Impact Test	105	>200
THERMAL	ASTM Test Method		
Melting Point (C/F)	D-3418	220 (428)	Conditioned
Heat Deflection @ 204 psi (1.8 MPa) (C/F)	D-648	65 (149)	-
Heat Deflection @ 66 psi (4.5 MPa) (C/F)	D-648	178 (352)	-
Coef. of Linear Thermal Expansion (room to 100°F)	E-831	0.83 X10 ⁻⁴	-
UL RATINGS	UL Test Method		
Flammability Rating, 1.5mm	UL94	V-2	-
Relative Temperature Index, 1.2mm	UL746B		105
Mechanical w/ Impact, C			105
Mechanical w/ Impact, C			130
ELECTRICAL	ASTM Test Method		
Volume Resistivity, 1.5mm	D-267	>1E13	-
Dielectric Strength, Sheet, 1.5mm	D-149	22	-

Processing Guidelines

Material Handling

Max. Water content: 0.15%
 Product is supplied in sealed containers and drying prior to molding is not required. If drying becomes necessary, a dehumidifying or desiccant dryer operating at 80 degC (176 degF) is recommended. Drying time is dependent on moisture level, but 2-4 hours is generally sufficient. Further information concerning safe handling procedures can be obtained from the Material Safety Data Sheet. Alternatively, please contact your BASF representative.

Typical Profile

Melt Temperature 240-280 degC (460-540 degF)
 Mold Temperature 65-90 degC (149-178 degF)
 Injection and Packing Pressure 35-125 bar (500-1800 psi)

Mold Temperatures

A mold temperature of 65-80 (degC (149-178 degF)) is recommended, but temperatures of as low as 10 degC (50 degF) can be used where applicable.

Pressures

Injection pressure controls the filling of the part and should be applied for 80% of run travel. Packing pressure affects the final part and can be used effectively in controlling sink marks and shrinkage. It should be applied and maintained until the gate area is completely frozen off.

Fill Rate

Fast fill rates are recommended to ensure uniform melt delivery to the cavity and prevent premature freezing.

Note

Although all statements and information in this publication are believed to be accurate and reliable, they are presented gratis and for guidance only, and risks and liability for results obtained by use of the products or application of the suggestions described are assumed by the user. NO WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ARE MADE REGARDING PRODUCTS DESCRIBED OR DESIGNS, DATA OR INFORMATION SET FORTH. Statements or suggestions concerning possible use of the products are made without representation or warranty that any such use is free of patent infringement and are not recommendations to infringe any patent. The user should not assume that toxicity data and safety measures are indicated or that other measures may not be required.

BASF Corporation
 Engineering Plastics
 1600 Bidde Avenue
 Wyandotte, MI 48192

General Information: 800-80-RESIN
 Technical Assistance: 800-521-TECH (724-324-5159)
 Web address: <http://www.plasticsportal.com/usa>

BASF Corporation
 Engineering Plastics
 1600 Bidde Avenue
 Wyandotte, MI 48192

General Information: 800-80-RESIN
 Technical Assistance: 800-521-TECH (724-324-5159)
 Web address: <http://www.plasticsportal.com/usa>

Appendix A.2

STANDARD DEVIATION						
<i>Throughput Set Value Dosing 5</i>	0.3519	1.5954	0.7126	0.0000	0.5693	0.0000
<i>Throughput Actual Value Dosing 5</i>	0.8370	1.6354	0.7896	26.7901	0.5981	9.5687
<i>Set Rotational Speed Screw ZSG</i>	0.1055	0.8347	0.0992	15.2223	0.0586	6.3338
<i>Torque ZSG</i>	4.0598	5.2120	4.9189	81.4812	7.3397	43.3375
<i>Resulting Force Servo Die</i>	0.3680	0.1722	0.3482	0.4749	0.2384	0.5704
<i>Force Servo Die Open</i>	0.4822	0.5162	0.4294	5.9733	0.4225	2.1149
<i>Force Servo Die Close</i>	0.5318	0.4701	0.4640	6.1070	0.3822	2.1729
<i>Actual Rotational Speed Screw ZSG</i>	0.1149	0.8335	0.1044	15.2714	0.1126	6.4146
<i>Internal Force ZSG</i>	0.0035	0.0177	0.0039	0.0029	0.0033	0.0031
<i>Internal Temperature ZSG</i>	0.2508	6.1047	0.4691	2.0843	0.6789	0.4034
<i>Temperature ZSG Zone 1</i>	0.2690	6.1172	0.3744	0.6258	0.2435	0.3713
<i>Temperature ZSG Zone 2</i>	0.2478	6.1597	0.1441	0.1934	0.1720	0.0980
<i>Temperature ZSG Zone 9</i>	0.0607	6.1230	0.5973	1.2864	1.1903	0.6273
<i>Temperature ZSG Zone 10</i>	0.0517	6.1229	0.0705	0.3137	0.5109	0.0969
<i>Temperature ZSG Zone 11</i>	0.7035	6.2436	0.1800	0.7661	0.3373	0.1604
<i>Temperature ZSG Zone 12</i>	0.2552	6.1796	0.2349	1.1274	0.2556	0.2263
<i>Temperature ZSG Zone 13</i>	0.0710	6.0095	0.0618	0.1404	0.1182	0.0732
<i>Temperature ZSG Zone 14</i>	0.1249	6.1025	0.2560	1.1225	0.2673	0.1667
<i>Temperature ZSG Zone 15</i>	0.1776	6.1223	0.1907	0.5421	0.3080	0.1934
<i>Temperature ZSG Zone 16</i>	0.2299	6.1225	0.1525	0.5405	0.3644	0.1589
<i>Rotational Speed Motor Shears</i>	24.5360	18.9730	17.8230	20.4261	17.1076	15.1917
<i>Pick-Up Belt Temperature</i>	364.1358	384.5255	359.5270	312.2360	360.7937	348.8935
<i>Positioning Belt 1 Temperature</i>	2.4753	6.3061	2.1954	8.4647	2.3818	2.0801
<i>Positioning Belt 2 Temperature</i>	0.1131	5.8794	0.0921	18.4468	0.0726	0.0986
<i>Positioning Belt 3 Temperature</i>	0.2718	5.8742	0.0847	17.0065	0.1000	0.1896
<i>Positioning Belt 4 Temperature</i>	0.5770	5.8992	0.1575	15.4414	0.1535	0.7930
<i>Positioning Belt 5 Temperature</i>	0.6113	5.8973	0.0984	16.6378	0.0617	0.5104
<i>Positioning Belt 6 Temperature</i>	0.2900	5.8923	0.1028	15.8270	0.1243	0.4791

