

NEWCASTLE UNIVERSITY

# Advancing the Analysis of Architectural Fabric Structures

---

A Thesis submitted for the Higher Degree of Doctor of Philosophy in  
the School of Civil Engineering and Geosciences, Newcastle University

**By**

**Nicola Smithies née Bartle**

**Supervisors:**

**Prof. P. D. Gosling**

**Dr. B. N. Bridgens**

**March 2015**



## Abstract

In current practice a plane stress framework comprising elastic moduli and Poisson's ratios is most commonly used to represent the mechanical properties of architectural fabrics. This is often done to enable structural analysis utilising commercially available, non-specialist, finite element packages. Plane stress material models endeavour to fit a flat plane to the highly non-linear stress strain response surface of architectural fabric.

Neural networks have been identified as a possible alternative to plane stress material models. Through a process of training they are capable of capturing the relationship between experimental input and output data. With the addition of historical inputs and internal variables it has been demonstrated that neural network models are capable of representing complex history dependant behaviour. The resulting neural network architectural fabric material models have been implemented within custom large strain finite element code. The finite element code, capable of using either a neural network or plane stress material model, utilises a dynamic relaxation solution algorithm and includes geodesic string control for soap film form-finding. Analytical FORM reliability analysis using implied stiffness matrices' derived from the equations of the neural network model has also been investigated.



## Acknowledgements

I would like to thank my supervisors Peter Gosling and Ben Bridgens for their guidance and encouragement. A particular thanks must go to Peter for his reassurance and humour during periods of discombobulation. I would also like to thank the technical staff William Cragie and Stuart Patterson along with my colleague Alex Colman for their help in the lab. I would also like to acknowledge Lei Zhang whose research code provided a basis for the Finite Element code presented in this thesis.

I would like to thank the Engineering and Physical Sciences Research Council for providing funding for this work.

Finally I would like to thank my husband Adam Smithies and my parents, Julie and Stephen Bartle. This PhD would not have been possible without their constant and unwavering support.



## Outline Contents

*Detailed tables of contents along with tables of figures and tables may be found at the beginning of each chapter.*

|   |     |
|---|-----|
| Abstract .....  | i   |
| Acknowledgements .....  | iii |
| Notation.....   | iii |
| <br>  |     |
| Chapter 1. Introduction.....  | 5   |
| 1.1    Background.....  | 5   |
| 1.2    Aim and Objectives.....                                      | 10  |
| 1.3    Scope .....  | 11  |
| 1.4    Thesis Structure.....  | 11  |
| 1.5    Conferences and Publications.....                            | 13  |
| <br>  |     |
| Chapter 2. Literature Review.....                                   | 15  |
| 2.1    Simulation of Architectural Fabric Structures .....          | 19  |
| 2.2    Determination and Representation of Material Behaviour ..... | 34  |
| 2.3    Neural Network Material Modelling .....                      | 56  |
| 2.4    Summary and Conclusions.....                                 | 66  |
| <br>  |     |
| Chapter 3. Finite Element Formulation.....                          | 71  |
| 3.1    Element Equations.....                                       | 78  |
| 3.2    Analysis Examples .....                                      | 108 |
| 3.3    Patterning.....  | 129 |
| 3.4    Summary and Conclusions.....                                 | 136 |

|  |     |
|--|-----|
| Chapter 4. Neural Network Training and Validation.....               | 139 |
| 4.1 Neural Networks .....  | 146 |
| 4.2 Biaxial Testing.....   | 155 |
| 4.3 Response Surface Neural Network Material Model.....              | 157 |
| 4.4 History Neural Network Material Model.....                       | 177 |
| 4.5 Summary and Conclusions.....                                     | 209 |
| Chapter 5. Neural Network Material Model Implementation.....         | 211 |
| 5.1 Implied Stiffness Matrix .....                                   | 218 |
| 5.2 Response Surface Style Network Implementation.....               | 231 |
| 5.3 History neural network implementation.....                       | 249 |
| 5.4 Summary and Conclusions.....                                     | 273 |
| Chapter 6. Reliability Analysis .....                                | 275 |
| 6.1 Limit State Functions and Statistical Variables .....            | 280 |
| 6.2 Solution Procedure.....  | 281 |
| 6.3 Sensitivity Analysis .....                                       | 282 |
| 6.4 Neural Network Variability.....                                  | 307 |
| 6.5 Response Surface Style Neural Network Reliability Analysis ..... | 310 |
| 6.6 Summary and Conclusions.....                                     | 323 |
| Chapter 7. Conclusions and Further Work.....                         | 325 |
| 7.1 Conclusions.....   | 329 |
| 7.2 Recommendations for Further Work .....                           | 335 |
| Bibliography .....   | 337 |
| Appendix A. Grasshoppper Mesh Generation.....                        | 345 |
| Appendix B. Full Biaxial History Network Results .....               | 357 |
| Appendix C. Neural Network Training Tool.....                        | 401 |

## Notation

### Finite Element Formulation

|                     |   |
|---------------------|---|
| $\{\sigma\}$        | Vector of element body stresses   |
| $\sigma_x$          | Stress along the local $X$ -axis (fabric warp direction)  |
| $\sigma_y$          | Stress along the local $Y$ -axis (fabric fill direction)  |
| $\tau_{xy}$         | Shear stress in the local co-ordinate system  |
| $\{\sigma_0\}$      | Vector of element initial stresses (pre-stress)   |
| $\{\varepsilon\}$   | Vector of element body strains  |
| $\varepsilon_x$     | Strain along the local $X$ -axis (fabric warp direction)  |
| $\varepsilon_y$     | Strain along the local $Y$ -axis (fabric fill direction)  |
| $\gamma_{xy}$       | Shear strain in the local co-ordinate system  |
| $\{\varepsilon_0\}$ | Vector of element initial strains   |
| $\{x\}_i$           | Element geometry array containing nodal co-ordinates $(x_i, y_i, z_i)$ in the global coordinate system                        |
| $\{C_o\}_i$         | Initial nodal coordinates in the global co-ordinate system of the $i^{\text{th}}$ element                                     |
| $\{C_t\}_i$         | Updated nodal coordinates in the global co-ordinate system of the $i^{\text{th}}$ element                                     |
| $x_i, y_i, z_i$     | Nodal coordinates in the global co-ordinate system at the $i^{\text{th}}$ node  |
| $X_i, Y_i, Z_i$     | Nodal coordinates in the element local co-ordinate system at the $i^{\text{th}}$ node   |
| $\{\delta\}_i$      | Displacement array containing nodal displacements $(u_i, v_i, w_i)$ aligned with the global co-ordinate system                |
| $u_i, v_i, w_i$     | Nodal displacements at the $i^{\text{th}}$ node aligned with the $x, y, z$ axis of the global co-ordinate system respectively |
| $U_i, V_i, W_i$     | Nodal displacements at the $i^{\text{th}}$ node aligned with the $X, Y, Z$ axis of the  |

local co-ordinate system respectively

|                      |  |
|----------------------|--|
| $[E]$                | Material stiffness matrix                                |
| $E_w$                | Young's modulus aligned with local fabric warp direction |
| $E_f$                | Young's modulus aligned with local fabric fill direction |
| $\nu_{fw}, \nu_{wf}$ | Poisson's ratio  |
| $G_{wf}$             | Shear modulus  |
| {P}                  | Vector of applied forces                                 |
| {R}                  | Vector of residual forces                                |
| {D}                  | Vector of combined displacements for system              |
| $[K_T]$              | System stiffness matrix                                  |
| $[K_E]$              | System elastic stiffness matrix                          |
| $[K_\sigma]$         | System geometric stiffness matrix                        |

## Element Formulation

|                       |   |
|-----------------------|---|
| $N_i$                 | Element shape functions of the $i^{\text{th}}$ node defined at a point in terms of natural co-ordinates               |
| $\xi_1, \xi_2, \xi_3$ | Natural co-ordinates  |
| $\xi, \eta$           | Independent natural co-ordinates  |
| $D_{N_{\xi,\eta}}$    | 2 by 6 array of element shape functions derivatives with respect to independent natural co-ordinates                  |
| $D_{N_{X,Y}}$         | 2 by 6 array of element shape functions derivatives with respect to local co-ordinates $X$ and $Y$                    |
| $T^m$                 | Transformation matrix between global and local co-ordinate system at a point defined in terms of natural co-ordinates |
| $l_i, m_i, n_i$       | Directional cosines between global and local co-ordinate axis   |
| $\theta$              | Angle between material warp direction and global $x$ -direction   |
| $\vec{\theta}_w$      | Basis vector describing material warp direction in the global $xy$ plane  |
| $\hat{Z}$             | Basis vector describing local $Z$ -direction  |

|                   |  |
|-------------------|--|
| $\hat{Y}$         | Basis vector describing local $Y$ -direction aligned with material fill direction          |
| $\hat{X}$         | Basis vector describing local $X$ -direction aligned with material warp direction          |
| $f_x, f_y, f_z$   | Nodal forces aligned with the $x, y, z$ axis of the global co-ordinate system respectively |
| $\Phi$            | General scalar quantity  |
| $J, \Gamma$       | Jacobian matrix and inverse Jacobian matrix  |
| $[\varepsilon_0]$ | Array of first order Green-Lagrange strain terms   |
| $[\varepsilon_L]$ | Array of second order Green-Lagrange strain terms  |
| $B$               | B-matrix of the finite element formulation   |
| $B_0$             | Linear B-matrix related to first order strain terms  |
| $B_L$             | Non-linear B-matrix related to second order strain terms                                   |
| $G$               | G-matrix of the finite element formulation   |
| $\{\Delta\}$      | Array of local displacement derivatives with respect to the local co-ordinate system       |
| $W_i$             | Gauss weight   |

### Cable Formulation

|                     |  |
|---------------------|--|
| $L_0$               | Initial cable length                     |
| $L$                 | Displaced cable length                   |
| $[\hat{c}]$         | Unit vector of cable directional cosines |
| $\{c_x, c_y, c_z\}$ | Cable directional cosines                |
| $F_{ca0}$           | Cable pre-stress                         |
| $F_{ca}$            | Cable force                              |
| $\{f_{ca}\}_i$      | Cable element residual force vector      |
| $T_k^{ca}$          | Cable transformation matrix              |
| $k_E^c$             | Cable local element stiffness            |
| $K_T^{ca}$          | Cable system stiffness matrix            |
| $K_E^{ca}$          | Cable elastic stiffness matrix           |
| $K_\sigma^{ca}$     | Cable geometric stiffness matrix         |

## Geodesic Control String Formulation

|                        |  |
|------------------------|--|
| $\hat{cs}$             | Unit vector of geodesic control string directional cosines |
| $\{cs_x, cs_y, cs_z\}$ | Geodesic control string directional cosines                |
| $\{f_{cs}\}_k$         | Geodesic control string residual force vector              |

## Wrinkling Procedure

|                           |  |
|---------------------------|--|
| $\sigma_{max,min}^p$      | Principle stresses   |
| $\varepsilon_{max,min}^p$ | Principle strains  |
| $\theta^p$                | Angle between maximum stress direction and local fabric warp direction |
| $[E_{mod}]$               | Modified material stiffness matrix                                     |
| $[E_{rot}]$               | Transformed material stiffness matrix                                  |
| $P$                       | Wrinkling penalisation factor  |
| $\sigma_{min}^{per}$      | Minimum permissible stress   |

## Neural Network Variables

|               |   |
|---------------|---|
| $I_i$         | Input to the $i^{th}$ input neuron  |
| $I_i^{max}$   | Maximum training input to the $i^{th}$ input neuron                         |
| $I_i^{min}$   | Minimum training input to the $i^{th}$ input neuron                         |
| $I_i^{NN}$    | Scaled output of the $i^{th}$ input neuron                                  |
| $NI$          | Number of neurons in the input layer  |
| $w_{ji}^{IH}$ | Connection weight between $i^{th}$ input neuron and $j^{th}$ hidden neuron  |
| $b_j^H$       | Bias to the $j^{th}$ hidden neuron  |
| $H_j$         | Output of the $j^{th}$ hidden neuron  |
| $NH$          | Number of neurons in the hidden layer                                       |
| $w_{kj}^{HO}$ | Connection weight between $j^{th}$ hidden neuron and $k^{th}$ output neuron |
| $b_k^O$       | Bias to the $k^{th}$ output neuron  |
| $O_k^{NN}$    | Scaled output of the $k^{th}$ output neuron                                 |
| $NO$          | Number of neurons in the output layer                                       |
| $O_k$         | Output of the $k^{th}$ output neuron  |
| $O_k^{max}$   | Maximum training output of the $k^{th}$ output neuron                       |
| $O_k^{min}$   | Minimum training output of the $k^{th}$ output neuron                       |
| $T_{k,c}$     | Target output of the $k^{th}$ output neuron                                 |
| $x_k^O$       | Input to the $k^{th}$ output neuron   |
| $x_j^H$       | Input to the $j^{th}$ hidden neuron   |

|                |   |
|----------------|---|
| $\xi_n$        | Historical stress stain internal variable       |
| $\Delta\eta_n$ | Historical stress strain step internal variable |
| $E^{implied}$  | Implied elastic stiffness matrix                |
| $SV_i$         | Historical input value                          |
| $IV_i$         | Internal historical input value                 |

### Residual Strain Removal

|                              |   |
|------------------------------|---|
| $\varepsilon_{l_{TP}}$       | Strain a load profile turning point           |
| $\varepsilon_{l_{TP}}^{Res}$ | Residual strain at load profile turning point |
| $\varepsilon_i^{RR}$         | Strain with residual strain removed           |

### Reliability Analysis

|                     |  |
|---------------------|--|
| $G_i(X_{si})$       | A limit state function                   |
| $X_{si}$            | Statistical basic variable               |
| $\sigma_{per}^w$    | Permissible fabric warp direction stress |
| $\sigma_{per}^f$    | Permissible fabric fill direction stress |
| $\delta_{per}$      | Permissible nodal displacement           |
| $t_{load}$          | Imposed load coefficient                 |
| $x_{si}^*$          | Design point value                       |
| $\mu_{X_{si}}^N$    | Mean of design point value               |
| $\sigma_{X_{si}}^N$ | Standard deviation of design point value |
| $x'_{si}^*$         | Design point in standard normal space    |
| $\alpha_i$          | Directional cosine                       |
| $\beta$             | Safety index                             |



## **Chapter 1.**

### **Introduction**



## **Contents**

|                                       |    |
|---------------------------------------|----|
| Chapter 1. Introduction.....          | 1  |
| 1.1 Background.....                   | 5  |
| 1.2 Aim and Objectives.....           | 10 |
| 1.3 Scope .....                       | 11 |
| 1.4 Thesis Structure.....             | 11 |
| 1.5 Conferences and publications..... | 13 |

## **Figures**

|  |   |
|--|---|
| Figure 1-1: ASU SkySong innovation Centre, Arizona USA [2] .....   | 5 |
| Figure 1-2: Ashford Designer Outlet, Kent, UK © Buro Happold ..... | 6 |
| Figure 1-3: Venezuelan Pavilion, Hanover, Germany.....             | 7 |
| Figure 1-4: Medinah Umbrellas Prophets Holy Mosque .....           | 7 |
| Figure 1-5: Spinoref.....  | 8 |
| Figure 1-6: Eastleigh Tennis Centre, Southampton, UK .....         | 8 |
| Figure 1-7: Microscopic image of PVC coated polyester fabric.....  | 9 |



## Chapter 1. Introduction

### 1.1 Background

Architectural fabric structures typically comprise of a doubly curved pre-stressed membrane supported by any combination of cables, ridge beams, hoops, edge clamps and masts as illustrated by Figure 1-1. A fabric membrane may also be supported pneumatically using positive air pressure either within the entire building envelope or as individual cladding units Figure 1-5 and Figure 1-6. The fabric membrane may form the entire building envelope or may be combined with other structural materials such as glass, wood or masonry to provide a full or partial enclosure [1].



**Figure 1-1: ASU SkySong innovation Centre, Arizona USA [2]**

Architectural fabrics have negligible bending and compression stiffness. Hence, fabric structures are designed with sufficient curvature to enable environmental loads to be resisted as tensile forces in the plane of the fabric. This differs greatly with conventional roofs in which loads are typically resisted by arch action or by stiffness in bending. The shape of the fabric canopy is vital to its ability to resist all applied loads in tension.



**Figure 1-2: Ashford Designer Outlet, Kent, UK © Buro Happold**

The most common use of tensile membrane structures is to create striking aesthetically pleasing long span shelters or enclosures, as may be required for sports stadia, shopping complexes and airports. This is mainly due to the weight of the material which is often treated as negligible for the purpose of design. In contrast self-weight is often the limiting factor in the design of more traditional concrete or steel long span structures. However, architectural fabric structures benefit from numerous additional advantages relevant for both large scale and smaller scale projects. These include high light transmission, reduced environmental impact and reduced requirement for 'heavy' support structure.

Architectural fabrics have also been successfully used to create temporary and deployable canopies, Figure 1-4 .



**Figure 1-3: Venezuelan Pavilion, Hanover, Germany**



**Figure 1-4: Medinah Umbrellas Prophets Holy Mosque**



**Figure 1-5: Temporary pneumatic structure**

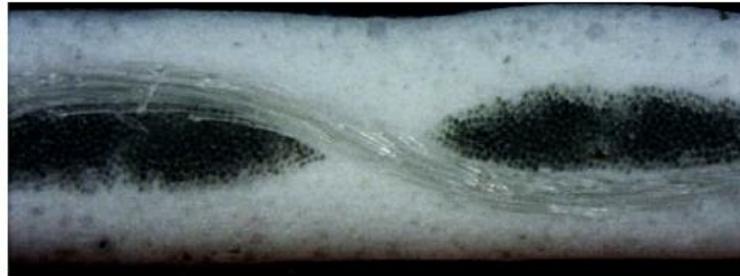


**Figure 1-6: Eastleigh Tennis Centre, Southampton, UK**

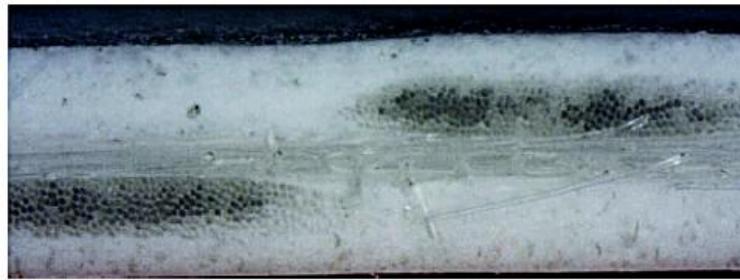
One of the main barriers to the use of architectural fabric in Europe, and worldwide, is a general lack of standardised design guidance which leads to a low level of readily available information for designers, fabricators and clients. A lack of knowledge prohibits the consideration of architectural fabric structures at the initial design stage. Lack of

knowledge combined with high levels of uncertainty with regard to fabric performance also leads to the necessity for specialist testing and design which often proves to be prohibitively expensive for smaller scale projects. There have also been a number of catastrophic failures of fabric architecture which have also reduced confidence in the sector [3, 4]. This leads to the conclusion that a more robust standardised design methodology is required, along with a better understanding of architectural fabric behaviour, to enable the wider dissemination of architectural fabric structures.

Architectural fabrics, sometimes referred to as structural or technical fabrics, typically comprise orthogonal woven yarns covered by one or more layers of coating. The most commonly used yarns are either polyester, produced through a melt spinning process, or glass yarns, which are drawn from a melt. Polyester yarns are generally coated with polyvinylchloride (PVC) and glass yarns by polytetrafluoroethylene (PTFE). The coating provides additional stability to the weave geometry, protects the yarns from the elements, and enables the creation of a weather tight envelope. Alternative yarn and coating combinations include Silicone coated glass fabrics and PTFE coated PTFE fabrics. Additional outer coatings may also be added to enable better joining, printing and self-cleaning properties [5].



(a)



(b)

**Figure 1-7: Microscopic image of PVC coated polyester fabric  
(a) Warp direction cross section (b) Fill direction cross section [6]**

In the simulation of tensile fabric structures it is common practice to represent membrane material as a homogeneous continuum described by a plane stress strain-stress

relationship, where the material characteristics are represented by Young's modulus and Poisson's ratio. Fitting a plane stress model to biaxial test data for typical architectural fabrics leads to inconsistencies between the physical and theoretical descriptions, with values of Poisson's ratio in excess of the compressibility limit of 0.5, and for some fabrics approaching 2.0. An alternative to the plane stress framework is therefore required to more accurately represent fabric behaviour.

Neural networks offer an exciting solution for the constitutive modelling of architectural fabrics as they are capable of capturing highly non-linear response. Furthermore, neural network material models effectively 'learn' the material response directly from experimental data and therefore no prior knowledge of the response is required.

## 1.2 Aim and Objectives

The aim of this research is to provide a means of linking material testing and structural analysis within a computational mechanics framework that accurately describes both material and structural behaviour. The key hypothesis is that it is possible to represent experimentally captured architectural fabric stress-strain response, including historical loading effects, using a neural network material model within large strain finite element analysis.

Neural network material models are selected based on their ability to capture highly non-linear functional mappings. This enables the accurate representation of the complex fabric strain-stress response to biaxial loading. In addition the proven ability of the neural network material models to capture the effects of historical loading is a distinct advantage over other previously published material models employed in the simulation of architectural fabric structures. In order to develop an accurate and implementable material model, fabric response must be captured experimentally in a way that represents fabric response in a real world situation.

In order to demonstrate the neural network material model it is vital that it be implemented within analysis code suitable for the simulation of architectural fabric structures. To that end a bespoke structural analysis tool will be developed.

It is anticipated that this initial research will lead to the development of tools enabling accurate whole life simulation of fabric structures. This could enable the simulation of membrane installation leading to improvements in installation techniques and allow the accurate prediction of the medium and long term performance of fabric structures.

The specific objectives are to:

- Adapt pre-existing Fortran large strain finite element code to provide plane-stress analysis comparisons and enable the implementation of the neural network material model.
- Capture the fabric material response through mechanical testing of architectural fabric.
- Develop a neural network approach to represent the relationship between biaxial fabric strain and fabric stress with residual strain and load history effects included.
- Implement and demonstrate the neural network material model.
- Demonstrate epistemic uncertainty introduced to neural network material models by random weight initiation and training data division.
- Perform reliability analysis using the implemented material model.

### 1.3 Scope

The first section of this research focuses on the development of the finite-element based analysis tool in preparation for the implementation of the neural network material model.

The second section is focused on the development of architectural fabric neural network material models. Fabric response is modelled with and without the inclusion of residual strain and load history effects.

The third and final section demonstrates the implementation of the neural network material models developed in the second section within the analysis tool of the first. Uncertainty is investigated and a reliability analysis protocol is developed.

### 1.4 Thesis Structure

#### Chapter 2. **Literature review.**

Gives an overview of current analysis methods applied to architectural fabric structure. The current practice for fabric testing and architectural material modelling, with specific focus on plane-stress material models used in industry, is explored. A need for an accurate non-linear material model is demonstrated. The advantages and disadvantages of neural network material models are summarised and the previous applications of neural networks to material modelling are reviewed.

#### Chapter 3. **Finite Element Formulation**

The formulation of a six node Linear Strain Triangle (LST) including large strain terms is presented and its implementation within finite-element analysis code is demonstrated.

## Chapter 4. Neural Network Training and Validation

The procedure for generating neural network material models for architectural fabric is presented. Initially fabric response is modelled with residual strain removed. This form of model effectively maps a single surface for the response in each of the material directions (warp and fill). Therefore, it is referred to as a response-surface neural network model. This model is comparable with existing plane-stress architectural fabric material models and utilises similar experimental data in its development. The neural network model is then advanced to include load history requiring the development of specialist testing. This form of model is referred to as a load-history neural network model. Both forms of neural network models are validated using experimental data not used in training.

## Chapter 5. Neural Network Material Model Implementation.

Implementation of the material models presented in Chapter 4 is described. Adaptation of the finite-element code to allow for the new material model is detailed including the development of iterative analysis to allow for inclusion of load history. Both forms of neural network material models are demonstrated using analysis of simple but realistic fabric structures including a conic and a hypar and compared with equivalent plane-stress analysis. Epistemic uncertainty introduced by the neural network is explored and a procedure for reliability analysis developed.

## Chapter 6. Reliability Analysis

The analytical partial derivatives required for probabilistic reliability analyses are derived and validated. A protocol for the determination of statistical descriptions for the variability of neural networks is proposed. The statistical variables are used in an adapted reliability analysis protocol in order to perform a reliability analysis of a realistic hypar structure.

## Chapter 7. Conclusion and Recommendations.

A summary of conclusions presented at the end of chapter is provided along with recommendations for further work.

## Bibliography

All references use a numeric format, A full list of reference arranged in the order they appear may be found at the back of this document.

## Appendix

All supporting material

## 1.5 Conferences and Publications

**IStructE Young Researchers Conference 2012** - Poster Presentation

**TensiNet Symposium 2013**, [RE]THINKING Lightweight structures, 2013, Istanbul - Presentation and paper: *A Neural Network Material Model for the Analysis of Fabric Structures*

**Structural Membranes 2013**, The sixth conference on Textile Composites and Inflatable Structures, Munich - Presentation: *Implementation of a Neural Network Material Model for the Analysis of Fabric Structures*

### Contributed to,

Gosling, P. D., et al. (2013). "Analysis and design of membrane structures: Results of a round robin exercise." *Engineering Structures* 48: 313-328.



## **Chapter 2.**

### **Literature Review**



## Contents

|   |    |
|---|----|
| Chapter 2. Literature Review.....                                   | 15 |
| 2.1    Simulation of Architectural Fabric Structures .....          | 19 |
| 2.2    Determination and Representation of Material Behaviour ..... | 34 |
| 2.3    Neural Network Material Modelling .....                      | 56 |
| 2.4    Summary and Conclusions.....                                 | 67 |

## Figures

|   |    |
|---|----|
| Figure 2-1: Updated Reference Strategy applied to a hypar [21].....   | 23 |
| Figure 2-2: Total, local, and basic isoparametric elements [32].....  | 26 |
| Figure 2-3: Reference system for membrane element [35].....   | 27 |
| Figure 2-4: Visualisation of FORM and SORM limit state approximations<br>in standard normal space (reproduced from [11]) .....          | 29 |
| Figure 2-5: Integration of finite element analysis with FORM reliability analysis<br>(reproduced from [39]) .....                       | 31 |
| Figure 2-6: Test methods used for the investigation of fabric shear response [65]. .....  | 37 |
| Figure 2-7: Specialised biaxial load history for the determination of<br>elastic constants [57] .....                                   | 42 |
| Figure 2-8: Warp and Fill response surfaces [67] .....  | 43 |
| Figure 2-9: FEM predictions compared with experimentally measured<br>data points [67] .....   | 45 |
| Figure 2-10: RMS and maximum absolute difference between experimental<br>data and model predictions [66] .....                          | 47 |
| Figure 2-11: Stress-stress-strain representation of ‘non-linear’ model<br>predictions (surface) and experimental data (dots) [66] ..... | 48 |
| Figure 2-12: Experimental results (markers) compared to predicted<br>results (lines) for $E_x$ , $G_{xy}$ and $v_{xy}/E_x$ [52].....    | 50 |
| Figure 2-13: Experimental data fit with two different rational Bezier curves [70] .....   | 52 |
| Figure 2-14: Elasto-plastic model representation of biaxial response<br>data compared with experimental data [6].....                   | 53 |
| Figure 2-15: A biological neuron and an artificial neuron the perceptron [98].....  | 57 |
| Figure 2-16: Single hidden layer feed-forward neural network.....   | 58 |

|  |    |
|--|----|
| Figure 2-17: Trained hysteretic concrete network material model output alongside experimental training data and analytical model prediction, [114] ..... | 62 |
| Figure 2-18: Trained hysteretic concrete network material model output alongside experimental data not used in training, [114].....                      | 62 |
| Figure 2-19: Neural networks trained to approximate $\sin(x)$ within two different ranges of $x$ [88].....   | 63 |
| Figure 2-20: Training and over-training of a neural network with data containing scatter [88] .....  | 64 |

## **Tables**

|  |    |
|--|----|
| Table 1: Summary of common ANN architectures with associated learning algorithms [97]..... | 60 |
|--|----|

## Chapter 2. Literature Review

In the first section of this chapter, numerical methods developed for the design of tensile membrane structures will be introduced and compared. The reliability analysis of fabric structures will also be addressed. This is done with a view to select an appropriate simulation procedure to develop and use in the implementation of the material model to be investigated in this thesis.

In the second section, the material models applied to architectural fabrics will be reviewed along with experimental methods used to capture the fabric strain-stress response. Particular attention is paid to the plane stress framework currently used as the industry standard in the design of fabric structures. Various methodologies applied in literature to address the shortcoming of the plane stress framework are also reviewed. The need for an improved material model will be established.

In the third and final section, artificial neural network material models are introduced. Neural network development and training is described along with the advantages and disadvantages associated with neural networks. The suitability of neural networks for the modelling of architectural fabrics will be established. A methodology for the application of neural network material modelling to architectural fabrics will be developed based on these findings.

### 2.1 Simulation of Architectural Fabric Structures

This section addresses the finite elements and solution algorithms typically used for the numerical simulation of tensile membrane structures along with their use within the design process. In general, the design process prior to construction may be defined as follows [7-9].

#### 1. Conceptual development

The site is assessed and a desired physical configuration is defined along with materials for both the membrane and support structure.

#### 2. Form finding

An equilibrated form is established using physical modelling or, more commonly in current practice, finite element analysis using a zero stiffness element. At this stage

the structural shape is optimised via the manipulation of boundary conditions and pre-stress.

### 3. Static load analysis

Appropriate surface pressures representing typical load cases, including snow loading and wind uplift, are applied to the form found structural model. In finite element analysis, material properties gathered through physical testing of chosen architectural fabrics should be used. However, due to the complexities and expense of material testing in many cases assumed stiffness values are used, at least for initial design. Depending on the results of static analysis it may be necessary to return to the form finding step in order to achieve acceptable stress and strain values as dictated by the strength and serviceability limits of the fabric material and its support structure. This process is likely to require numerous iterations. Current practice uses a permissible stress approach. However, more detailed reliability approaches may be adopted in order to validate the final configuration.

### 4. Patterning

The final form found configuration is used in conjunction with experimentally derived compensation values to develop cutting patterns for fabric panels that may be joined together on or off site prior to installation. Compensation values are applied in order to achieve the final structural form at the required pre-stress. The material direction within each panel and the position of seams should be taken into account within the simulation.

### 5. Construction sequencing

A sequence and method of assembly is established, and where possible simulated. Key to this sequence is the fixing of support and membrane components in place and application of pre-stress to the required level. This information will impact on the design of connection and support details.

Architectural fabrics possess negligible compressive stiffness. Fundamental to the ability of a fabric structure to resist load purely in tension is the formation of either an anticlastic or synclastic surface geometry. Anticlastic geometry comprises opposing curves that generally follow the warp and fill directions of the architectural fabric. This form of surface is stabilised by uniform pre-stress applied in the plane of the membrane by boundary restraints such as cables. Synclastic geometry comprises spherical type surfaces that are stabilised by pneumatic or hydraulic pressure normal to the membrane surface. Such doubly curved surfaces are based on minimal surfaces. In mathematical terms, a minimal surface has zero mean curvature at any point. This leads to a locally minimised surface area formed between boundary conditions [1].

A physical model of a minimal surface may be formed by a soap film that has uniform surface tension. Frei Otto and his team at the Institute for Lightweight Structures are generally referred to as the pioneers of Soap film experiments.

*"The study of soap bubbles greatly helps the understanding of pneumatic structures. In each soap bubble or agglomeration of soap bubbles the membrane stresses are equal at each point and in every direction, if we neglect infinitesimal stress differences caused by the weight of the bubble skin"* [10, pg.11]

Certain boundary conditions prohibit the formation of a minimal surface. In this situation a pseudo minimal surface may be achieved using architectural fabric by allowing localised stress variations. For example, the formation of a minimal surface between upper and lower rings to form a conic type structure is limited by necking where the mid plane radius reduces as the distance between the rings increases. This effect has been shown to be reduced by the application of a higher level of stress in the meridinal direction than in the circumferential direction [11].

Unlike in the case of traditional structural forms where geometry is directly prescribed, the geometry of a tensile membrane structure is governed by the prescribed boundary conditions and desired stress state. An initial form-finding analysis is required followed by the generation of a suitable cutting pattern to reproduce that form. Prior to the development of non-linear computational methods, physical models were used in the development of geometry. Given a sufficiently accurate physical model, patterns or component geometry may be measured directly from the model, and used in hand calculations of forces [12]. These calculations offer only approximations as they cannot account for the effect of displacement on force. With the invention and subsequent development of computer-aided design in the late sixties, it became possible to generate minimal surfaces using computer methods and perform detailed stress analysis. Since then the power of commercially available computers has increased exponentially allowing the generation of increasingly complex numerical models.

The large displacement finite element formulation required for the simulation of fabric structures leads to state equations for equilibrium that are non-linear and cannot be solved in a single step. The principal approaches to solving the state equations of the finite element method for fabric structures may be divided in to two basic groups – matrix and vector methods. Vector methods are also referred to as dynamic methods. The key difference between matrix methods and vector methods is that the condition of

equilibrium and compatibility for the entire structure remains coupled throughout the analysis for the former and are decoupled until equilibrium for the latter.

Argyris undertook some of the earliest work on matrix methods. He and his colleagues applied the Newton-Raphson method to the form-finding of cable-nets [13]. This method utilised material properties leading to additional computational cost and non-minimal surface solutions. Since this early work, attempts have been made to improve the suitability of this method when applied to form-finding, typically by the removal of material stiffness from the virtual work equation solved for equilibrium.

The Force Density method was developed by Schek at Stuttgart University for the form finding of general networks [14]. This method makes use of the ratio between element forces and length, or ‘force densities’, to drive convergence to equilibrium. It simplifies the geometrically non-linear problem to a set of linear equations. Linkwitz described the development of the ‘Stuttgart-direct-approach’ which combines least squares, graph-theory and numerical mathematics with force-densities [15]. This approach has been demonstrated using a simple form-finding exercise on a quadrilateral network restrained in four corners. It was shown that it is the proportional relationship between mesh and boundary cable force densities that governs the final form of the structure rather than their absolute values. One of the primary weaknesses of the method is that the coefficients used to drive convergence are a function of element length and so the shape of the initial mesh has an impact on the final solution. The force-densities are also physically meaningless and the values prescribed are typically based only on the experience of the engineer. The Force Density Method has been extended to incorporate membrane elements [16, 17] to allow for the representation of shear resistance.

In order to perform static analysis, the finite difference terms are eliminated through the application of material properties to the finite difference equations describing the final form found configuration. The equilibrium equations are reported to become identical to the equilibrium equations of the finite element method [17]. This procedure is used by the Easy Lightweight Structure Design software package [18].

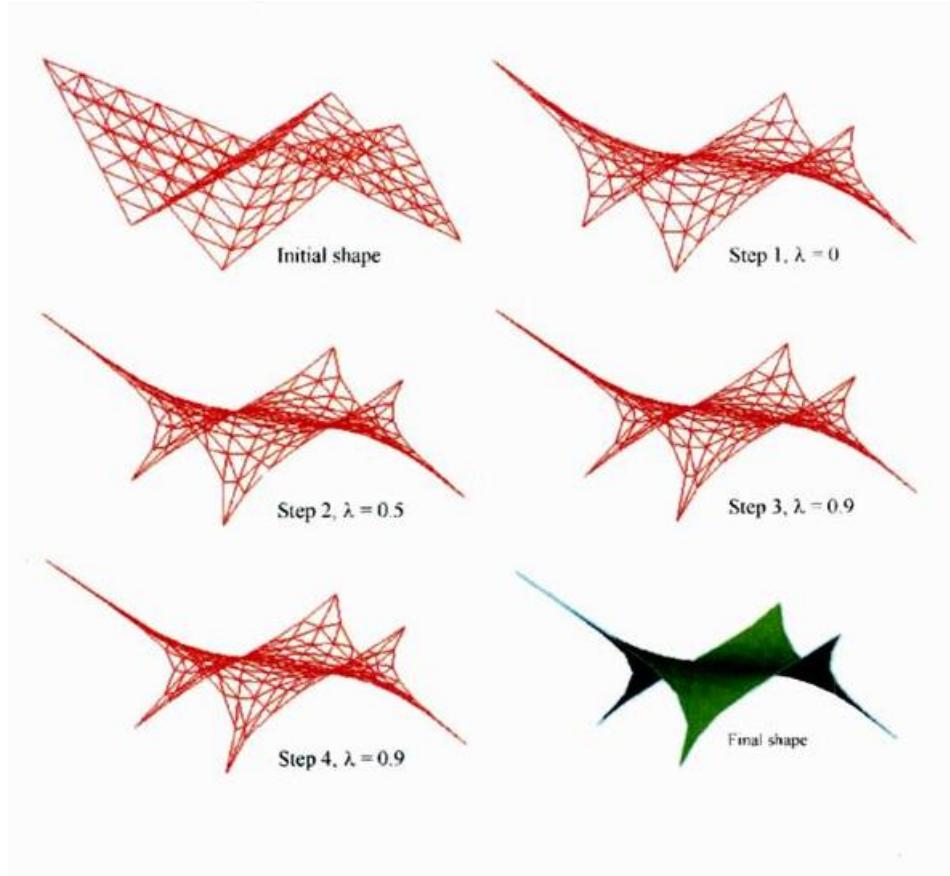
When using a matrix method, such as the Newton-Raphson method, to solve the state equations of equilibrium with a zero stiffness matrix, the lack of in plane stiffness leads to floating nodes in the mesh surface. This leads to a singular structural stiffness matrix. The Updated Reference Strategy (URS) [19, 20] solved this problem by adding a term to the virtual work equation that artificially introduces in plane stiffness. The equation for virtual work for a minimal surface with an isotropic stress field, given by,

$$\delta w_\lambda = st \int_A (\lambda \det FF^{-T}) : \delta F dA = 0 \quad 0 < \lambda < 1 \quad (2-1)$$

becomes,

$$\delta w_\lambda = st \int_A (\lambda \det FF^{-T} + (1 - \lambda)F) : \delta F dA = 0 \quad 0 < \lambda < 1 \quad (2-2)$$

The solution found using equation (2-2) does not represent the true minimal surface while  $\lambda$  is less than 1. However URS employs an iterative process where the geometry solution to equation (2-2) is found with increasing values of  $\lambda$ . The geometry from the previous step is used as the reference or initial geometry until  $\lambda$  is equal to 1 and the final minimal surface has been found, Figure 2-1.



**Figure 2-1: Updated Reference Strategy applied to a hypar [21].**

Bonet and Mahaney also proposed the use of an ‘artificial constitutive model’ to reduce the problems of singularity caused by zero in-plane stiffness in a flat mesh [22]. This technique proposed the simultaneous minimisation of area and shear distortion to reduce mesh distortion. The authors demonstrated a positive effect on mesh regularity for the

classical form finding problem of Scherk's surface using a quadrilateral mesh. However, it was noted that limited difference was observed when using a triangular mesh.

Early work on vector methods was undertaken by Barnes and Wakefield. Following the work of Day (1965), a Dynamic Relaxation algorithm was applied to the analysis various tension structures using a CST element that transformed the continuum membrane problem into a discrete cable analysis [23, 24]. The Dynamic Relaxation algorithm is based on Newton's second law of motion. The force, or out-of-balance (residual) force is given by the sum of applied load and structural element forces at a specific node. The mass is an assumed lumped mass at that node. As the objective is to find the equilibrium condition and not the true dynamic behaviour of the structure, a fictitious mass is adopted, and is often defined to optimise convergence. Barnes describes the development and use of Dynamic Relaxation for form-finding and analysis of tension structures in [25]. Dynamic relaxation is employed in both GSA and Tensyl analysis packages.

Recently particle methods of form finding have been explored as an alternative to the classical finite element method [26, 27]. In [27] the a particle method is described for use in the form finding of gravity loaded shell structures using dynamic relaxation. This method could equally be applied to form finding of a tension structure. It is noted that the method is less computationally efficient than the finite element method but allows form finding without the definition of a grid.