<i>Temperature</i>						
<i>Safety Tunnel Temperature</i>	0.5355	5.9000	0.1676	15.5231	0.2119	0.8129
<i>Working Tunnel 1 Temperature</i>	0.2078	5.8960	0.1243	6.5838	0.0977	0.1815
<i>Working Tunnel 2 Temperature</i>	1.9644	5.9654	0.4841	4.6187	0.7029	1.8771
<i>Working Tunnel 3 Temperature</i>	0.8783	5.9014	0.3037	4.5386	0.3234	1.3520
<i>Actual Speed Pick-Up Belt</i>	0.9345	5.9068	0.2297	4.8145	0.1939	1.7679
<i>Actual Speed Melt Compound</i>	7.6851	6.1737	9.9541	22.4933	9.0162	27.4428
<i>Actual Speed Positioning Belt 1</i>	40.1015	40.2671	40.6155	40.5273	41.3192	46.6435
<i>Actual Speed Wagon</i>	7.4507	5.6122	9.5169	8.5655	8.3647	9.8737
<i>Set Rotational Speed Screw ZSE</i>	0.3942	1.7870	0.7906	29.8134	0.6376	10.2670
<i>Torque ZSE</i>	14.9461	17.3104	16.1261	197.0795	13.5010	78.8789
<i>Actual Rotational Speed Screw ZSE</i>	0.4526	1.7703	0.8056	29.9668	0.6884	10.3686
<i>Internal Force ZSE</i>	0.0037	0.0063	0.0028	0.0032	0.0024	0.0009
<i>Internal Temperature ZSE</i>	0.2628	6.2036	0.3511	0.2581	0.2521	0.2631
<i>Temperature ZSE Zone 1</i>	0.2402	6.0403	0.0782	1.0305	0.1368	0.0879
<i>Temperature ZSE Zone 2</i>	1.4524	6.2663	1.5091	1.4752	1.4833	1.5924
<i>Temperature ZSE Zone 2</i>	0.1009	6.1226	0.0592	0.1242	0.0702	0.0661
<i>Temperature ZSE Zone 3</i>	0.1276	6.1226	0.0809	0.1433	0.1406	0.0669
<i>Temperature ZSE Zone 4</i>	0.1121	6.1225	0.0697	0.1085	0.0873	0.0613
<i>Temperature ZSE Zone 5</i>	0.6120	5.7073	0.2625	0.2255	0.9431	0.6713
<i>Temperature ZSE Zone 6</i>	0.4666	6.0399	0.5864	0.9108	0.8264	0.6027
<i>Temperature ZSE Zone 7</i>	0.6184	6.2292	0.7003	1.2286	0.4862	0.4292
<i>Temperature ZSE Zone 8</i>	0.1912	6.1229	0.2001	0.2928	0.2328	0.2913
<i>Temperature ZSE Zone 9</i>	0.1154	6.1223	0.1251	0.1510	0.1186	0.1388
<i>Temperature ZSE Zone 10</i>	0.2628	6.2036	0.3511	0.2581	0.2521	0.2631

Appendix A.3

EAN VALUE		Trial1	Trial2	Trial3	Trial4	Trial5
Throughput Set Value Dosing 5	[kg/hr]	67.01	63.52	60.05	56.22	52.38
Throughput Actual Value Dosing 5	[kg/hr]	67.10	63.56	43.46	56.23	50.68
Set Rotational Speed Screw ZSG	[kg/hr]	36.66	36.27	28.13	36.09	35.04
Torque ZSG	[Nm]	164.22	184.58	156.29	275.55	266.84
Resulting Force Servo Die	[kN]	0.64	0.31	0.37	0.41	0.54
Force Servo Die Open	[kN]	13.49	13.71	10.01	13.36	13.05
Force Servo Die Close	[kN]	14.13	13.93	10.28	13.75	13.51
Actual Rotational Speed Screw ZSG	[U/min]	36.56	36.19	28.17	35.99	35.02
Internal Force ZSG	[bar]	0.76	0.75	0.75	0.75	0.75
Internal Temperature ZSG	[°C]	269.09	270.39	271.45	276.36	278.59
Temperature ZSG Zone 1	[°C]	269.57	269.47	270.04	268.91	269.51
Temperature ZSG Zone 2	[°C]	271.51	271.88	271.90	272.14	272.42
Temperature ZSG Zone 9	[°C]	269.88	271.21	272.92	276.25	279.46
Temperature ZSG Zone 10	[°C]	269.88	270.02	269.98	269.93	270.03
Temperature ZSG Zone 11	[°C]	275.21	275.54	275.82	279.20	279.25
Temperature ZSG Zone 12	[°C]	272.39	272.98	273.21	275.55	275.06
Temperature ZSG Zone 13	[°C]	264.87	265.01	264.95	265.00	265.00
Temperature ZSG Zone 14	[°C]	268.99	269.79	269.88	271.91	271.87
Temperature ZSG Zone 15	[°C]	269.86	270.01	269.82	269.98	270.00
Temperature ZSG Zone 16	[°C]	269.88	270.03	269.78	269.95	269.97
Rotational Speed Motor Shears	[U/min]	114.68	178.50	248.52	309.42	365.64
Pick-Up Belt Temperature	[°C]	56.11	50.26	37.45	50.19	47.59
Positioning Belt 1 Temperature	[°C]	261.56	261.27	258.19	254.89	252.25
Positioning Belt 2 Temperature	[°C]	259.15	259.30	252.99	249.64	249.62
Positioning Belt 3 Temperature	[°C]	258.93	259.12	253.25	249.42	249.35
Positioning Belt 4 Temperature	[°C]	259.98	260.03	254.71	249.91	249.90
Positioning Belt 5 Temperature	[°C]	259.93	260.01	254.06	250.01	249.97
Positioning Belt 6 Temperature	[°C]	259.71	259.88	254.51	249.93	249.89
Safety Tunnel Temperature	[°C]	259.98	260.01	254.73	249.96	249.92
Working Tunnel 1 Temperature	[°C]	259.89	260.01	257.81	250.27	250.09
Working Tunnel 2 Temperature	[°C]	258.03	259.12	257.58	249.11	249.02
Working Tunnel 3 Temperature	[°C]	258.94	259.55	258.01	249.61	249.39
Actual Speed Pick-Up Belt	[mm/s]	259.51	259.90	258.12	249.91	249.80
Actual Speed Melt Compound	[mm/s]	11.53	11.79	10.77	11.71	15.18
Actual Speed Positioning Belt 1	[mm/s]	17.76	18.23	15.44	18.28	21.48
Actual Speed Wagon	[mm/s]	0.33	-0.04	0.18	0.11	-0.15
Set Rotational Speed Screw ZSE	[U/min]	75.06	71.15	52.41	62.97	56.79
Torque ZSE	[Nm]	488.95	471.67	336.41	492.18	463.51