A comparative review of available numerical methods for structural form finding of general networks is presented in [28]. A useful framework is established in order to allow clear comparison of the methods. The methods reviewed in [28] are divided in to three general categories. Stiffness matrix methods, that use both the elastic stiffness geometric stiffness matrices. Material independent geometric stiffness methods which employ matrix methods to solve the limit state equations. The final category, dynamic equilibrium methods, are also material independent that find static equilibrium by finding the equivalent steady-state solution of dynamic equilibrium.

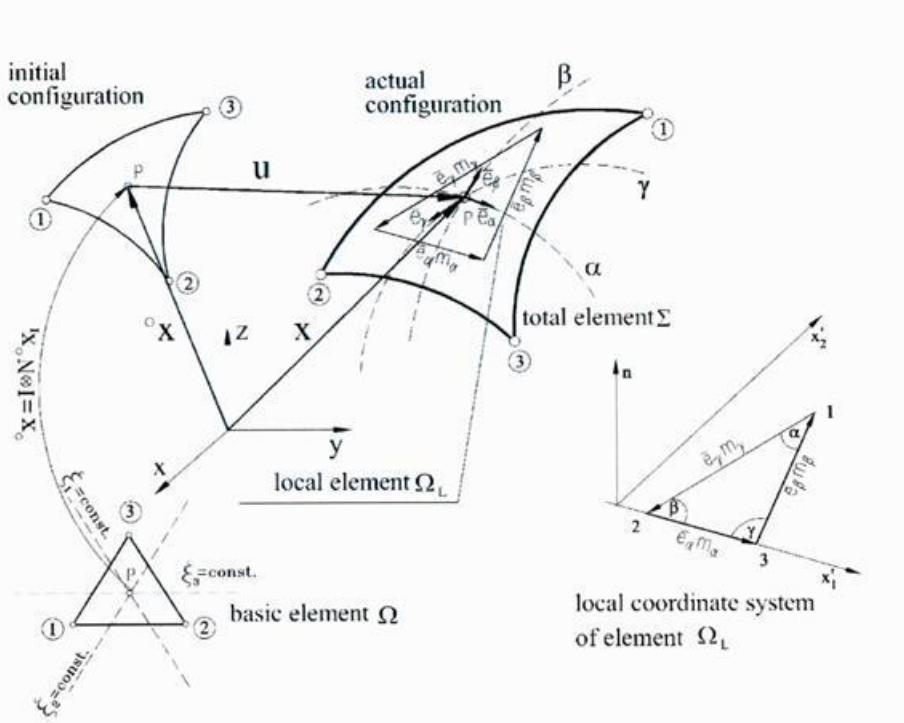
Geometric stiffness methods in general, and the multi-step form finding method with force adjustment in particular, were found to be more computationally efficient than dynamic equilibrium and stiffness matrix methods for form finding a minimal surface net. When form finding a network with non-uniform forces a single method or category of methods is not identified as being superior. It is also stated that application of the cable-net analogy is not easily applied to the simulation of a membrane due to the disregard of shear effects. This review offers some very useful insights into the similarities and differences between

the various numerical form finding methods available but cannot be used in isolation to inform a decision on the best method to be used for form finding with membrane elements.

Wakefield identified key attributes required for a fabric analysis package [29]. These include: control of surface stress during form finding, the ability to control fabric weave orientation during both form finding and analysis, the availability of specialist membrane elements, and the integration of patterning into the form-finding process. The elements required for the simulation of fabric structures include membrane elements, geodesic string elements, and slip cables, as well as more typical beam elements used to simulate supporting structures.

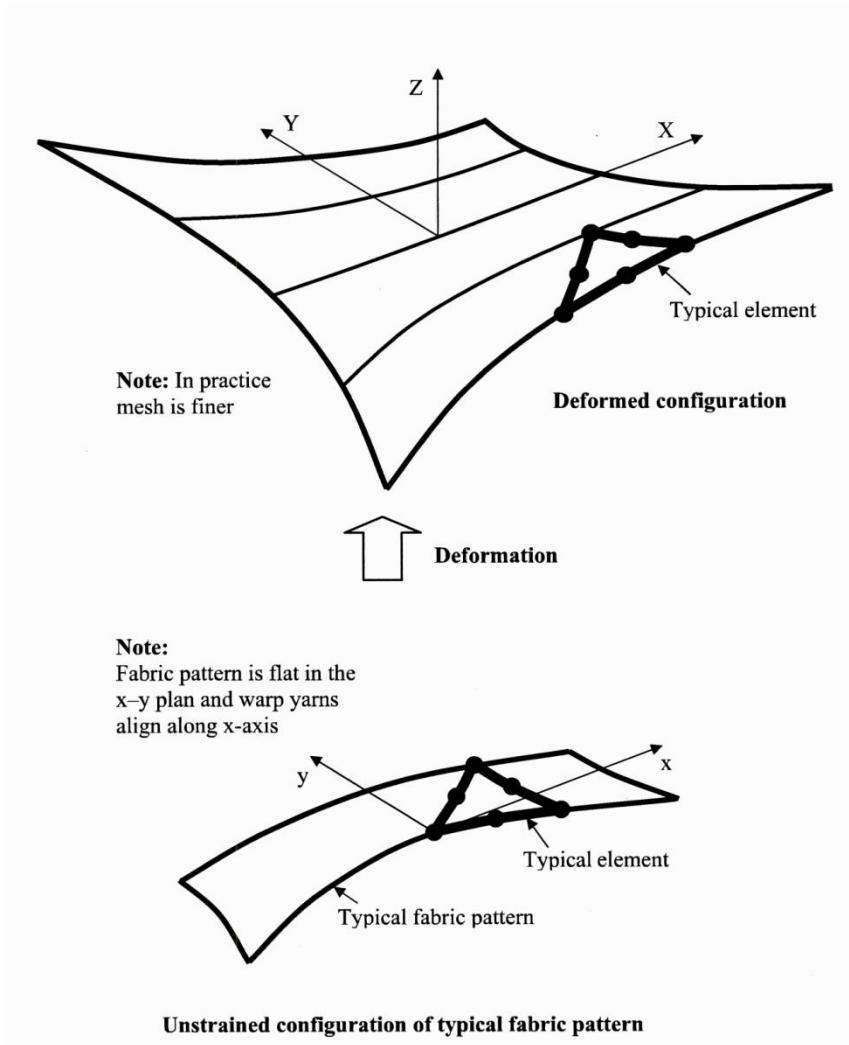
Early work in the development of a specialist membrane element focused on the three-node constant strain triangle (CST) formulation. A detailed review of this work may be found in [30] and [31]. An improved CST element has been proposed with the inclusion of higher order terms to overcome deficiencies in the basic formulation [31]. The CST element continues to be used and is often used to benchmark higher-order elements including 6 & 8-node isoperimetric finite element formulations based on the linear strains. These elements attempt to overcome some of the shortcomings of the CST elements through the ability to represent a curved geometry and generate smoother stress distributions and transitions, particularly in areas exhibiting steep stress gradients. It was found in [31] that an isoperimetric element formulation generates a smoother stress distribution than the large strain CST formulation with only a small increase in computational expense.

Isoparametric formulations utilise natural co-ordinates to maintain inter-element compatibility by mapping the curved element onto standardised flat configuration defined by area or natural co-ordinates. In [32] and [31] the element geometry is transformed from the local Cartesian co-ordinate system on the tangent element plane to an oblique curvilinear material co-ordinate system, Figure 2-2. Alternatively in [30, 33] an 8 node quadrilateral element is developed with the local material co-ordinate system defined by the tangents to the element surface at each element Gauss point aligned with the fabric warp direction using a user defined angle. A detailed description of the development of isoparametric elements is presented in [34] and has been used in the development of a number of specialist membrane element formulations including [30] and more recently [35].



**Figure 2-2: Total, local, and basic isoparametric elements [32]**

A fully integrated design and analysis finite element methodology is presented in [35]. This includes formulations for both slip cable and membrane finite elements. The key novel feature described in this methodology is the fact that the membrane element accounts for development of the final 3-dimensional shape from initially flat panels, Figure 2-3. The form finding process via minimal surfaces is rejected. The initial shapes of the fabric panels are defined by hand and are then geometrically refined in an iterative process based on the results of multiple analysis runs. This process is continued until a permissible stress state is found. Three examples are presented - the inflation of a square membrane, the analysis of a flat membrane bounded by cables, and a simple conic with edges cables. This method severely limits the available geometries and the trial and error process is likely to be time consuming. An alternative approach would be to perform patterning analysis in order to determine the magnitude of locked in stress in conjunction with the development of flat patterned panels.



**Figure 2-3: Reference system for membrane element [35]**

This paper highlights the importance of patterning consideration in the early stages of model development. It also explores the likelihood of non-uniform stress fields resulting from the development of flat geometry into doubly curved geometry. However, the material model used to describe the material response is absolutely critical for the generation of useful, meaningful, results from this type of simulation. In [35] the authors used a constitutive model based on a unit cell of fabric in place of the more typical plane stress constitutive model.

A comprehensive review of current design practice involving numerous contributors from both industry and academia was undertaken in [36]. The review involved a comparative

study of the form-finding and load analysis of a set of simple but realistic fabrics structures by each of the contributors. The results generated highlight the vast variability between the reactions, stresses, and displacement produced by the various simulation procedures employed within the industry.

In the simulation and design of complex systems such as a fabric structure not all, if any, variables in the structural model can be identified with absolute certainty. These uncertainties may be categorised into 2 groups – epistemic uncertainty and aleatoric uncertainty. Aleatoric uncertainty is a statistical uncertainty. This group contains the inherent variability associated with a physical or simulated process; for example, uncertainty in material behaviour introduced by a manufacturing process. This type of uncertainty may be measured in terms of statistics generated through repeated simulations or physical tests. Epistemic or systematic uncertainties are variables or relationships that are identified as unknown in the process but are not accounted for in the model.

Once it is accepted that both aleatoric and epistemic uncertainties exist within structural design it becomes important that this uncertainty is quantified in some way. Uncertainty quantification aims to identify sources of uncertainty and categorise them with a view to reduce all uncertainties to aleatoric uncertainties. Sources of uncertainty in the simulation and design of fabric structures include variability of actions, material properties and geometry, uncertainties in the load and resistance effects due to essential simplification of the system, and lack of knowledge concerning material behaviour and actions in real world conditions [16]. Reliability analysis of a structure aims to account for the identified uncertainties and to ultimately quantify the reliability (or safety) of a structure in terms of a probability of failure or a safety index.

In the Eurocodes the definition for reliability is given as:

*"...the ability of a structure to comply with given requirements under specified conditions during the intended life, for which it was designed. In quantitative sense reliability may be defined as the complement of the probability of failure" [16]*

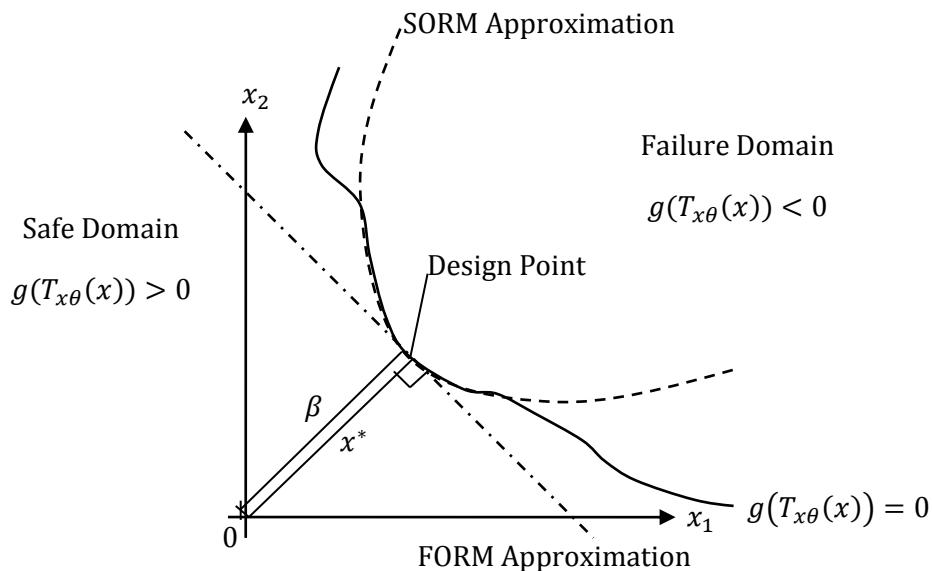
A reliability analysis requires that the structural behaviour be defined in terms of a finite number of statistical variables. In the analysis of fabric structures these variables may include, but are not limited to, loading, mechanical properties and geometry [8]. It is also required that the failure condition is defined as a differentiable function , which is positive in the safe domain and negative in the failure domain. This function is known as the limit

state function and may be categorised as either an ultimate limit state, defining total structural failure or collapse, or a serviceability limit state, defining a point at which the structure is no longer fit for purpose [20].

The failure probability of a structural system may be calculated using either approximate methods or simulation methods. Approximate methods assume a particular form of the limit state function around the “design point” (e.g. linear in the case of FORM and parabolic in the case of SORM) to enable calculation of the reliability index  $\beta$ , defined as the minimum distance from an origin point to the limit state surface in standard normal space. The probability of failure,  $P_F$ , is given by,

$$P_F = \int_{g(\theta) < 0} h(\theta) d\theta \approx \hat{P}_F = \Phi(-|x^*|) = \Phi(-\beta), \quad (2-3)$$

where  $h(\theta)$  represents the model defined in terms of uncertain parameters,  $\theta$ ,  $x^*$  represent design points transformed in standard normal space, and  $\Phi(\cdot)$  is the standard normal cumulative density function [37]. FORM and SORM approximations of a limit state surface are illustrated in Figure 2-4.

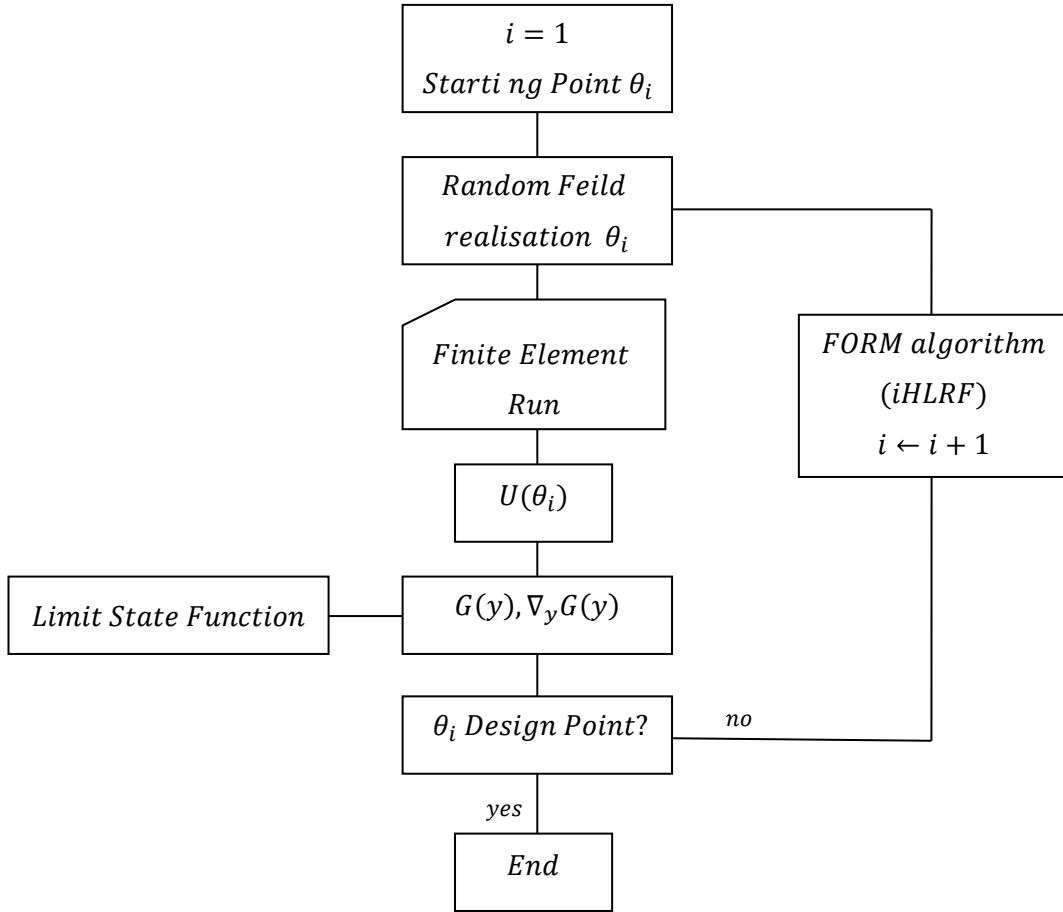


**Figure 2-4: Visualisation of FORM and SORM limit state approximations in standard normal space (reproduced from [11])**

Simulation methods use multiple simulation runs with procedurally generated sets of uncertain variables in order to assess the probability of failure. Monte-Carlo simulation is the most widely used method in this category [11]. Importance Sampling aims to improve on the efficiency of reliability analysis via the Monte Carlo simulation by generating a greater number of samples in the region of the failure domain that accounts for the

majority of the probability of failure. It is demonstrated in [38] that Importance Sampling is ineffective for high-dimensional reliability analysis due to an inability to generate sufficient samples in the important region of the failure domain. The validity of the design point used in FORM and SORM is also shown to be affected by high-dimensional non-linear problems.

In the simplest case of an approximate method where both the structural behaviour and limit state function is defined by a linear equation, the reliability index  $\beta$  may be found in a single step. Where the structural behaviour is described by a more complex non-linear approximation, i.e. defined implicitly from a finite element analysis, the reliability analysis becomes an optimisation problem where  $\beta$  is minimised within the standard normal space iteratively, Figure 2-5. A number of different optimisation algorithms for FORM, namely the fast probability integration method, hypersphere method, successive approach method and directional cosines method, are directly compared to Monte Carlo simulation in [16]. It is found that the directional cosines method shows the quickest convergence and has the added benefit of providing sensitivity information about the significance of the uncertain variables, allowing for simplification of the problem by the removal of stochastic variables with low sensitivities with respect to the limit state function. A commonly used directional cosine optimisation algorithm is the improved Hassofer-Lind, Rackwitz-Fiessler (iHLRF) algorithm. It is required that the uncertainty variables are normally distributed. Where they are not normally distributed, they must be transformed to a normal distribution prior to their transformation to a standard normal variable. This is achieved via the Rackwitz-Fiessler transformation [31].



**Figure 2-5: Integration of finite element analysis with FORM reliability analysis (reproduced from [39])**

FORM and SORM reliability methods rely on the assumption that a single global design point, the point at the minimum distance from the origin in standard normal space, exists on the limit state surface. In practice, there may be potentially many local minima on the surface of the limit state function, any or all of which may be found when using a typical gradient driven optimisation algorithm and a range of starting points. One solution to this problem, proposed in [40], is to force the optimisation algorithm to find multiple points by constructing 'barriers' around previously found solutions by moving the design point away from the origin. This enables an assessment to be made of the likelihood that the true global point has been found.

Finite element analyses, especially for non-linear problems, are computationally expensive. Therefore, the number of times the finite element model is called to calculate the value of the limit state function needs to be limited, if only for practical reasons, [19]. FORM and SORM have the advantage that they typically require only 5-10 evaluations of the limit state function (and therefore finite element analyses) to converge to a value of  $\beta$

with sufficient accuracy, [41]. The combination of the computationally expensive finite element analysis with a generally low probability of failure makes simulation methods, such as Monte Carlo simulation, unworkable because of the requirement for a number of simulations in the order of  $10^5$ .

Limitations of FORM and SORM are addressed in [19] and [41]. Convergence issues are introduced by discontinuous non-linear failure surfaces. One suggested remedy is the use of smooth material models. The generation of trial points located too far into the failure in the early stages of the search may cause the finite element analysis to collapse or convergence to take too many steps. It is proposed that either limiting the step size or restricting the allowable position of the trial points in the initial steps of the search offers a solution to this problem. Collapse or failure of the finite element analysis is likely to occur when one or more statistical variable deviates far from their initial mean value. It is suggested that limiting the step size is acceptable, as this area of the failure region is unlikely to contain the design point. Therefore these regions may be avoided without reducing the accuracy of the estimated failure probability.

FORM implemented with a specialist fabric finite element code is demonstrated in [8]. This paper is a continuation of the work undertaken in [31]. Reliability factors are obtained for two examples, a simple conceptual hypar and the real Doncaster Education City Crèche canopy. The reliability values obtained are compared with Eurocode guidance as well as deterministic limit state design using stress reduction factors. The main motivation for the development of reliability analysis for fabric structures is to enable the assessment of the reliability of a structure without the over simplification introduced by the use of an all-encompassing stress-reduction factor. Reliability analysis provides a formal framework for defining structural safety. As shown the Round Robin paper, [36], there is no consensus for the value of the stress reduction factor, essentially because it is based on serendipity.

The design process applied to fabric structures is highly specialised. Fabric structures resist all applied load in tension. Therefore, a doubly curved surface geometry is required in order to resist out of plane loads. The geometric form of a fabric structure is not directly prescribed, rather a form finding analysis is required in order to determine the form that is dependent on the applied pre-stress and boundary conditions. A minimal surface, such as that formed by a soap film, has been identified as an ideal form for a fabric structure.

Definition of a zero elastic stiffness in finite element analysis leads to floating nodes in the plane of the fabric. This makes the equilibrium solution to the virtual work equation

singular. The finite difference method, which reduces the geometrically non-linear problem to a set of linear equations, uses finite-differences to drive convergence. The use of finite differences is not ideal as the final solution is dependent on the initial configuration. The updated reference strategy introduces an artificial in-plane stiffness term to the virtual work equation. An iterative form finding process is undertaken, during which the in plane stiffness term is gradually reduced to zero. Dynamic relaxation utilises a vector method to solve the limit state equation where the condition of equilibrium and compatibility is decoupled. Geodesic control elements may be employed to control the final mesh configuration. These are useful for the definition and control of fabric panels which may be used for patterning after the form finding process is complete.

Two main element formulations have been applied to the simulation of architectural fabrics, the constant strain triangular element and both triangular and quadrilateral higher order isoparametric elements. Isoparametric elements have the advantage of allowing curved geometry and are more suitable for the capture of steep stress gradients. The isoparametric element formulation may be enhanced to include the local material direction by the inclusion of an angle term describing the material direction in the local coordinate transformation. In this thesis an isoparametric large strain triangular element formulation will be employed with a dynamic relaxation solution algorithm. The use of dynamic relaxation is advantageous when implementing an alternative material model to the 3x3 plane stress elastic stiffness matrix, as the final solution is not determined directly from the structural stiffness matrix. Only the diagonal terms of the global stiffness matrix are used as damping coefficients. The solution is driven by the current out of balance force determined from the current level of stress given by the material model for the current level of strain.

FORM reliability analysis has been applied to fabric structures. The use of a directional-cosine optimisation method, such as the improved Hassofer-Lind, Rackwitz-Fiessler (iHLRF) algorithm, enable the assessment of the significance of the uncertain variables. The reliability analysis is complicated by a non -linear limit state function. Mitigations may be required in order to stabilise the analysis procedure and the presence of multiple design points or a high degree of non-linearity may make the application of FORM analysis infeasible. The analysis procedure should be validated using a simulation reliability procedure such as a Monte Carlo simulation or Importance Sampling to ensure that a single minimum design point can be identified.

## 2.2 Determination and Representation of Material Behaviour

The static analysis of a tensile fabric structure requires a material model along with boundary conditions and initial geometry, determined through form finding. The material model typically takes the form of a constitutive equation that relates values of strain to stress or vice versa. The accuracy of such material models is heavily dependent on assumptions made to simplify the material response. The various models may be divided into three disparate categories - elastic matrix models, mathematical models and mechanical models. Common to all material models is the requirement to accurately capture the fabric material response obtained from physical testing in order to validate and, in the majority of cases, calibrate the model.

Architectural fabrics exhibit a number of distinct responses under the application of load. They exhibit different mechanical characteristics in the warp and weft yarn directions. The fabric response contains nonlinearity's and is greatly influenced by the ratio of loading in the two directions. After the initial application and removal of load, unrecoverable strains alter the fabric response . This effect is generally reduced but remains in subsequent loading and unloading cycles [42-44]. The well documented nonlinear stress strain response of architectural fabric is attributed to the internal structure of the fabric and the combination of deformation mechanisms therein. These have been studied at length since the 1980s. The mechanisms include crimp interchange, yarn extension and crushing, coating extensions, friction between yarn filaments and friction between yarns and coating [42, 45].

Many testing methodologies for fabrics were first developed for the determination of mechanical properties of uncoated woven fabrics used in the clothing industry [46]. In more recent years testing methodologies have been developed for fabrics used in engineering applications including woven fabrics used in the production of composites [47] as well as architectural fabrics. However, only a limited number of testing standards exist for architectural fabrics.

The simplest, and one of the most widely used, testing methodologies is uniaxial testing performed on strips of fabric. ASTM D 5035-11, 'Standard Test Method for Breaking Force and Elongation of Textile Fabrics (Strip Method)' [48] and ISO 1421:1998, '*Rubber- or plastics-coated fabrics -- Determination of tensile strength and elongation at break*' [49] along with [50, 51] offer standardised uniaxial testing methodologies. Uniaxial tests are performed in order to assess material strength, determine stiffness characteristics [52,

53], and investigate creep[54], the effects of strain rate[55, 56], load history and temperature[55, 56].

For uniaxial testing, multiple samples are cut from a roll of fabric. The number of samples required is dependent upon the property under investigation. Where an average value is required, 5 samples are deemed sufficient. Significantly more are required in order to investigate the distribution of measured values [57]. ASTM D 5035-11 states that samples should be aligned with the warp direction of the fabric and, if required, additional samples may be aligned with the fill direction. However, when testing architectural fabrics additional strips aligned at 'off-axis' angles from the warp direction are often also tested in order to investigate shear response [52, 53] and failure criteria[55]. The majority of European standard uniaxial testing is based on a 50mm wide strip. However, the European Design Guide recommends a strip width of 100mm in order achieve more consistent results due to the reduced influence of number of threads [57]. Samples are typically affixed into a tensile testing machine via jaw clamps or capstan fixtures which reduce the effect of sample damage at the restraints [58]. A constant rate of deformation is applied to the strip and elongation and load is recorded. Samples may be tested directly to failure or in a repetitive sequence of loading and unloading, referred to as cyclic loading, depending on the mechanical property under investigation. Samples may also be pre-conditioned via cyclic loading prior to testing for mechanical properties.

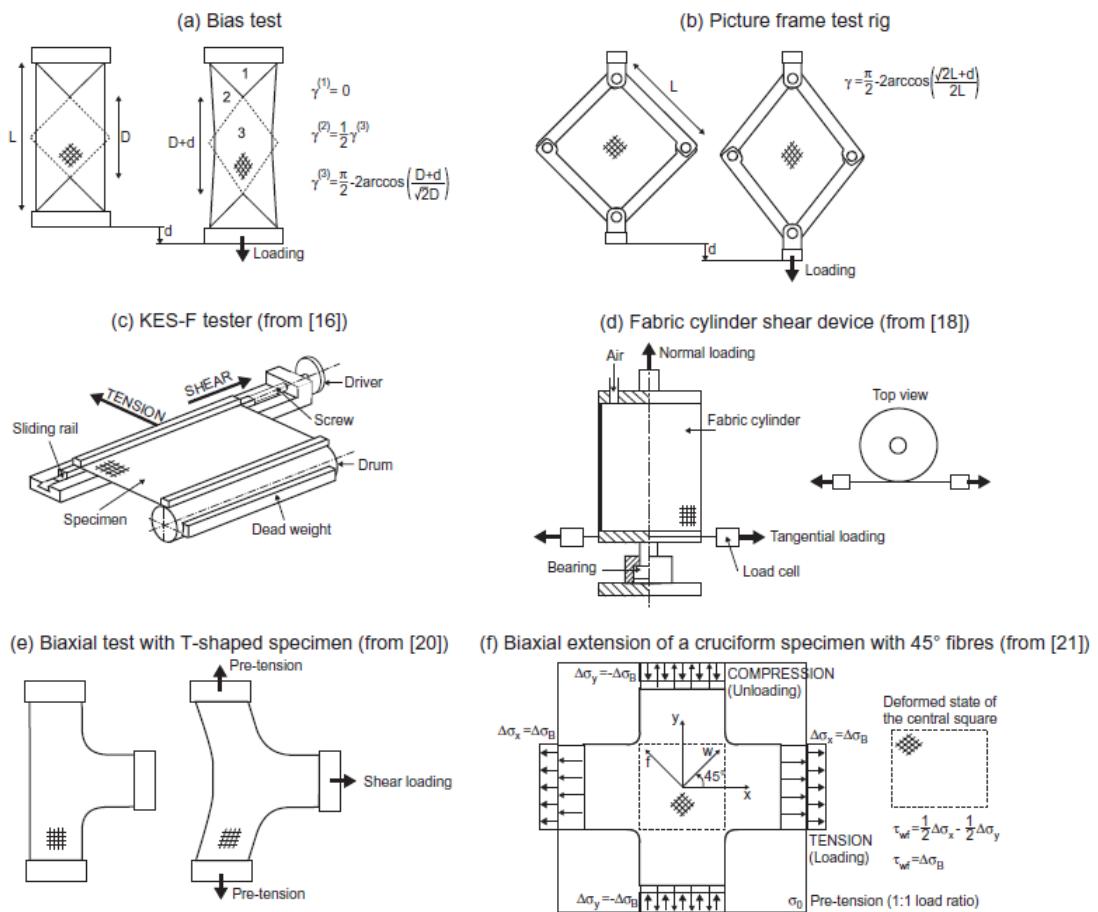
Biaxial testing is used to assess the response of fabric to varying stress ratios in the warp and fill directions and is typically used to obtain stiffness characteristics. The most common form of biaxial testing is performed on flat samples clamped along four edges within a specialist biaxial testing machine. These tests aim to induce stresses in line with the warp and fill directions, thus removing any shear effect. Various biaxial testing rigs have been developed each with individual features that aim to improve the homogeneity of the induced strain field whilst allowing for large displacements. Initial tests were performed on square samples. Later, arms were added to form the now more common cruciform sample in order to reduce the restraint produced by the clamp plate. Subsequently, slits within the cruciform arms were added to further reduce this affect [59]. A number of test rigs aim to further reduce restraint by clamping and individually loading each strip within the cruciform arm [57]. Strain is generally measured at the centre of the fabric sample and the stress is derived from the applied load via a reduction factor to account for reduction in load between the sample's arms and the centre [55].

Due to the complexity of the response of woven fabric to biaxial load, biaxial tests are typically load controlled as oppose to strain controlled. The most common use of biaxial testing is for the production of elastic stiffness constants. MSAJ/M-02-1995, '*Testing Method for Elastic Constants of Membrane Materials*' [43] provides the only international standard methodology for biaxial testing for elastic constants. Biaxial testing is also used to determine compensation values, which account for initial fabric response during installation. Carefully considered load profile design for biaxial testing is of vital importance in order to capture fabric response relevant. The profile should contain a relevant load range and, for the determination of medium to long term properties, include mechanical conditioning. Bridgens presented a protocol that aimed to explore the conditioned response of architectural fabric through the derivation of a more detailed response surface. This was achieved by testing the fabric using a radial load regime including 12 distinct load ratios preceded by conditioning load cycles [59, 60]. In contrast, Blum recommended a profile containing 3 cycles of loading in the warp direction while the fill is held at pre-stress followed by 3 cycles where the warp is held at pre-stress and the fill is loaded, in effect applying only two load ratios[57].

Alternative forms of biaxial testing to the in-plane test include the bursting test and cylinder test. Both use pneumatic pressure to induce biaxial stress within the sample. Bursting tests are performed on circular samples with fully clamped edges. The sample is inflated to failure and the internal pressure is used to calculate the biaxial forces within the membrane. This form of testing is more typically undertaken on non-woven membrane materials such as rubber[10] and ETFE foils [61] which do not exhibit such strong anisotropic mechanical behaviour. Cylinder tests are undertaken using a cylindrical sample that is clamped at each end and inflated to induce radial stress. Movement of the clamped ends allow for the independent application of axial load and torsion [46]. These tests may also be referred to as combined tests as shear may be induced through torsion at various biaxial loads. Kabche *et al.* presented an inflated cylinder test with combined tension and torsion in order to investigate the effect of inflation pressure on axial and shear stiffness moduli for the design of air beams [62]. This method of testing is developed with a specific structural design task in mind.

Similar to the aforementioned 'off-axis' uniaxial strip testing for the determination of shear characteristics, 'off-axis' biaxial tests have also been used [57]. Picture frame testing involves clamping the sample into a frame hinged at each corner. The frame may then be installed into a deformation machine and laterally displaced, effectively squashing the frame in order to induce shear stress [46].

Combined shear testing methodologies endeavour to test fabrics under a combination of biaxial and shear loading. The use of a T-shaped specimen allows shear stress to be applied with controlled levels of axial pre-stress using a biaxial testing rig [63]. Other combined biaxial shear testing methods using specialist shear testing equipment include the cylinder test described previously, picture frame testing combined with initial biaxial conditioning and loading using a biaxial rig [64], shear ramp testing using a specialist load profile [65] and the KES-F tester. Galliot and Luchsinger provide a succinct overview of published shear tests which are illustrated in Figure 2-6 [65].



**Figure 2-6: Test methods used for the investigation of fabric shear response [65].**

The aim of all mechanical testing is to induce a uniform and measurable strain field at a known load in order to infer an applied uniform stress field. This is somewhat difficult as an in depth understanding of the stress field induced by a particular test requires an accurate understanding of how load is transferred from the restrained boundaries of the sample. This is often investigated using finite element simulation. However, an accurate material model is required in order to produce meaningful results. During testing, the applied load is typically measured via load transducers, and strain data is gathered using linear transducers or strain gauges adhered to the sample's surface. These methods of

strain measurement have the drawback of only measuring strain across a finite area of the sample and only in a single direction. This leads to an inability to capture non-uniformity.

Recent advances in high-resolution digital cameras and computing has enabled the use optical measurement techniques such as digital image correlation in the field of material testing. Digital image correlation uses grey value digital images captured by two or more high resolution cameras. If the imaging parameter of each camera's sensor along with the sensor's orientation with respect to each other sensor is known it is possible to calculate the position of any point captured in two or more images. Through the application of a stochastic speckle pattern to the surface of the sample, this method may be used to measure sample contours, displacements and calculate strain during loading. Individual points within the speckle pattern are identified in each image via a correlation algorithm, hence the name digital image correlation. Other optical measurement methods use a series of targets affixed to the sample surface in the place of a speckle pattern. These methods produce similar data to that produced using linear transducers with the advantage of removing the necessity to physically fix a device to the sample. Full-field image analysis measurement techniques have enabled researchers to gain an improved understanding of strain, and therefore stress distribution, throughout a sample[53].

Typically, a plane stress framework that comprises of elastic moduli and Poisson's ratio is used to represent the mechanical properties of fabrics [43, 57, 66-68]. This is often done because fabric analysis codes such as GSA [69] and Easy [18] expect the material properties to be presented in a plane stress format, and do not have capabilities permitting other types of constitutive models [59]. Due to the well-documented non-linear behaviour of coated woven architectural fabrics [43, 67] various post-processing procedures are required in order to derive plane stress moduli from experimental data. A key post-processing procedure is the removal of residual strain (or accumulated creep strain) from the experimental data. This is done to remove distortion of the strain-strain-stress response surface as defined by selected experimental load ratio arms. This enables the fabric behaviour to be represented by a single surface, whether defined directly or using plane stress coefficients.

The plane stress framework is a simplified form of Hooke's stress-strain compliance relationship for an orthotropic material in which it is assumed that all stresses related to the z direction are negligible and therefore can be set to zero. The compliance matrix is given by (2-4)

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix} = \begin{bmatrix} 1/E_{xx} & -v_{yx}/E_{yy} & 0 \\ -v_{xy}/E_{xx} & 1/E_{yy} & 0 \\ 0 & 0 & 1/2G_{xy} \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix}, \quad (2-4)$$

where  $\varepsilon$  denotes strain,  $\sigma$  stress,  $E$  Young's moduli,  $v$  Poisson's ratio and  $G$  shear moduli the subscripts x, y and z refer to the principle axis of the material. The strain to stress stiffness matrix, given in (2-5), is found from the inverse of the orthotropic plane stress compliance matrix.

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \begin{bmatrix} E_{xx}/(1-v_{yx}v_{xy}) & E_{xx}v_{yx}/(1-v_{yx}v_{xy}) & 0 \\ E_{yy}v_{xy}/(1-v_{yx}v_{xy}) & E_{yy}/(1-v_{yx}v_{xy}) & 0 \\ 0 & 0 & 2G_{xy} \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix} \quad (2-5)$$

Due to the symmetry of the compliance and stiffness matrices the reciprocal relationships (2-6) and (2-7) hold, such that,

$$\frac{v_{xy}}{E_x} = \frac{v_{yx}}{E_y} \quad (2-6)$$

$$\frac{E_x v_{yx}}{1-v_{yx}v_{xy}} = \frac{E_y v_{xy}}{1-v_{yx}v_{xy}} \quad (2-7)$$

When considering fabric, the principle directions are taken to be the warp and fill directions and to reflect this the following simplified nomenclature will be used,

$$E_{xx} = E_w, E_{yy} = E_f, v_{xy} = v_{wf}, v_{yx} = v_{fw}$$

where the subscripts  $w$  and  $f$  refer to the warp and fill directions of the coated woven fabric. In current industrial practice, plane stress elastic constants are derived from biaxial test data and measures are taken to ensure loading is applied solely along the warp fill yarn direction. Therefore, in-plane shear stress is generally assumed to be negligible, rendering the 3x3 compliance and stiffness matrices (2) and (3) to be 2x2 matrices. This is only true if you assume that shear and direct stiffness are uncoupled, hence the zeros in the 3x3 plane stress compliance matrix. This may not be a valid assumption for woven fabrics.

The MSAJ standard describes a method for the derivation of elastic constants, that, '*satisfy a reciprocal relationship of anisotropic elastic material*' within a prescribed load range, from load-strain curves obtained for a set of 5 load ratios [43, p.19]. Four parameters are obtained in this process, tensile stiffness in the warp direction,  $E_w$ , and the fill direction,

$E_f$ , and corresponding Poisson's ratio in the warp,  $\nu_w$ , and weft,  $\nu_f$ , directions. Notation here is simplified, in MSAJ Tensile stiffness is expressed as the Young's moduli multiplied by the thickness, here  $E_w$  and  $E_f$  incorporate the thickness. It is noted in MSAJ that the load-strain curves obtained via biaxial testing typically do not conform to a reciprocal relationship, (2-6) and (2-7), and must therefore be replaced by curves that do.

The plane stress constitutive model is fitted to the experimental data using either a least squares method or a method of best approximation where the difference between stresses or strains predicted by the constitutive model and experimentally measured values are minimised. When the difference between experimental and predicted stress is minimised MSAJ gives the relationship between stress and strain in terms of Young's moduli and Poisson's ratio as follows,

$$\begin{aligned}\sigma_w &= E_{11}\varepsilon_w + E_{12}\varepsilon_f \\ \sigma_y &= E_{21}\varepsilon_w + E_{22}\varepsilon_f\end{aligned}\quad (2-8)$$

where,

$$E_{11} = \frac{E_w}{v}, E_{22} = \frac{E_f}{v}, E_{12} = E_{21} = \frac{\nu_{fw}E_w}{v} = \frac{\nu_{wf}E_f}{v} \text{ and } v = 1 - \nu_{wf}\nu_{fw}$$

This relationship is identical to the stiffness matrix equation(2-5) under the assumption of zero applied shear stress. When the difference between experimental and predicted strain is minimised the relationship becomes,

$$\begin{aligned}\varepsilon_x &= E_{11}\sigma_x - E_{12}\sigma_y \\ \varepsilon_y &= E_{22}\sigma_y - E_{21}\sigma_x\end{aligned}\quad (2-9)$$

where,

$$E_{11} = \frac{1}{E_x}, E_{22} = \frac{1}{E_y}, E_{12} = E_{21} = \frac{\nu_y}{E_y} = \frac{\nu_x}{E_x}$$

As expected this relationship is identical to the compliance matrix, equation (2-4), with shear stress assumed to be zero.