<i>Actual Rotational Speed Screw ZSE</i>	<i>[U/min]</i>	74.79	71.35	52.48	62.91	56.55
<i>Internal Force ZSE</i>	<i>[Bar]</i>	-0.21	-0.21	-0.21	-0.22	-0.22
<i>Internal Temperature ZSE</i>	<i>[°C]</i>	273.45	274.08	274.26	273.09	272.63
<i>Temperature ZSE Zone 1</i>	<i>[°C]</i>	266.24	266.39	266.98	266.50	266.43
<i>Temperature ZSE Zone 2</i>	<i>[°C]</i>	269.82	270.00	270.11	269.97	270.01
<i>Temperature ZSE Zone 2</i>	<i>[°C]</i>	269.86	270.00	270.03	269.98	270.00
<i>Temperature ZSE Zone 3</i>	<i>[°C]</i>	269.86	270.00	270.02	269.99	270.01
<i>Temperature ZSE Zone 4</i>	<i>[°C]</i>	269.86	270.00	270.01	270.01	270.01
<i>Temperature ZSE Zone 5</i>	<i>[°C]</i>	251.57	251.97	251.95	248.39	250.55
<i>Temperature ZSE Zone 6</i>	<i>[°C]</i>	266.19	266.14	266.37	266.33	266.13
<i>Temperature ZSE Zone 7</i>	<i>[°C]</i>	274.04	274.99	274.73	273.64	275.00
<i>Temperature ZSE Zone 8</i>	<i>[°C]</i>	269.88	270.00	269.99	270.06	270.01
<i>Temperature ZSE Zone 9</i>	<i>[°C]</i>	269.87	270.01	269.96	270.01	270.00
<i>Temperature ZSE Zone 10</i>	<i>[°C]</i>	273.45	274.08	274.26	273.09	272.63

Appendix B

Appendix B.1

Tension Test Processing Parameters					
Specimen	1		Final width	25	mm
Specimen label	H1-T-C-L		Final thickness	3	mm
Width	25.1	mm	Final length	50	mm
Thickness	3.2	mm	Final diameter	14	mm
Length	50	mm	Final Inner diameter	8	mm
Diameter	14	mm	Final outer diameter	10	mm
Inner diameter	8	mm	Final wall thickness	1	mm
Outer diameter	10	mm	Final area	0.75	cm ²
Wall thickness	1	mm	Final linear density	100	tex
Area	0.8032	cm ²	Type		
Linear density	100	tex	Operator	Ying	
Sled weight	10	N	Sample Identification	Henkel	
Loading span	10	mm	Machine model	Instron 8800	
Support span	100	mm	Rate 1	2 mm/min	
Span ratio	2		Humidity	0.49	
Fixture type	4-point		Temperature	23C	
Extensometer	0.05	m	Reference Number	Oct 29th, 2014	

Appendix B.2

<i>Fiber Weight</i>	<i>Fiber Orientation</i>	<i>Specimen Number</i>	<i>Young's modulus</i>	<i>Tensile Strength</i>	<i>YM</i>	TS
%					<i>normalized</i>	normalized
25	0	1	163.9929	192.2186	0.633708	0.633708
25	0	2	175.761	183.1407	0.58675	0.58675
25	0	3	192.8338	197.4853	0.660952	0.660952
25	0	4	233.8792	179.2662	0.566709	0.566709
25	0	5	243.9888	208.5161	0.718011	0.718011
25	90	1	128.5817	109.4259	0.205442	0.205442
25	90	2	105.3603	101.0734	0.162236	0.162236
25	90	3	111.4122	119.7388	0.258788	0.258788
25	90	4	93.65933	106.2884	0.189213	0.189213
30	0	1	308.6791	213.9192	0.74596	0.74596
30	0	2	233.1917	227.564	0.816541	0.816541
30	0	3	223.8519	208.9298	0.720151	0.720151

30	0	4	243.3876	198.9086	0.668314	0.668314
30	0	5	201.5158	212.5412	0.738832	0.738832
30	90	1	123.2578	127.8231	0.300606	0.300606
30	90	2	102.5207	84.63088	0.077184	0.077184
30	90	3	141.2559	137.2595	0.349418	0.349418
30	90	4	94.56492	109.0475	0.203485	0.203485
35	0	1	231.2057	184.656	0.594589	0.594589
35	0	2	285.2188	190.4354	0.624484	0.624484
35	0	3	215.4965	213.3801	0.743171	0.743171
35	0	4	247.2964	216.1841	0.757676	0.757676
35	0	5	232.6405	201.5783	0.682124	0.682124
35	90	1	94.0076	108.8007	0.202208	0.202208
35	90	2	102.9919	103.3278	0.173898	0.173898
35	90	3	123.4473	112.3698	0.22067	0.22067
35	90	4	97.50725	117.134	0.245314	0.245314
40	0	1	267.3084	185.8378	0.600702	0.600702
40	0	2	270.1817	231.1717	0.835203	0.835203
40	0	3	204.2819	176.6107	0.552972	0.552972
40	0	4	316.9048	252.6693	0.946405	0.946405
40	0	5	309.9997	194.3213	0.644585	0.644585
40	90	1	137.946	101.1056	0.162403	0.162403
40	90	2	140.9496	72.07844	0.012253	0.012253
40	90	3	177.1003	86.45568	0.086623	0.086623
40	90	4	105.6971	99.2853	0.152987	0.152987
45	0	1	296.7338	173.8167	0.53852	0.53852
45	0	2	271.8612	232.0374	0.839681	0.839681
45	0	3	401.1213	205.9707	0.704845	0.704845
45	0	4	239.9607	155.3034	0.442755	0.442755
45	0	5	287.4778	240.0088	0.880915	0.880915
45	90	1	127.548	98.50387	0.148945	0.148945
45	90	2	105.2286	103.0466	0.172443	0.172443
45	90	3	103.6183	85.01565	0.079174	0.079174
45	90	4	150.4958	89.87189	0.104294	0.104294
50	0	1	297.2958	252.5146	0.945604	0.945604
50	0	2	388.1345	204.8349	0.698969	0.698969
50	0	3	286.4889	263.0304	1	1
50	0	4	278.2419	206.0701	0.705359	0.705359
50	0	5	274.271	210.8878	0.730279	0.730279
50	90	1	133.362	75.75994	0.031296	0.031296
50	90	2	110.7088	75.41561	0.029515	0.029515

50	90	3	136.2055	69.70969	0	0
50	90	4	161.9398	84.97261	0.078951	0.078951
nor: normalized data.						

Appendix B.3

	<i>Young's modulus</i>		<i>Tensile Strength</i>	
	0	90	0	90
Trial1 /25%	163.9929	128.5817	192.2186243	109.4259461
Trial1 /25%	175.761	105.3603	183.1406883	101.0733562
Trial1 /25%	192.8338	111.4122	197.4853065	119.7387694
Trial1 /25%	233.8792	93.65933	179.2661837	106.2883953
Trial1 /25%	243.9888		208.5160621	
Trial2 /30%	308.6791	123.2578	213.919201	127.8230809
Trial2 /30%	233.1917	102.5207	227.5639931	84.6308767
Trial2 /30%	223.8519	141.2559	208.9297767	137.2594616
Trial2 /30%	243.3876	94.56492	198.9086162	109.0474838
Trial2 /30%	201.5158		212.5412088	
Trial3 / 35%	231.2057	94.0076	184.6559997	108.8007313
Trial3 / 35%	285.2188	102.9919	190.4353796	103.3278268
Trial3 / 35%	215.4965	123.4473	213.3801053	112.3698217
Trial3 / 35%	247.2964	97.50725	216.1841369	117.1340377
Trial3 / 35%	232.6405		201.5783097	
Trial4 / 40%	267.3084	137.946	185.8377753	101.1056396
Trial4 / 40%	270.1817	140.9496	231.1716862	72.0784358
Trial4 / 40%	204.2819	177.1003	176.610679	86.45567703
Trial4 / 40%	316.9048	105.6971	252.6693212	99.28529644
Trial4 / 40%	309.9997		194.3213376	
Trial5/ 45%	296.7338	127.548	173.8167413	98.50387297
Trial5/ 45%	271.8612	105.2286	232.0373507	103.0465759
Trial5/ 45%	401.1213	103.6183	205.9707416	85.01565045
Trial5/ 45%	239.9607	150.4958	155.3034121	89.8718872
Trial5/ 45%	287.4778		240.0088396	
Trial6 / 50%	297.2958	133.362	252.5145652	75.7599355
Trial6 / 50%	388.1345	110.7088	204.8349047	75.41561171
Trial6 / 50%	286.4889	136.2055	263.0303819	69.70968684
Trial6 / 50%	278.2419	161.9398	206.0700962	84.97261065
Trial6 / 50%	274.271		210.8878269	

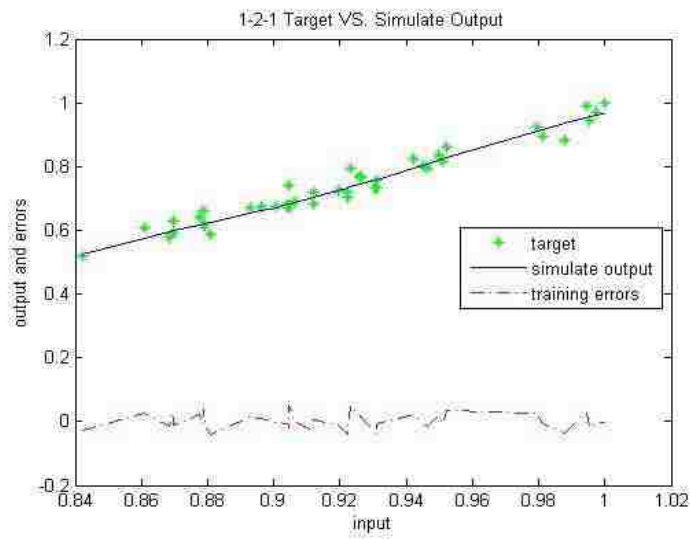
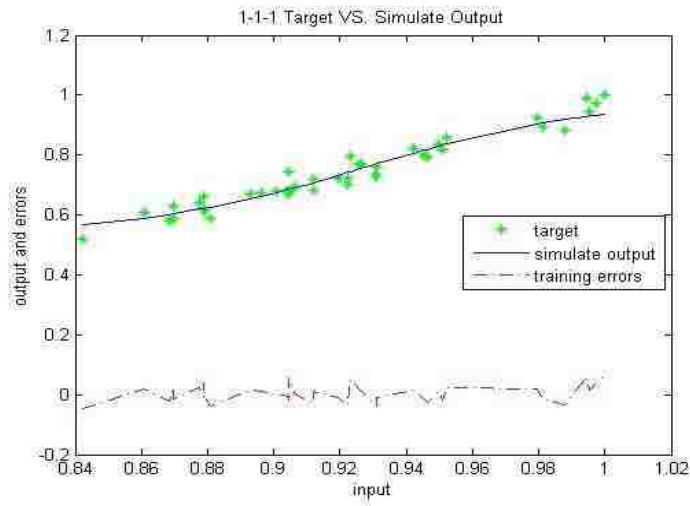
Appendix B.4