Gosling and Bridgens explored the effectiveness of plane-stress theory using the compliance matrix, equation (2) [68]. The results of pairs of biaxial tests for ten different coated woven fabrics, five PVC coated polyester fabrics and five PTFE coated glass fibre fabrics, were combined into single data sets and used to calculate values of Young's moduli and Poisson's ratio for each material. This was achieved by minimising the root mean

square difference between the predicted plane stress values and the test results. Unlike in the MSAJ standard, no constraints were placed on either the values on Young's moduli or Poisson's ratio and as a result the values determined did not adhere to the reciprocal relationship (4). The following reason was given for this inconsistency with plane stress theory '*fundamental for this research is the fact coated woven fabrics are not homogeneous materials: the interaction of warp and fill yarns and the behaviour of the twisted yarn structure mean that they are better described as mechanisms. It is this mechanical interaction which causes the elastic moduli and Poisson's ratios not to fit the relationship for a homogeneous material*' [68, p.220].

On comparing the predicted results with experimental data using response surfaces it was found that '*overall the correlation between the test data plane stress representation is good*' [68, p.218]. This was especially the case for the PVC-polyester fabrics where the variation between the test data and the model was found to be only twice the variation between the two tests used in the production of the elastic constants. The improved fit achieved by the PVC-polyester model was attributed to the greater effect of mechanical preconditioning on PVC-coated polyester compared to PTFE coated glass fibre fabrics. PTFE coated glass fibre fabrics also show more consistency between tests.

Blum *et al.* [57] proposed a biaxial test evaluation method which utilises elastic and interaction moduli for given stress ranges in the form,

$$\begin{bmatrix} \Delta\sigma_{11} \\ \Delta\sigma_{22} \end{bmatrix} = \begin{bmatrix} E_{1111} & E_{1122} \\ E_{1122} & E_{2222} \end{bmatrix} \begin{bmatrix} \Delta\varepsilon_{11} \\ \Delta\varepsilon_{22} \end{bmatrix} \quad (2-10)$$

where  $\Delta\sigma_{11}$  and  $\Delta\sigma_{22}$  represent change in stress in the warp and fill direction respectively and  $\Delta\varepsilon_{11}$  and  $\Delta\varepsilon_{22}$  represent change in strain in the warp and fill directions respectively.  $E_{1111}$  and  $E_{2222}$  are the elastic moduli in the warp and fill directions and  $E_{1122}$  is an interaction moduli representing the stiffness interaction between the warp and fill directions and may be used alongside the elastic moduli to define two Poisson's ratios  $\nu_{12}$  and  $\nu_{21}$ ,

$$\nu_{12} = \frac{E_{1122}}{E_{1111}} \quad (2-11)$$

$$\nu_{21} = \frac{E_{1122}}{E_{2222}} \quad (2-12)$$

The inverse compliance matrix, which relates strains to stresses, is given by Blum *et al.* as,

$$\begin{bmatrix} \Delta\epsilon_{11} \\ \Delta\epsilon_{22} \end{bmatrix} = \begin{bmatrix} C_{1111} & C_{1122} \\ C_{1122} & C_{2222} \end{bmatrix} \begin{bmatrix} \Delta\sigma_{11} \\ \Delta\sigma_{22} \end{bmatrix} \quad (2-13)$$

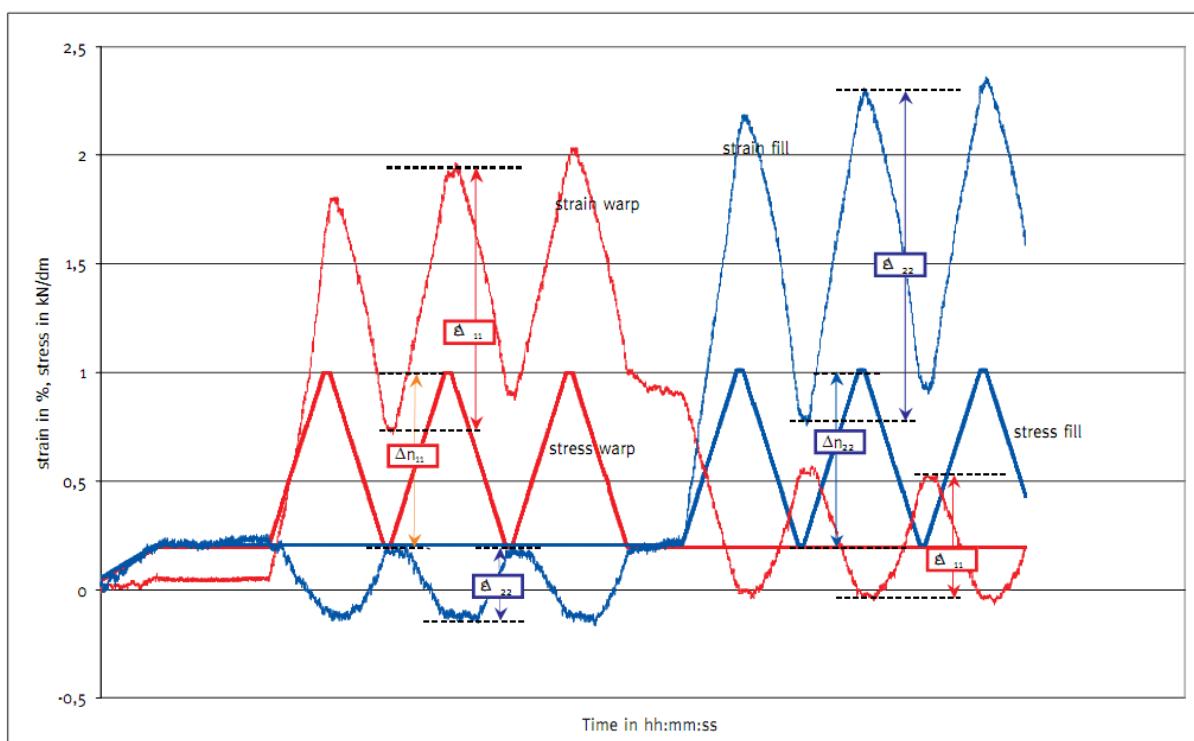
where  $C_{1111}$  denotes compliance in the warp direction,  $C_{2222}$  compliance in the fill direction and  $C_{1122}$  compliance interaction between warp and fill.

The use of change in stress and strain, as opposed to absolute values, leads to the linearization of the fabrics non-linear behaviour between specified intervals. The elastic moduli may be assessed in this way through a specialised load profile, Figure 2-7, that is designed to allow the elastic constants to be decoupled. The warp and fill directions are loaded separately in order to produce the following pairs of equations,

$$\begin{aligned} \Delta\sigma_{11} &= E_{1111}\Delta\epsilon_{11} + E_{1122}\Delta\epsilon_{22} \\ \Delta\sigma_{22} &= 0 = E_{1122}\Delta\epsilon_{11} + E_{2222}\Delta\epsilon_{22} \end{aligned} \quad (2-14)$$

and

$$\begin{aligned} \Delta\sigma_{11} &= 0 = E_{1111}\Delta\epsilon_{11} + E_{1122}\Delta\epsilon_{22} \\ \Delta\sigma_{22} &= E_{1122}\Delta\epsilon_{11} + E_{2222}\Delta\epsilon_{22} \end{aligned} \quad (2-15)$$



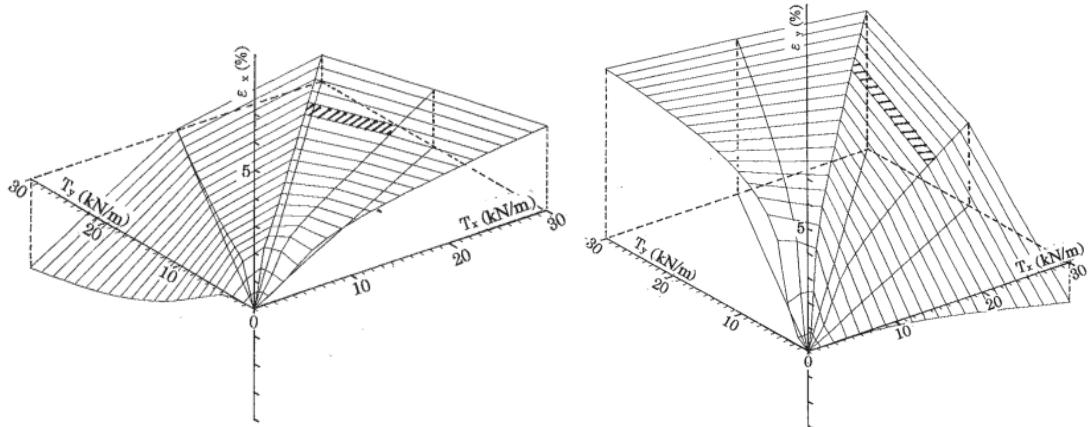
**Figure 2-7: Specialised biaxial load history for the determination of elastic constants**  
[57]

The four equations (2-14) and (2-15) are solved simultaneously to find the three unknowns  $E_{1111}$ ,  $E_{2222}$  and  $E_{1122}$ . Unless the material is isotropic this method will produce two different values of  $E_{1122}$  and differing values of Poisson's ratio that do not adhere to the reciprocal relationship (2-7). For this reason an average of the two  $E_{1122}$  terms is generally used to give a single value [70]. This leads to a further simplification of the derived fabric response beyond linearisation.

Minami [67] extends the use of multi-step linear approximation through the use of response surfaces. A pair of response surfaces relating to warp and fill strain in terms of warp and fill stress were developed from biaxial stress-strain curves obtained for 0:1, 1:2, 1:1, 2:1 and 1:0 load ratios. The surfaces are split into quadrilateral or triangular elements with the corners of each element positioned on experimentally measured stress and strain values (Figure 2-8).

Within each element it is assumed that the stresses and strains conform to plane stress linear elastic theory. Minami uses a stiffness matrix similar to that of Blum *et al.* [57] with two additional constants  $e_x$  and  $e_y$ ,

$$\begin{bmatrix} T_x \\ T_y \end{bmatrix} = \begin{bmatrix} E_{xx} & E_{xy} \\ E_{xy} & E_{yy} \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \end{bmatrix} + \begin{bmatrix} e_x \\ e_y \end{bmatrix} \quad (2-16)$$



**Figure 2-8: Warp and Fill response surfaces [67]**

The five unknown elastic constants ( $E_{xx}$ ,  $E_{yy}$ ,  $E_{xy}$ ,  $e_x$  and  $e_y$ ) are found for each element through a process of minimising the sum of the squares (2-17),

$$S_s = \sum_{n=1}^N [(T_x - E_{xx}\varepsilon_x - E_{xy}\varepsilon_y - e_x)^2 + (T_y - E_{xy}\varepsilon_x - E_{yy}\varepsilon_y - e_y)^2]_n \quad (2-17)$$

such that,

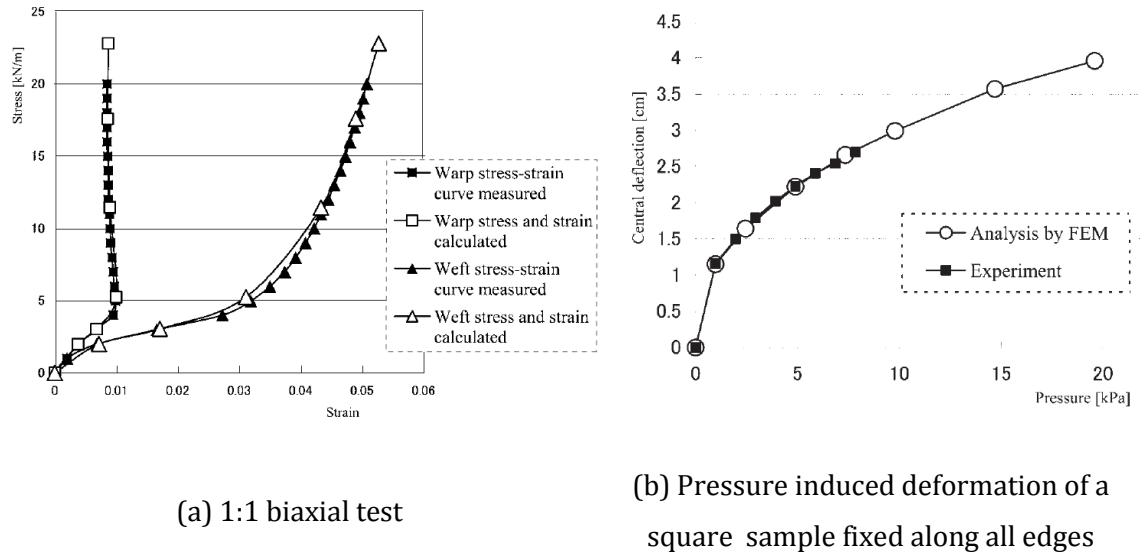
$$\frac{\partial S_s}{\partial E_{xx}} = \frac{\partial S_s}{\partial E_{yy}} = \frac{\partial S_s}{\partial E_{xy}} = \frac{\partial S_s}{\partial e_x} = \frac{\partial S_s}{\partial e_y} = 0 \quad (2-18)$$

This process is similar to that proposed in the MSAJ [43]. Where a more accurate load strain relationship is required it is suggested in [43] that a multistep linear approximation should be used to determine material constants. Once the elastic constants have been derived for each element they are stored in a look up table that may be sourced by a finite element programme. Two simulations using a fundamental non-linear finite element method of two physical tests were reported to demonstrate the implementation. The simulation and experimental results are compared to determine the accuracy of the non-linear material model. A linear equation was used to calculate in plane shear stress for each element, as in,

$$\tau_{xy} = G\gamma \quad (2-19)$$

where  $\tau_{xy}$  denotes in plane shear stress,  $G=49\text{N/m}$  defines shear rigidity and is determined through standard testing [71] and  $\gamma$  denotes shear strain.

The first physical test to be simulated was a cruciform specimen subjected to 1:1 biaxial loading, Figure 2-9. The second physical test simulated was the lateral deformation of a square sample, restrained along all four edges, induced by pneumatic pressure. The deflection was calculated in the centre of the specimen, Figure 2-9b. While the implementation examples presented demonstrate excellent agreement between simulation and experimentally measured data neither explore the fabric response away from 1:1 load ratio. This method introduces a level non-linearity but the accuracy of the model is highly dependent on the size of the element. Larger elements may lead to a failure to capture discontinuities in the behaviour of woven architectural fabric.



**Figure 2-9: FEM predictions compared with experimentally measured data points [67]**

Unlike the methods of MSAJ and Blum *et al.*, the method presented by Minami to determine elastic constants includes stress strain data collected during initial loading without the application of mechanical conditioning. It is however noted that for the analysis of in situ architectural fabric '*the biaxial elongation property surface formed with the biaxial stress-strain curves measured after adequate iterative loading should be used*' [67, p.195].

The compliance matrix (2-4) is used by Galliot and Luchsinger as the starting point for the development of a simple non-linear material model for PVC-polyester fabrics that does not rely of linearization of the stress-strain curve [66]. The stress-strain response of a fabric was investigated using the specialised load history proposed by Blum *et al.* (Figure 1) and a number of observations were made including the fact that the '*Young's moduli are almost linear functions of the normalized load ratio while the Poisson's ratio is rather independent of the load ratio*' [66, p.442]. The normalised load ratios are given by,

$$\gamma_w = \frac{\sigma_w}{\sqrt{\sigma_w^2 + \sigma_f^2}}$$

$$\gamma_f = \frac{\sigma_f}{\sqrt{\sigma_w^2 + \sigma_f^2}} \quad (2-20)$$

The model therefore has three key assumptions, the first of which is common to all of the coated fabric material models discussed in this section – that for given a load ratio the

material behaviour is linear elastic plane stress orthotropic. Therefore, the material behaviour may be described by the compliance matrix (2-4) that conforms to the reciprocal relationship (4) and may be expressed as,

$$\begin{bmatrix} \Delta\epsilon_w \\ \Delta\epsilon_f \end{bmatrix} = \begin{bmatrix} 1/E_w(\gamma_w) & -v_{wf}/E_w(\gamma_w) \\ -v_{wf}/E_w(\gamma_w) & 1/E_f(\gamma_f) \end{bmatrix} \begin{bmatrix} \Delta\sigma_w \\ \Delta\sigma_f \end{bmatrix} \quad (2-21)$$

However, as proposed by Galliot and Luchsinger, this linear model is extended to become non-linear by the second two assumptions;

1. The Young's moduli  $E_w$  and  $E_f$  may be expressed as linear functions of the load ratios  $\gamma_w$  and  $\gamma_f$ ,

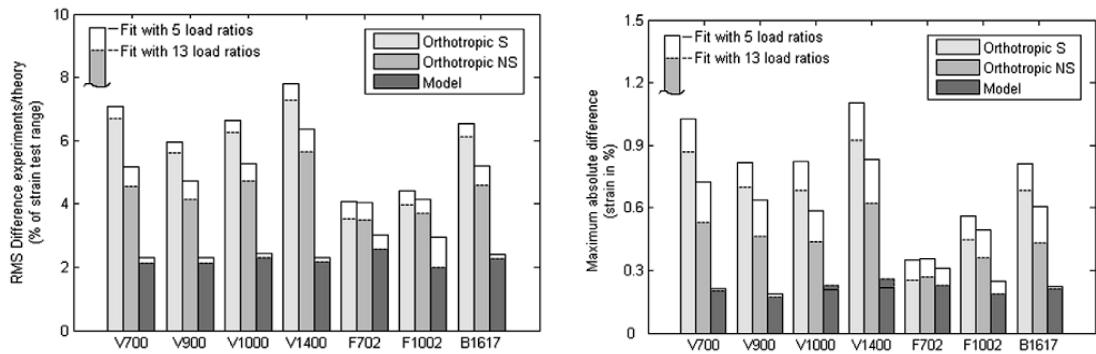
$$\begin{aligned} E_w(\gamma_w) &= \Delta E_w \left( \gamma_w - \frac{1}{\sqrt{2}} \right) + E_w^{1:1} \\ E_f(\gamma_f) &= \Delta E_f \left( \gamma_f - \frac{1}{\sqrt{2}} \right) + E_f^{1:1} \end{aligned} \quad (2-22)$$

where  $E_w^{1:1}$  and  $E_f^{1:1}$  are the warp and fill moduli given by a 1:1 load ratio ( $\gamma_{f,w} = 1/\sqrt{2}$ ) and  $\Delta E_w$  and  $\Delta E_f$  are the variation of young's moduli across the entire range of tested load ratios.

2. The in-plane Poisson's ratio  $v_{fw}$  is independent of the load ratios  $\gamma_w$  and  $\gamma_f$

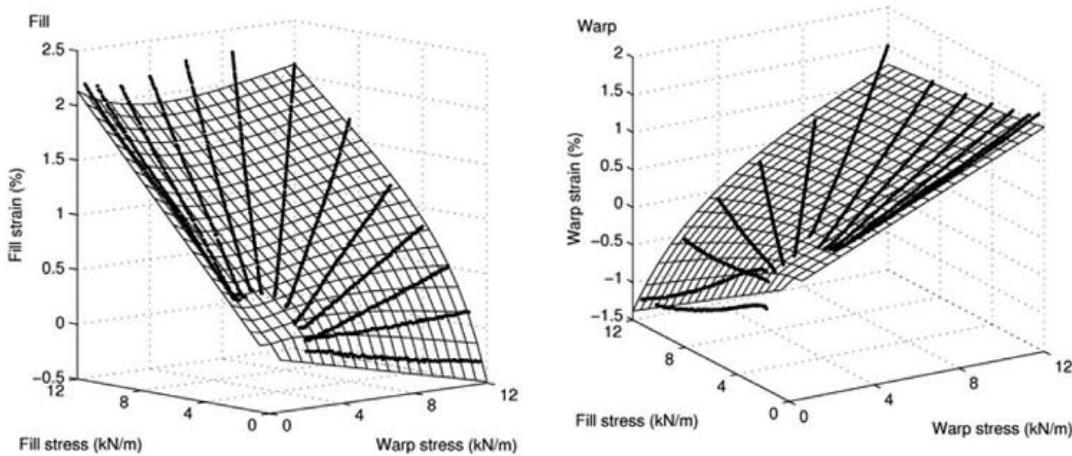
The model parameters for seven types of PVC-polyester fabric were obtained from experimental data using a least square fit method performed using two sets of load ratios, one containing five and another containing thirteen different ratios. The experimental procedure included mechanical preconditioning with only the fifth and final load cycle for each load ratio used to determine fabric properties. The value of Poisson's ratio was constrained to less than 0.5. The allowed range of Poisson's ratio is an important consideration when applying the plane stress framework to architectural fabric. An isotropic elastic material with a Poisson's ratio of 0.5 is typically considered to be incompressible. In the previously described plane-stress material models Poisson's ratio regularly exceeds 0.5 [43] and in some cases exceeds 1 [68]. The required constraint is that the product of the two orthotropic Poisson's ratio must not be greater than 1 in order to avoid a case where a positive strain results in a negative stress. This would be unacceptable when implemented within a finite element simulation.

In comparison to other material models such as a standard plane stress orthotropic model (labelled orthotropic S in Figure 2-10) and the orthotropic linear model with two independent Poisson's ratios [68] (labelled orthotropic NS in Figure 2-10), it was found that the lowest RMS and lowest maximum absolute difference between experimental and model parameters was achieved by the non-linear model for all fabrics tested. The non-linear model performance is also shown to be less sensitive to the number of load ratios used for fitting. This may suggest that less data is required to generate an accurate model, (Figure 2-10).



**Figure 2-10: RMS and maximum absolute difference between experimental data and model predictions [66]**

To further investigate the effectiveness of the non-linear model, response surfaces were plotted using the model and compared with experimental data, Figure 2-11. It was concluded by Galliot and Luchsinger that '*it clearly appears that the experimental curves do not lie all on a plane and thus linear models have limited capabilities. For the proposed non-linear model, a curved surface enables a better representation of the experimental material characteristics*' [66, p.443]. While an improved fit has been shown by the comparison to the plane stress linear models it is still evident in Figure 2-11 that error between the experimental and predict material response remains.



**Figure 2-11: Stress-stress-strain representation of ‘non-linear’ model predictions (surface) and experimental data (dots) [66]**

Common to all the Plane Stress approximations discussed, with the exception of the procedure proposed by Blum *et al.*, is the use of some form of minimising the root mean square difference between the proposed material model and experimental data in order to find a series of elastic constants. The main source of differences between the models is in the application of constraints which force the experimental data to conform to the orthotropic plane stress constitutive equations, and in particular the reciprocal relationships (2-6) and (2-7).

The post-processing of stress-strain curves from which the elastic constants are calculated also varies from method to method. MSAJ [43] and Bridgens and Gosling [68] proposed the simplest methods of elastic constant approximation where single sets of elastic constants are produced for each set of test data. Minami, Blum *et al* and in some cases MSAJ used methods which created multi-linear approximations of the non-linear stress strain response and assumed a plane stress response for each linear section. This produced sets of multiple elastic constants for each stress-strain data set. Minami takes this concept further by creating response surfaces made up of linear elements each with a set of elastic constants. Alternatively, Galliot *et al.* introduced non-linear terms by expressing Young’s moduli in terms of a load ratio term. Both Minami and Galliot *et al.* successfully implement their adapted plane-stress models into FEM software.

All of the plane stress models discussed adhere to the reciprocal relationships (4) and (5) except for Bridgens & Gosling who do not impose any constraints. Nevertheless, Bridgens and Gosling observe a generally good correlation between predicted and experimentally measured values. Luchsinger *et al* introduce the greatest level of constraint and alone do not allow Poisson’s ratios greater than 0.5. However, their non-linear model is shown to

perform better than both a simple plane stress orthotropic model, similar to that of MSAJ, and Bridgens & Gosling's unconstrained model.

In contrast to the above plane stress models which used biaxial test data, Chen *et al.* [52] investigated whether architectural fabrics could be treated as an orthotropic material using off-axial constitutive equations (2-23) derived for anisotropic lamina composites [72]. Uniaxial testing was undertaken on samples of plain weave PVC coated polyester biased to the warp direction in the range 0° to 90° at 15° intervals. Each biased sample was initially tested to failure to find the ultimate strength. A second sample was then taken to 20% of the determined ultimate strength. Load was applied at a constant rate, held and released at a constant rate three times. It was found that after the first two load cycles the third load cycle produced a more linear stress-strain response. This was attributed to permanent deformation within the sample leading to reductions in both shear deformation and crimp interchange in subsequent load cycles. Deformation was measured in the loading and transverse direction.

The mechanical properties including Young's modulus ( $E$ ), shear modulus ( $G$ ) and Poisson's ratio ( $\nu$ ) were derived from the third load cycle results and compared to predicted values derived from the off-axial constitutive relationship (2-23).

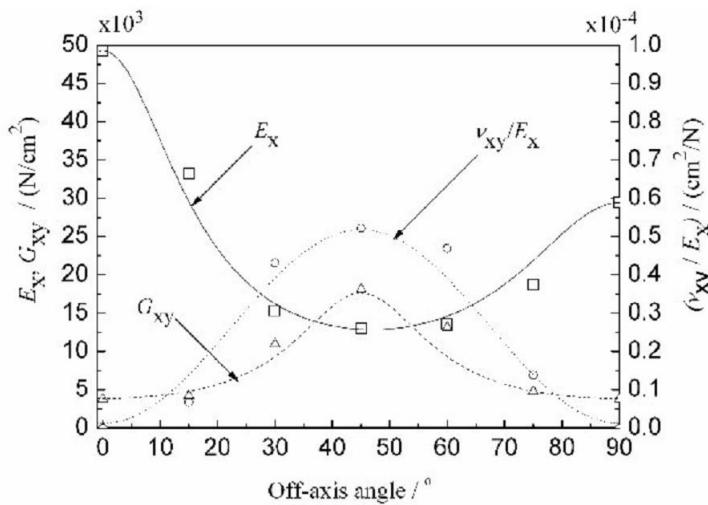
$$\begin{aligned} \frac{1}{E_x} &= \frac{\cos^4 \theta}{E_1} + \left( \frac{1}{G_{12}} - \frac{2\nu_{12}}{E_1} \right) \cos^2 \theta \sin^2 \theta + \frac{\sin^4 \theta}{E_2} \\ \frac{\nu_{xy}}{E_x} &= \frac{\nu_{12}}{E_1} + \left( \frac{1}{E_1} + \frac{1}{E_2} - \frac{1}{G_{12}} + \frac{2\nu_{12}}{E_1} \right) \cos^2 \theta \sin^2 \theta \\ \frac{1}{G_{xy}} &= \frac{1}{G_{12}} + \left( \frac{1}{E_1} + \frac{1}{E_2} - \frac{1}{G_{12}} + \frac{2\nu_{12}}{E_1} \right) 4 \cos^2 \theta \sin^2 \theta \end{aligned} \quad (2-23)$$

$$G_{12} = \frac{E_{45}}{2(1+\nu_{45})} \quad (2-24)$$

Subscript 1 and 2 represent the fabrics warp and fill directions respectively, x and y denote the direction of loading and the direction perpendicular to it respectively,  $\theta$  is the bias angle of the fabric with respect to the warp direction. The principal shear modulus was derived from the 45° bias test using equation (2-24). The shear moduli for all other samples were calculated from the third equation in (2-23) which was simplified by combining it with the second equation to give (2-25). This enables the shear modulus to be calculated using experimentally derived Poisson's ratio and elastic moduli.

$$\frac{1}{G_{xy}} = \frac{1}{G_{12}} + 4 \left( \frac{\nu_{12}}{E_1} - \frac{\nu_{xy}}{E_x} \right) \quad (2-25)$$

In order to make predictions using the constitutive equations (2-23),  $E_1$ ,  $E_2$ ,  $E_{45}$ ,  $\nu_{12}$  and  $\nu_{45}$  (and therefore  $G_{12}$ ) are required. By substituting  $\theta = 0^\circ$  into (2-23)  $E_1$  can be identified as Young's modulus in the warp direction. Likewise by substituting  $\theta = 90^\circ$ ,  $E_2$  is Young's modulus in the fill direction.  $\nu_{12}$  and  $G_{12}$  are the warp direction Poisson's ratio and shear modulus, respectively.  $E_{45}$  and  $\nu_{45}$  are Young's modulus and Poisson's ratio related to loading at  $45^\circ$  bias to the warp direction ( $\theta = 45^\circ$ ). The experimental results are compared with the predictions of the constitutive equations in Figure 2-12.



**Figure 2-12: Experimental results (markers) compared to predicted results (lines) for  $E_x$ ,  $G_{xy}$  and  $v_{xy}/E_x$  [52]**

From this study it is concluded that '*...coated fabrics could be treated as an orthotropic and elastic material if loading was less than 20% of the ultimate tensile stress*' [52, p.373]. This conclusion is also dependent on the fact that the fabric has been mechanically conditioned via cyclic loading. This pre-conditioning replicates the typical behaviour of in-situ fabric and is therefore relevant for structural analysis [60]. However, this study is based solely on uniaxial loading conditions, a loading condition rarely, if ever, found in fabric structures, and does not account for the complexities introduced by biaxial loading [46]. It is also noted that this study focuses on a single specimen of architectural fabric and therefore does not take into account variation in weave pattern, manufacturing process, yarn material or coating properties. The conclusion that coated fabrics in general can be treated as orthotropic and elastic is arguably too broad. PTFE coated fabrics, for instance, have been found to exhibit substantial non-linearity even after mechanical conditioning [68].

The failure of plane stress models to accurately represent architectural fabric material behaviour was identified very early and led to the investigation of alternative methods. Prior to the development of the plane stress model described above Day proposed the representation of biaxial response using the relationship between mean and difference of the principle stresses and strains first developed to represent soil mechanics [44]. The model is defined by the equation of (2-26) where  $\sigma_x$  and  $\sigma_y$  is warp and fill stress respectively and  $\varepsilon_x$  and  $\varepsilon_y$  is warp and fill strain. Due to a low shear stiffness of fabric, the shear stiffness is assumed to be independent. The functions represented by  $f^1$  to  $f^4$  are determined iteratively from points on 1:1, 5:1 and 1:5 load ration stress strain curves.

$$\sigma_a = \frac{(\sigma_x + \sigma_y)}{2}$$

$$\varepsilon_a = \frac{(\varepsilon_x + \varepsilon_y)}{2}$$

$$T = \frac{(\sigma_y - \sigma_x)}{2}$$

$$g = \frac{(\varepsilon_x - \varepsilon_y)}{2}$$

$$\sigma_a = f^1(\varepsilon_a) + f^2(g)$$

$$T = f^3(\varepsilon_a) + f^4(g) \quad (2-26)$$

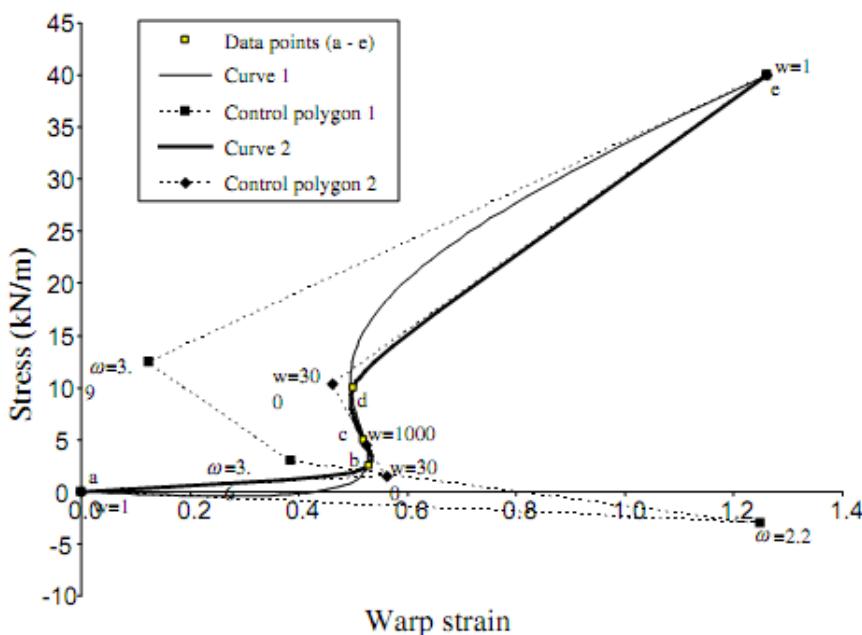
Bridgens and Gosling revisited this method [70] and concluded that it provided a good solution for the representation of up to six non-linear curves. Further testing is required to establish whether this method provides accurate interpolation between the three curves used for the derivation of the model.

Bridgens and Gosling also presented a method of direct stress-strain representation via response surfaces developed from experimental data using spline curves, [68, 70]. Spline functions offer an alternative to polynomial functions for the definition of smooth curved surfaces. The suitability of three types of curve have been assessed for the generation of response surfaces from biaxial test data namely Bezier curves, B-splines and NURBS.

Spline functions are proposed as they offer an '*intrinsic interpretation of test data*' [70, p.1914]. A key advantage of spline curves over more simple polynomial functions is their ability to represent rapid changes in gradient or discontinuities, both of which are features of Biaxial test data. Bezier curves were successfully fit to typical stress strain data using trial and error methods. An additional advantage highlighted is the ability of NURBS functions to represent any number of independent variables thus making them well suited

for the inclusion of factors such as woven fabric shear behaviour which has been shown to be non-linear, hysteretic and discontinuous.

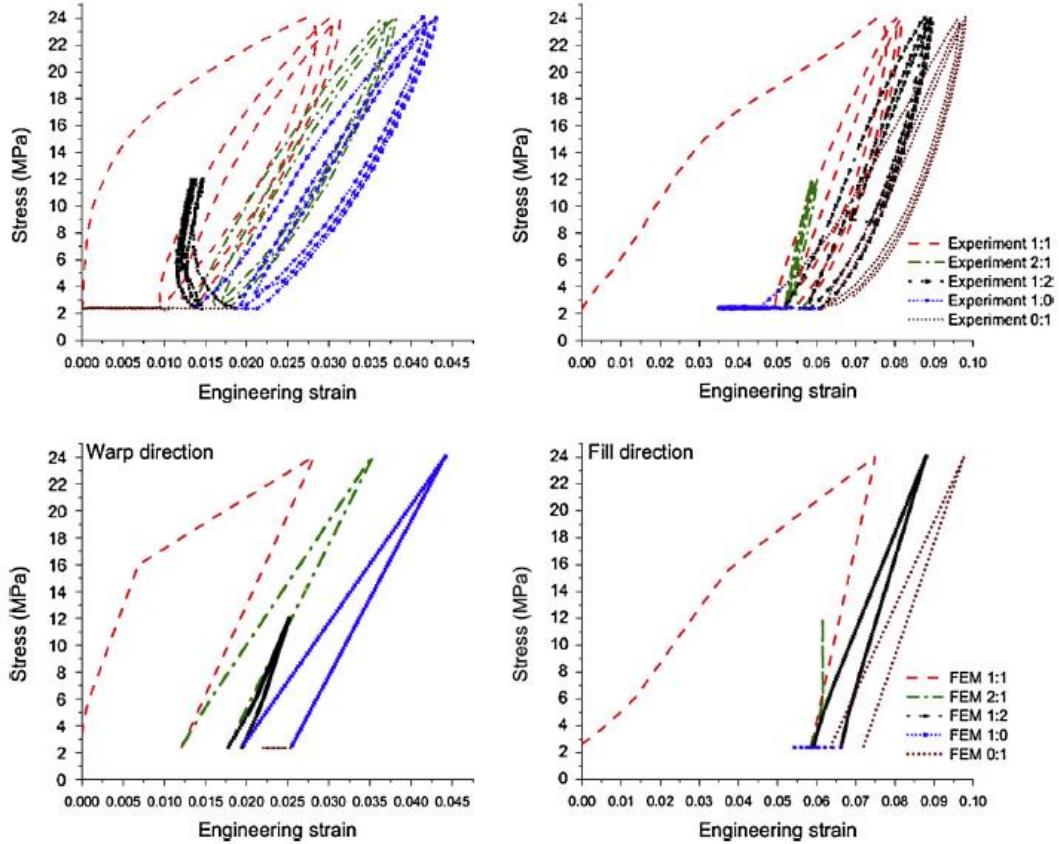
A major unresolved flaw in the use of spline curves is however highlighted in Figure 2-13. It was found that whilst spline curves may provide a close fit to data points, they are non-unique. Varying the control point locations each with a mean square offset from the data of zero can generate a wide range of interpolating curves. This non-uniqueness of the curves leads to the conclusion that additional criteria are required to produce unique curves that reliably represent the fabrics behaviour.



**Figure 2-13: Experimental data fit with two different rational Bezier curves [70]**

Very recently a elasto-plastic material model has been published which uses 11 variables determined form a combination of biaxial and uniaxial test data [6]. The important noval feature of this material model is the representation of permanent strain, as well as the representation of orthotropic and non-linear behaviour. The model is implemented in the finite element code Abaqus. The model is validated using both uniaxial and biaxial experimental data. the model produces a reasonable fit to the uniaxial experimental data. However, the response is somewhat simplified and multiple cycles have not been demonstrated. The model is validated for a biaxial case using a simulation of quarter of a biaxial sample undergoing a 1:1 load cycle. The simulation and experimental results again are shown to be in reasonable agreement but again the response is simplified. Strain recovery and negative strain is particularly underestimated. The ability of the model to

generalise other load ratio combinations has not yet been demonstrated due to a lack of available data.



**Figure 2-14: Elasto-plastic model representation of biaxial response data compared with experimental data [6].**

An alternative approach to using only experimental data to derive stiffness parameters for constitutive equations is to model the internal structure of the fabric, attempting to capture the various mechanisms that contribute to the overall response. The key displacement mechanism to represent is crimp interchange. Crimp interchange is the process of yarn straightening in the direction of maximum load increasing the crimp in the orthogonal direction. Typically, models will represent a unit-cell, the smallest possible repeatable unit, within the fabric. In the case of plain weave fabrics the unit cell is formed by two orthogonally crossed yarns, creating half a wavelength of each the warp and fill. The models are derived from the geometry of the unit-cell in combination with material properties of the constituent yarns and coating material, if present.

The models can generally be placed into one of two groups [59]. The first group require biaxial experimental data to determine various parameters and aim to fill the gaps between measured data points on the stress-strain surface [73, 74]. The second group

uses yarn and coating properties along with fabric geometry to predict the stress-strain response or ultimate strength of the fabric without the need for biaxial testing [75-78]. Since the late 1990s work has been undertaken in modelling the unit-cell or a group of unit cells using 3-dimensional finite element analysis [79-81]. These models enable a reduction in limiting assumptions but rely on the accurate definition of properties for the constituent materials. The majority of models in this group have been developed to investigate the mechanical behaviour of woven panels used in heat formed composites. All models are validated, if not optimised, using experimental data and therefore some testing is unavoidable.

Due to high computational cost and linking issues, mechanical models are rarely used directly as constitutive models within the finite element simulation of fabric structures, one exception is presented in [42]. They are more frequently used to produce coefficients for the plane stress constitutive equations classically used. Mechanical material modelling is outside the scope of this body of work. However, work in this area has provided important insight into the overall behaviour of architectural fabrics. Furthermore, it highlights the complexity of fabric behaviour and the need for consideration of this when undertaking structural analysis.

Architectural fabric strain-stress response is complex and non-linear. Specialist biaxial and uniaxial testing methodologies are used to capture the direct strain-stress response of architectural fabric for the development and validation of fabric material models. These methodologies will be employed to capture the experimental data required by the material model to be developed in this thesis. Biaxial load profiles using various combinations of load ratios have been proposed in order to explore the full material response and provide data for the fitting of constitutive models. Residual strain is typically removed prior to fitting. Comprehensive load profiles including additional non-standard features will be used to capture the full fabric response. The removal of residual strain in post processing may be avoided if the model is designed to capture load history effects. Shear stress is generally assumed to be decoupled from direct stress and is measured in separate physical tests. The assumption of decoupled shear behaviour will be maintained in order to allow the use of established biaxial testing methodologies. However, it is anticipated that combined direct stress and shear response will be incorporated into the proposed material mode at a later date.

A variety of fabric material models based on the plane stress framework have been developed. It has been demonstrated that an improved fit to experimental data may be

achieved by disregarding the plane stress reciprocal relationship which constrains Poisson's ratio. The plane stress framework has been further manipulated to include non-linearity by adding additional terms. In one method the strain-strain-stress response surface is divided into separate elements with unique elastic constants and in another the elastic moduli are modified based on the current load ratio. Elastic matrix models based on the plane stress framework are limited by the use of relatively small numbers of variables and classical plane stress assumptions that do not apply to the mechanical behaviour of woven coated fabrics.