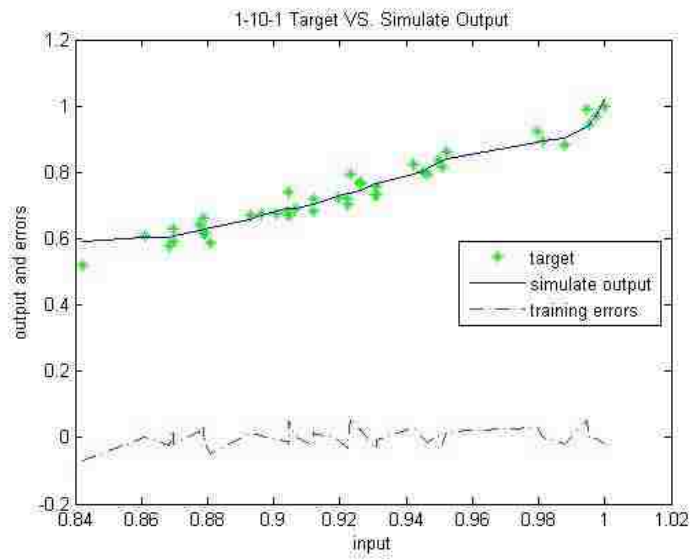
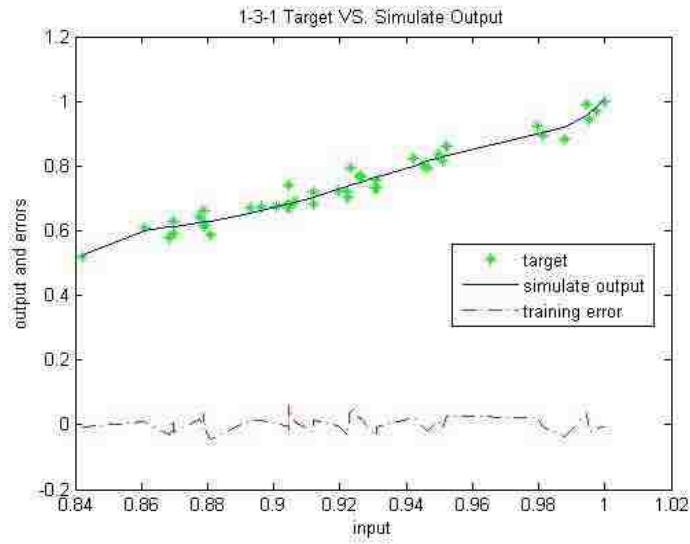
	<i>Young's modulus</i>		<i>Tensile Strength</i>	
	0	90	0	90
Trial1 /25%	0	0.726039	0.342674	0.79722
Trial1 /25%	0.049628	0.594919	0.258406	0.736367
Trial1 /25%	0.121626	0.629091	0.391563	0.872353
Trial1 /25%	0.294719	0.528849	0.22244	
Trial1 /25%	0.337353		0.493958	0.774361
Trial2 /30%	0.61016	0.695978	0.544114	0.931252
Trial2 /30%	0.29182	0.578885	0.670775	0.616576
Trial2 /30%	0.252433	0.797604	0.497799	1
Trial2 /30%	0.334817	0.533962	0.404775	0.794462
Trial2 /30%	0.158239		0.531323	
Trial3 / 35%	0.283445	0.530816	0.272472	0.792665
Trial3 / 35%	0.511225	0.581545	0.32612	0.752792
Trial3 / 35%	0.217197	0.697047	0.53911	0.818667
Trial3 / 35%	0.351301	0.550576	0.565139	0.853377
Trial3 / 35%	0.289496		0.429557	
Trial4 / 40%	0.435694	0.778914	0.283442	0.736602
Trial4 / 40%	0.447812	0.795874	0.704264	0.525125
Trial4 / 40%	0.169904	1	0.19779	0.62987
Trial4 / 40%	0.644849	0.59682	0.903821	0.72334
Trial4 / 40%	0.615729		0.362193	
Trial5/ 45%	0.559785	0.720202	0.171854	0.717647
Trial5/ 45%	0.454894	0.594175	0.7123	0.750743
Trial5/ 45%	1	0.585083	0.470331	0.619379
Trial5/ 45%	0.320366	0.849777	0	0.654759
Trial5/ 45%	0.520751		0.786297	
Trial6 / 50%	0.562155	0.753031	0.902385	0.551947
Trial6 / 50%	0.945233	0.625119	0.459787	0.549438
Trial6 / 50%	0.516581	0.769087	1	0.507868
Trial6 / 50%	0.481802	0.914396	0.471253	0.619066
Trial6 / 50%	0.465057		0.515975	

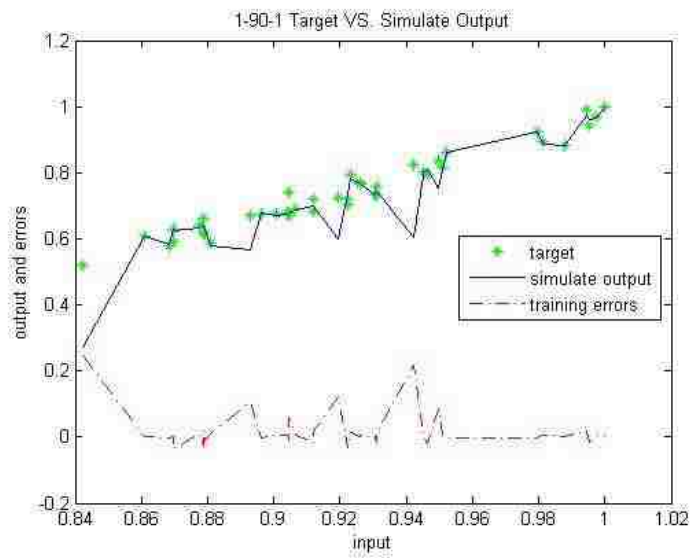
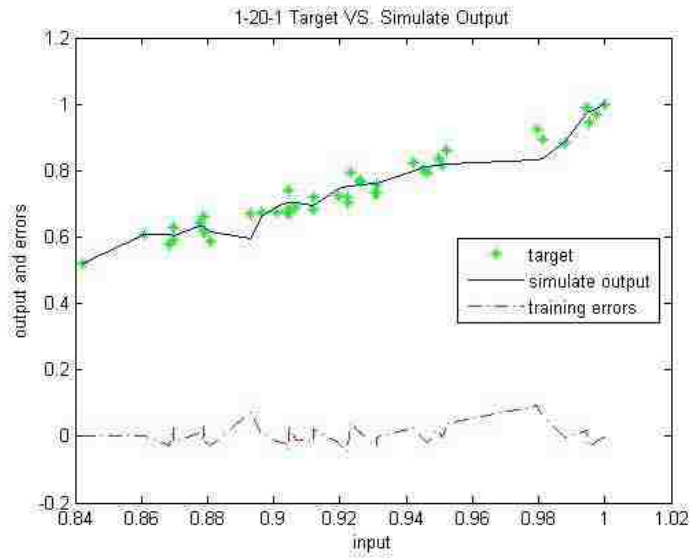
Appendix C

Appendix C.1

Plot of different numbers of hidden neurons







Appendix C.2

Cement (component 1) (kg in a m ³ mixture)	Concrete compressive strength (MPa, megapascals)	Cement (normalized)	Concrete compressive strength(normalized)
200.0	7.84	0.84213	0.519598
225.0	11.17	0.86085	0.608933
236.0	9.99	0.868437	0.580784
238.0	12.05	0.869778	0.628122
238.1	10.34	0.869845	0.589361
250.0	12.73	0.877597	0.642023
252.0	11.47	0.878863	0.615543
252.0	13.71	0.878863	0.660713

252.5	11.48	0.879178	0.615937
255.0	10.22	0.880744	0.586568
275.0	14.20	0.892746	0.669564
281.0	14.50	0.896176	0.674777
289.0	14.60	0.900638	0.676463
295.0	14.80	0.903904	0.679884
295.8	14.84	0.904335	0.680694
296.0	14.20	0.904442	0.669564
296.0	18.91	0.904442	0.741819
300.0	15.58	0.906576	0.692833
310.0	14.99	0.911787	0.683155
310.0	17.24	0.911787	0.718513
325.0	17.54	0.919298	0.722814
331.0	16.26	0.922205	0.703763
331.0	17.44	0.922205	0.721322
333.0	23.40	0.923163	0.7956
339.0	20.97	0.926001	0.767843
339.0	21.16	0.926001	0.770156
339.2	21.18	0.926095	0.770386
349.0	18.13	0.930622	0.731108
350.0	18.13	0.931077	0.731108
350.0	20.28	0.931077	0.759406
375.0	26.06	0.942043	0.822736
382.0	24.00	0.944982	0.801942
382.0	24.00	0.944982	0.801942
382.5	24.07	0.94519	0.802623
385.0	23.22	0.946226	0.793615
393.0	27.74	0.949494	0.838501
397.0	25.45	0.951104	0.816717
400.0	30.14	0.952301	0.85945
475.0	38.60	0.979615	0.921869
480.0	34.57	0.981279	0.894018
500.0	33.21	0.987768	0.883909
522.0	50.51	0.994612	0.989707
525.0	42.42	0.995522	0.945677
531.3	46.90	0.997418	0.970981
540.0	52.61	1	1