Methods that entirely move away from plane stress assumptions were proposed as early as 1986. A model based on the relationship between mean and difference of principle stresses and strains showed promising results. The concept of direct stress-strain representation using a response surface type model has also been explored. Extrapolation between experimental data to generate a smooth continuous response surface is required in order to implement this modelling methodology. Spline curves have shown to be most promising but are sensitive to the location of control points leading to a large variation between curves fit to the same data. Artificial neural network material modelling, presented in the following section, offers a solution for the generalisation of fabric response from biaxial experimental data. Neural networks trained using comprehensive experimental data and have been demonstrated to successfully capture and generalise non-linear and history dependant material response for a variety of materials. Furthermore, they are suitable for implementation in finite element code.

## 2.3 Neural Network Material Modelling

Advances in computing technology have allowed for the exploration of a number of computer methods based on artificial intelligence concepts. Genetic Algorithms and Neural Networks have been used in a number of engineering applications and both have been used in constitutive modelling.

Genetic Algorithms (GAs) are an optimisation heuristic based on the principles of natural selection. The process or evolution begins with a population of individuals, usually randomly generated. Each individual member has a set of characteristics or chromosomes, typically expressed in binary form, which can be altered or mutated. The process is iterative; individuals are assessed using a fitness function, usually the objective function of the optimisation. The individuals deemed to be less fit are discarded and the remaining population is used to generate a new population through random combination and mutation. The process is halted after a set number of generations or once the population exhibits a specified level of fitness. Genetic Programming (GP) is a specific form of a Genetic Algorithm where the objective function itself is obtained via evolution. In the case of GP each individual of the population is a process or function.

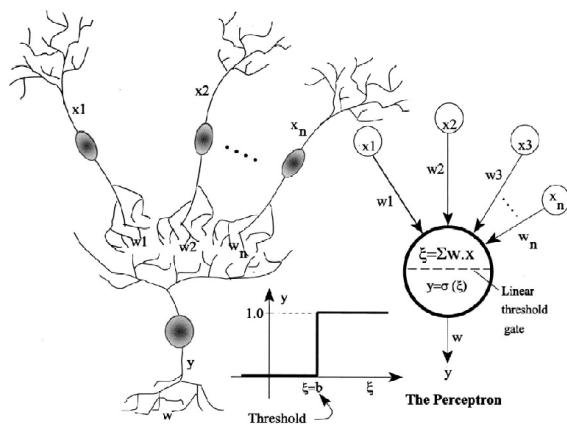
GAs and GP have been used to define constitutive equations for a number of engineering materials but have primarily been employed in the area of soils [82, 83] and composites [84, 85]. They have also recently been used to explore the cyclic behaviour of steel [86, 87]. GAs are typically selected where a complex function is required to represent the material response that is defined by several unknown parameters.

Artificial Neural Networks (ANNs) belong to the ‘soft computing’ [88] or ‘intelligent method’ [89] paradigm and are based on a highly simplified biological neuron such as may be found in the brain. Their basic structure comprises layers of interconnected processing units referred to as artificial neurons or nodes. Each neuron sums the input signals from the previous layer along with a bias, performs some form of transfer function, and passes it to each neuron of the next layer via a weighted connection. The weights and biases are determined through a process of training and it is within the weights and biases that the relationship between input and output is captured. ANNs have been successfully applied to a wide range of real world problems including prediction [90, 91], control [92], pattern recognition [93] and optimisation [94, 95]. It is however the ability of an ANN to approximate a function which is of most relevance to constitutive modelling of architectural fabric. This is where an ANN is trained to represent the functional mapping of input signals, for example strains, to corresponding outputs, such as stresses [96]. The

basics of ANNs have been widely discussed in the literature. Useful overviews are provided in [97] and [98]. A brief summary is provided in the following paragraphs.

To understand the development of the artificial neuron a basic understanding of a biological neuron is advantageous. An extremely simplified biological neuron, shown in Figure 2-15(a), comprises a cell body or soma and two types of connective branches - the axon and the dendrites. The dendrites receive signals and transfer them to the cell body. The axon receives signals from the cell body and transfers them via synapses (microscopic gaps) to the dendrites of neighbouring neurons [97]. Within the axon the signal is electrical. However at the synapse the signal is converted to a chemical neurotransmitter. The neurotransmitter is diffused across the synaptic gap to the dendrites of neighbouring neurons, in turn causing that neuron to produce new electrical signals. The quantity of the neurotransmitter produced and therefore the strength of the chemical signal is proportional to the strength of the signal reaching the synapse. The magnitude of the signal produced by any neuron is dependent on the intensity of signal received from feeding neurons, the strength of connecting synapses, and the threshold of the neuron. Due to the huge number of connections associated with each neuron and the vast number of neurons within the network, this biological system is capable of completing massively complex tasks, such as facial recognition, within a fraction of a second.

Though massively simplified, the basic process of a biological neuron is used to inform the development of the artificial neuron. Within an artificial neuron, the activity of the cell body is represented by the transfer function,  $f$ , and bias,  $b$ . The axon and dendrites are represented by the connections, and the synapses by the weights  $w$  [98]. Figure 2-15 shows a biological neuron (a) alongside its equivalent artificial counterpart often referred to as the perceptron (b).



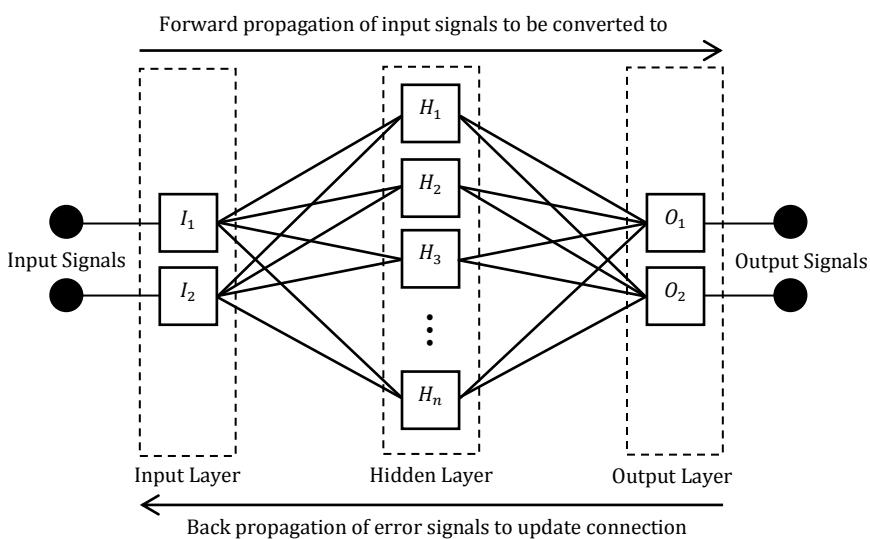
**Figure 2-15: A biological neuron and an artificial neuron the perceptron [98]**

Figure 2-15 (b) depicts a single perceptron. The perceptron sums the weighted output signals from each neuron of the previous layer, adds a bias signal and passes the result through an activation function. The function may be any differentiable function.

Commonly used activation functions include threshold, linear, sigmoid and Gaussian functions. The sigmoid transfer function is the most commonly used transfer function for constitutive modelling [97, 99-102].

The architecture of an ANN generally takes on a layered form with each layer containing a number of neurons that receive weighted signals and output a response. ANNs are typically categorised according to the organisation of the connections. A feed-forward network is organised in such a way that the signal is passed consecutively from one layer to the next. Networks belonging to this group include the single layer perceptron, multilayer perceptron and radial basis function nets. A recurrent or feedback network may contain one or more nodes that are connected to nodes either in the same or previous layers. Networks belonging to this group include competitive networks [103], Kohonen's SOM [104], Hopfield network [105] and ART models [106].

A graphical representation of a general multilayer feed forward neural network is shown in Figure 2-16. The network contains an input layer made up of a node for each input value that is connected to a subsequent layer known as the hidden layer. This layer contains a user-defined number of neurons. The network may be built to contain any number of additional hidden layers although one is often sufficient. The final layer of a standard feed forward neural network is the output layer; this layer contains a neuron for each network output.



**Figure 2-16: Single hidden layer feed-forward neural network**

The first feature of network training is the training paradigm. This depends on the form of training data presented to the network. Supervised training occurs when a network is presented with the correct answers corresponding to each given input. The learning process aims to minimise the error between network output and the correct answer. During unsupervised training the network learns solely from the underlying structure or correlations within a set of inputs and organises them accordingly. Networks may also be trained using a hybrid of both supervised and unsupervised learning. Functional mappings, such as those performed by ANN constitutive models, require supervised training.

The second feature of network training is the learning rule used to inform how network weights are to be updated. Well known learning rules include error-correction, Boltzmann, Hebbian and competitive. A learning algorithm is developed to update the weights according to the selected rule. The selection of the learning rule and the subsequent learning algorithm is dependent upon the architecture of the neural network and available training data, which is in turn dependent on the task to be performed by the neural network. Table 1 compiled by Jain *et al*, 1996 gives a good overview of existing network architectures along with their associated learning algorithms and applications. As this review is concerned with function approximation for constitutive modelling, the relevant architectures include single or multilayer perceptron and radial basis-function networks. These along with their associated training algorithms will be the focus of this review. For details on other network architectures and learning algorithms see [97, 98].

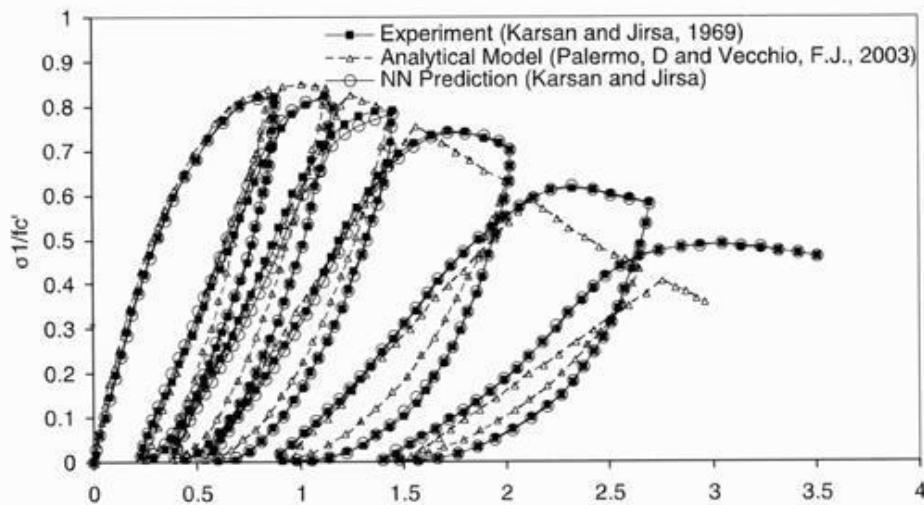
| Paradigm     | Learning rule                    | Architecture                    | Learning algorithm   | Task  |
|--------------|----------------------------------|---------------------------------|--|---|
| Supervised   | Error-correction                 | Single or multilayer perceptron | Perceptron learning algorithms<br>Back-propagation<br>Adaline and Madaline | Pattern classification, <b>Function approximation</b> , Prediction, Control |
|              | Boltzmann                        | Recurrent                       | Boltzmann learning algorithm   | Pattern classification  |
|              | Hebbian                          | Multilayer feed-forward         | Linear discriminant analysis   | Data analysis   |
|              |                                  |                                 |  | Pattern classification  |
|              | Competitive                      | Competitive                     | Learning vector quantization   | Within-class categorisation, Data compression                               |
|              |                                  | ART network                     |  | Pattern classification  |
| Unsupervised | Error-correction                 | Multilayer feed-forward         | Sammon's projection  | Data analysis   |
|              | Hebbian                          | Feed-forward or competitive     | Principal component analysis   | Data analysis, Data compression   |
|              |                                  | Hopfield Network                | Associative memory learning  | Associative memory  |
|              | Competitive                      | Competitive                     | Vector quantization  | Categorisation, Data compression  |
|              |                                  | Kohonen's SOM                   | Kohonen's SOM  | Categorisation, Data analysis   |
|              |                                  | ART networks                    | ART1, ART2   | Categorisation  |
| Hybrid       | Error-correction and competitive | RBF network                     | RBF learning algorithm   | Pattern classification, <b>Function approximation</b> , Prediction, Control |

**Table 1: Summary of common ANN architectures with associated learning algorithms [97]**

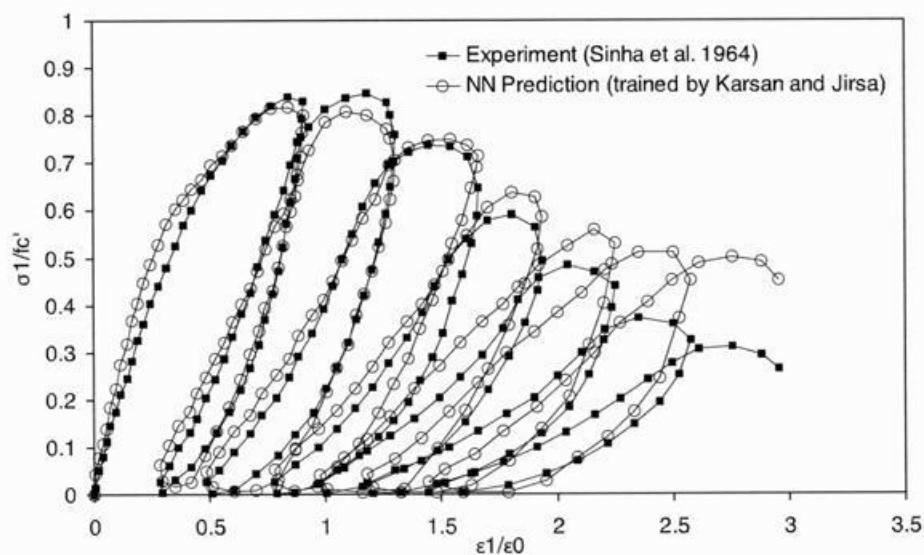
Multilayer feed-forward networks trained using back-propagation training algorithms are by far the most commonly used network in material modelling. As shown in Figure 2-16 input signals are propagated forward through the network to be converted to an output signal via the neurons and weighted connections. During this phase all connection weights are held constant. The error between the network output and the target output is then calculated and propagated backwards through the network. This error is used to update the weights in the direction of the steepest gradient. The gradient is determined from the partial derivatives of the total network error with respect to the network weights or biases. This process of forward propagation of inputs and backward propagation of error is repeated until the network reaches some level of convergence.

The key fundamental advantages of ANNs over other material models include learning ability, generalisation, ability to cope with fuzzy or discontinuous data and adaptability [97, 98, 107]. In contrast the key weaknesses are a limited ability to extrapolate, overfitting, poor definition of uncertainty with regard to prediction, and the 'black-box' nature of ANNs [88, 108]. ANNs have been used to describe the characteristics of numerous engineering materials including concrete [109, 110], soil [111, 112], steel [89] and composite materials [113]. Neural networks show particular promise in modelling materials that exhibit a highly complex non-linear response that may be history dependent [100, 114, 115].

A neural network with additional load history inputs is demonstrated in [114] to be capable of capturing the cyclic behaviour of plain concrete under uniaxial cyclic loading. The trained network output was compared to the experimental data used in training alongside an analytical model for the same behaviour. The network was further tested using unseen data not used in network training in order to illustrate the networks ability to generalise the response.



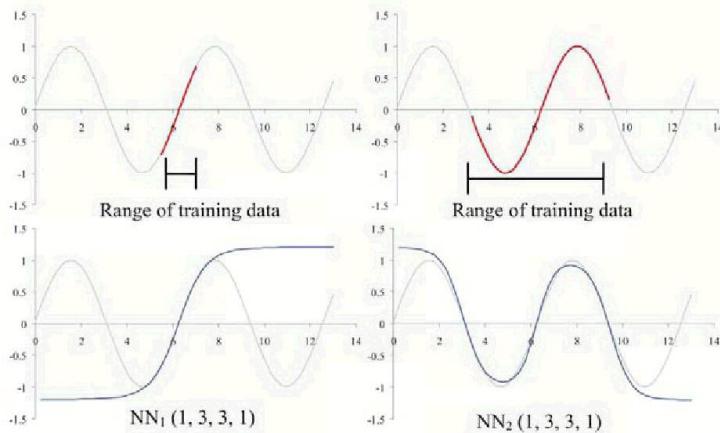
**Figure 2-17: Trained hysteretic concrete network material model output alongside experimental training data and analytical model prediction, [114]**



**Figure 2-18: Trained hysteretic concrete network material model output alongside experimental data not used in training, [114]**

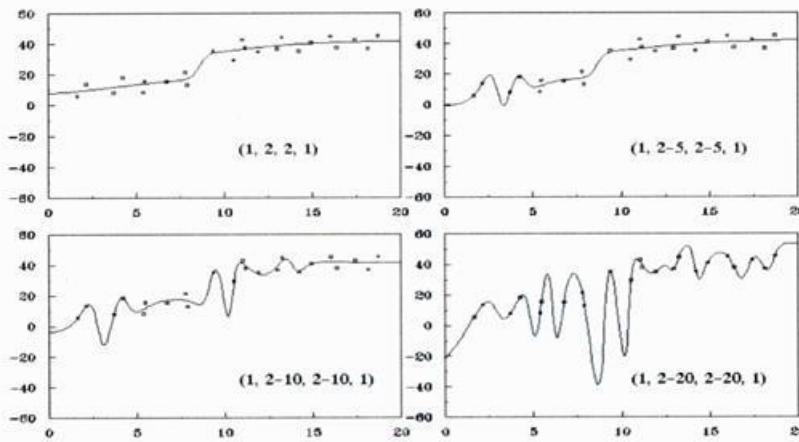
Given that feed-forward, back-propagation ANNs effectively learn a functional mapping through a process of training using sets of input data with known outputs, the need for prior knowledge and limiting assumptions regarding the modelled response is, for practical purposes, removed. However, knowledge of the material response is required in order to effectively design the architecture of the network, gather comprehensive training data, and implement a successful training algorithm. In the case of material modelling, training data may be gathered through traditional material experiments or, alternatively, through the monitoring of an in-situ structure.

ANNs have been shown to perform poorly when presented with inputs that are outside the domain of the training data – e.g. extrapolation. This problem is illustrated clearly in Figure 2-19 where a basic ANN is trained to map a sine curve using selected sections of the curve [88]. This weakness leads to the need for the network designer to have an in depth understanding of the required domain. However, limited guidance is available in the definition of a comprehensive data set. Designer experience is commonly relied upon. One alternative technique employed in constitutive modelling is the use of global load-displacement data in combination with a finite element model and a partially trained ANN material model to iteratively produce increasingly accurate stress-strain pairs for network training [99, 116, 117]. This form of training is termed ‘auto progressive’ or ‘self-learning’.



**Figure 2-19: Neural networks trained to approximate  $\sin(x)$  within two different ranges of  $x$  [88]**

Once trained an ANN will gain the ability to generalise the response and produce outputs from previously unseen inputs within the domain of the training data. This ability to generalise allows the network to robustly represent a function based on noisy or incomplete data. An associated problem is over-fitting. This occurs when a network no longer maintains the ability to generalise a response and effectively learns the noise or scatter within the training data. Over-fitting may occur due to insufficient training data, over training, or may be caused by unnecessarily large or complex network architecture. This issue is illustrated in Figure 2-20 where a scattered data set is approximated by increasingly large neural networks [88]. Numerous techniques have been employed to prevent the over-fitting phenomena including early stopping during training [118] and adaptive training where the neurons are automatically added as required during network training [17].



**Figure 2-20: Training and over-training of a neural network with data containing scatter [88]**

Neural Network material models are often viewed as 'black-box' models as their internal parameters have no specific physical meaning. The lack of physical meaning is necessary to allow for the networks key advantage of requiring no prior assumption about material response. There have, however, been numerous studies which utilise ANN models to gain insight into a material's behaviour [17].

Finally ANNs are adaptable in that they may undergo further training as additional training data becomes available. Specifically within the area of material modelling, ANNs have also been shown to be suitable for implementation within readily available finite element code. Implementation of neural network constitutive models in finite element analysis has been demonstrated in [119-121]. A constitutive model performs two functions. Firstly it provides the means to determine current stresses from current strain. This is relatively straightforward as current stress may be taken directly as the network output. The second function is contributing to the generation of a global stiffness matrix typically derived from the element B-matrix with the constitutive matrix in an equation of the form (2-27). This is problematic, as it requires the constitutive equation to be in the form of a square matrix compatible with the Element B-matrix.

$$K_E = \sum \int_V [B]^T [E][B] dV \quad (2-27)$$

The derivation of an implied elastic stiffness matrix from the partial derivatives of the neural network equations is proposed and demonstrated in [119]. Alternatively, a feed forward neural network may be trained to directly generate elastic constants used to populate and establish elastic matrix. This is demonstrated in the modelling of the nonlinear spring back response of steel sheet metal in [102]. The use of network

derivatives is used in the implementation of a fuzzy neural network trained to represent uncertain time dependant material response in a fuzzy finite element analysis [106].

Neural networks have been demonstrated to be suitable for the material modelling of a wide variety of engineering materials. The back-propagation training algorithm is almost exclusively used for the training of constitutive material models using experimentally gathered strain-stress data. Physical testing may be coupled with a non linear finite element simulation to facilitate the training of complex constitutive models from more easily obtained boundary data. A relatively simple network architecture comprising a single hidden layer with a tan-sigmoid function can be used to capture non-linear material response.

The disadvantages of neural networks include poor extrapolation outside the bounds of the supplied training data, over fitting leading to poor generalisation and uncertainty introduced by the training process. All of these features should be kept in mind during the development of a neural network material model. The design and generation of a comprehensive training data set should eliminate the requirement for the network to extrapolate beyond the training data domain. Early stopping during training is a relatively straight forward way to control over-fitting. Thorough testing using 'unseen' data not used in training is also vital for the validation of the material model and should reveal if over-fitting has occurred. The training of multiple networks using the same training data will allow network uncertainty to be investigated and quantified.



## 2.4 Summary and Conclusions

A holistic approach is required for the design of fabric structures due to the dependence of the structural form on restraint conditions, applied pre-stress, patterning, and applied loading. A complete fabric analysis package should be capable of form finding under prescribed pre-stress, take into account the expected patterning at the earliest opportunity, and be capable of resolving geometrically non-linear state equations.

Dynamic relaxation has been demonstrated to be a robust solution algorithm and will be used in the code presented in this thesis.

The membrane element formulation should include large strain assumptions and control of material direction. This is vital to enable the material warp and fill directions to be aligned with panel geometry. Mesh control during form finding is also essential in order to maintain regular mesh distribution and produce feasible panel geometries. Geodesic control elements have been successfully demonstrated for this purpose. 6 node isoparametric large strain linear strain triangle finite elements have been demonstrated to be suitable for the representation of architectural fabric. They are able to represent curved geometry and are more efficient in the approximation of steep stress gradients than the more simplistic constant strain triangle element.

Reliability analysis of fabric structures enables the designer to assess the acceptability of a design by providing an estimate of probability of failure or the safety index. FORM reliability is a well-established methodology and has been implemented with a specialist fabric finite element code. When assessing the feasibility of the implementation of FORM, a number of issues should be considered and mitigated against. These include discontinuous failure surfaces, the presence of multiple local design points, and instability of the finite element caused by a design point located too far into the failure region.

The collection of comprehensive response data is required for the calibration of all current material models. This is done through physical testing, most commonly based on either uniaxial strip testing or biaxial testing on cruciform samples. Cruciform specimen testing enables the capture of biaxial mechanical effects, including crimp interchange, which massively influences the behaviour of in situ architectural fabric. Shear testing is further challenging, and undertaken using a range of methods. The picture frame test shows promise as it allows the shear test to be undertaken with varying levels of direct stress applied. Typically, shear behaviour is decoupled from direct stress response and is represented in the majority of material models by a single modulus.

The constitutive material model used in conjunction with any specialist fabric finite element package greatly affects the effectiveness of the simulation. Current best practice relies on the plane stress framework with its compatibility with readily available finite element codes. A variety of modifications to the plane stress approach have been proposed, including the removal of the reciprocal relationship between Poisson's ratio terms of the compliance matrix and the addition of various parameters that aim to add non-linearity to the inherently linear plane stress model. One such model adapts the plane stress material description to a response surface style model. A recent elasto-plastic fabric material model has been demonstrated to capture non-linear behaviour and permanent strain. However, the model performs poorly when representing load ratios which result in negative strain and does not capture strain recovery.

A pure response surface type material mode offers promise due to the removal of limiting plane stress assumptions and offers the opportunity to use experimental response data directly to model fabric response. The key problem to solve is the interpolation between available data points in order to model the response away from tested load ratios.

Neural networks, with one or more hidden layers and a sigmoid transfer function, have been demonstrated to be capable of capturing constitutive relationships between stress and strain for a variety of materials. They are capable of learning material response directly from experimental data and have been shown to have the ability to generalise. Neural network material models are capable of accurately reproducing material behaviour in 'unseen' regions of the material response. However, it is important to note that networks are unreliable in regions outside the training data. Careful training data set design is vital to ensure that the network has sufficient data to represent the response area required for simulation.

With the addition of historical inputs and internal variables, artificial neural network material models have also been demonstrated to be capable of capturing history dependant behaviour. This enables the effects of accumulated residual strain under cyclic loading to be assessed in a finite element simulation. It is also noted that neural networks can be implemented directly within finite element analysis. The current stress may be obtained directly from the network model presented with current strain, and the global stiffness matrix may be found using the implied stiffness matrix, which can be derived directly from the network equations.

Therefore, in this thesis a feed forward neural network architecture with a single hidden layer with a tan-sigmoid transfer function will be used for the development of fabric

neural network material models. A back propagation training algorithm will be used for network training. Additional network inputs will be used to facilitate the representation of history dependant fabric behaviour.

The 'self-learning' method of training, using a finite element simulation coupled with a physical experiment, shows good potential for the capture of complex material response from a relatively straightforward physical test. However, it increases the complexity of network training. For this initial investigation into the development of an architectural fabric material model the strain-stress training data will be obtained using established fabric testing methodologies. Shear response will be assumed to be decoupled from the direct stress response and will not be included in the network model at this stage of development.

In order to assess the ability of the trained networks to accurately generalise fabric response, and to identify cases of over-fitting, the networks will be tested with 'unseen' data not used in training. The number of nodes required in the hidden layer will be determined through the training of sets of networks with varying numbers of nodes. The variability between network models trained with the same data will also be investigated through the training of sets of multiple networks.

The trained and validated network material models will be implemented in the dynamic relaxation, large strain, finite element code using an implied stiffness matrix. The network will be used directly to determine current stress from strain, and the implied stiffness matrix will be used to generate the system stiffness matrix used to determine the damping coefficients for the dynamic relaxation algorithm.



**Chapter 3.**  
**Finite Element Formulation**



## Contents

|  |     |
|--|-----|
| Chapter 3. Finite Element Formulation.....                     | 71  |
| 3.1 Element Equations.....                                     | 78  |
| 3.1.1 Application of the assumed displacement field [25] ..... | 79  |
| 3.1.2 Dynamic Relaxation solution algorithm .....              | 80  |
| 3.1.3 Initial and updated nodal configurations .....           | 83  |
| 3.1.4 Natural co-ordinate system and shape functions [34]..... | 84  |
| 3.1.5 Element local co-ordinate system [30].....               | 87  |
| 3.1.6 The Jacobian matrix [30].....                            | 88  |
| 3.1.7 Element B-matrix [30,34] .....                           | 89  |
| 3.1.8 Element initial stress stiffness matrix [30,122].....    | 93  |
| 3.1.9 Numerical integration for triangles [31,34] .....        | 95  |
| 3.1.10 Cable elements.....                                     | 96  |
| 3.1.11 Form finding with geodesic control.....                 | 98  |
| 3.1.12 Procedure overview .....                                | 102 |
| 3.1.13 Wrinkling procedure.....                                | 104 |
| 3.2 Analysis Examples .....                                    | 108 |
| 3.2.1 Patch test.....  | 108 |
| 3.2.2 Square based conic simulation .....                      | 112 |
| 3.2.3 Hypar simulation.....                                    | 121 |
| 3.3 Patterning.....  | 129 |
| 3.4 Summary and Conclusions.....                               | 136 |

## Figures

|   |    |
|---|----|
| Figure 3-1: Determination of $t^*$ [25].....                              | 83 |
| Figure 3-2: Mapping between global and natural co-ordinates.....          | 85 |
| Figure 3-3: Natural (area) co-ordinates $\xi_i$ for a triangle [34] ..... | 86 |
| Figure 3-4: Approximate Gauss point positions within element.....         | 96 |

|   |     |
|---|-----|
| Figure 3-5: Definition of linear cable element.....                           | 97  |
| Figure 3-6: Conic mesh with panels, seams (red) and centre lines (blue) ..... | 99  |
| Figure 3-7: (a) Element normals, $Z_i$ , and surface normals, $\vec{N}$       |     |
| (b) Surface basis vectors.....  | 101 |
| Figure 3-8: Summary of Dynamic Relaxation algorithm.....                      | 103 |
| Figure 3-9: Wrinkling procedure summary .....                                 | 107 |
| Figure 3-10: 32 element patch for validation.....                             | 108 |
| Figure 3-11: Patch test results using $BL = \frac{1}{2}AG$ .....              | 111 |
| Figure 3-12: Square based conic.....  | 112 |
| Figure 3-13: Initial conic mesh.....  | 112 |
| Figure 3-14: Form found conic mesh.....                                       | 113 |
| Figure 3-15: Initial refined conic mesh .....                                 | 114 |
| Figure 3-16: Conic prestress results .....                                    | 116 |
| Figure 3-17: Conic result snow load.....                                      | 117 |
| Figure 3-18: Refined conic result snow load .....                             | 118 |
| Figure 3-19: Conic results wind load .....                                    | 119 |
| Figure 3-20: Conic results wind load .....                                    | 120 |
| Figure 3-21: Hypar.....   | 121 |
| Figure 3-22: Initial hypar mesh.....  | 121 |
| Figure 3-23: Form found hypar mesh.....                                       | 122 |
| Figure 3-24: Hypar prestress results .....                                    | 125 |
| Figure 3-25: Hypar result snow load .....                                     | 126 |
| Figure 3-26: Hypar result wind load .....                                     | 127 |
| Figure 3-27: Hypar result wind load with wrinkling procedure.....             | 128 |
| Figure 3-28: Patterning of a hypar.....                                       | 132 |
| Figure 3-29: Patterning analysis output.....                                  | 133 |

## Tables

|  |     |
|--|-----|
| Table 3-1: Gauss point area co-ordinates and weights.....  | 95  |
| Table 3-2: Analysis results using $BL = AG$ and updated nodal configuration for residual force calculation alongside expected results. ....            | 110 |
| Table 3-3: Analysis results using $BL = \frac{1}{2}AG$ and updated nodal configuration for residual force calculation alongside expected results. .... | 110 |
| Table 3-4: Analysis results using $BL = \frac{1}{2}AG$ and initial nodal configuration for residual force calculation alongside expected results. .... | 110 |

|  |     |
|--|-----|
| Table 3-5: Conic material properties .....             | 114 |
| Table 3-6: Conic results summary.....                  | 115 |
| Table 3-7: Hypar material properties.....              | 123 |
| Table 3-8: Hypar analysis results summary.....         | 123 |
| Table 3-9: Initial and patterned panel comparison..... | 134 |
| Table 3-10: Seam compatibility.....                    | 135 |



## Chapter 3. Finite Element Formulation

The material models developed in the course of this PhD. are implemented in Fortran finite element code developed from code initially proposed by Zhang [31]. Two element formulations were presented by Zhang. The first was a three node constant strain triangle (CST) element which includes higher order terms. The second was a six node linear strain triangle (LST) element which utilised curvilinear co-ordinates in-order to define the local coordinate system.

The six node LST formulation was selected for this work due to the advantages demonstrated by the higher order element. However, difficulties were presented by the formulation, and more critically the existing Fortran code, with regard to aligning the local coordinate system with the local material direction. Due to the highly orthotropic nature of architectural fabric material response this is highly influential in the accurate simulation of a fabric structures. Zhang concluded that the inclusion of curvature terms in the numerical representation of membrane surface lead to more accurate calculation of strain from displacement. However, the inclusion of curvature terms may lead to discontinuities in the simulated membrane at the nodes producing a quilted type pattern. Therefore, the formulation for the LST element was replaced by one adapted from work presented by Gosling on the development of an isoparametric eight node quadratic element [30].

It has been demonstrated that neural network material models may be implemented within commercially available finite element software packages. However, the development of custom code enables a higher level of control and the implementation of analytical sensitivity and reliability analysis. It is also the aim of this project to develop a robust computational mechanics framework that accurately describes both material and structural behaviour. To this end an element developed specifically for the representation of architectural fabric is required.

### 3.1 Element Equations

In the case of the analysis of thin elastic continua the three terms of stress and corresponding strain terms describing plane stress are sufficient to represent an element's behaviour. Therefore the vector of element body stresses,  $\{\sigma\}$ , is written as

$$\{\sigma\}^T = \{\sigma_x, \sigma_y, \tau_{xy}\} \quad (3.1)$$

Where  $\sigma_x$  and  $\sigma_y$  are principle direct stresses and  $\tau_{xy}$  is the principle shear strain. The corresponding strain vector is

$$\{\varepsilon\}^T = \{\varepsilon_x, \varepsilon_y, \gamma_{xy}\} \quad (3.2)$$

As demonstrated in [30, 31, 34] the element stiffness matrices and element load vectors for a discretised system may be derived from the potential energy expression for an individual element of that system.

$$\Pi_{pe} = \int_V \left( \frac{1}{2} \{\varepsilon\}^T [E] \{\varepsilon\} - \{\varepsilon\}^T [E] \{\varepsilon_0\} + \{\varepsilon\}^T \{\sigma_0\} \right) dV - \int_V \{\delta\}^T \{F\} dV - \int_S \{\delta\}^T \{\Phi\} dS \quad (3.3)$$

The first integral containing system strain,  $\{\varepsilon\}$ , initial strain,  $\{\varepsilon_0\}$ , initial stress,  $\{\sigma_0\}$ , and  $[E]$ , an elastic matrix containing appropriate material properties, represents strain energy per unit volume. The second and third integral containing body forces,  $\{F\}$ , and surface tractions,  $\{\Phi\}$ , represent work done by  $\{F\}$  and  $\{\Phi\}$  as the body deforms [34].

As discussed in Chapter 2 a plane stress frame work is typically used to represent the architectural fabric strain-stress relationship within a finite element formulation. The plane stress material stiffness matrix  $[E]$  which relates current strain to stress is given by,

$$\{\sigma\} = [E]\{\varepsilon\} = \begin{bmatrix} \frac{E_w}{(1-v_{wf}v_{fw})} & \frac{E_w v_{fw}}{(1-v_{wf}v_{fw})} & 0 \\ \frac{E_f v_{wf}}{(1-v_{wf}v_{fw})} & \frac{E_f}{(1-v_{wf}v_{fw})} & 0 \\ 0 & 0 & G_{wf} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \quad (3.4)$$

where  $E_w$  and  $E_f$  are values of Young's modulus aligned with the local warp and fill directions respectively,  $v_{fw}$  and  $v_{wf}$  are values of Poisson's ratio and  $G_{wf}$  is the shear modulus. This form of material model will be used in the initial development of the finite element code but will later be replaced by a neural network material model.

### 3.1.1 Application of the assumed displacement field [25]

The adoption of potential energy infers that the displacements are the primary unknown. As expanded upon later in Section 3.1.4 the displacements at any arbitrary point within a body may be defined in terms of a normalised interpolation scheme such that

$$\{\delta\} = \{u, v, w\} \approx [N]\{\delta\}_i \quad (3.5)$$

Using the definition of displacement in (3.5) the strain terms of (3.2) may be expressed in matrix form as

$$\{\varepsilon\} = [B]\{\delta\}_i \quad (3.6)$$

Where  $[B]$  contains differentials of the assumed displacement field  $[N]\{\delta\}_i$ .

Assuming that body forces and surface traction forces are zero and substituting (3.6) into (3.3) yields the following expression for the potential energy of an individual element

$$\begin{aligned} \Pi_{pe} = & \\ & \frac{1}{2}\{\delta\}_i^T \left[ \int_V [B]^T [E] [B] dV \right] \{\delta\}_i - \{\delta\}_i^T \int_V [B]^T [E] [\varepsilon_0] dV + \\ & \{\delta\}_i^T \int_V [B]^T [\sigma_0] dV \end{aligned} \quad (3.7)$$

The total potential energies of the individual elements are summed to find the total potential energy of the discretised system. Therefore, the total potential energy of a system made up of  $m$  elements is given by

$$\Pi_{ps} = \left[ \sum_{i=1}^{i=m} \Pi_{pe} i \right] - \{D\}^T \{P\} \quad (3.8)$$

The second term of (3.8) accounts for work done by concentrated forces,  $\{P\}$ , applied to the system. The vector  $\{D\}$  contains the combined nodal displacements of the entire system obtain by the vector summation of the corresponding terms of the element displacement vectors,  $\{\delta\}_i$ . The element matrices contained within the square bracket may be combined in the same way yielding

$$\begin{aligned} \Pi_{ps} = & \frac{1}{2}\{D\}^T \left[ \sum_{i=1}^{i=m} \left[ \int_V [B]^T [E] [B] dV \right]_i \right] \{D\} - \{D\}^T \sum_{i=1}^{i=m} \left[ \int_V [B]^T [E] [\varepsilon_0] dV \right]_i + \\ & \{D\}^T \sum_{i=1}^{i=m} \left[ \int_V [B]^T [\sigma_0] dV \right]_i - \{D\}^T \{P\} \end{aligned} \quad (3.9)$$

In order to establish an expression for the equilibrium of the system (3.9) is differentiated with respect to the nodal degrees of freedom  $1 \rightarrow n$  and set to zero. This is written as,

$$\frac{\partial \Pi_{ps}}{\partial D_1} = 0, \frac{\partial \Pi_{ps}}{\partial D_2} = 0, \dots, \frac{\partial \Pi_{ps}}{\partial D_n} = 0 \quad (3.10)$$

The expression of equilibrium for the system is therefore found by applying (3.10) to (3.9).

$$\begin{aligned} \frac{\partial \Pi_{ps}}{\partial D_{1 \rightarrow n}} = \{R\} &= \sum_{i=1}^{i=m} \left[ \int_V [B]^T [E] [B] dV \right]_i \{D\} - \sum_{i=1}^{i=m} \left[ \int_V [B]^T [E] [\varepsilon_0] dV \right]_i + \\ &\quad \sum_{i=1}^{i=m} \left[ \int_V [B]^T [\sigma_0] dV \right]_i + \{P\} = \sum_{i=1}^{i=m} \left[ \int_V [B]^T [\sigma] dV \right]_i - \{P\} = 0 \end{aligned} \quad (3.11)$$

Where

$$[\sigma] = [E]([\varepsilon] - [\varepsilon_0]) + [\sigma_0] = [E]([B]\{D\} - [\varepsilon_0]) + [\sigma_0] \quad (3.12)$$

Any non-zero values given by (3.11) constitute the out of balance or residual force vector,  $\{R\}$ .

The second derivative of (3.9) yields the stiffness matrix,  $[K_T]$  given by,

$$\frac{\partial^2 \Pi_{ps}}{\partial D_{1 \rightarrow n}^2} = [K_T] = \sum_{i=1}^{i=m} \left[ \int_V d[B]^T [\sigma] dV \right]_i + \sum_{i=1}^{i=m} \left[ \int_V [B]^T d[\sigma] dV \right]_i \quad (3.13)$$

Taking the definition for stress in (3.12) the stress derivative is given by

$$d[\sigma] = [E]d[\varepsilon] = [E][B] \quad (3.14)$$

Using (3.14) with (3.13) the elastic stiffness matrix,  $K_E$ , is given by

$$[K_E] = \sum_{i=1}^{i=m} \left[ \int_V [B]^T [E] [B] dV \right]_i \quad (3.15)$$

The remaining term in (3.13) is the geometric stiffness matrix,  $K_\sigma$ , given by

$$[K_\sigma] = \sum_{i=1}^{i=m} \left[ \int_V d[B]^T [\sigma] dV \right]_i \quad (3.16)$$

The geometric stiffness matrix, critical for form finding when the material stiffness defined by  $[E]$  is set to zero, will be explicitly defined later in Section 3.1.8.