Appendix D

```
clc
clear
%% Import the data of 25%
[~, ~, raw] = xlsread('C:\Users\user\Documents\MATLAB\gogogo\WT25wt.xls','Sheet3','A3:BC2542');
data = reshape([raw{:}],size(raw));
WT25wt = dataset;
WT25wt.VarName1 = data(:,1);
WT25wt.ThroughputSetValueDosing5 = data(:,2);
WT25wt.ThroughputActualValueDosi = data(:,3);
WT25wt.SetRotationalSpeedScrewZS = data(:,4);
WT25wt.TorqueZSG = data(:,5);
WT25wt.ResultingForceServoDie = data(:,6);
WT25wt.ForceServoDieOpen = data(:,7);
WT25wt.ForceServoDieClose = data(:,8);
WT25wt.ActualRotationalSpeedScre = data(:,9);
WT25wt.InternalForceZSG = data(:,10);
WT25wt.InternalTemperatureZSG = data(:,11);
WT25wt.TemperatureZSGZone1 = data(:,12);
WT25wt.TemperatureZSGZone2 = data(:,13);
WT25wt.TemperatureZSGZone9 = data(:,14);
WT25wt.TemperatureZSGZone10 = data(:,15);
WT25wt.TemperatureZSGZone11 = data(:,16);
WT25wt.TemperatureZSGZone12 = data(:,17);
WT25wt.TemperatureZSGZone13 = data(:,18);
WT25wt.TemperatureZSGZone14 = data(:,19);
WT25wt.TemperatureZSGZone15 = data(:,20);
WT25wt.TemperatureZSGZone16 = data(:,21);
WT25wt.ActualCycleNumber = data(:,22);
WT25wt.RotationalSpeedMotorShear = data(:,23);
WT25wt.PickUpBeltTemperature = data(:,24);
WT25wt.PositioningBelt1Temperatu = data(:,25);
WT25wt.PositioningBelt2Temperatu = data(:,26);
WT25wt.PositioningBelt3Temperatu = data(:,27);
WT25wt.PositioningBelt4Temperatu = data(:,28);
WT25wt.PositioningBelt5Temperatu = data(:,29);
WT25wt.PositioningBelt6Temperatu = data(:,30);
WT25wt.SafetyTunnelTemperature = data(:,31);
WT25wt.WorkingTunnel1Temperature = data(:,32);
WT25wt.WorkingTunnel2Temperature = data(:,33);
```

```

WT25wt.WorkingTunnel3Temperature = data(:,34);
WT25wt.ActualSpeedPickUpBelt = data(:,35);
% WT25wt.ActualSpeedMeltCompound = data(:,36);
WT25wt.ActualSpeedPositioningBel = data(:,37);
WT25wt.ActualSpeedWagon = data(:,38);
WT25wt.SetRotationalSpeedScrewZS1 = data(:,39);
WT25wt.TorqueZSE = data(:,40);
WT25wt.ActualRotationalSpeedScre1 = data(:,41);
WT25wt.InternalForceZSE = data(:,42);
WT25wt.InternalTemperatureZSE = data(:,43);
WT25wt.VarName44 = data(:,44);
WT25wt.TemperatureZSEZone1 = data(:,45);
WT25wt.TemperatureZSEZone2 = data(:,46);
WT25wt.TemperatureZSEZone3 = data(:,47);
WT25wt.TemperatureZSEZone4 = data(:,48);
WT25wt.TemperatureZSEZone5 = data(:,49);
WT25wt.TemperatureZSEZone6 = data(:,50);
WT25wt.TemperatureZSEZone7 = data(:,51);
WT25wt.TemperatureZSEZone8 = data(:,52);
WT25wt.TemperatureZSEZone9 = data(:,53);
WT25wt.TemperatureZSEZone10 = data(:,54);
% WT25wt.TemperatureZSEZone11 = data(:,55);
clearvars dataraw;
%% Import the data of 30%,35%,40%,45%,50% are the same method
%% get the mean and standard deviation of the input
wt25=double(WT25wt);wt30=double(WT30wt);wt35=double(WT35wt);wt40=double(WT40wt);
wt45=double(WT45wt);wt50=double(WT50wt);
Processparametername(1,:)=WT50wt(1,:);Processparametername(1,:)=[];
wt25mean=mean(wt25,1);wt30mean=mean(wt30,1);wt35mean=mean(wt35,1);
wt40mean=mean(wt40,1);wt45mean=mean(wt45,1);wt50mean=mean(wt50,1);
wt25std=std(wt25,1);wt30std=std(wt30,1);wt35std=std(wt35,1);
wt40std=std(wt40,1);wt45std=std(wt45,1);wt50std=std(wt50,1);
%% obtain the input of mean and std
input_mean=[wt25mean; wt30mean; wt35mean;wt40mean; wt45mean; wt50mean];
input_std=[wt25std; wt30std; wt35std;wt40std; wt45std; wt50std];
input_mean(:,1)=[];
input_std(:,1)=[];
%% load target from the excel files
%% Import the data
[~, ~, raw] = xlsread('C:\Users\user\Documents\MATLAB\gogogo\Orgnaized ANN data.xlsx','Raw Data Standard
Deviation','BF1:BG57');

```

```

raw(cellfun(@(x) ~isempty(x) && isnumeric(x) && isnan(x),raw)) = {};
R = cellfun(@(x) ~isnumeric(x) && ~islogical(x),raw); % Find non-numeric cells
raw(R) = {NaN};
data = reshape([raw{:}],size(raw));
OrgnaizedANNdataS1 = table;
OrgnaizedANNdataS1.Youngsmodulus = data(:,1);
OrgnaizedANNdataS1.TensileStrength = data(:,2);
clearvars datarawR;
Youngsmodulus=double(OrgnaizedANNdataS1.Youngsmodulus);
TensileStrength=double(OrgnaizedANNdataS1.TensileStrength);
%%
Youngsmodulus(1:3,:)=[];
TensileStrength(1:3,:)=[];
a11mean=mean(Youngsmodulus(1:5,:));
a12mean=mean(Youngsmodulus(6:9,:));
a21mean=mean(Youngsmodulus(10:14,:));
a22mean=mean(Youngsmodulus(15:18,:));
a31mean=mean(Youngsmodulus(19:22,:));
a32mean=mean(Youngsmodulus(23:27,:));
a41mean=mean(Youngsmodulus(28:32,:));
a42mean=mean(Youngsmodulus(33:36,:));
a51mean=mean(Youngsmodulus(37:41,:));
a52mean=mean(Youngsmodulus(42:45,:));
a61mean=mean(Youngsmodulus(46:50,:));
a62mean=mean(Youngsmodulus(51:54,:));
Youngsmodulus0mean=[a11mean; a21mean; a31mean; a41mean; a51mean; a61mean];
Youngsmodulus90mean=[a12mean;a22mean;a32mean;a42mean;a52mean;a62mean];
a11std=std(Youngsmodulus(1:5,:));
a12std=std(Youngsmodulus(6:9,:));
a21std=std(Youngsmodulus(10:14,:));
a22std=std(Youngsmodulus(15:18,:));
a31std=std(Youngsmodulus(19:22,:));
a32std=std(Youngsmodulus(23:27,:));
a41std=std(Youngsmodulus(28:32,:));
a42std=std(Youngsmodulus(33:36,:));
a51std=std(Youngsmodulus(37:41,:));
a52std=std(Youngsmodulus(42:45,:));
a61std=std(Youngsmodulus(46:50,:));
a62std=std(Youngsmodulus(51:54,:));
Youngsmodulus0std=[a11std;a21std;a31std;a41std;a51std;a61std];
Youngsmodulus90std=[a12std;a22std;a32std;a42std;a52std;a62std];