### 3.1.2 Dynamic Relaxation solution algorithm

Where an analysis is geometrically non-linear an incremental solution algorithm is required in order to solve the static equilibrium equation (3.11). During the process of dynamic relaxation all nodes undergo artificially damped pseudo oscillations controlled by the element stiffness and nodal out of balance forces. The system oscillates about the equilibrium position coming to rest only when the kinetic and potential energy of the entire system has dissipated. Thus the static equilibrium position is found. Dynamic relaxation was first applied to the form finding and analysis of tension structures in the late 1960s.

Both Zhang, [26], and Gosling, [25], employed dynamic relaxation as their solution algorithm. Zhang also implemented the commonly used Newton-Raphson method with his LST element to enable comparison. It was found that the Dynamic Relaxation algorithm was considerably less computationally expensive than the Newton-Raphson method when

implemented in conjunction with the higher order LST element. Dynamic Relaxation does not use the complete system stiffness matrix and therefore also offers considerable advantages when implementing a Neural Network material model, see Chapter 5.

The following algorithm equations included for completeness, they are reproduced from [30]. The D'Alembert principle is used to describe motion of the system,

$$P_{pq} = M_{pq} \ddot{\delta}_{pq} + C \dot{\delta}_{pq} + K_{pq} \delta_{pq} \quad (3.17)$$

or

$$P_{pq} - K_{pq} \delta_{pq} = M_{pq} \ddot{\delta}_{pq} + C \dot{\delta}_{pq} \quad (3.18)$$

and

$$R_{pq} = M_{pq} \ddot{\delta}_{pq} + C \dot{\delta}_{pq} \quad (3.19)$$

The subscripts  $pq$  refer to the  $p^{th}$  node in the  $q^{th}$  direction ( $q = 1 \rightarrow 3$  corresponding to the global axis directions  $\{x, y, z\}$ ). The external load vector,  $P_{pq}$ , includes terms representing the effects of surface pressures and initial strain. The nodal stiffness,  $K_{pq}$ , is taken from the diagonal terms of the element stiffness matrix.  $R_{pq}$  denotes the nodal out of balance forces and  $M_{pq}$  the fictitious nodal masses. Nodal acceleration, velocity and displacement is denoted by  $\ddot{\delta}_{pq}$ ,  $\dot{\delta}_{pq}$  and  $\delta_{pq}$  respectively.

Kinetic damping has been shown to be more stable and efficient than the viscous damping approach in (3.17). When using kinetic damping the un-damped oscillation of the system is monitored and when a local kinetic energy peak is reached all velocity components are reset to zero. The geometry is then updated and the process repeated until the kinetic and potential energy is minimised and static equilibrium is achieved. Using this approach (3.19) becomes

$$P_{pq} - K_{pq} \delta_{pq} = R_{pq} = M_{pq} \ddot{\delta}_{pq} \quad (3.20)$$

Using finite difference the acceleration,  $\ddot{\delta}_{pq}$ , is derived form change in velocity over the time increment  $\Delta t$ .

$$\ddot{\delta}_{pq} = \frac{\dot{\delta}_{pq}^{t+\frac{\Delta t}{2}} - \dot{\delta}_{pq}^{t-\frac{\Delta t}{2}}}{\Delta t} \quad (3.21)$$

A recurrent equation for current nodal velocity,  $\dot{\delta}_{pq}^{t+\frac{\Delta t}{2}}$ , is found by substitution of (3.21) into (3.20) and rearranging to give,

$$\dot{\delta}_{pq}^{t+\frac{\Delta t}{2}} = \dot{\delta}_{pq}^{t-\frac{\Delta t}{2}} + R_{pq}^t \frac{\Delta t}{M_{pq}} \quad (3.22)$$

To ensure numerical stability the following expression for  $M_{pq}$  is suggested [30, 25]

$$M_{pq} \geq \frac{K_{pq}\Delta t^2}{2} \quad (3.23)$$

Substitution of (3.23) into (3.22) yields,

$$\dot{\delta}_{pq}^{t+\frac{\Delta t}{2}} = \dot{\delta}_{pq}^{t-\frac{\Delta t}{2}} + R_{pq}^t \left[ \frac{2}{\Delta t \cdot K_{pq}} \right] \quad (3.24)$$

The current nodal displacement, at time  $t + \frac{\Delta t}{2}$ , is calculated from the current

velocity,  $\dot{\delta}_{pq}^{t+\frac{\Delta t}{2}}$ , using

$$\delta_{pq}^{t+\frac{\delta t}{2}} = \dot{\delta}_{pq}^{t+\frac{\Delta t}{2}} \Delta t \quad (3.25)$$

In order to monitor the kinetic energy it must be calculated at each pseudo oscillation

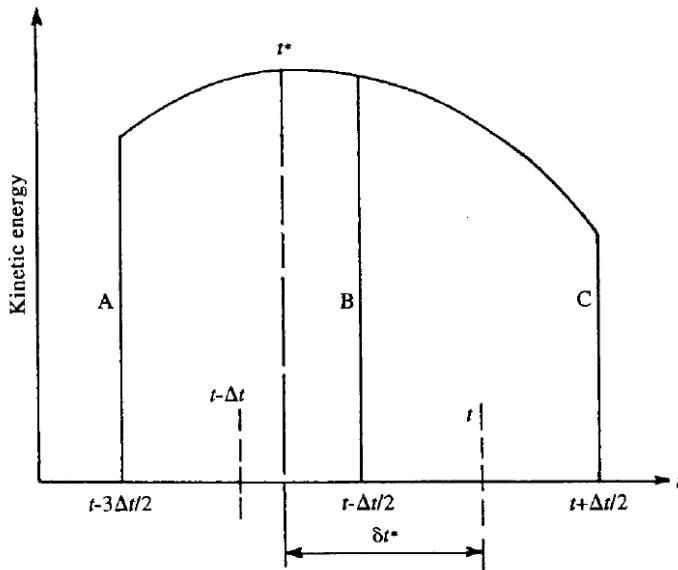
defined by (3.24) and (3.25). Current and previous kinetic energy,  $U_k^{t+\frac{\Delta t}{2}}$  and  $U_k^{t-\frac{\Delta t}{2}}$  are given by,

$$U_k^{t+\frac{\Delta t}{2}} = \frac{1}{2} \sum_{p=1}^{p=N} \sum_{q=1}^{q=3} M_{pq} \left( \dot{\delta}_{pq}^{t+\frac{\Delta t}{2}} \right)^2 \quad (3.26)$$

And,

$$U_k^{t-\frac{\Delta t}{2}} = \frac{1}{2} \sum_{p=1}^{p=N} \sum_{q=1}^{q=3} M_{pq} \left( \dot{\delta}_{pq}^{t-\frac{\Delta t}{2}} \right)^2 \quad (3.27)$$

A kinetic energy peak is detected when  $U_k^{t+\frac{\Delta t}{2}}$  is less than  $U_k^{t-\frac{\Delta t}{2}}$  at time  $t + \frac{\Delta t}{2}$ . However, the true kinetic energy peak will have occurred sometime before  $t + \frac{\Delta t}{2}$ .



**Figure 3-1: Determination of  $t^*$  [25]**

As demonstrated in Figure 3-1 the approximate time,  $t^*$ , of the true energy peak may be found by fitting a parabola through the current and two previous energy peaks. The improved estimate of time is given by,

$$t^* = t - \delta t^* \quad (3.28)$$

Where,

$$\delta t^* = \Delta t \cdot \frac{B-C}{((B-C)-(A-B))} = \Delta t \cdot \frac{U_k^{t-\frac{\Delta t}{2}} - U_k^{t+\frac{\Delta t}{2}}}{2U_k^{t-\frac{\Delta t}{2}} - U_k^{t+\frac{\Delta t}{2}} - U_k^{t-\frac{3\Delta t}{2}}} = \Delta t \cdot \alpha \quad (3.29)$$

Therefore, an improved estimate of the position of the system at the kinetic energy peak is given by

$$\delta t^* = \dot{\delta}_{pq}^{t+\frac{\Delta t}{2}} - \dot{\delta}_{pq}^{t+\frac{\Delta t}{2}} \cdot \Delta t - \dot{\delta}_{pq}^{t-\frac{\Delta t}{2}} \cdot \delta t^* \quad (3.30)$$

Substitution of (3.24) and (3.29) into (3.30) yields

$$\delta t^* = \dot{\delta}_{pq}^{t+\frac{\Delta t}{2}} - (1 + \alpha) \dot{\delta}_{pq}^{t+\frac{\Delta t}{2}} \cdot \Delta t + \Delta t \cdot \alpha \cdot R_{pq}^t \left[ \frac{2}{\delta t \cdot K_{pq}} \right] \quad (3.31)$$

Through recurrent use of (3.24) and (3.25) the condition of static equilibrium in the system may be found.

### 3.1.3 Initial and updated nodal configurations

The current stress is induced by the current strain. As shown in (3.6) strain is the product of the current displacement and shape function derivatives contained within the B-matrix. For the determination of current strain the element B-matrix is derived from the initial

element nodal configuration,  $\{C_o\}_i$ , and the current element nodal displacements,  $\delta^t$ . This will be referred to as the initial element B-matrix,  ${}_0B$ . Therefore (3.6) becomes,

$$\{\varepsilon\} = [{}_0B]\{\delta^t\}_i \quad (3.32)$$

The residual force vector is always calculated at the updated nodal configuration calculated at each energy peak. The updated nodal co-ordinates at time  $t$ ,  $\{C_t\}_i$ , of an element may be defined by the dot product of the initial nodal co-ordinates and the element global nodal displacements.

$$\begin{aligned} \{C_t\}_i &= \{C_o\}_i + \{\delta^t\}_i = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \\ x_5 & y_5 & z_5 \\ x_6 & y_6 & z_6 \end{bmatrix} + \begin{bmatrix} u_1 & v_1 & w_1 \\ u_2 & v_2 & w_2 \\ u_3 & v_3 & w_3 \\ u_4 & v_4 & w_4 \\ u_5 & v_5 & w_5 \\ u_6 & v_6 & w_6 \end{bmatrix} = \\ &\begin{bmatrix} x_1 + u_1 & y_1 + v_1 & z_1 + w_1 \\ x_2 + u_2 & y_2 + v_2 & z_2 + w_2 \\ x_3 + u_3 & y_3 + v_3 & z_3 + w_3 \\ x_4 + u_4 & y_4 + v_4 & z_4 + w_4 \\ x_5 + u_5 & y_5 + v_5 & z_5 + w_5 \\ x_6 + u_6 & y_6 + v_6 & z_6 + w_6 \end{bmatrix} \end{aligned} \quad (3.33)$$

For the determination of current residual force the element B-matrix is derived from the updated element nodal configuration. As the nodal co-ordinates have been updated the nodal displacements become zero. This will be referred to as the updated element B-matrix,  ${}_tB$ . Using these element B-matrix definitions the residual force equation (3.11) becomes,

$$\{R\} = \sum_{i=1}^{i=m} \left[ \int_V [{}_tB]^T [\sigma] dV \right]_i - \{P\} \quad (3.34)$$

Therefore, the elastic stiffness matrix (3.15) becomes,

$$[K_E] = \sum_{i=1}^{i=m} \left[ \int_V [{}_tB]^T [E] [{}_0B] dV \right]_i \quad (3.35)$$

The geometric stiffness matrix (3.16) becomes

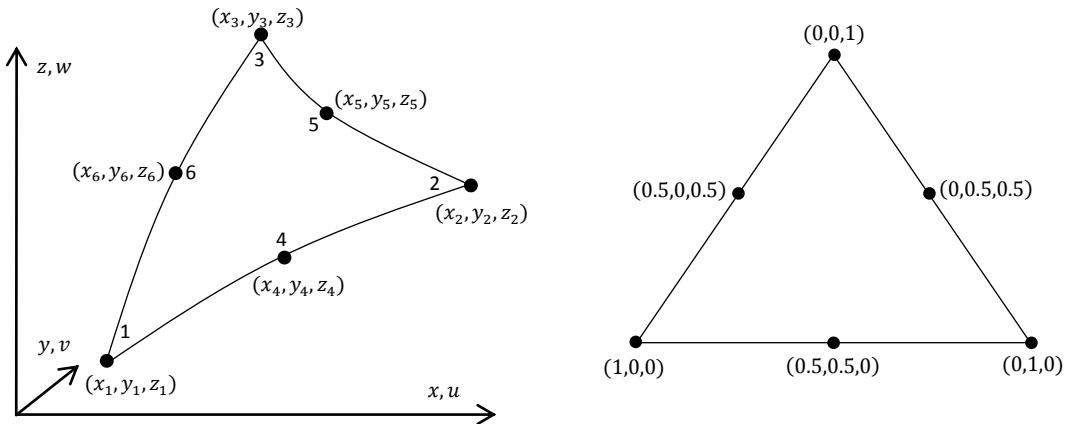
$$[K_\sigma] = \sum_{i=1}^{i=m} \left[ \int_V d[{}_tB]^T [\sigma] dV \right]_i \quad (3.36)$$

### 3.1.4 Natural co-ordinate system and shape functions [34]

An isoparametric element utilises shape functions based on a single natural coordinate system to interpolate both the global element geometry,  $\{x\}_i$ , and displacements,  $\{\delta\}_i$ . The co-ordinates or displacements at any point within the element may therefore be found by (3.37).

$$x = \sum_{i=1}^{i=6} N_i x_i, \quad \delta = \sum_{i=1}^{i=6} N_i \delta_i \quad (3.37)$$

The 6 element nodes are labelled  $i = 1 \rightarrow 6$ . The shape function,  $\{N_i\}$ , is calculated at the point of interest defined in terms of the natural co-ordinates. The displacement components  $u$ ,  $v$  and  $w$  are represented by  $\delta_i$ , similar expressions may be derived for the global  $y$  and  $z$  co-ordinates.



**Figure 3-2: Mapping between global and natural co-ordinates**

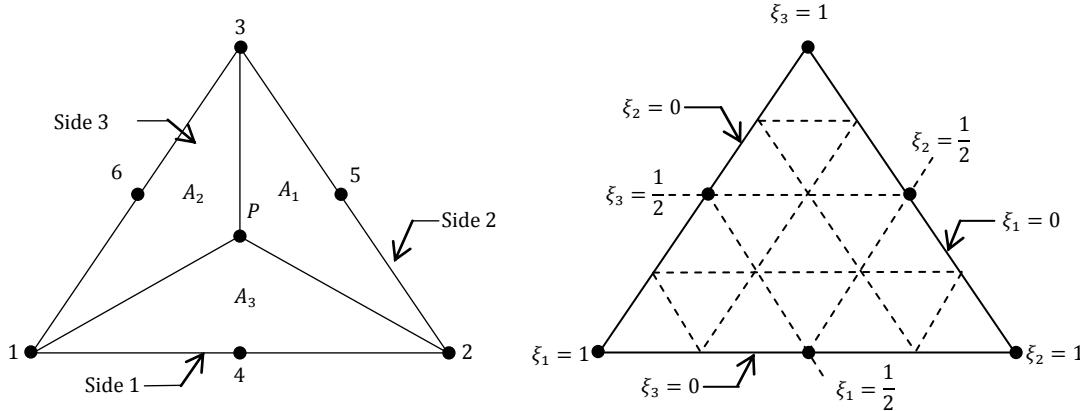
An isoparametric formulation is selected as it facilitates the generation of non-uniform elements with curved boundaries. This is enabled through the mapping of the irregular element in the global system to a regular element in the natural coordinate system as demonstrated in Figure 3-2.

In the case of a triangle the natural co-ordinates are provided by area co-ordinates as illustrated in Figure 3-3.

Point  $P$  divides a triangle into three subareas  $A_1$ ,  $A_2$  and  $A_3$ . Area co-ordinates are defined by the ratios between total area,  $A$ , and subarea as shown in (3.38) with the constraints in (3.39).

$$\xi_1 = \frac{A_1}{A} \quad \xi_2 = \frac{A_2}{A} \quad \xi_3 = \frac{A_3}{A} \quad (3.38)$$

$$A_1 + A_2 + A_3 = A \text{ therefore } \xi_1 + \xi_2 + \xi_3 = 1 \quad (3.39)$$



**Figure 3-3: Natural (area) co-ordinates  $\xi_i$  for a triangle [34]**

The shape functions of a 6 node quadratic triangular element expressed in terms of area coordinate are given in (3.40) or in vector form (3.41).

$$\begin{aligned} N_1 &= \xi_1(2\xi_1 - 1) & N_2 &= \xi_2(2\xi_2 - 1) & N_3 &= \xi_3(2\xi_3 - 1) \\ N_4 &= 4\xi_1\xi_2 & N_5 &= 4\xi_2\xi_3 & N_6 &= 4\xi_3\xi_1 \end{aligned} \quad (3.40)$$

$$N_i = \{\xi_1(2\xi_1 - 1) \quad \xi_2(2\xi_2 - 1) \quad \xi_3(2\xi_3 - 1) \quad 4\xi_1\xi_2 \quad 4\xi_2\xi_3 \quad 4\xi_3\xi_1\} \quad (3.41)$$

Due to the constraints shown in (3.39) the three area co-ordinates are not independent and therefore may be redefined in terms of only  $\xi_1$  and  $\xi_2$  (3.42).

$$\xi_1 = \xi \quad \xi_2 = \eta \quad \xi_3 = 1 - \xi - \eta \quad (3.42)$$

The derivatives of the shape function vector (3.41) with respect to the area co-ordinates, required for the definition of the B matrix and definition of local direction base vectors, are found using the chain rule and simplified by the partial derivatives of (3.42) with respect to  $\xi$  and  $\eta$ .

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial \xi_1} \frac{\partial \xi_1}{\partial \xi} + \frac{\partial N_i}{\partial \xi_2} \frac{\partial \xi_2}{\partial \xi} + \frac{\partial N_i}{\partial \xi_3} \frac{\partial \xi_3}{\partial \xi} = \frac{\partial N_i}{\partial \xi_1} - \frac{\partial N_i}{\partial \xi_3} \quad (3.43)$$

$$\frac{\partial N_i}{\partial \eta} = \frac{\partial N_i}{\partial \xi_1} \frac{\partial \xi_1}{\partial \eta} + \frac{\partial N_i}{\partial \xi_2} \frac{\partial \xi_2}{\partial \eta} + \frac{\partial N_i}{\partial \xi_3} \frac{\partial \xi_3}{\partial \eta} = \frac{\partial N_i}{\partial \xi_2} - \frac{\partial N_i}{\partial \xi_3} \quad (3.44)$$

Partially differentiating (3.41) with respect to the area co-ordinates  $\xi_1$ ,  $\xi_2$  and  $\xi_3$  gives,

$$\frac{\partial N_i}{\partial \xi_1} = [4\xi_1 - 1 \quad 0 \quad 0 \quad 4\xi_2 \quad 0 \quad 4\xi_3] \quad (3.45)$$

$$\frac{\partial N_i}{\partial \xi_2} = [0 \quad 4\xi_2 - 1 \quad 0 \quad 4\xi_1 \quad 4\xi_3 \quad 0] \quad (3.46)$$

$$\frac{\partial N_i}{\partial \xi_3} = [0 \quad 0 \quad 4\xi_3 - 1 \quad 0 \quad 4\xi_2 \quad 4\xi_1] \quad (3.47)$$

Substituting (3.45)-(3.47) in to (3.43) and (3.44) gives,

$$\frac{\partial N_i}{\partial \xi} = [4\xi_1 - 1 \quad 0 \quad -4\xi_3 + 1 \quad 4\xi_2 \quad -4\xi_2 \quad 4(\xi_3 - \xi_1)] \quad (3.48)$$

$$\frac{\partial N_i}{\partial \eta} = [0 \quad 4\xi_2 - 1 \quad -4\xi_3 + 1 \quad 4\xi_1 \quad 4(\xi_3 - \xi_2) \quad -4\xi_1] \quad (3.49)$$

For convenience these equations may be arranged into the 6 by 2 matrix  $[D_{N_{\xi,\eta}}]$ ,

$$D_{N_{\xi,\eta}} = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} & \frac{\partial N_4}{\partial \xi} & \frac{\partial N_5}{\partial \xi} & \frac{\partial N_6}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta} & \frac{\partial N_5}{\partial \eta} & \frac{\partial N_6}{\partial \eta} \end{bmatrix} \\ = \begin{bmatrix} 4\xi_1 - 1 & 0 & -4\xi_3 + 1 & 4\xi_2 & -4\xi_2 & 4(\xi_3 - \xi_1) \\ 0 & 4\xi_2 - 1 & -4\xi_3 + 1 & 4\xi_1 & 4(\xi_3 - \xi_2) & -4\xi_1 \end{bmatrix} \quad (3.50)$$

### 3.1.5 Element local co-ordinate system [30]

The membrane transformation matrix,  $[T^m]$ , is defined in terms of directional cosines between the global and local coordinate systems.

$$T^m = \begin{bmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix} \quad (3.51)$$

The directional cosines contained in  $T$  are derived from the basis vectors aligned with the natural coordinate system along with the material direction. The natural coordinate basis vectors are the product of the derivatives of the shape functions given in (3.48) and (3.49) and the global element nodal co-ordinates.

As demonstrated in [30] base vectors  $\vec{\xi}$  aligned natural coordinate direction may be defined by,

$$\vec{\xi} = \left[ \frac{\partial x}{\partial \xi} \hat{i} + \frac{\partial y}{\partial \xi} \hat{j} + \frac{\partial z}{\partial \xi} \hat{k} \right] \quad (3.52)$$

The basis vectors for the  $\xi$  and  $\eta$  natural coordinate directions may be defined in terms of the shape functions by substituting (3.48) into (3.52) in the same form as (3.37).

$$\vec{\xi} = \left[ \sum_{i=1}^{i=6} \frac{\partial N_i}{\partial \xi} x_i \hat{i} + \sum_{i=1}^{i=6} \frac{\partial N_i}{\partial \xi} y_i \hat{j} + \sum_{i=1}^{i=6} \frac{\partial N_i}{\partial \xi} z_i \hat{k} \right] \quad (3.53)$$

$$\vec{\eta} = \left[ \sum_{i=1}^{i=6} \frac{\partial N_i}{\partial \eta} x_i \hat{i} + \sum_{i=1}^{i=6} \frac{\partial N_i}{\partial \eta} y_i \hat{j} + \sum_{i=1}^{i=6} \frac{\partial N_i}{\partial \eta} z_i \hat{k} \right] \quad (3.54)$$

Due to the curvature of the element in three dimensions these two base vectors may not be orthogonal. However, they define a plane tangential to the element surface at any

arbitrary point. The cross product of the natural coordinate basis vectors produces the orthogonal vector,  $\vec{Z}$ , aligned the local Z-direction. This vector is normalised to give  $\hat{Z}$ .

$$\vec{Z} = \vec{\xi} \times \vec{\eta} \quad (3.55)$$

$$\hat{Z} = \frac{\vec{Z}}{\|\vec{Z}\|} \text{ or } \hat{Z} = \frac{\vec{\xi} \times \vec{\eta}}{\|\vec{\xi} \times \vec{\eta}\|} = [n_1 \hat{i} + n_2 \hat{j} + n_3 \hat{k}] \quad (3.56)$$

The material direction is defined in terms of an angle,  $\theta$ , referenced to the global  $x$  direction. Using  $\theta$  global material warp and fill direction basis vectors,  $\vec{\theta}_w$  and  $\vec{\theta}_f$ , may be defined in the global  $xy$ -plane ( $z = 0$ ).

$$\vec{\theta}_w = [\cos \theta \hat{i} + \sin \theta \hat{j} + 0 \hat{k}] \text{ and } \vec{\theta}_f = [-\sin \theta \hat{i} + \cos \theta \hat{j} + 0 \hat{k}] \quad (3.57)$$

The cross product of the global fill direction basis vector,  $\vec{\theta}_f$ , and the local Z-direction basis vector,  $\hat{Z}$ , gives the basis vector,  $\hat{X}$ , aligned with the local warp material direction in the local coordinate plane. This is also referred to as the local  $X$ -direction basis vector.

$$\hat{X} = \frac{\vec{\theta}_f \times \vec{Z}}{\|\vec{\theta}_f \times \vec{Z}\|} = [l_1 \hat{i} + l_2 \hat{j} + l_3 \hat{k}] \quad (3.58)$$

The cross product of the local  $X$ -direction basis vector,  $\hat{X}$ , and  $Z$ -direction base vector,  $\hat{Z}$ , completes the local orthogonal coordinate system giving the local  $Y$ -direction basis vector,  $\hat{Y}$ , aligned with the local fill material direction in the local coordinate  $XY$ -plane.

$$\hat{Y} = \frac{\hat{Z} \times \hat{X}}{\|\hat{Z} \times \hat{X}\|} = [m_1 \hat{i} + m_2 \hat{j} + m_3 \hat{k}] \quad (3.59)$$

The local basis vectors transform global co-ordinates  $\{x \ y \ z\}_i$ , forces  $\{f_x \ f_y \ f_z\}_i$  and displacements  $\{u \ v \ w\}_i$  to the local coordinate systems.

### 3.1.6 The Jacobian matrix [30]

The displacements are expressed as functions of the normalised natural co-ordinates. However, strains are required in terms of the element local coordinate system. Therefore, it is required that differentiation with respect to the natural co-ordinates is related to differentiation with respect to the local co-ordinates via a change in variable. The Jacobian accounts for the change in variable between the distorted element expressed in terms of local co-ordinates and the normalised element expressed in terms of natural co-ordinates.

The Jacobian may be derived by taking a general scalar quantity,  $\Phi$ , (for example of displacements in the global  $x$  direction) at a position in the element defined by  $(\xi, \eta)$ . If the position of  $\Phi$  is moved from  $(\xi, \eta)$  to  $(\xi + d\xi, \eta + d\eta)$  the change in  $\Phi$ ,  $d\Phi$ , is given by,

$$d\Phi = \frac{\partial \Phi}{\partial \xi} \cdot d\xi + \frac{\partial \Phi}{\partial \eta} \cdot d\eta \quad (3.60)$$

A change in position of  $d\xi$  implies a shift in position given as  $\vec{\xi} \cdot d\xi$ . Likewise a change in position of  $d\eta$  implies a shift in position gives as  $\vec{\eta} \cdot d\eta$ . Resolving these shifts into the local  $X$  and  $Y$  directions gives,

$$dX_\xi = \vec{\xi} \cdot \hat{X} d\xi \text{ and } dY_\xi = \vec{\xi} \cdot \hat{Y} d\xi \quad (3.61)$$

$$dX_\eta = \vec{\eta} \cdot \hat{X} d\eta \text{ and } dY_\eta = \vec{\eta} \cdot \hat{Y} d\eta \quad (3.62)$$

Rewriting (3.60) in terms of the local coordinate system gives,

$$d\Phi = \frac{\partial \Phi}{\partial X} \cdot dX + \frac{\partial \Phi}{\partial Y} \cdot dY \quad (3.63)$$

where,  $dX = dX_\xi + dX_\eta$  and  $dY = dY_\xi + dY_\eta$

Substitution of (3.61) and (3.62) into (3.63) gives,

$$d\Phi = \left[ \vec{\xi} \cdot \hat{X} \frac{\partial \Phi}{\partial X} + \vec{\xi} \cdot \hat{Y} \frac{\partial \Phi}{\partial Y} \right] d\xi + \left[ \vec{\eta} \cdot \hat{X} \frac{\partial \Phi}{\partial X} + \vec{\eta} \cdot \hat{Y} \frac{\partial \Phi}{\partial Y} \right] d\eta = \frac{\partial \Phi}{\partial \xi} \cdot d\xi + \frac{\partial \Phi}{\partial \eta} \cdot d\eta \quad (3.64)$$

Collection of the common terms in (3.64) leads to the matrix formulation,

$$\begin{bmatrix} \frac{\partial \Phi}{\partial \xi} \\ \frac{\partial \Phi}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \vec{\xi} \cdot \hat{X} & \vec{\xi} \cdot \hat{Y} \\ \vec{\eta} \cdot \hat{X} & \vec{\eta} \cdot \hat{Y} \end{bmatrix} \begin{bmatrix} \frac{\partial \Phi}{\partial X} \\ \frac{\partial \Phi}{\partial Y} \end{bmatrix} \quad (3.65)$$

Or more usefully,

$$\begin{bmatrix} \frac{\partial \Phi}{\partial X} \\ \frac{\partial \Phi}{\partial Y} \end{bmatrix} = \frac{1}{\det[J]} \begin{bmatrix} \vec{\eta} \cdot \hat{Y} & -\vec{\xi} \cdot \hat{Y} \\ -\vec{\eta} \cdot \hat{X} & \vec{\xi} \cdot \hat{X} \end{bmatrix} \begin{bmatrix} \frac{\partial \Phi}{\partial \xi} \\ \frac{\partial \Phi}{\partial \eta} \end{bmatrix} \quad (3.66)$$

$$J = \begin{bmatrix} \vec{\xi} \cdot \hat{X} & \vec{\xi} \cdot \hat{Y} \\ \vec{\eta} \cdot \hat{X} & \vec{\eta} \cdot \hat{Y} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \quad (3.67)$$

The determinant of the Jacobian is constant throughout the element. In the case of a rectangular element the determinant returns the area of the element. Therefore, for a triangular element it returns twice the area.

### 3.1.7 Element B-matrix [30,34]

Strain is defined using the Green-Lagrange definition for large strain (3.68)-(3.70).

$$\varepsilon_x = \frac{\partial U}{\partial X} + \frac{1}{2} \left[ \left( \frac{\partial U}{\partial X} \right)^2 + \left( \frac{\partial V}{\partial X} \right)^2 + \left( \frac{\partial W}{\partial X} \right)^2 \right] \quad (3.68)$$

$$\varepsilon_y = \frac{\partial v}{\partial y} + \frac{1}{2} \left[ \left( \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 + \left( \frac{\partial w}{\partial y} \right)^2 \right] \quad (3.69)$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} + \frac{1}{2} \left[ \frac{\partial u}{\partial x} \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \frac{\partial v}{\partial y} + \frac{\partial w}{\partial x} \frac{\partial w}{\partial y} \right] \quad (3.70)$$

The initial first order terms represent the standard definition of engineering strain,

$$[\varepsilon_0] = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{bmatrix} \quad (3.71)$$

The additional second order terms become significant as strain becomes large,

$$[\varepsilon_L] = \frac{1}{2} \begin{bmatrix} \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial x} \right)^2 + \left( \frac{\partial w}{\partial x} \right)^2 \\ \left( \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2 + \left( \frac{\partial w}{\partial y} \right)^2 \\ \frac{\partial u}{\partial x} \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \frac{\partial v}{\partial y} + \frac{\partial w}{\partial x} \frac{\partial w}{\partial y} \end{bmatrix} \quad (3.72)$$

The B-matrix is the relationship between strain and displacement. The B-matrix consists of two distinct parts. The linear term,  $B_0$ , is independent of current displacements accounts for  $\varepsilon_0$ . The non-linear strain term  $\varepsilon_L$  which is dependent on displacement is accounted for by  $B_L$ . This leads to the following definition,

$$[\varepsilon] = [\varepsilon_0] + [\varepsilon_L] = [B_0 + B_L]\{\delta_i\} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} \quad (3.73)$$

### 3.1.7.1 First order linear terms of the B-matrix, $B_0$ [30,34]

The value of displacement in the local  $X$  direction,  $U$ , is interpolated from the global displacements at each node via the shape functions calculated at the point of interest and transformed into the local system via the unit vector  $\hat{X}$ .

$$U = \hat{X}^T [N]\{\delta_i\} = \hat{X}^T \left[ \sum_{i=1}^{i=6} N_i u_i \hat{i} + \sum_{i=1}^{i=6} N_i v_i \hat{j} + \sum_{i=1}^{i=6} N_i w_i \hat{k} \right] \quad (3.74)$$

The first order linear contribution to the total strain may be derived from the derivative of (3.74) with respect to  $X$ .

$$\frac{\partial U}{\partial X} = \hat{X}^T \frac{\partial [N]}{\partial X} \{\delta_i\} = \hat{X}^T \left[ \sum_{i=1}^{i=6} \frac{\partial N_i}{\partial X} u_i \hat{i} + \sum_{i=1}^{i=6} \frac{\partial N_i}{\partial X} v_i \hat{j} + \sum_{i=1}^{i=6} \frac{\partial N_i}{\partial X} w_i \hat{k} \right] \quad (3.75)$$

In agreement with [30] the additional terms which arise from the chain rule are assumed to be zero and are therefore not included in (1.37).

Substituting the directional cosine form of,  $\hat{X}^T$ , shown in (3.59) into (3.75) yields,

$$\frac{\partial U}{\partial X} = \sum_{i=1}^6 l_1 \frac{\partial N_i}{\partial X} u_i + \sum_{i=1}^6 l_2 \frac{\partial N_i}{\partial X} v_i + \sum_{i=1}^6 l_3 \frac{\partial N_i}{\partial X} w_i \quad (3.76)$$

Similarly it can be shown that,

$$\frac{\partial U}{\partial Y} = \sum_{i=1}^6 m_1 \frac{\partial N_i}{\partial Y} u_i + \sum_{i=1}^6 m_2 \frac{\partial N_i}{\partial Y} v_i + \sum_{i=1}^6 m_3 \frac{\partial N_i}{\partial Y} w_i \quad (3.77)$$

$$\frac{\partial V}{\partial X} = \sum_{i=1}^6 l_1 \frac{\partial N_i}{\partial X} u_i + \sum_{i=1}^6 l_2 \frac{\partial N_i}{\partial X} v_i + \sum_{i=1}^6 l_3 \frac{\partial N_i}{\partial X} w_i \quad (3.78)$$

$$\frac{\partial V}{\partial Y} = \sum_{i=1}^6 m_1 \frac{\partial N_i}{\partial Y} u_i + \sum_{i=1}^6 m_2 \frac{\partial N_i}{\partial Y} v_i + \sum_{i=1}^6 m_3 \frac{\partial N_i}{\partial Y} w_i \quad (3.79)$$

Substituting (3.76)-(3.79) for the first order terms of (3.68)-(3.70) yields the definition of linear strain. By taking out the displacement terms and arranging them into the global displacement array,  $\{\delta_i\}$ , (3.81) leaves the linear 3 by 18 B-matrix in the following form,

$$[B_0] =$$

$$\begin{bmatrix} l_1 \frac{\partial N_1}{\partial X} & l_2 \frac{\partial N_1}{\partial X} & l_3 \frac{\partial N_1}{\partial X} & \dots & l_3 \frac{\partial N_6}{\partial X} \\ m_1 \frac{\partial N_1}{\partial Y} & m_2 \frac{\partial N_1}{\partial Y} & m_3 \frac{\partial N_1}{\partial Y} & \dots & m_3 \frac{\partial N_6}{\partial Y} \\ l_1 \frac{\partial N_1}{\partial Y} + m_1 \frac{\partial N_1}{\partial X} & l_2 \frac{\partial N_1}{\partial Y} + m_2 \frac{\partial N_1}{\partial X} & l_3 \frac{\partial N_1}{\partial Y} + m_3 \frac{\partial N_1}{\partial X} & \dots & l_3 \frac{\partial N_6}{\partial Y} + m_3 \frac{\partial N_6}{\partial X} \end{bmatrix} \quad (3.80)$$

$$\{\delta_i\}^T = \{u_1 \ v_1 \ w_1 \ \dots \ u_6 \ v_6 \ w_6\} \quad (3.81)$$

As shown in Section 3.1.6 the inverse of the Jacobian,  $J^{-1}$ , relates derivatives of a scalar quantity with respect to the natural co-ordinates  $(\xi, \eta)$  to derivatives with respect to the local coordinate system  $(X, Y)$ . Therefore, the derivatives of the shape functions,  $[N_i]$ , with respect to the local co-ordinates may be found by matrix multiplication of the inverse of the Jacobian (3.66) with the matrix of shape function derivatives with respect to natural co-ordinates (3.50),

$$D_{N_{X,Y}} = [\Gamma] \begin{bmatrix} D_{N_{\xi,\eta}} \end{bmatrix} \quad (3.82)$$

where,

$$[\Gamma] = \frac{1}{\det[J]} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{21} & \Gamma_{22} \end{bmatrix} \quad (3.83)$$

Shown explicitly (3.82) becomes,

$$\begin{aligned} D_{N_{X,Y}} &= \begin{bmatrix} \Gamma_{11} \frac{\partial N_1}{\partial \xi} + \Gamma_{12} \frac{\partial N_1}{\partial \eta} & \dots & \dots & \dots & \dots & \Gamma_{11} \frac{\partial N_6}{\partial \xi} + \Gamma_{12} \frac{\partial N_6}{\partial \eta} \\ \Gamma_{21} \frac{\partial N_1}{\partial \xi} + \Gamma_{22} \frac{\partial N_1}{\partial \eta} & \dots & \dots & \dots & \dots & \Gamma_{21} \frac{\partial N_6}{\partial \xi} + \Gamma_{22} \frac{\partial N_6}{\partial \eta} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial N_1}{\partial X} & \frac{\partial N_2}{\partial X} & \frac{\partial N_3}{\partial X} & \frac{\partial N_4}{\partial X} & \frac{\partial N_5}{\partial X} & \frac{\partial N_6}{\partial X} \\ \frac{\partial N_1}{\partial Y} & \frac{\partial N_2}{\partial Y} & \frac{\partial N_3}{\partial Y} & \frac{\partial N_4}{\partial Y} & \frac{\partial N_5}{\partial Y} & \frac{\partial N_6}{\partial Y} \end{bmatrix} \end{aligned} \quad (3.84)$$

### 3.1.7.2 Second order displacement dependent terms of the B-matrix, $B_L$

The following derivation of the displacement dependant B-matrix is based on the derivation presented by Zienkiewicz [122] and is used by Gosling [30].

The second order strain terms defined in (3.72) may be rearranged into the form,

$$[\varepsilon_L] = \frac{1}{2} \begin{bmatrix} \{\Delta_X\}^T & \{0\} \\ \{0\} & \{\Delta_Y\}^T \\ \{\Delta_Y\}^T & \{\Delta_X\}^T \end{bmatrix} \begin{bmatrix} \{\Delta_X\} \\ \{\Delta_Y\} \end{bmatrix} = \frac{1}{2} [A][\Delta] \quad (3.85)$$

where,

$$\{\Delta_X\}^T = \left[ \frac{\partial U}{\partial X} \quad \frac{\partial V}{\partial X} \quad \frac{\partial W}{\partial X} \right] \text{ and } \{\Delta_Y\}^T = \left[ \frac{\partial U}{\partial Y} \quad \frac{\partial V}{\partial Y} \quad \frac{\partial W}{\partial Y} \right] \quad (3.86)$$

The 6 by 18 G-matrix relates local displacement derivatives to global nodal displacements in the form,

$$[\Delta] = \begin{bmatrix} \{\Delta_X\} \\ \{\Delta_Y\} \end{bmatrix} = [G]\{\delta\}_i = \begin{bmatrix} \frac{\partial U}{\partial X} \\ \frac{\partial V}{\partial X} \\ \frac{\partial W}{\partial X} \\ \frac{\partial X}{\partial U} \\ \frac{\partial X}{\partial V} \\ \frac{\partial X}{\partial W} \\ \frac{\partial Y}{\partial U} \\ \frac{\partial Y}{\partial V} \\ \frac{\partial Y}{\partial W} \end{bmatrix} \quad (3.87)$$

and is assembled in a similar fashion to (3.80),

$$[G] = \begin{bmatrix} l_1 \frac{\partial N_1}{\partial X} & l_2 \frac{\partial N_1}{\partial X} & l_3 \frac{\partial N_1}{\partial X} & \dots & l_3 \frac{\partial N_6}{\partial X} \\ m_1 \frac{\partial N_1}{\partial X} & m_2 \frac{\partial N_1}{\partial X} & m_3 \frac{\partial N_1}{\partial X} & \dots & m_3 \frac{\partial N_6}{\partial X} \\ n_1 \frac{\partial N_1}{\partial X} & n_2 \frac{\partial N_1}{\partial X} & n_3 \frac{\partial N_1}{\partial X} & \dots & n_3 \frac{\partial N_6}{\partial X} \\ l_1 \frac{\partial N_1}{\partial Y} & l_2 \frac{\partial N_1}{\partial Y} & l_3 \frac{\partial N_1}{\partial Y} & \dots & l_3 \frac{\partial N_6}{\partial Y} \\ m_1 \frac{\partial N_1}{\partial Y} & m_2 \frac{\partial N_1}{\partial Y} & m_3 \frac{\partial N_1}{\partial Y} & \dots & m_3 \frac{\partial N_6}{\partial Y} \\ n_1 \frac{\partial N_1}{\partial Y} & n_2 \frac{\partial N_1}{\partial Y} & n_3 \frac{\partial N_1}{\partial Y} & \dots & n_3 \frac{\partial N_6}{\partial Y} \end{bmatrix} \quad (3.88)$$

Taking the variation of (3.85) gives,

$$d[\varepsilon_L] = \frac{1}{2} d[A][\Delta] + \frac{1}{2} [A]d[\Delta] \quad (3.89)$$

Inspection of (3.85) and (3.89) leads to the following property

$$\begin{aligned}
 d[A][\Delta] &= \frac{1}{2} \begin{bmatrix} d\{\Delta_X\}^T & \{0\} \\ \{0\} & d\{\Delta_Y\}^T \\ d\{\Delta_Y\}^T & d\{\Delta_X\}^T \end{bmatrix} \begin{bmatrix} \{\Delta_X\} \\ \{\Delta_Y\} \end{bmatrix} \\
 &= \frac{1}{2} \begin{bmatrix} \{\Delta_X\}^T & \{0\} \\ \{0\} & \{\Delta_Y\}^T \\ \{\Delta_Y\}^T & \{\Delta_X\}^T \end{bmatrix} \begin{bmatrix} d\{\Delta_X\} \\ d\{\Delta_Y\} \end{bmatrix} = [A]d[\Delta]
 \end{aligned} \tag{3.90}$$

Substitution of (3.90) into (3.89) yields,

$$d[\varepsilon_L] = \frac{1}{2}d[A][\Delta] + \frac{1}{2}[A]d[\Delta] = [A]d[\Delta] = [A][G]d\{\delta\}_i \tag{3.91}$$

The relationship between the second-order strain terms and displacement is given by,

$$d[\varepsilon_L] = [B_L]d\{\delta\}_i \tag{3.92}$$

Therefore  $B_L$  is given by,

$$[B_L] = [A][G] \tag{3.93}$$

### 3.1.7.3 Alternative second order displacement dependent terms of the B-matrix, $B_L$

Patch test results presented later in Section 3.2.1 indicate that the cancellation of the  $\frac{1}{2}$  in (3.85) through the method of variation leads to a discrepancy between Green-Lagrange definition for large strain (3.68)-(3.70) and the strain calculated by the formulation. An alternative formulation is therefore presented here.

The terms of the  $[\Delta]$  matrix are found from (3.87) and may be arranged into the 3 by 6  $[A]$  matrix defined by (3.85). Therefore, direct substitution of (3.87) into (3.85) gives,

$$[\varepsilon_L] = \frac{1}{2} \begin{bmatrix} \{\Delta_X\}^T & \{0\} \\ \{0\} & \{\Delta_Y\}^T \\ \{\Delta_Y\}^T & \{\Delta_X\}^T \end{bmatrix} \begin{bmatrix} \{\Delta_X\} \\ \{\Delta_Y\} \end{bmatrix} = \frac{1}{2}[A][G]\{\delta\}_i \tag{3.94}$$

Therefore  $B_L$  is given by,

$$[B_L] = \frac{1}{2}[A][G] \tag{3.95}$$

### 3.1.8 Element initial stress stiffness matrix [30,122]

According to the principle of virtual work the following expression for the initial or geometric stiffness matrix,  $[K_\sigma]^e$ , is suggested by Zienkiewicz [122]. The formulation uses the revised definition for  $B_L$  that includes the  $\frac{1}{2}$  term.

$$[K_\sigma^e]d\{\delta\}_i = \int_V d[B_L]^T\{\sigma\} dV \quad (3.96)$$

Taking the definition for  $B_L$ , (3.95), and applying a variation with respect to the displacement vector,  $\{\delta\}_i$ , gives

$$d[B_L]^T = \left\{ \frac{1}{2}d[A]G + \frac{1}{2}[A]d[G] \right\}^T \quad (3.97)$$

Assuming that the current geometry is held constant  $d[G] = 0$  and noting that

$$\left\{ \frac{1}{2}d[A]G \right\}^T = \frac{1}{2}[G]^T d[A]^T \text{ substitution of (3.97) into (3.96) yields,}$$

$$[K_\sigma^e]d\{\delta\}_i = \int_V \frac{1}{2}[G]^T d[A]^T \{\sigma\} dV \quad (3.98)$$

Using the definition for  $[A]$  given in (3.85)  $d[A]^T\{\sigma\}$  may be written as

$$\begin{aligned} d[A]^T\{\sigma\} &= \begin{bmatrix} d\{\Delta_X\} & 0 & d\{\Delta_Y\} \\ 0 & d\{\Delta_X\} & d\{\Delta_Y\} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \\ &= \begin{bmatrix} d\{\Delta_X\}\sigma_x + d\{\Delta_Y\}\tau_{xy} \\ d\{\Delta_Y\}\sigma_y + d\{\Delta_X\}\tau_{xy} \end{bmatrix} \\ &= \begin{bmatrix} \sigma_x[I_3]d\{\Delta_X\} + \tau_{xy}[I_3]d\{\Delta_Y\} \\ \tau_{xy}[I_3]d\{\Delta_X\} + \sigma_y[I_3]d\{\Delta_Y\} \end{bmatrix} \quad (3.99) \\ &= \begin{bmatrix} \sigma_x[I_3] & \tau_{xy}[I_3] \\ \tau_{xy}[I_3] & \sigma_y[I_3] \end{bmatrix} \begin{bmatrix} d\{\Delta_X\} \\ d\{\Delta_Y\} \end{bmatrix} \\ &= \begin{bmatrix} \sigma_x[I_3] & \tau_{xy}[I_3] \\ \tau_{xy}[I_3] & \sigma_y[I_3] \end{bmatrix} d\{\Delta\} \end{aligned}$$

Where  $[I_3]$  is a 3 by 3 identity matrix.

Now using the definition for  $\{\Delta\}$  given in (3.87) and again noting that  $d[G] = 0$  (3.99) may be rewritten as

$$d[A]^T\{\sigma\} = [M][G]d\{\delta\}_i \quad (3.100)$$

Where

$$[M] = \begin{bmatrix} \sigma_x[I_3] & \tau_{xy}[I_3] \\ \tau_{xy}[I_3] & \sigma_y[I_3] \end{bmatrix} \quad (3.101)$$

Substitution of (3.100) into (3.98) leads to the final expression for the initial (or geometric) stiffness matrix.

$$[K_{\sigma}^e] = \int_V \frac{1}{2} [G]^T [M][G] dV \quad (3.102)$$

### 3.1.9 Numerical integration for triangles [31,34]

As the integrations within (3.11) and (3.102) are difficult to evaluate analytically due to the complexity of the expressions numerical integration is employed. As demonstrated in [34] the quadrature rule for a function  $\phi$  of area coordinates  $\{\xi_1, \xi_2, \xi_3\}$  is given by

$$\int_A \phi dA = \frac{1}{2} \sum_{i=1}^n W_i J_i \phi_i \quad (3.103)$$

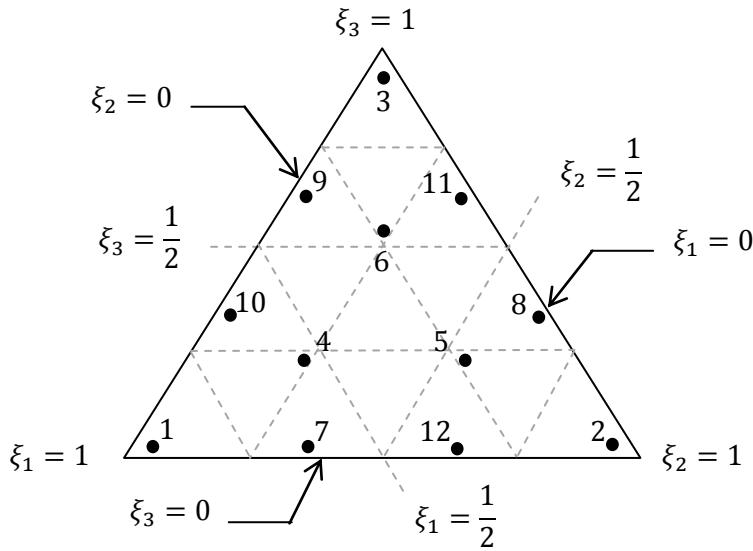
Where  $\phi_i$  is the value of  $\phi$  at a Gauss point in the triangle,  $W_i$  is the weight corresponding to that point and  $n$  is the number of Gauss points. The Jacobian,  $J_i$ , is calculated at each sample point using (3.67).

In the formulation presented by Zhang [31] it is proposed that 12 Gauss points should be used in order to maintain accuracy effected by element distortions. The area co-ordinates and weights of the sample points used are listed in Table 3-1 and shown in

Figure 3-4.

| Point<br><i>i</i> | Area co-ordinates |              |              | Weight<br><i>W<sub>i</sub></i> |
|-------------------|-------------------|--------------|--------------|--------------------------------|
|                   | $\xi_1$           | $\xi_2$      | $\xi_3$      |                                |
| <b>1</b>          | 0.8738219710      | 0.0630890144 | 0.0630890144 | 0.0508449063                   |
| <b>2</b>          | 0.0630890144      | 0.8738219710 | 0.0630890144 | 0.0508449063                   |
| <b>3</b>          | 0.0630890144      | 0.0630890144 | 0.8738219710 | 0.0508449063                   |
| <b>4</b>          | 0.5014265096      | 0.2492867451 | 0.2492867451 | 0.1167862757                   |
| <b>5</b>          | 0.2492867451      | 0.5014265096 | 0.2492867451 | 0.1167862757                   |
| <b>6</b>          | 0.2492867451      | 0.2492867451 | 0.5014265096 | 0.1167862757                   |
| <b>7</b>          | 0.6365024991      | 0.3013524510 | 0.0531450498 | 0.0825810756                   |
| <b>8</b>          | 0.0531450498      | 0.6365024991 | 0.3013524510 | 0.0825810756                   |
| <b>9</b>          | 0.3013524510      | 0.0531450498 | 0.6365024991 | 0.0825810756                   |
| <b>10</b>         | 0.6365024991      | 0.0531450498 | 0.3013524510 | 0.0825810756                   |
| <b>11</b>         | 0.0531450498      | 0.3013524510 | 0.6365024991 | 0.0825810756                   |
| <b>12</b>         | 0.3013524510      | 0.6365024991 | 0.0531450498 | 0.0825810756                   |

**Table 3-1: Gauss point area co-ordinates and weights**



**Figure 3-4: Approximate Gauss point positions within element**

### 3.1.10 Cable elements

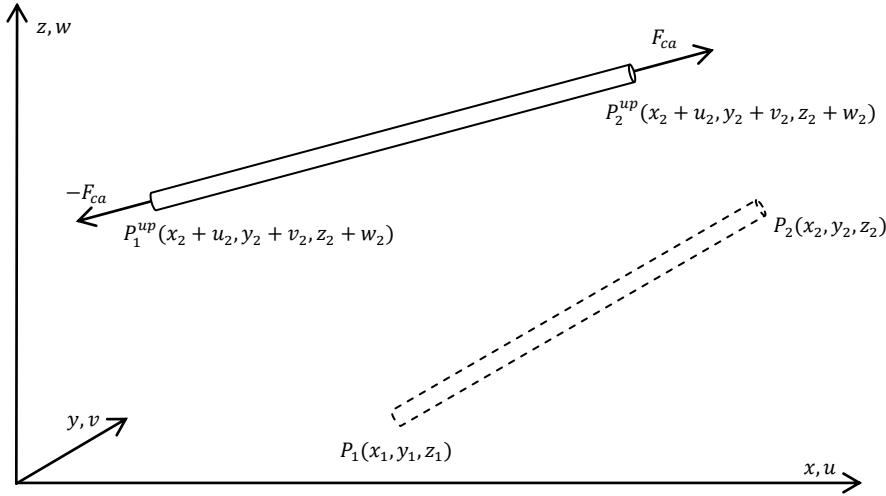
The edges of fabric structures are often restrained by prestressed edge cables. Therefore, an appropriate cable element is required. A classical linear element formulation used in both [31] and [30] is employed with the assumption of large displacements.

As depicted in Figure 3-5 the cable comprises two nodes defined in the global co-ordinate system as  $(x_1, y_1, z_1)$  and  $(x_2, y_2, z_2)$  respectively. The nodes undergo global displacements  $(u_1, v_1, w_1)$  and  $(u_2, v_2, w_2)$ . The initial element length,  $L_0$ , is given by

$$L_0 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (3.104)$$

and the displaced length,  $L$ , by

$$L = \sqrt{(x_2 - x_1 + u_2 - u_1)^2 + (y_2 - y_1 + v_2 - v_1)^2 + (z_2 - z_1 + w_2 - w_1)^2} \quad (3.105)$$



**Figure 3-5: Definition of linear cable element**

The change in length in the local element direction,  $\Delta_{ca}$ , may be expressed in terms of the initial and displaced lengths

$$\Delta_{ca} = (L - L_0) \quad (3.106)$$

The direction of the element is defined by the unit vector  $\hat{c}$  containing the direction cosines of the element,  $\{c_x, c_y, c_z\}$ . Due to the assumption of large displacement the updated nodal co-ordinates are used in the definition of the directional cosines.

$$[\hat{c}] = \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} = \frac{1}{L} \begin{bmatrix} (x_2 - x_1 + u_2 - u_1) \\ (y_2 - y_1 + v_2 - v_1) \\ (z_2 - z_1 + w_2 - w_1) \end{bmatrix} \quad (3.107)$$

The cable force,  $F_{ca}$ , acting along the cable axis is given by

$$F_{ca} = F_{ca0} + \frac{EA}{L_0} (L - L_0) = F_{ca0} + k_E^c \Delta_{ca} \quad (3.108)$$

Where  $F_{ca0}$  is the cable prestress, and  $\frac{EA}{L_0}$  is the local stiffness modulus of the element,  $k_E^c$ .

The cable element residual forces at each node in the global co-ordinate system are combined into a single vector,  $\{f_{ca}\}_i$ , and are given by

$$\{f_{ca}\}_i = \begin{bmatrix} \{f_{ca}\}_{P_1} \\ \{f_{ca}\}_{P_2} \end{bmatrix} = \begin{bmatrix} -\hat{c} F_{ca} \\ \hat{c} F_{ca} \end{bmatrix} \quad (3.109)$$

Therefore, the cable transformation matrix,  $T_k^{ca}$ , which transforms the cable force from the element local co-ordinate system to the global co-ordinate system may be expressed as,

$$T_k^{ca} = \begin{bmatrix} -c_x \\ -c_y \\ -c_z \\ c_x \\ c_y \\ c_z \end{bmatrix} = \frac{1}{L} \begin{bmatrix} -(x_2 - x_1 + u_2 - u_1) \\ -(y_2 - y_1 + v_2 - v_1) \\ -(z_2 - z_1 + w_2 - w_1) \\ (x_2 - x_1 + u_2 - u_1) \\ (y_2 - y_1 + v_2 - v_1) \\ (z_2 - z_1 + w_2 - w_1) \end{bmatrix} \quad (3.110)$$

The cable element force vectors are assembled into a full system vector which is added to the residual force equation (3.34) to give,

$$\{R\} = \sum_{i=1}^{i=m} \left[ \int_V [tB]^T [\sigma] dV \right]_i + \sum_{i=1}^{i=m} [f_{ca}]_i - \{P\} \quad (3.111)$$

The cable elastic stiffness matrix in the global co-ordinate system,  $K_E^{ca}$ , is given by,

$$K_E^{ca} = T_k^c T k_E^{ca} T_k^c = \frac{EA}{L_0} \begin{bmatrix} [\hat{e}][\hat{e}]^T & -[\hat{e}][\hat{e}]^T \\ -[\hat{e}][\hat{e}]^T & [\hat{e}][\hat{e}]^T \end{bmatrix} \quad (3.112)$$

The geometric stiffness matrix  $K_\sigma^c$  which accounts for stiffness induced by the element's change in orientation is derived in [30] as

$$K_\sigma^{ca} = \frac{F_{cao}}{L_0} \begin{bmatrix} ([I_3] - [\hat{e}][\hat{e}]^T) & -([I_3] - [\hat{e}][\hat{e}]^T) \\ -([I_3] - [\hat{e}][\hat{e}]^T) & ([I_3] - [\hat{e}][\hat{e}]^T) \end{bmatrix} \quad (3.113)$$

The total element stiffness matrix,  $K_T^{ca}$ , is given by the summation of (3.112) and (3.113) yielding,

$$K_T^{ca} = K_E^{ca} + K_\sigma^{ca} = \begin{bmatrix} (K_{Tsub}^{ca}) & -(K_{Tsub}^{ca}) \\ -(K_{Tsub}^{ca}) & (K_{Tsub}^{ca}) \end{bmatrix}$$

where

$$K_{Tsub}^{ca} = \frac{EA - F_{cao}}{L_0} \cdot [\hat{e}][\hat{e}]^T + \frac{F_{cao}}{L_0} \cdot [I_3] \quad (3.114)$$

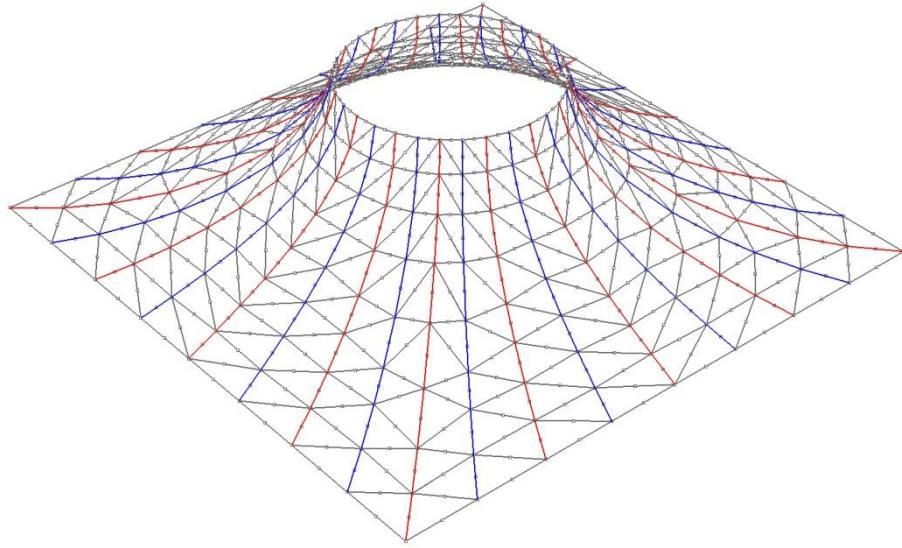
The cable contribution to stiffness is also assembled into a full system array and is added to the total stiffness (3.13) to give,

$$[K_T] = [K_\sigma] + [K_E] + \sum_{i=1}^{i=m} [K_T^{ca}]_i \quad (3.115)$$

### 3.1.11 Form finding with geodesic control

As the form of a fabric structure is a product of the applied prestress and restraints an initial form finding process must be undertaken prior to any load analysis to find the structures initial equilibrated form. As discussed in Chapter 2 the initial form may be investigated using physical modelling with low stiffness materials or soap films. With the rise in computer simulation physical modelling has been replaced by computer simulation

using a soap film analogy. Further improvements in efficiency may be achieved through parametric design used to define initial mesh geometry



**Figure 3-6: Conic mesh with panels, seams (red) and centre lines (blue)**

A fabric membrane is typically fabricated from panels welded along seams as shown in Figure 3-6. The seams and centre lines of the panels follow geodesic paths over the membrane surface. The material is generally aligned so the warp direction follows the centre line of each panel. As form finding is undertaken with zero stiffness, additional control is required in order to maintain the geodesic paths of the panel edges and centre lines. This may be done by including special tensioned control elements similar to cable elements. The following control element definition and computational procedure is adapted from the warp string control element presented in [25] for a 3 node constant strain element.

The residual forces within the control string acting in the global coordinate system directions are derived in much the same way as the global cable residual forces. The force acting in the local control string direction is transformed to the global via direction cosines,  $\hat{cs}$ , defined using the nodal co-ordinates at each end of the control string.

$$[\hat{cs}] = \begin{bmatrix} cs_x \\ cs_y \\ cs_z \end{bmatrix} = \frac{1}{L} \begin{bmatrix} (x_2 - x_1 + u_2 - u_1) \\ (y_2 - y_1 + v_2 - v_1) \\ (z_2 - z_1 + w_2 - w_1) \end{bmatrix} \quad (3.116)$$

The control string element is only used during form finding and therefore always has zero stiffness. The tension force within the imaginary control element is arbitrary. Through trial and error a value of 10kN/m was found to generate sufficient mesh control.

$$F_{cs} = F_{cs0} = 10 \quad (3.117)$$

The control string residual forces at each node of a control string  $k$  in the global co-ordinate system are combined into a single vector,  $\{f_{cs}\}_k$ , given by

$$\{f_{cs}\}_k = \begin{bmatrix} \{f_{cs}\}_{p_1} \\ \{f_{cs}\}_{p_2} \end{bmatrix} = \begin{bmatrix} -\hat{c}\hat{s}F_{cs} \\ \hat{c}\hat{s}F_{cs} \end{bmatrix} \quad (3.118)$$

The resulting control string force vectors may then be combined into a full system array by simple addition to give the final residual force at each node.

The control elements should not affect the shape of the final membrane only the alignment of mesh along the panel edges and centre lines. Therefore, the residual forces at each node in the string are resolved into orthogonal components aligned with a plane tangential to the surface at that node. Any residual force component acting normal to the membrane surface is zeroed. In order to prevent nodal drift along the string the residual force component aligned with the string direction in the local surface co-ordinate system is also zeroed. This leaves only residual forces acting perpendicular to the string within the membrane surface. These residual forces force the string to remain aligned with the desired geodesic path. At the end nodes, for example those situated in boundary cables, the residual force must be entirely zeroed to prevent distortion of the mesh boundary.

The local membrane surface coordinate system is defined using a vector normal to the surface found from the average of the normal vectors of all connected elements. As the surface of each element is curved the element normal vectors are defined at the specific node within the element using the normalised natural co-ordinate system. The process is similar that described in Section 3.1.5 for the development of the membrane transformation matrix.

The natural co-ordinates of each node within the element are illustrated in Figure 3-2. The tangent plane to the element surface at a node is derived from the natural co-ordinate basis vectors at the node given by,

$$\vec{\xi} = \left[ \sum_{i=1}^6 \frac{\partial N_i}{\partial \xi} x_i \hat{i} + \sum_{i=1}^6 \frac{\partial N_i}{\partial \eta} y_i \hat{j} + \sum_{i=1}^6 \frac{\partial N_i}{\partial \zeta} z_i \hat{k} \right] \quad (3.119)$$

$$\vec{\eta} = \left[ \sum_{i=1}^6 \frac{\partial N_i}{\partial \eta} x_i \hat{i} + \sum_{i=1}^6 \frac{\partial N_i}{\partial \zeta} y_i \hat{j} + \sum_{i=1}^6 \frac{\partial N_i}{\partial \xi} z_i \hat{k} \right] \quad (3.120)$$

The cross product of the natural coordinate basis vectors produces the orthogonal vector,  $\vec{Z}$ , aligned with the local Z-direction normal to the element surface.

$$\vec{Z} = \vec{\xi} \times \vec{\eta} \quad (3.121)$$

$$\hat{Z} = \frac{\vec{z}}{\|\vec{z}\|} \text{ or } \hat{Z} = \frac{\vec{\xi} \times \vec{\eta}}{\|\vec{\xi} \times \vec{\eta}\|} \quad (3.122)$$

As demonstrated in Figure 3-7(a) any node within the control string will be connected to 2 or more elements. An approximation for the normal vector to the surface at a node is given by the mean of all element normal vectors associated with that node.

$$\hat{N} = \frac{\sum_{j=1}^{j=n} \hat{Z}_{i,j}}{n} = [n_1 \hat{i} + n_2 \hat{j} + n_3 \hat{k}] \quad (3.123)$$

Where  $\hat{Z}_{i,j}$  denotes the normal vector at node  $i$  in element  $j$ ,  $n$  is the total number of elements connected to the control string at node  $i$ .

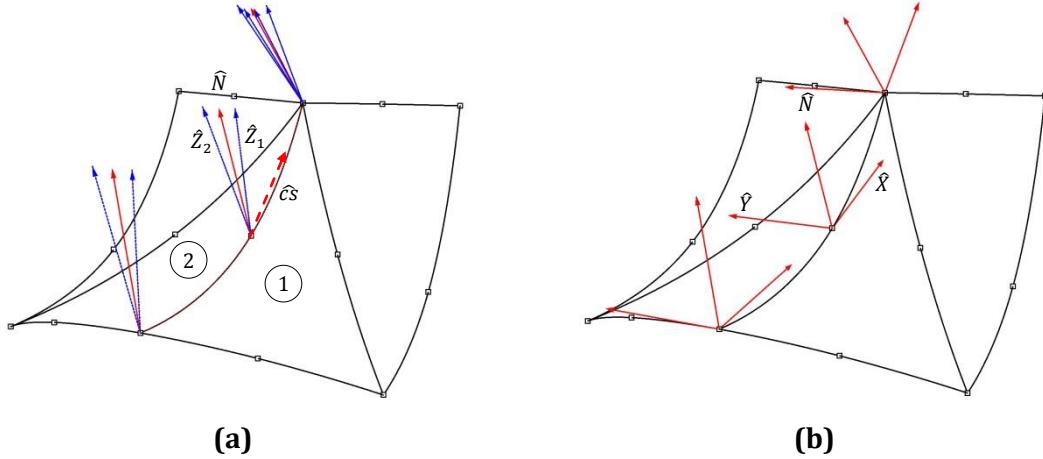
To define the remaining basis vectors, illustrated in Figure 3-7(b), the element direction vector,  $\vec{cs}$ , is used. The cross product of the surface normal vector and the string direction vector gives the basis vector perpendicular to the string in the tangent plane of the surface.

$$\vec{Y} = \hat{N} \times \vec{cs} \quad (3.124)$$

$$\hat{Y} = \frac{\vec{Y}}{\|\vec{Y}\|} \text{ or } \hat{Y} = \frac{\hat{N} \times \vec{cs}}{\|\hat{N} \times \vec{cs}\|} = [m_1 \hat{i} + m_2 \hat{j} + m_3 \hat{k}] \quad (3.125)$$

The remaining basis vector is easily found from the cross product of the normal and perpendicular basis vectors,

$$\hat{X} = \frac{\hat{Y} \times \hat{Z}}{\|\hat{Y} \times \hat{Z}\|} = [l_1 \hat{i} + l_2 \hat{j} + l_3 \hat{k}] \quad (3.126)$$



**Figure 3-7: (a) Element normals,  $\hat{Z}_i$ , and surface normals,  $\hat{N}$  (b) Surface basis vectors**

The surface transformation matrix at node  $i$ ,  $[T_i^{CS}]$ , may then be defined in terms of directional cosines between the global and local surface coordinate systems.

$$T_i^{cs} = \begin{bmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix} \quad (3.127)$$

The control string residual force component acting normal to the membrane surface in the local surface co-ordinate system is given by

$$\{F_{csN}\}_i = \hat{N}_i^T \{f_{cs}\}_i \quad (3.128)$$

The control string residual force component acting within the surface tangent plane aligned with the control string in the local surface co-ordinate system is given by

$$\{F_{csX}\}_i = \hat{X}_i^T \{f_{cs}\}_i \quad (3.129)$$

Therefore, the control string residual force component acting within the surface tangent plane perpendicular to the control string in the global co-ordinate system is given by

$$\{f_{csy}\}_i = \{f_{cs}\}_i - \{F_{csN}\}_i \hat{N}_i - \{F_{csX}\}_i \hat{X}_i \quad (3.130)$$

The control string element residual force vectors are assembled into a full system vector which is added to the residual force equation (3.111) to give,

$$\{R\} = \sum_{i=1}^{i=m} \left[ \int_V [{}_t B]^T [\sigma] dV \right]_i + \sum_{i=1}^{i=m} [f_{cal}]_i + \sum_{i=1}^{i=m} [f_{csy}]_i - \{P\} \quad (3.131)$$

### 3.1.12 Dynamic Relaxation procedure overview

The iterative procedure as summarised by Barnes, [25], is shown in Figure 3-8.

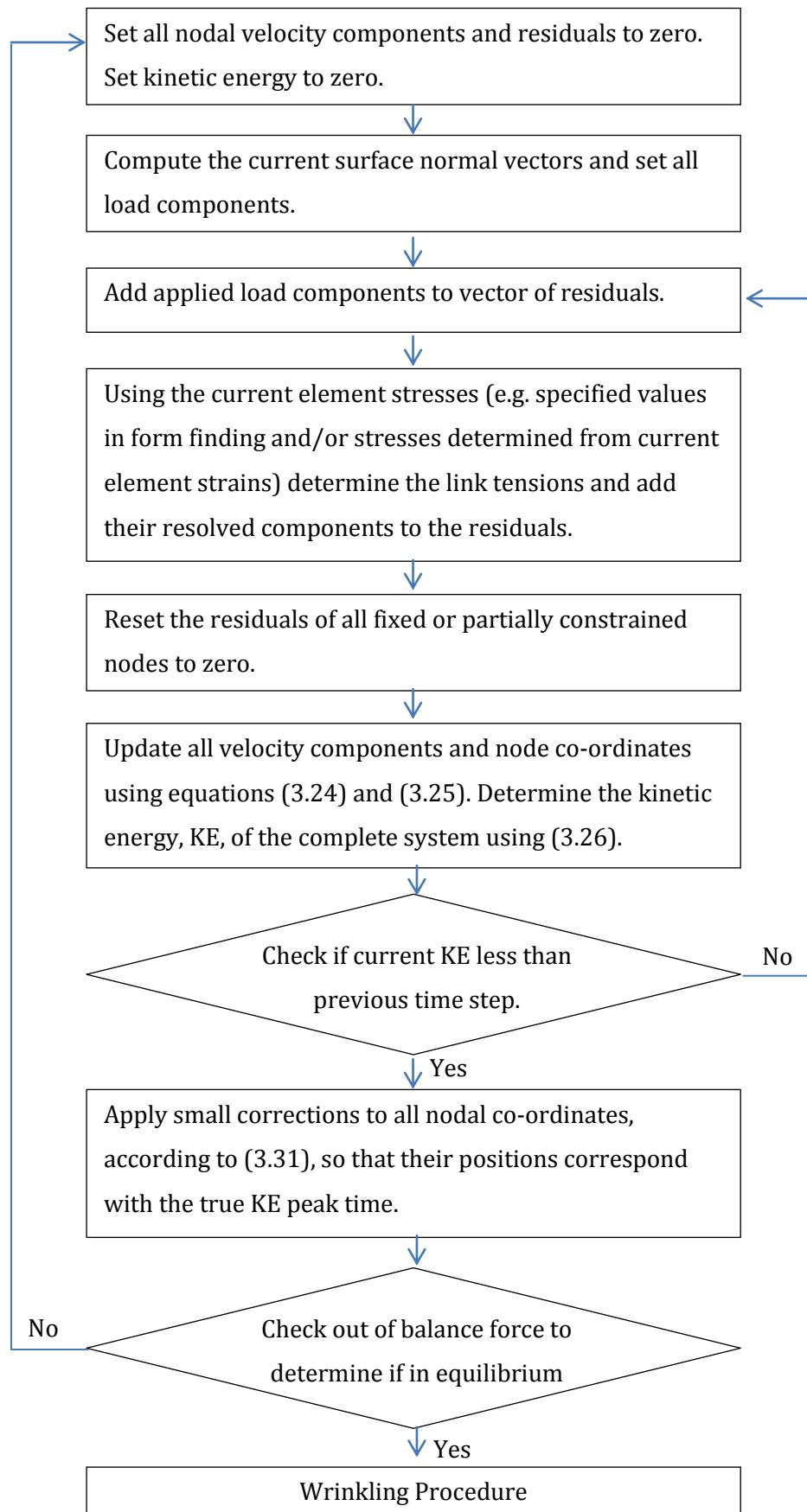


Figure 3-8: Summary of Dynamic Relaxation algorithm

### 3.1.13 Wrinkling procedure

During Load analysis it is possible that the structure may undergo sufficient negative strain to cause certain areas to exhibit negative stress. To account for the negligible compressive stiffness of Architectural Fabric it is necessary to modify the material model where a simulation produces such areas of negative stress. The following wrinkling procedure is based on those presented in [31] and [123].

The principle stresses, i.e. the maximum and minimum stress at a gauss point are given by,

$$\sigma_{max,min}^p = \frac{1}{2}(\sigma_y + \sigma_x) \pm \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + (\tau_{xy})^2} \quad (3.132)$$

where  $\sigma_w$  and  $\sigma_f$  are the warp and fill stresses at that gauss point and  $\tau_{xy}$  is the shear stress. Similarly the principle strains are given by,

$$\varepsilon_{max,min}^p = \frac{1}{2}(\varepsilon_y + \varepsilon_x) \pm \sqrt{\left(\frac{\varepsilon_x - \varepsilon_y}{2}\right)^2 + (\gamma_{xy})^2} \quad (3.133)$$

where  $\varepsilon_w$  and  $\varepsilon_f$  are the warp and fill strains at that gauss point and  $\gamma_{xy}$  is the shear strain.

The angle between the direction of maximum stress and the local warp direction is given by,

$$\theta^p = \frac{1}{2} \tan^{-1} \left( \frac{2\tau_{xy}}{\sigma_x - \sigma_y} \right) \quad (3.134)$$

The fabric membrane may be deemed to be in one of three different states. The first state is taut where the membrane has positive stress in all directions. The second state is wrinkled where the membrane has negative stress in some directions but positive in others. The final state is slack where the material undergoes negative stress in all directions. Three different wrinkling criterion are suggested,

1. Stress criterion based solely on principle stress,

$$\sigma_{min}^p > 0: \quad 'taut state' \quad (3.135)$$

$$\sigma_{min}^p \leq 0 \text{ and } \sigma_{max}^p > 0: \quad 'wrinkled state' \quad (3.136)$$

$$\text{otherwise:} \quad 'slack state' \quad (3.137)$$

This criterion is straightforward but a case where negative maximum stress may occur in combination with a positive minimum strain, thus a wrinkled state is incorrectly defined as slack.

2. Strain criterion based solely on principle strain,

$$\varepsilon_{min}^p > 0: \quad 'taut state' \quad (3.138)$$

$$\varepsilon_{min}^p \leq 0 \text{ and } \varepsilon_{max}^p > 0: \quad 'wrinkled state' \quad (3.139)$$

$$\text{otherwise:} \quad 'slack state' \quad (3.140)$$

Again this criterion is straightforward. However, due to the effect of Poisson's ratio, it has been found that a taunted state may be misjudged as wrinkled where a negative minimum strain may coincide with positive minimum stress.

3. Mixed criterion based on a combination of principle stress and strain,

$$\sigma_{min}^p > 0: \quad 'taut state' \quad (3.141)$$

$$\sigma_{min}^p \leq 0 \text{ and } \varepsilon_{max}^p > 0: \quad 'wrinkled state' \quad (3.142)$$

$$\text{otherwise:} \quad 'slack state' \quad (3.143)$$

This criterion is deemed to have overcome the shortfalls of the individual stress and strain criterion. During testing it was found that both the stress and mixed criterion achieved reasonable results however the strain criterion failed to correctly identify area of wrinkling. It is noted that in [123] the normal component of strain to the surface is used in place of the principle strain in the criterion. Again in testing this lead to instabilities in the formulation. Therefore, the formulation presented is in closer agreement to [31].

The procedure is initialised after convergence is reached using the unmodified material stiffness matrix,  $[E]$ . Each Gauss point is then inspected and assigned a state depending on the selected wrinkling criterion. Where the membrane is deemed to be taut after the first analyses run the material stiffness matrix remains unmodified,

$$[E_{mod}] = [E] \quad (3.144)$$

Where the membrane is deemed to be wrinkled the material stiffness matrix is modified in the direction of the minimum principle stress, i.e. normal to the wrinkle. The material stiffness matrix is first transformed to be aligned with the minimum principle stress using,

$$[E_{rot}] = [T^E(-\theta^p)][E][T^E(-\theta^p)]^T \quad (3.145)$$

where,

$$[T^E(\theta^p)] = \begin{bmatrix} c^2 & s^2 & -2sc \\ s^2 & c^2 & 2sc \\ sc & -sc & c^2 - s^2 \end{bmatrix}, c = \cos(\theta^p) \text{ and } s = \sin(\theta^p) \quad (3.146)$$

The rotated material stiffness matrix is then modified using a penalisation parameter,  $P$ , as follows,

$$[E_{rot,mod}] = \begin{bmatrix} E_{rot,11} & P \cdot E_{rot,12} & E_{rot,13} \\ P \cdot E_{rot,21} & P \cdot E_{rot,22} & P \cdot E_{rot,23} \\ E_{rot,31} & P \cdot E_{rot,32} & E_{rot,33} \end{bmatrix} \quad (3.147)$$

Finally the material stiffness matrix is transformed back to the original local material system using

$$[E_{mod}] = [T^E(\theta^p)][E_{rot,mod}][T^E(\theta^p)]^T \quad (3.148)$$

Where the membrane is deemed to be slack the material stiffness matrix is penalised in all directions

$$[E_{mod}] = P \cdot [E] \quad (3.149)$$

In [123] the following definition for the penalisation parameter is suggested to improve convergence stability,

$$P_\sigma = \frac{\sigma_{min}^{per}}{\sigma_{min}^p} \rightarrow \begin{cases} P_\sigma < P & \rightarrow P = P_\sigma \\ P_\sigma > P & \rightarrow P = P \\ P_\sigma > 1 \text{ or } P_\sigma < 0 & \rightarrow P = 1 \end{cases} \quad (3.150)$$

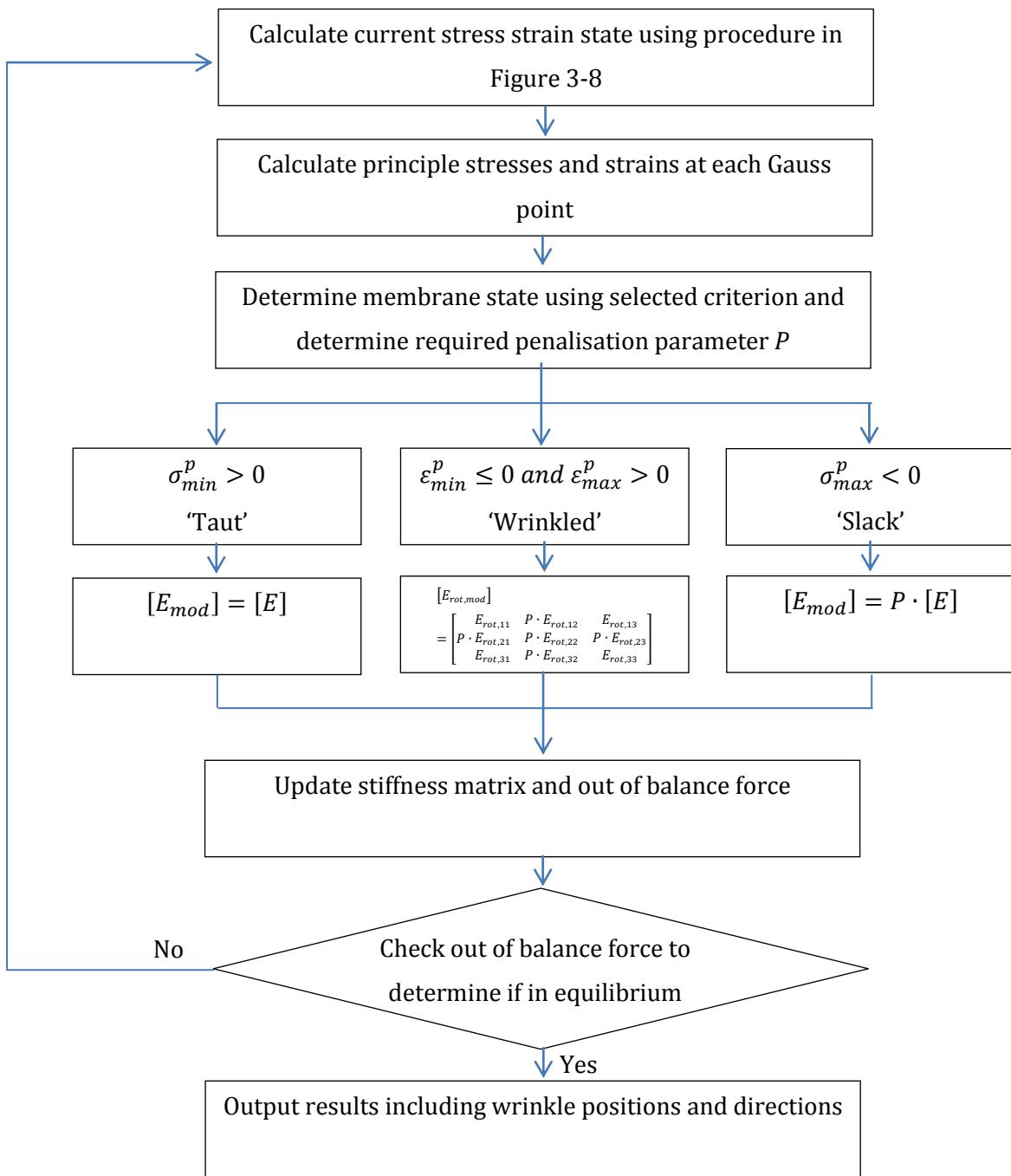
where  $\sigma_{min}^{per}$  is the maximum permissible compressive stress and  $\sigma_{min}^p$  is the current minimum stress. This allows a small amount of compression in the structure.