```

```

%%
b11mean=mean(TensileStrength(1:5,:));
b12mean=mean(TensileStrength(6:9,:));
b21mean=mean(TensileStrength(10:14,:));
b22mean=mean(TensileStrength(15:18,:));
b31mean=mean(TensileStrength(19:22,:));
b32mean=mean(TensileStrength(23:27,:));
b41mean=mean(TensileStrength(28:32,:));
b42mean=mean(TensileStrength(33:36,:));
b51mean=mean(TensileStrength(37:41,:));
b52mean=mean(TensileStrength(42:45,:));
b61mean=mean(TensileStrength(46:50,:));
b62mean=mean(TensileStrength(51:54,:));
TensileStrength0mean=[b11mean; b21mean; b31mean; b41mean; b51mean; b61mean];
TensileStrength90mean=[b12mean;b22mean;b32mean;b42mean;b52mean;b62mean];
b11std=std(TensileStrength(1:5,:));
b12std=std(TensileStrength(6:9,:));
b21std=std(TensileStrength(10:14,:));
b22std=std(TensileStrength(15:18,:));
b31std=std(TensileStrength(19:22,:));
b32std=std(TensileStrength(23:27,:));
b41std=std(TensileStrength(28:32,:));
b42std=std(TensileStrength(33:36,:));
b51std=std(TensileStrength(37:41,:));
b52std=std(TensileStrength(42:45,:));
b61std=std(TensileStrength(46:50,:));
b62std=std(TensileStrength(51:54,:));
TensileStrength0std=[b11std;b21std;b31std;b41std;b51std;b61std];
TensileStrength90std=[b12std;b22std;b32std;b42std;b52std;b62std];
%%
%% organized input and target
input=input_mean';      target=[Youngsmodulus0mean      Youngsmodulus90mean      TensileStrength0mean
TensileStrength90mean]';
% input=input_std'; target=[Youngsmodulus0std Youngsmodulus90std TensileStrength0std TensileStrength90std]';
clearvars -except input target input_mean input_std
%% normalize data
[inputn,inputps]=mapminmax(input,0,1);
[targetn,targetps]=mapminmax(target,0,1);
%% build neural network
[r,c]=size(inputn);
P=inputn;T=targetn;

```

```

S=2*r+1;
% S=104;
% % while S<=120 ;
net=newff(P,T,S);
% net=newcfc(P,T,S);
% net=newelm(P,T,S);
% %Training function selet
net.layers{1}.transferFcn = 'logsig';
net.layers{2}.transferFcn = 'purelin';
net.trainFcn='trainlm';
[net,tr]=train(net,inputn,targetn);
outputn=sim(net,inputn
%reverse normolize
output=mapminmax('reverse',outputn,targetps,0,1);
%network error figure
plotperform(tr);
errors=output-target;
errorn=outputn-targetn;
figure(1)
plot(1:6,errors(1,:),'bo',1:6,errors(2,:),'g+',1:6,errors(3,:),'rx',1:6,errors(4,:),'c*')
title('BP Network Simulate error','fontsize',12)
ylabel('Error','fontsize',12)
xlabel('Output Sample','fontsize',12)
figure(2)
plot(1:6,output,'s',1:6,target,'-.+');
title('Comparison on Target to Output')
ylabel('Output (Youngs modulus and Tensil Strength)')
xlabel('Output Samples')
figure(3)
plot(target,output,'o');
axis([0 500 0 500]);
title('Regression of Target and Output');
ylabel('Output');
xlabel('Target');
figure(4)
plot(targetn,outputn,'s');
axis([0 2.5 0 2.5]);
title('Regression of Normalized Target and Output');
ylabel('Normalized Output');
xlabel('Normalized Target');
hiddenlayerweight=net.iw{1,1};

```

```

outputlayerweight=net.lw{2,1};
hiddenlayerbias=net.b{1,1};
outputlayerbias=net.b{2,1};
bestperform=tr.best_perf;
tr;
performn=perform(net,targetn,outputn);
performance(S,:)=performn
xmean=mean(inputn,2)';
[rm,cm]=size(xmean);
i=1;
while i<=S
x(i,:)=xmean(1,:);
i=i+1;
end
a=hiddenlayerweight;
a0=hiddenlayerbias;
b=outputlayerweight;
b1=b(1,:);b2=b(2,:);b3=b(3,:);b4=b(4,:);
z=outputn;
% clearvars -except a a0 b x z b1
%calculate u
u=sum(a.*x,2)+a0;
%calculate sensitivity
%dz1dx
ii=1;
while ii<= cm
dz1dx(ii,:)=sum(b1.*a(:,ii).*((1+exp(-u)).^2));
ii=ii+1;
end
%dz2dx
ii=1;
while ii<= cm
dz2dx(ii,:)=sum(b2.*a(:,ii).*((1+exp(-u)).^2));
ii=ii+1;
end
%dz3dx
ii=1;
while ii<= cm
dz3dx(ii,:)=sum(b3.*a(:,ii).*((1+exp(-u)).^2));
ii=ii+1;
end

```

```
%dz4dx
ii=1;
while ii<= cm
dz4dx(ii,:)=sum(b4.*a(:,ii).*((1+exp(-u)).^2));
ii=ii+1;
end
S
dzdx=[dz1dx/S dz2dx/S dz3dx/S dz4dx/S];
% clearvars -except dzdx10 a b u
```

VITA AUCTORIS

NAME: Xu Zhang

PLACE OF BIRTH: Yinchuan, Ningxia, China

YEAR OF BIRTH: 1989

EDUCATION: No. 9 Middle/High School, Yinchuan, Ningxia, China, 2001

Beijing Jiaotong University, B.Sc., Beijing, China, 2007

University of Windsor, M.Eng (Automotive Option), Windsor, ON, 2011

University of Windsor, M.A.Sc. Mechanical Engineering, Windsor, ON, 2013