The structure is then re-analysed using the modified elastic matrix at each Gauss point and again the state of the membrane at each gauss point is assessed. If the membrane has undergone a change of state i.e. from wrinkled or slack to taut the penalisation parameter is modified to take this into account using.

$$\text{new state is 'taut'} \rightarrow \begin{cases} \text{old state is 'taut'} \rightarrow & [E_{mod}] = [E] \\ \text{old state is not 'taut'} \rightarrow & P = P_{old} \cdot \beta \\ & \text{new state} = \text{old state} \end{cases} \quad (3.151)$$

where  $\beta = 10$  as recommended in [123]. In [123] the unmodified material stiffness matrix is used in conjunction with strain state derived using the modified stiffness matrix to calculate an effective principle stress for the assessment of the wrinkling criterion. In testing this consistently lead to instability in the analysis. Therefore, in the procedure presented here the current stress calculated using the modified material stiffness matrix is always used in the wrinkling criterion assessment.

Wrinkling analysis continues until the convergence criterion is met after the material stiffness matrix has been modified. In other words the out of balance force is not significantly altered by the application of any wrinkling modification required. This will occur when all compressive stresses greater than maximum permissible compressive stress have been eliminated.

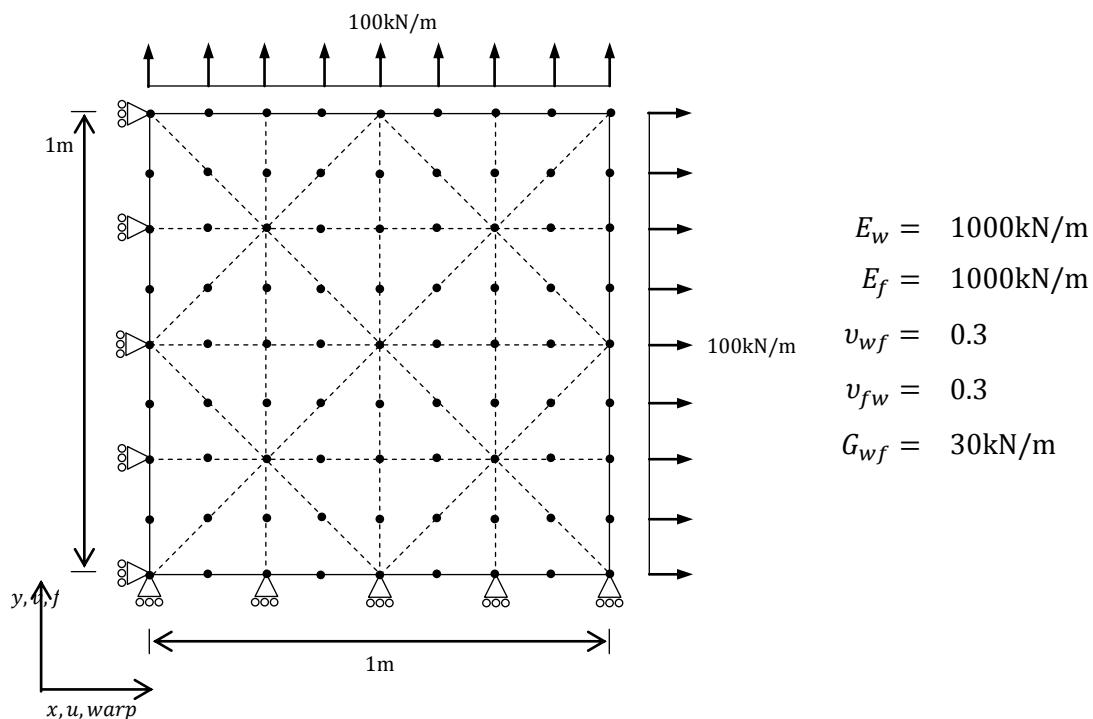
**Figure 3-9: Wrinkling procedure summary**

## 3.2 Analysis Examples

### 3.2.1 Patch test

The analysis procedure developed above is validated using a simple patch test. A 1 metre square patch of fabric is discretised into 81 elements and restrained along two edges by rolling restraints. Uniformly distributed loads (UDL) of 100kN/m are applied along the free edges. The warp material direction is aligned with the global  $x$ -direction and the fill material direction with the global  $y$ -direction. No prestress is applied to the material.

Therefore, no form finding is required.



**Figure 3-10: 32 element patch for validation**

The patch is given the plane stress material properties shown in Figure 3-10 alongside the mesh definition and boundary conditions. Substitution of the values into the plane stress stiffness matrix yields,

$$\{\sigma\} = [E]\{\varepsilon\} = \begin{bmatrix} \frac{E_w}{(1-v_{wf}v_{fw})} & \frac{E_wv_{fw}}{(1-v_{wf}v_{fw})} & 0 \\ \frac{E_fv_{wf}}{(1-v_{wf}v_{fw})} & \frac{E_f}{(1-v_{wf}v_{fw})} & 0 \\ 0 & 0 & G_{wf} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{1000}{0.91} & \frac{300}{0.91} & 0 \\ \frac{300}{0.91} & \frac{1000}{0.91} & 0 \\ 0 & 0 & 30 \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{wy} \end{Bmatrix} \quad (3.152)$$

The loads are applied only in the plane of the material. Therefore, no out of plane forces are present and the standard definition for green's strain may be used to relate displacement to strain as follows.

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{1}{2} \left( \frac{(\text{o}l_x + \Delta l_x)^2}{\text{o}l_x^2} - 1 \right) \\ \frac{1}{2} \left( \frac{(\text{o}l_y + \Delta l_y)^2}{\text{o}l_y^2} - 1 \right) \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{1}{2} \left( \frac{(1+u)^2}{1^2} - 1 \right) \\ \frac{1}{2} \left( \frac{(1+v)^2}{1^2} - 1 \right) \\ 0 \end{Bmatrix} \quad (3.153)$$

As all loading is aligned with the material directions no shear strain and hence shear stress is introduced. The patch will undergo equal displacement in the principal directions due to the applied load. The resulting stress in the patch after displacement is given by

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{100}{(1+u)} \\ \frac{100}{(1+v)} \\ 0 \end{Bmatrix} \quad (3.154)$$

Three full analyses are completed. The first analysis uses the displacement dependent B matrix (3.93) developed in Section 3.1.7.2 and the updated nodal configuration for residual force calculation (3.34). This analysis yields the results shown in Figure 3-11(a-f). The results alongside the expected results given by equations (3.152) to (3.154) are shown in Table 3-2.

The second analysis uses the displacement dependent B matrix (3.95) developed in Section 3.1.7.3 and the updated nodal configuration for residual force calculation (3.34). This analysis yields uniform stress and strain results similar to those demonstrated in Figure 3-11 (a-f). The results alongside the expected results given by equations (3.152) to (3.154) are shown in Table 3-3.

The third analysis again uses the displacement dependent B matrix (3.95) developed in Section 3.1.7.3 but does not use the updated co-ordinate nodal configuration. Again this analysis yields uniform stress and strain results similar to those demonstrated in Figure 3-11 (a-f). The results alongside the expected results given by equations (3.152) to (3.154) are shown in Table 3-4.

All values are reported to 4 significant figures, the error between the analysis and the result from (3.152) is attributed to rounding error.

|  | <b>Analysis</b> | <b>(3.152)</b> | <b>(3.153)</b> | <b>(3.154)</b> |
|--|-----------------|----------------|----------------|----------------|
| <b>Warp Stress, <math>\sigma_x</math></b>      | 94.16kN/m       | 94.14kN/m      | -              | 94.16kN/m      |
| <b>Fill Stress, <math>\sigma_y</math></b>      | 94.16kN/m       | 94.14kN/m      | -              | 94.16kN/m      |
| <b>Warp Strain, <math>\varepsilon_x</math></b> | 0.0659          | 0.0659         | 0.0640         | -              |
| <b>Fill Strain, <math>\varepsilon_y</math></b> | 0.0659          | 0.0659         | 0.0640         | -              |
| <b>Warp Displacement, <math>u</math></b>       | 0.0621m         | -              | 0.0621m        | 0.0621m        |
| <b>Fill Displacement, <math>v</math></b>       | 0.0621m         | -              | 0.0621m        | 0.0621m        |

**Table 3-2:** Analysis results using  $[B_L] = [A][G]$  and updated nodal configuration for residual force calculation alongside expected results.

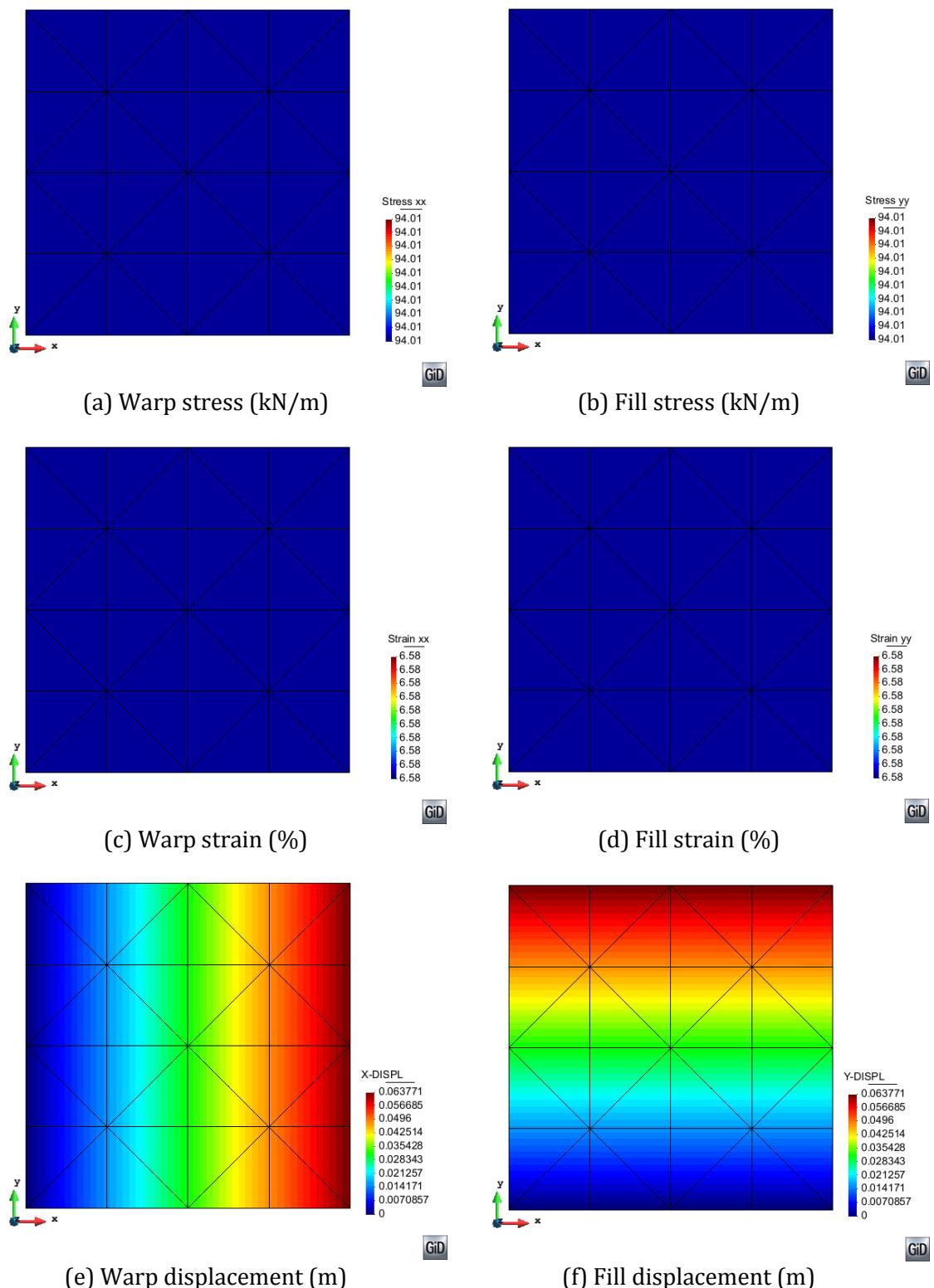
|  | <b>Analysis</b> | <b>(3.152)</b> | <b>(3.153)</b> | <b>(3.154)</b> |
|--|-----------------|----------------|----------------|----------------|
| <b>Warp Stress, <math>\sigma_x</math></b>      | 94.01kN/m       | 94.00kN/m      | -              | 94.01kN/m      |
| <b>Fill Stress, <math>\sigma_y</math></b>      | 94.01kN/m       | 94.00kN/m      | -              | 94.01kN/m      |
| <b>Warp Strain, <math>\varepsilon_x</math></b> | 0.0658          | 0.0658         | 0.0658         | -              |
| <b>Fill Strain, <math>\varepsilon_y</math></b> | 0.0658          | 0.0658         | 0.0658         | -              |
| <b>Warp Displacement, <math>u</math></b>       | 0.0638m         | -              | 0.0638m        | 0.0638m        |
| <b>Fill Displacement, <math>v</math></b>       | 0.0638m         | -              | 0.0638m        | 0.0638m        |

**Table 3-3:** Analysis results using  $[B_L] = \frac{1}{2}[A][G]$  and updated nodal configuration for residual force calculation alongside expected results.

|  | <b>Analysis</b> | <b>(3.152)</b> | <b>(3.153)</b> | <b>(3.154)</b> |
|--|-----------------|----------------|----------------|----------------|
| <b>Warp Stress, <math>\sigma_x</math></b>      | 96.82kN/m       | 96.86kN/m      | -              | 93.84kN/m      |
| <b>Fill Stress, <math>\sigma_y</math></b>      | 96.82kN/m       | 96.86kN/m      | -              | 93.84kN/m      |
| <b>Warp Strain, <math>\varepsilon_x</math></b> | 0.0678          | 0.0678         | 0.0678         | -              |
| <b>Fill Strain, <math>\varepsilon_y</math></b> | 0.0678          | 0.0678         | 0.0678         | -              |
| <b>Warp Displacement, <math>u</math></b>       | 0.0656m         | -              | 0.0656m        | 0.0656m        |
| <b>Fill Displacement, <math>v</math></b>       | 0.0656m         | -              | 0.0656m        | 0.0656m        |

**Table 3-4:** Analysis results using  $[B_L] = \frac{1}{2}[A][G]$  and initial nodal configuration for residual force calculation alongside expected results.

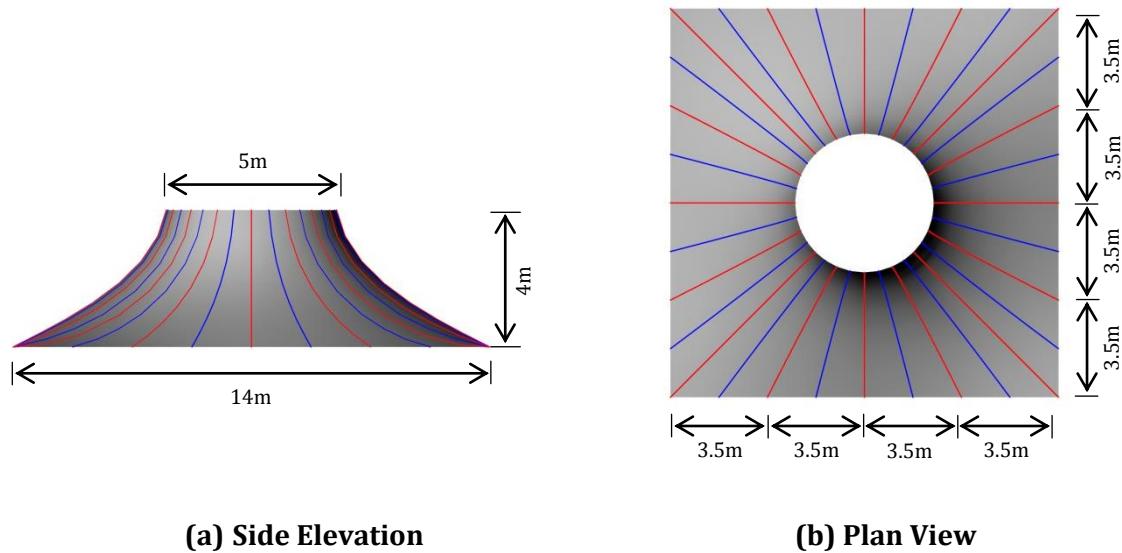
The strain result produced by the analysis using (3.93) is not consistent with (3.153) shown in Table 3-2. As demonstrated in Table 3-4 the analysis which uses (3.95) but does not use the updated nodal configuration produces a stress result which is not consistent with (3.114). Only the results given by the analysis using (3.95) and the updated nodal configuration for calculation of residual force produces consistent results across the board.



**Figure 3-11: Patch test results using  $[B_L] = \frac{1}{2} [A][G]$**

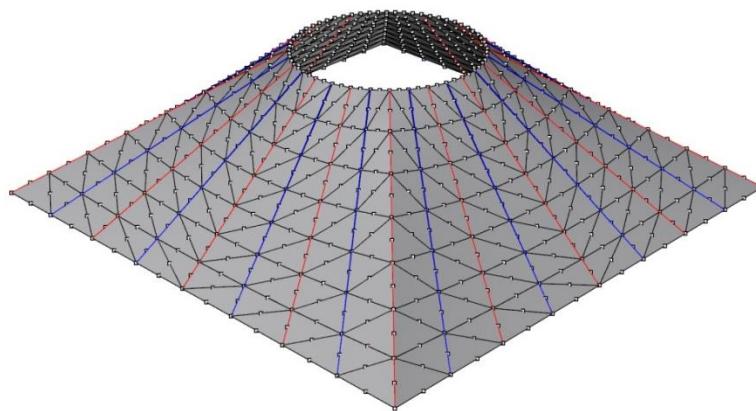
### 3.2.2 Square based conic simulation

To demonstrate the effectiveness of the geodesic control string elements form finding simulations are performed with and without control elements on a simple square based conic. The conic is based on the conic example with equal prestress presented in [36] and is shown in Figure 3-12. The base is 14m square with a 5m diameter hoop 4m above the base, the conic is fully fixed along all edges. A prestress of 4kN/m is applied in the warp and fill material directions. The meshes in this thesis have been generated using Grasshopper an algorithmic modelling plug in for Rhino 3D. For an overview of the mesh generation and analysis procedure for the conic structure see Appendix A.



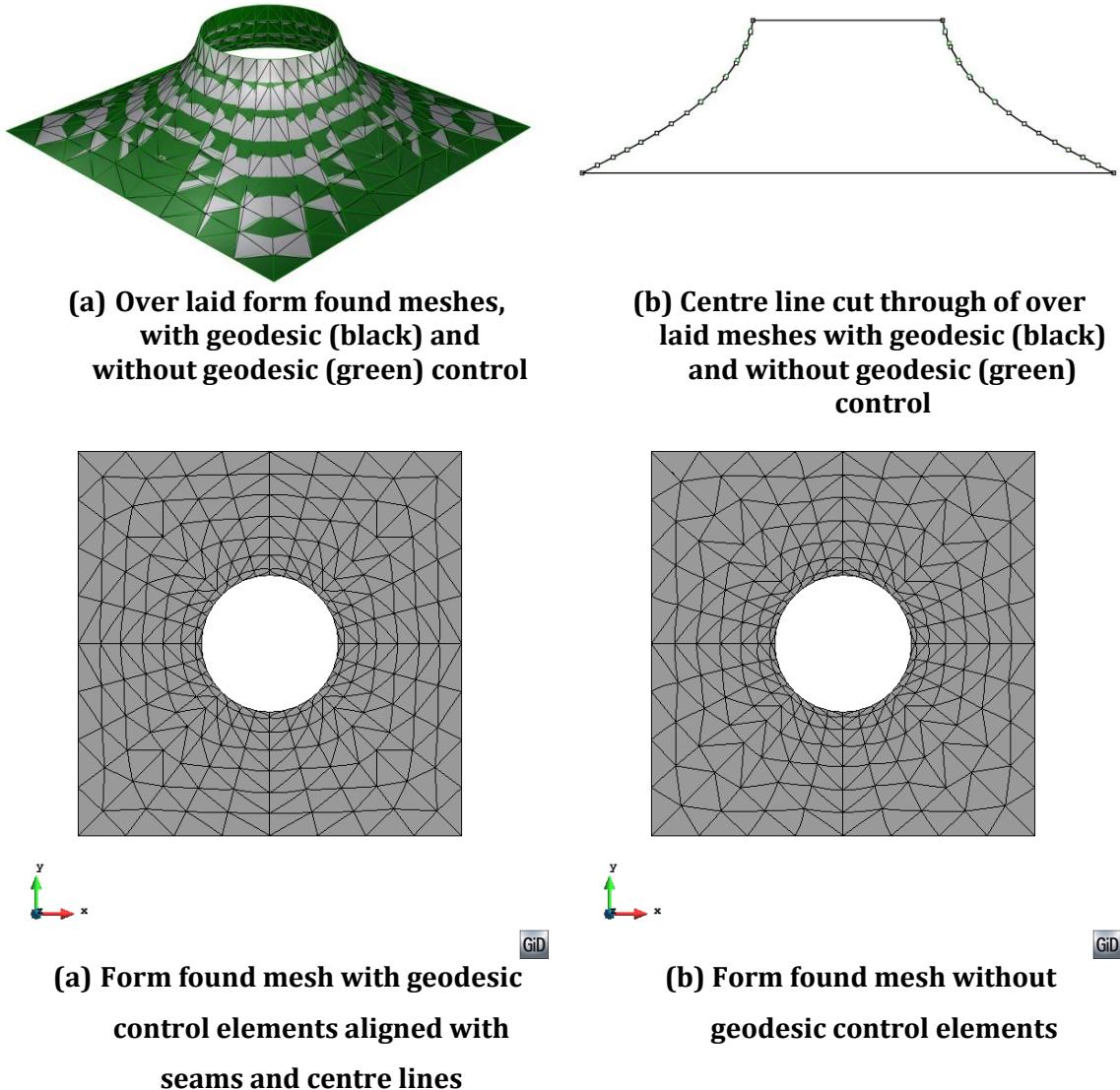
**Figure 3-12: Square based conic**

The conic is assembled from 16 panels the edges of which are shown in red in Figure 3-12 the centre lines of the panels are shown in blue and define the material warp direction. The initial mesh made up of 448 elements defined by 960 nodes is shown in Figure 3-13.



**Figure 3-13: Initial conic mesh**

The conic is formed found with and without geodesic control elements the resulting mesh configurations are shown in Figure 3-14(a-c).



**Figure 3-14: Form found conic mesh**

Figure 3-14(a) demonstrates that the mesh surface is approximately aligned within the constraints of the mesh. The resulting seam edges and centre lines are more clearly shown with and without geodesic control elements in Figure 3-14(b) and (c) respectively. It can be seen that the seams and centre lines become distorted without the control elements.

The material properties shown in Table 3-5 are applied to the form found mesh and 3 load analyses are undertaken to demonstrate the formulation with material properties. The first analysis checks the uniformity of prestress by applying no load. The second load case simulates a snow load of 0.6kN/m acting vertically .The third load case simulates wind uplift load of 1kN/m applied perpendicular to the surface.

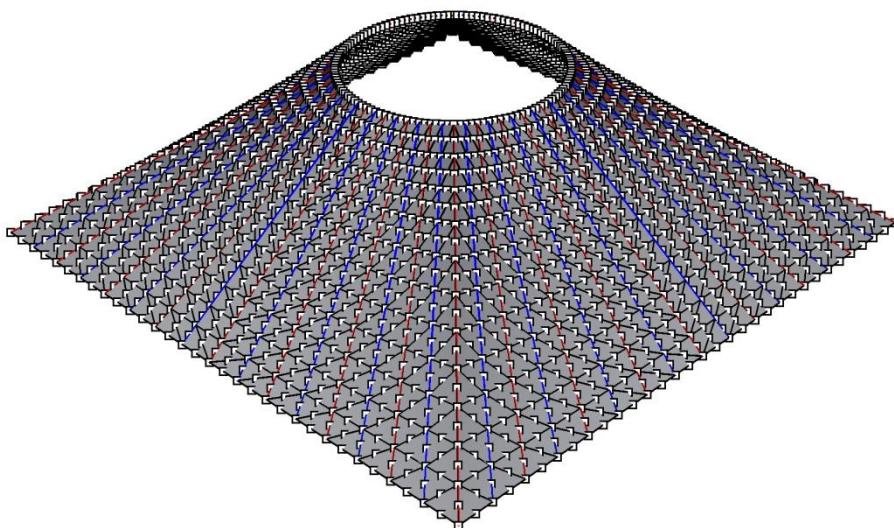
| $E_w$   | $E_f$   | $v_{wf}$ | $v_{fw}$ | $G_{wf}$ |
|---------|---------|----------|----------|----------|
| 600kN/m | 600kN/m | 0.4      | 0.4      | 30kN/m   |

**Table 3-5: Conic material properties**

Figure 3-16 demonstrates that the mesh has been found to an accuracy better than  $\pm 0.002\text{kN/m}$ . This is achieved by form finding using a convergence criteria or maximum out of balance force of  $0.001\text{kN}$ . The analysis with material properties is completed using a maximum out of balance force of  $0.0005\text{kN}$ . The convergence criteria is selected as a compromise between computation speed and solution accuracy. A summary of results may also be seen in Table 3-6.

Figure 3-17 showing global snow load results indicates that the maximum stress occurs in warp direction in the top third of the conic. The maximum fill stress also occurs in this area. Localised areas of negative warp strain occur at the base of each panel seam this negative strain is caused by the inconsistency of material direction within neighbouring panels. This phenomenon would not be observed where a continuous radial definition of material direction is used. Peak levels of shear stress are also observed at panel boundaries.

In order to investigate the effect of panel width the form finding and analysis process is repeated using a refined mesh and the panels measuring half the width of the initial conic. The conic is assembled from 32 panels as shown in Figure 3-15. The panel edges are shown in red the centre lines which define the material direction of each panel are shown in blue. The initial mesh made up of 1408 elements defined by 2944 nodes is shown in Figure 3-15.

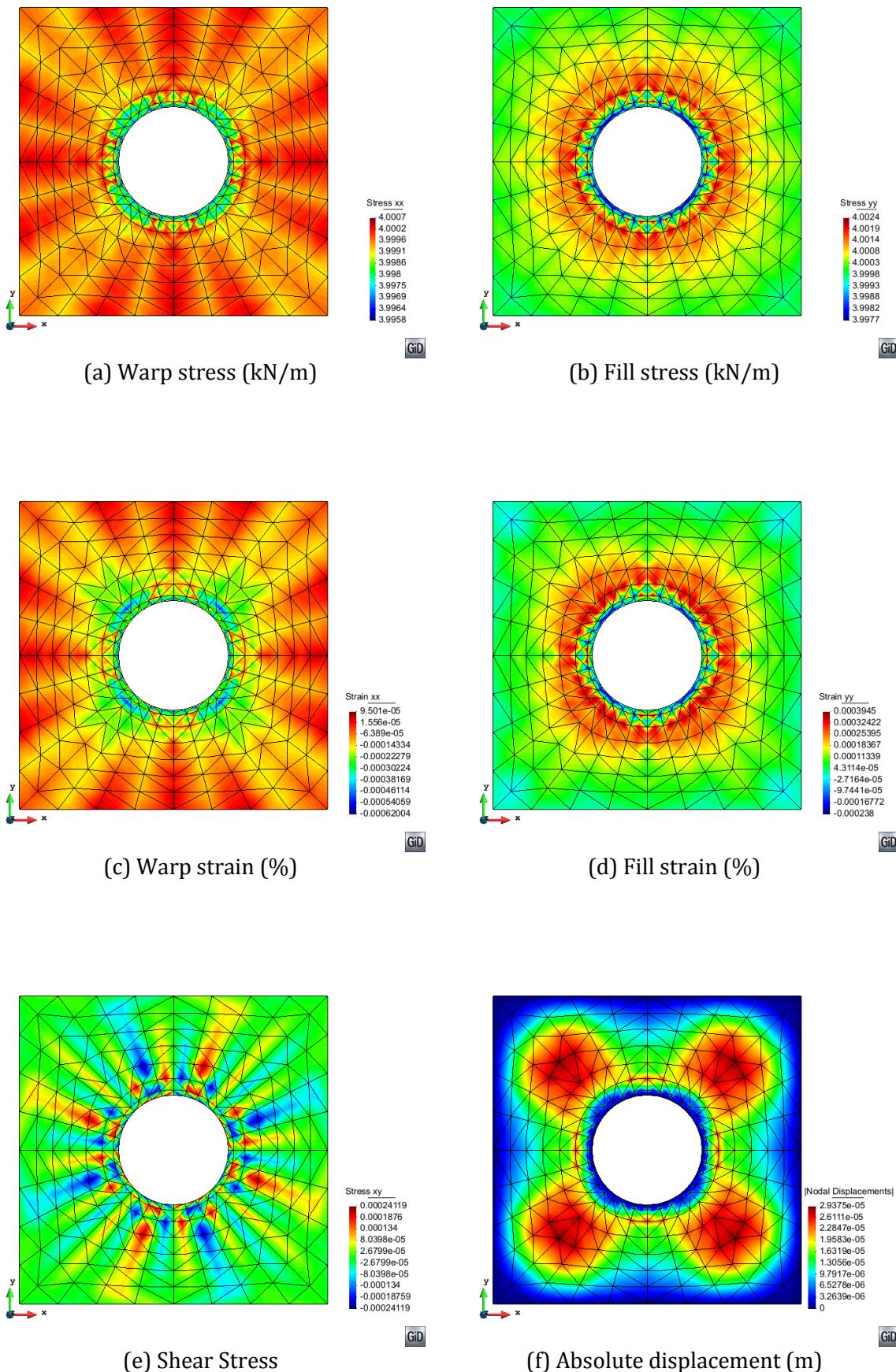
**Figure 3-15: Initial refined conic mesh**

The results of the analysis are shown in Figure 3-18. The areas of negative strain once again occur at the base of each seam however at a reduced level. The peaks in shear stress also follow the same distribution but at a reduced level. It is noted that refinement of mesh may have some effect on stress levels however the reduction is notable.

Figure 3-19 showing projected wind load results indicates that the maximum stress occurs around the central region of the conic in the radial fill direction. Again concentrated strains this time positive are observed in the warp direction at the seam at the base. Significant negative stresses are observed close to the hoop. As architectural fabrics have no stiffness in compression the wrinkling procedure is initiated in order to reduce the stiffness of elements undergoing wrinkling. This leads to the results shown in Figure 3-20. It may be observed that the top section of the conic has lost all warp prestress and the reduction of stiffness leads to unacceptably large strain, in the range of 8% in the warp direction.

|                       | Warp Stress (kN/m) |         | Fill Stress (kN/m) |        | Displacement (m) |         |          |
|-----------------------|--------------------|---------|--------------------|--------|------------------|---------|----------|
|                       | Max                | Min     | Max                | Min    | z                |         | Absolute |
|                       |                    |         |                    |        | Max              | Min     | Max      |
| Prestress             | 3.9957             | 4.0012  | 4.0035             | 3.9972 | 2.74e-5          | 0       | 2.94e-5  |
| Snow Load             | 9.6240             | 1.8271  | 8.5199             | 1.3974 | 0                | -0.1148 | 0.1156   |
| Snow Load (refined)   | 9.2074             | 2.6476  | 8.6334             | 1.3684 | 0                | -0.1100 | 0.1110   |
| Wind Load             | 10.733             | -2.6278 | 12.974             | 1.2002 | 0.1811           | 0       | 0.2025   |
| Wind Load (Wrinkling) | 11.464             | 0.0225  | 13.3050            | 2.5188 | 0.1860           | 0       | 0.2030   |

**Table 3-6: Conic results summary**

**Figure 3-16: Conic prestress results**

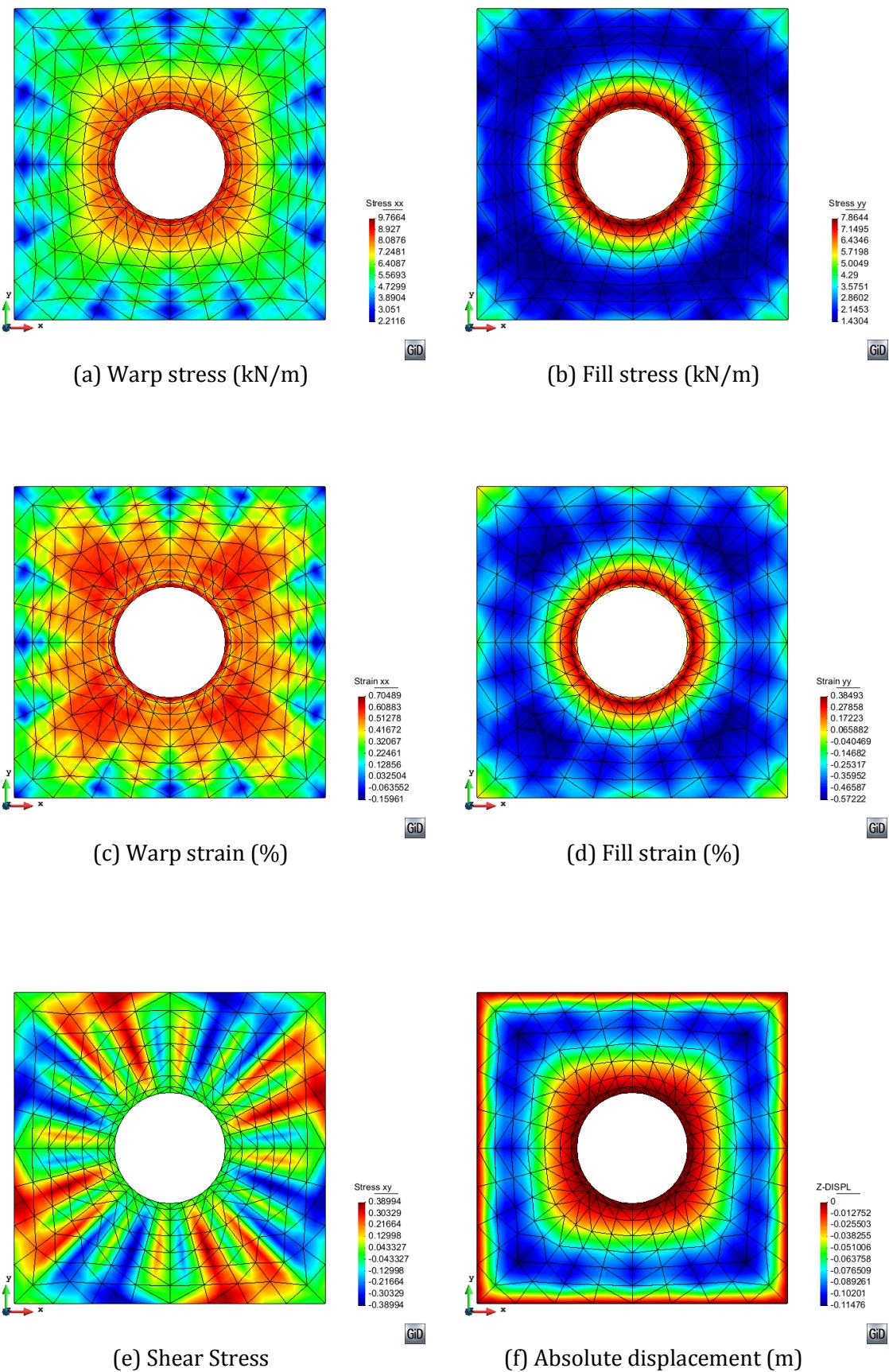


Figure 3-17: Conic result snow load

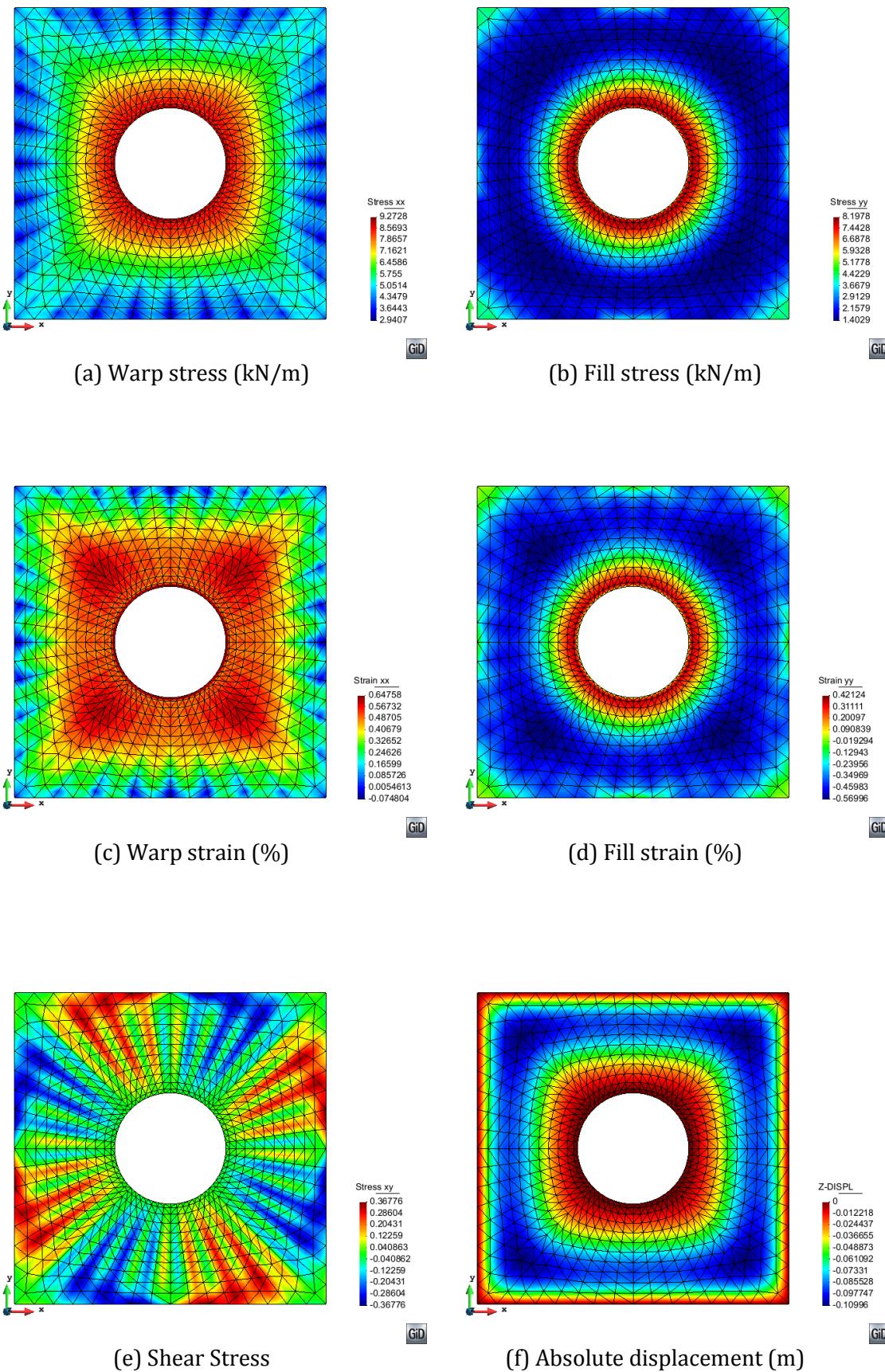
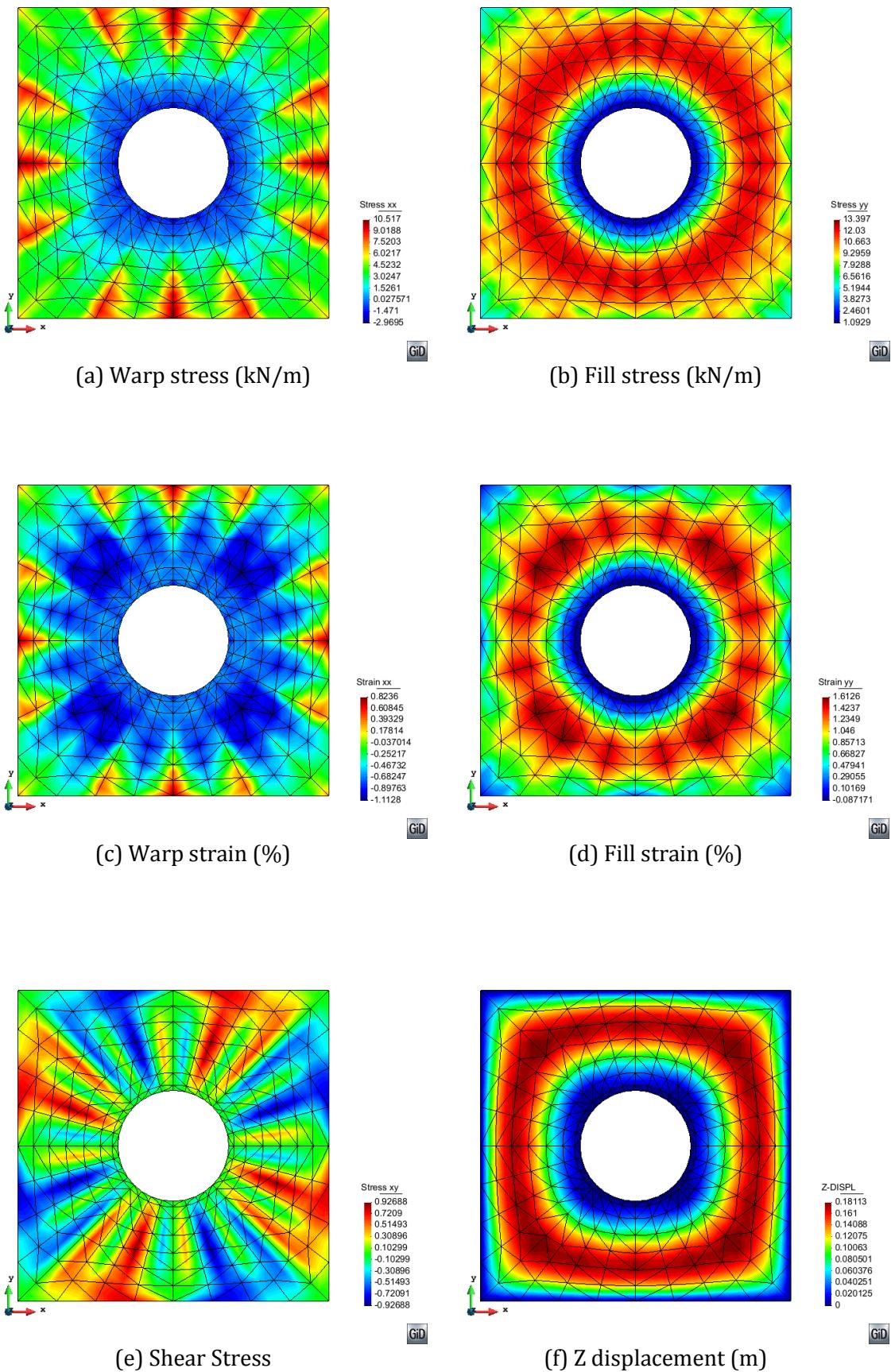
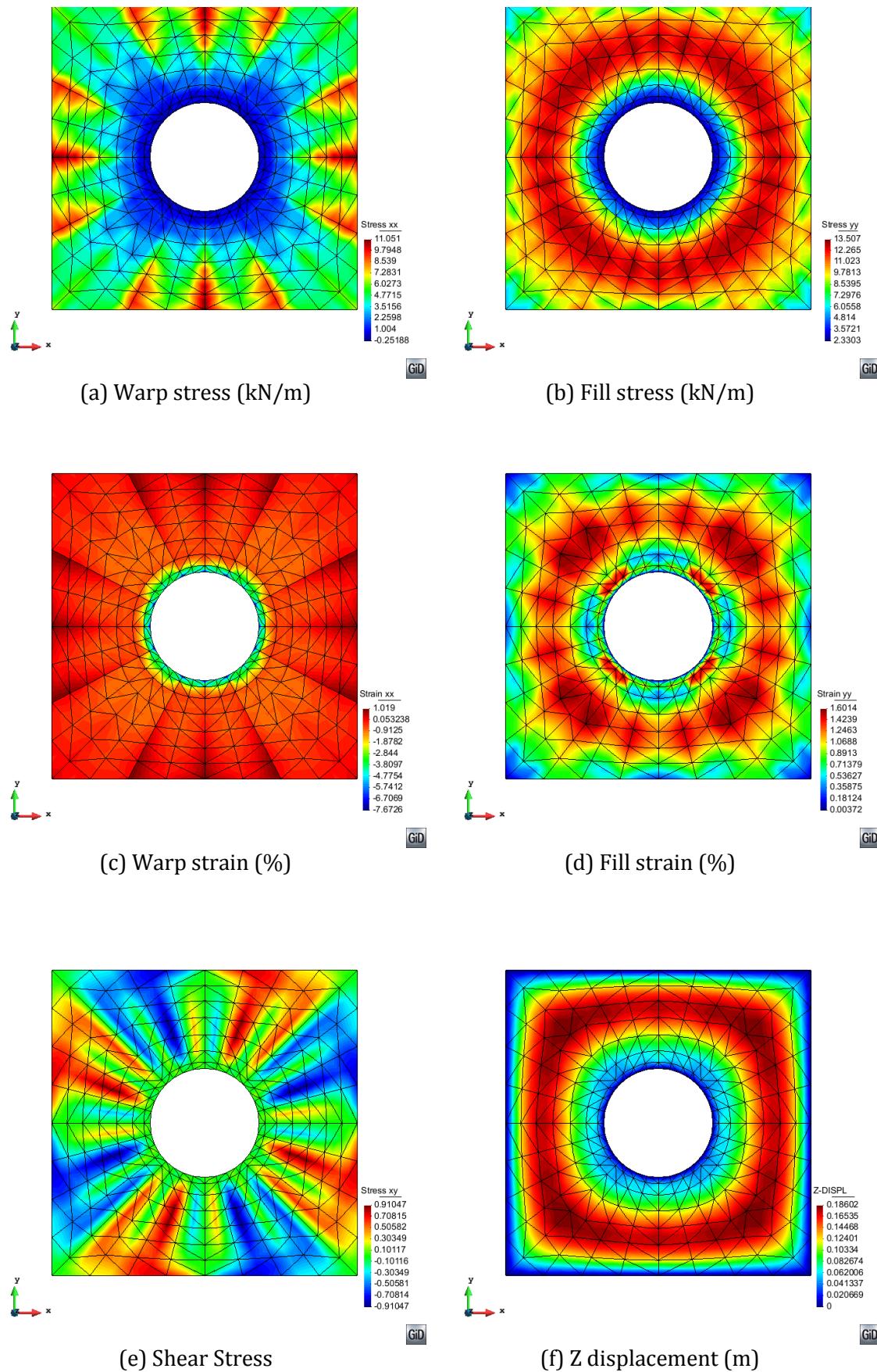


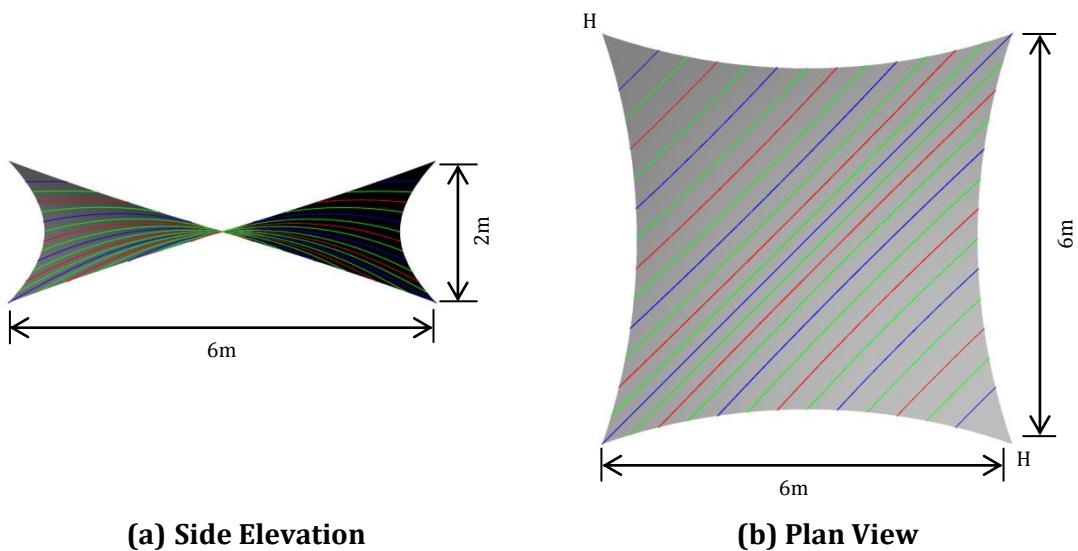
Figure 3-18: Refined conic result snow load

**Figure 3-19: Conic results wind load**

**Figure 3-20: Conic results wind load**

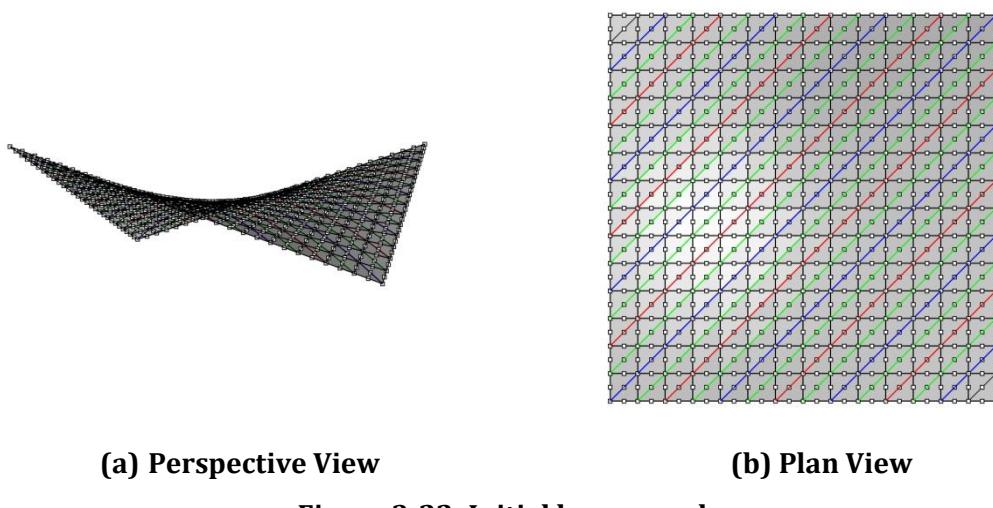
### 3.2.3 Hypar simulation

To further demonstrate the effectiveness of the geodesic control string elements form finding simulations are performed with and without control elements on a simple hypar. The hypar is based on the hypar example with equal prestress presented in [36] and is shown in Figure 3-21. The hypar is 6m square with high points at 2m. The structure is fully fixed at each corner and has 12mm diameter edge cables with a prestress of 30kN. A prestress of 3kN/m is applied to the membrane in the warp and fill material directions.



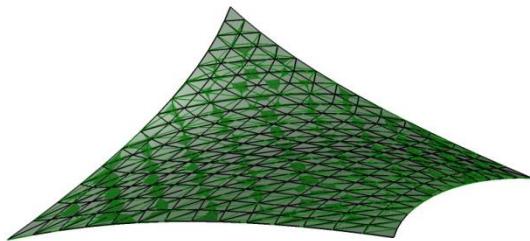
**Figure 3-21: Hypar**

The hypar is assembled from 7 panels the edges of which are shown in red in Figure 3-12 the centre lines of the panels are shown in blue and define the material warp direction. Additional geodesic control elements are shown in green. The initial mesh made up of 392 elements defined by 841 nodes is shown in Figure 3-22.

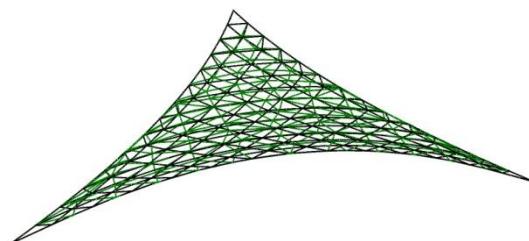


**Figure 3-22: Initial hypar mesh**

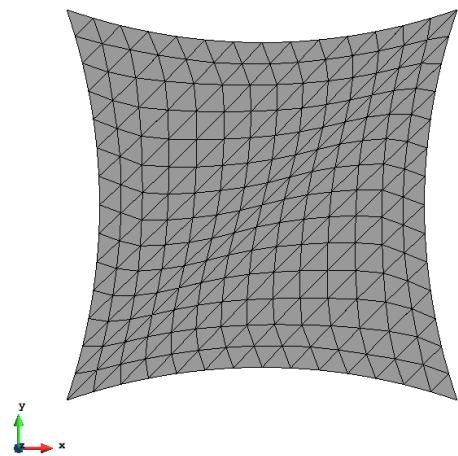
The hypar is form-found with and without geodesic control elements the resulting mesh configurations are shown in Figure 3-23(a-d).



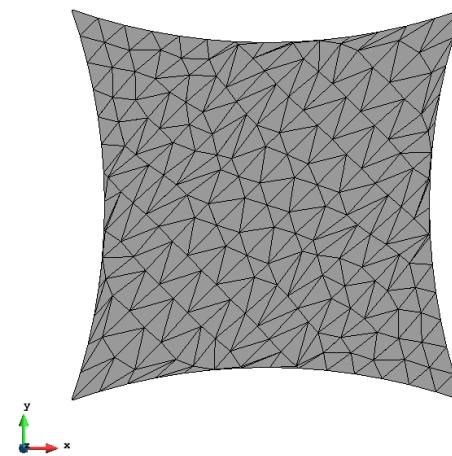
**(a) Over laid form found meshes, with geodesic (black) without and geodesic (green) control**



**(b) Diagonal cut through of over laid meshes with geodesic (black) and without geodesic (green) control**



**(c) Form found mesh with geodesic control elements aligned with seams and centre lines**



**(d) Form found mesh without geodesic control elements**

**Figure 3-23: Form found hypar mesh**

Figure 3-23 (a) and (b) demonstrate that the form found mesh surfaces are equivalent. Differences occur due to the positioning of the nodes however the overall shape is the same with and without geodesic control elements. The resulting meshes are more clearly shown with and without geodesic control elements in Figure 3-23 (c) and (d) respectively. It can be seen that the seams and centre lines become distorted without the control elements. Without control elements numerous LST elements at the mesh edges collapse completely, this leads to a failure to converge at lower convergence criteria or when using a more refined mesh.

As in the case of the conic the material properties shown in Table 3-7 are applied to the form found mesh and 3 load analyses are undertaken to demonstrate the formulation with material properties. The first analysis checks the uniformity of prestress by applying no load. The second load case simulates a wind uplift load of 1kN/m applied perpendicular to the surface. The third load case simulates a snow load of 0.6kN/m acting in the global z direction.

| $E_w$   | $E_f$   | $v_{wf}$ | $v_{fw}$ | $G_{wf}$ |
|---------|---------|----------|----------|----------|
| 600kN/m | 600kN/m | 0.4      | 0.4      | 30kN/m   |

**Table 3-7: Hypar material properties**

Figure 3-24 demonstrates that the mesh has been form found to an accuracy better than  $\pm 0.012\text{kN/m}$ . This is achieved by form finding using a convergence criteria or maximum out of balance force of 0.001kN. The analysis with material properties is completed using a maximum out of balance force of 0.0005kN. Convergence criteria is selected as a compromise between computation speed and solution accuracy. All numerical simulations provide only approximate solutions to the physical real world problems.

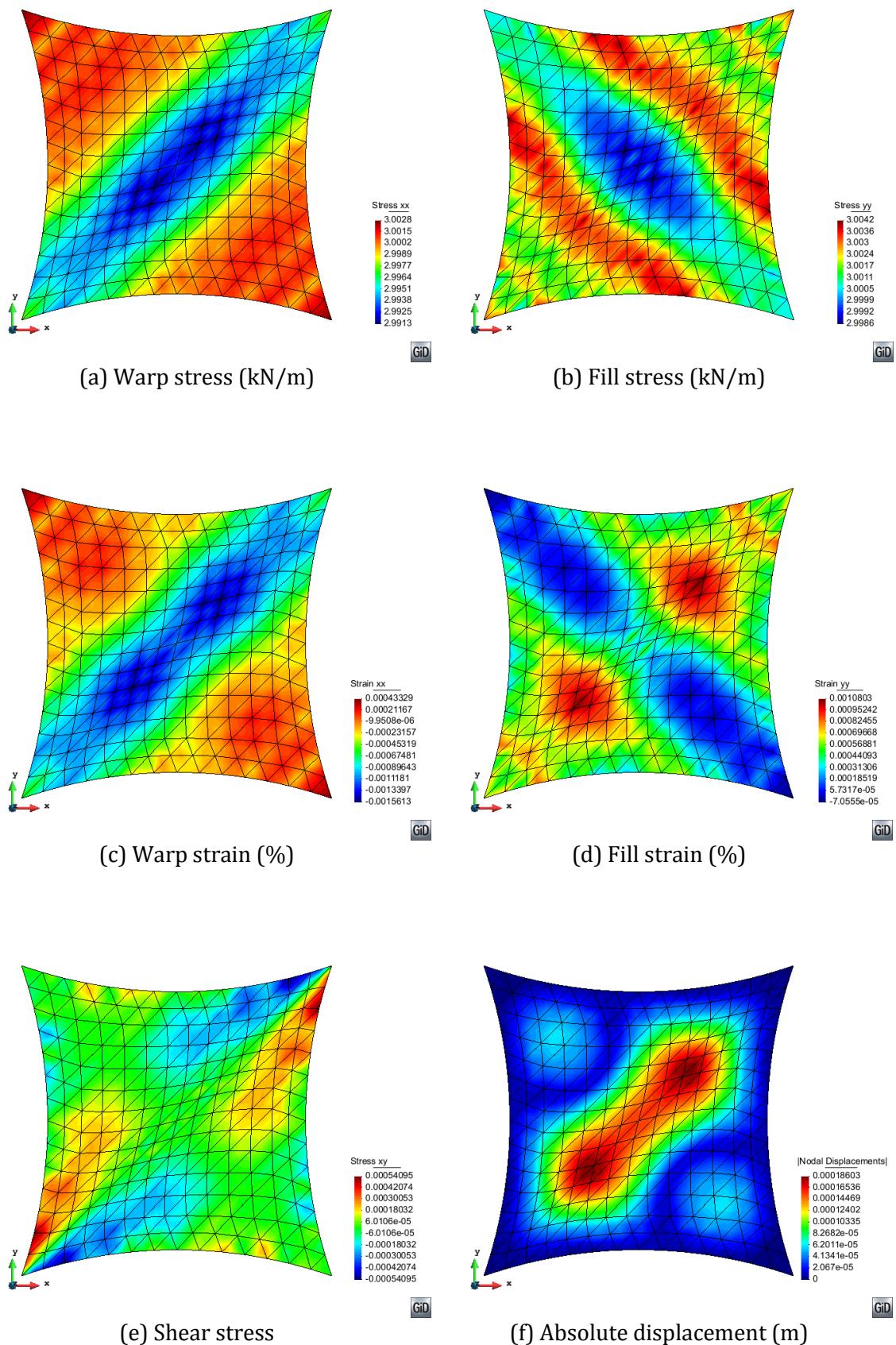
Figure 3-25 showing global snow load results indicates that the maximum stress occurs in fill direction between the hypar high points. There is a rapid stress change at the high point corners which may be a feature for consideration for design. As expected maximum displacement occurs in the centre of the membrane.

Figure 3-26 showing projected wind load results indicates that the maximum stress occurs between the low points in the warp direction. Concentrated stresses and strains in both material directions may be seen acting in the centre of the hypar in line with the warp direction. These stress concentrations are also seen in the Z-direction results. Negative stresses in the fill direction indicate that the membrane is in a wrinkled state.

|                              | Warp Stress (kN/m) |        | Fill Stress (kN/m) |         | Displacement (m) |         |          |
|------------------------------|--------------------|--------|--------------------|---------|------------------|---------|----------|
|                              | Max                | Min    | Max                | Min     | z                |         | Absolute |
|                              |                    |        |                    |         | Max              | Min     |          |
| <b>Prestress</b>             | 3.0026             | 2.9897 | 3.0044             | 2.9981  | 5.86e-5          | -0.0002 | 0.0002   |
| <b>Snow Load</b>             | 3.3081             | 0.5754 | 5.8744             | 3.069   | 0                | -0.0649 | 0.0649   |
| <b>Wind Load</b>             | 8.9381             | 3.1718 | 3.8495             | -0.2189 | 0.1192           | 0       | 0.1193   |
| <b>Wind Load (Wrinkling)</b> | 9.0208             | 3.6119 | 4.2710             | 0.3863  | 0.1104           | 0       | 0.1104   |

**Table 3-8: Hypar analysis results summary**

As the membrane has no resistance to negative or compressive loading a wrinkling procedure is implemented to reduce the stiffness of the material in the wrinkle direction. The results of the wrinkling procedure may be seen in Figure 3-27. The results are not completely satisfactory as uneven stress and strain fields are observed. This indicates that a more robust wrinkling procedure may be required to allow the smoother redistribution of stress. However, the wrinkles demonstrated by the Z-direction displacements have been substantially reduced indicating that the hypar will remain unwrinkled where compressive stiffness is eliminated. This further highlights the requirement for a more consistent material model.

**Figure 3-24: Hypar prestress results**

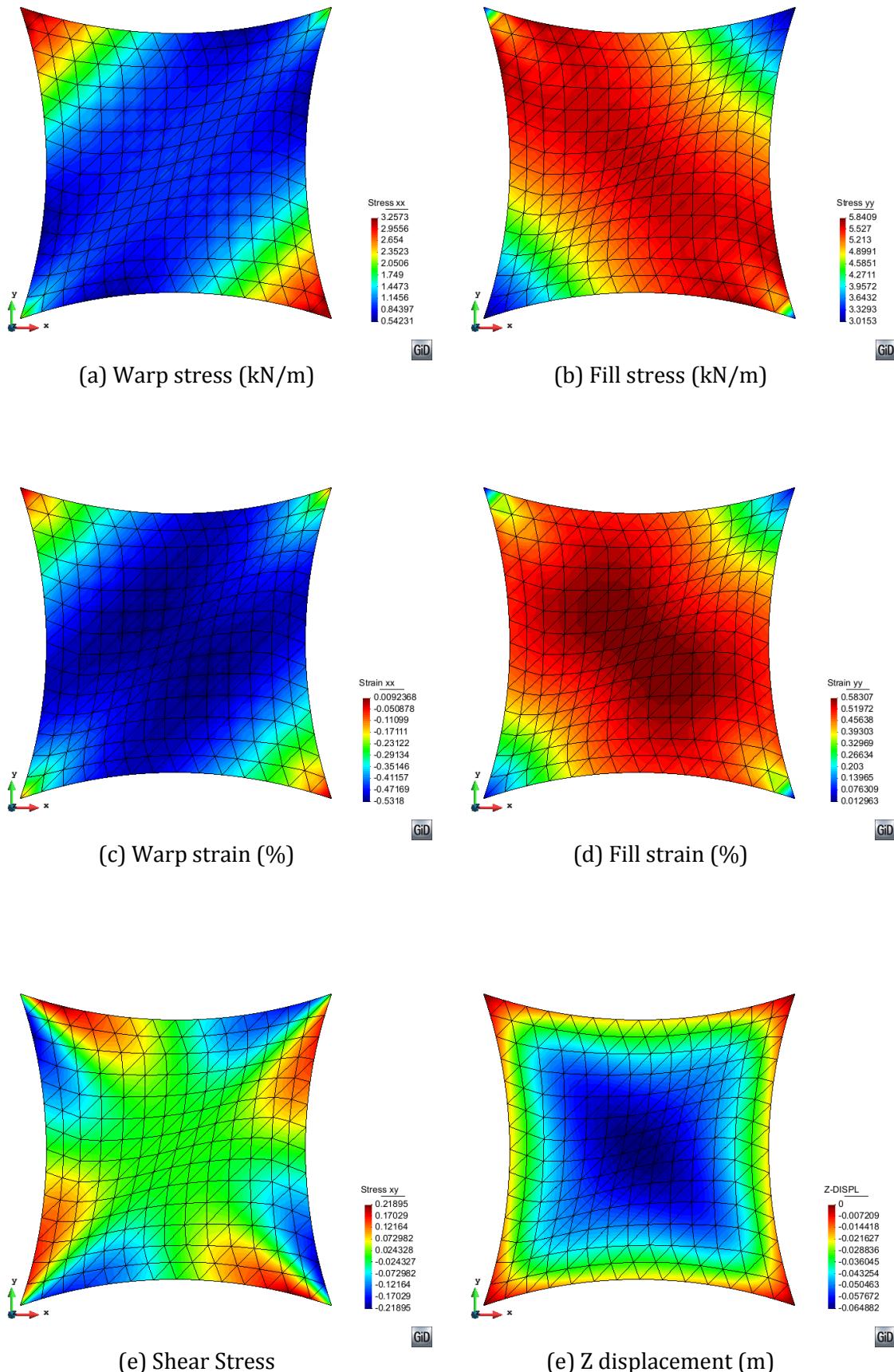


Figure 3-25: Hypar result snow load

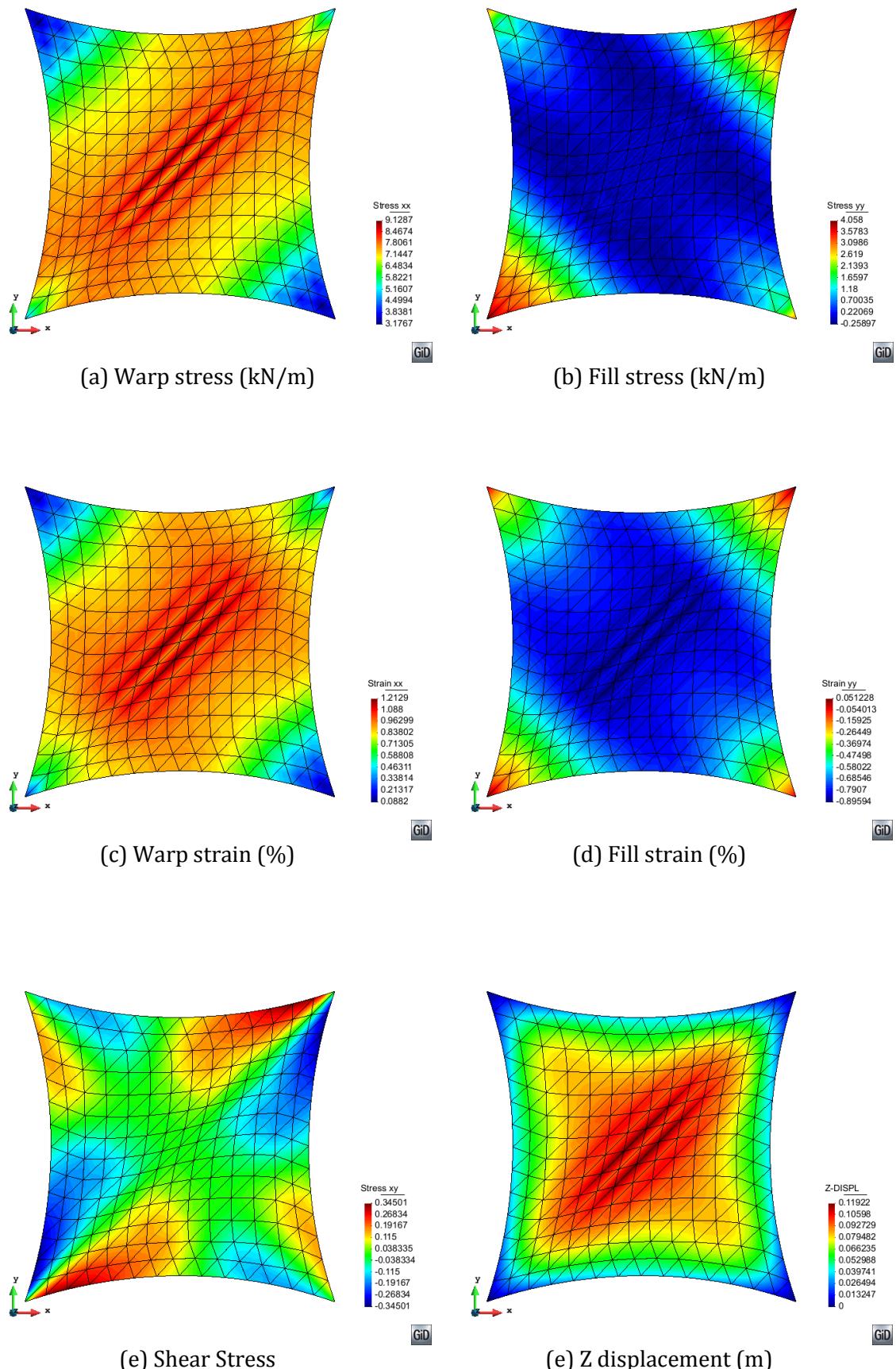
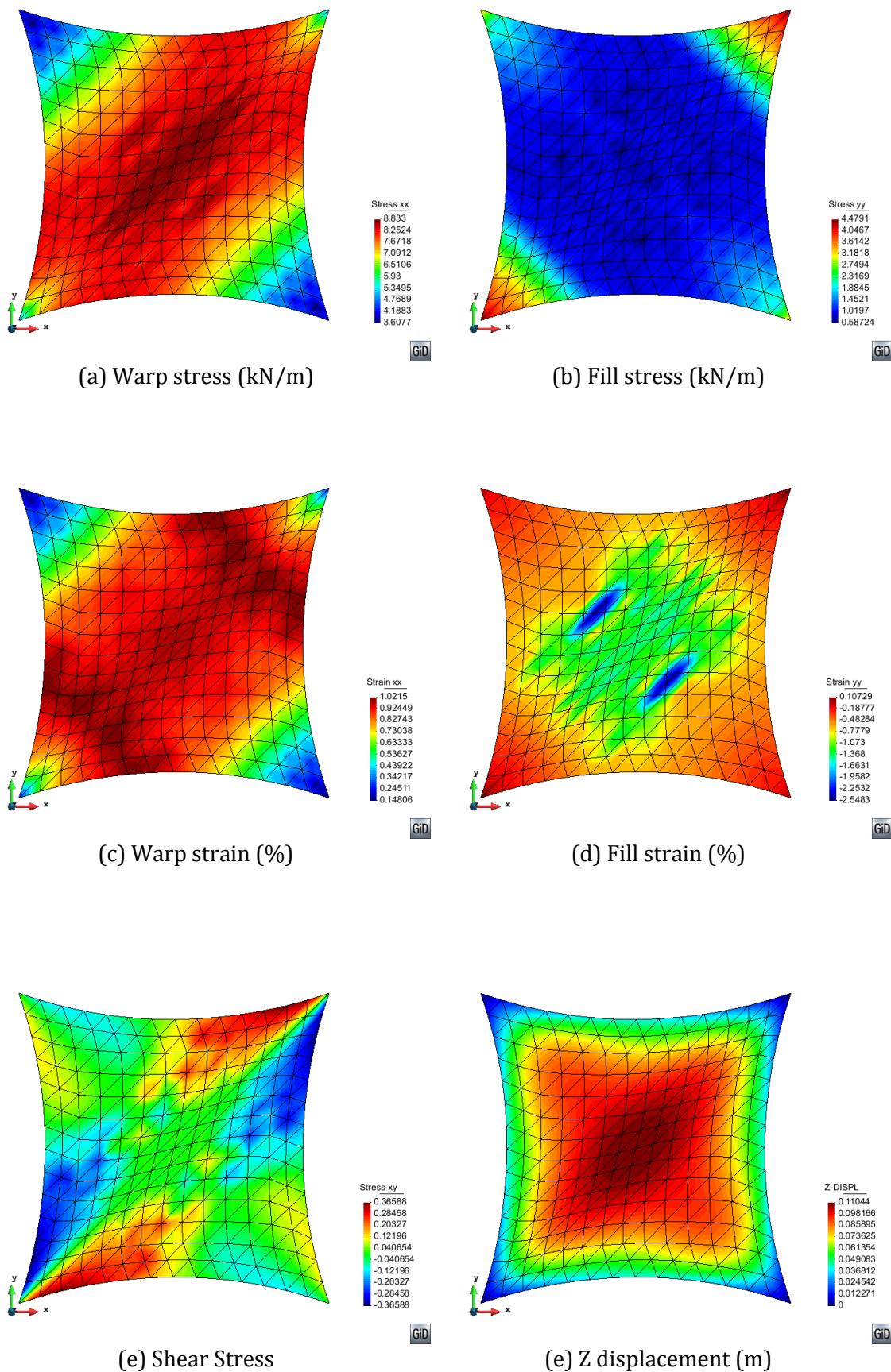


Figure 3-26: Hypar result wind load

**Figure 3-27: Hypar result wind load with wrinkling procedure**

### 3.3 Patterning

1. The final step in the design process of a fabric structure is the generation of cutting patterns. These cutting patterns are used to cut the constituent flat panels from rolls of fabric. These panels are then connected together and prestressed into their final form. In the case of a final form with highly doubly curved geometry the development of the flat panel into final 3-dimensional shape may induce uneven direct stresses and high shear stresses. Patterning using a finite element methodology allows for the investigation of the development of these 'locked-in' forces via the reverse process of flattening a 3-dimension doubly curved panel to flat cutting pattern panel. The patterning procedure, summarised in  $\{P\} = 0$  and the B matrix is based on the deformed Z=0 plane configuration, is then used to calculate the equivalent nodal loads from the stress calculated in step 5. All Z-direction equivalent nodal loads are set to zero.
2. A dynamic relaxation run is completed using the current Z=0 plane mesh and equivalent nodal loads calculated in step 6. At the kinetic energy peak the Z=0 plane mesh is updated using the resulting X and Y displacements.
3. A new applied displacement vector may then be found by subtracting the current Z=0 plane mesh geometry from the initial 3-dimensional panel mesh.
4. Step 4 to 8 is repeated until the equivalent nodal forces derived in 6 converge to close to zero (i.e.  $\{R\} < 0.00001$ ).

, is driven by internal stresses induced in the mesh elements when the mesh is projected to a flat plane.

The patterning procedure is as follows,

5. A single 3-dimensional meshed panel is isolated from the full form found mesh under consideration. The panel edges and central seam have been controlled during form finding via geodesic string elements. Depending on the configuration of the initial 3-dimensional mesh a rigid body transformation may be performed on the panel mesh order to minimise the initial applied displacement required to perform the initial flattening (step 3). This mesh is referred to as the initial 3-dimensional panel mesh.
6. All stresses within the element are set to zero and the central point of the mesh is fully restrained,

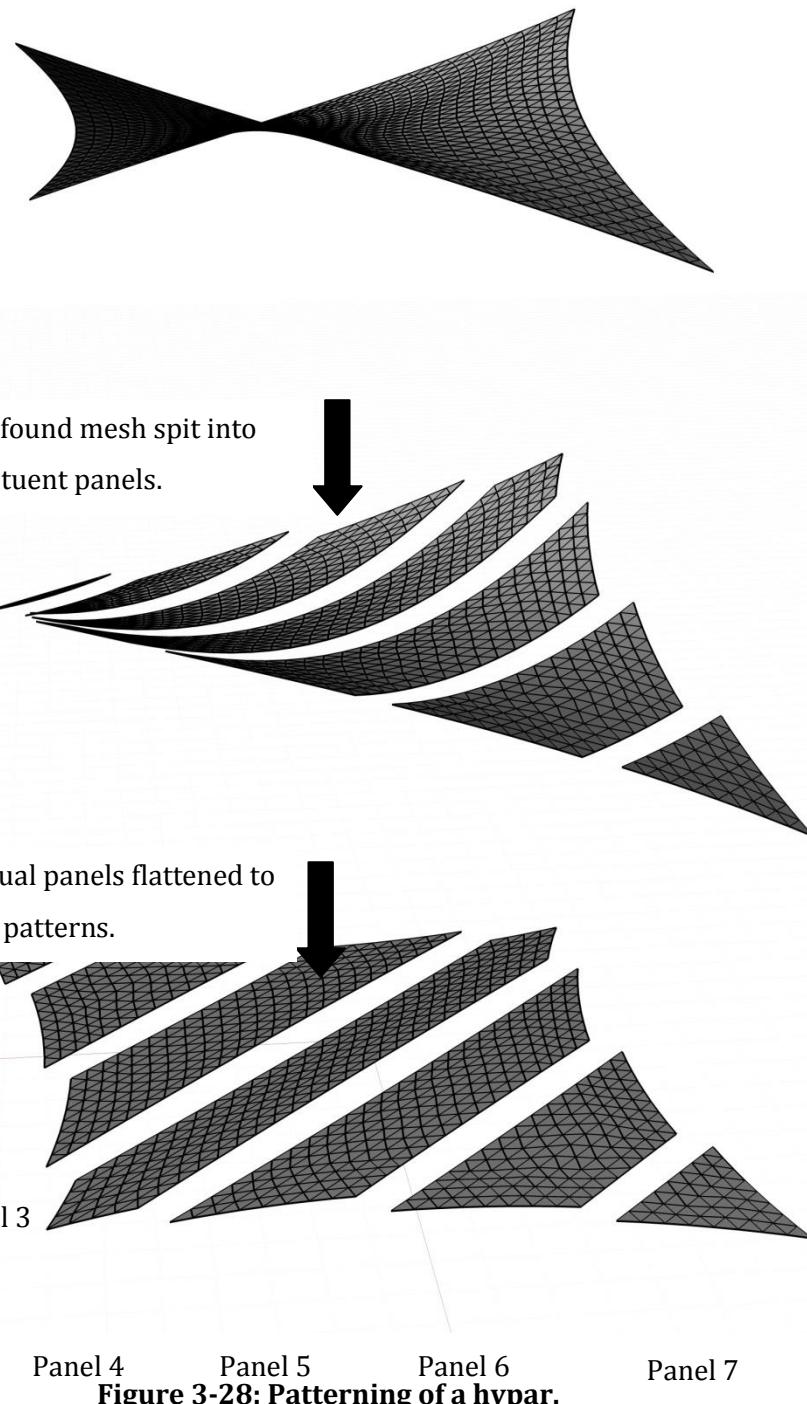
7. A vector of applied Z- displacements is derived to transform all nodes in the initial 3-dimensional mesh to the same plane, for example the Z=0 plane. The resulting mesh is referred to as Z=0 plane mesh.
8. Equation (3.32),  $\{\varepsilon\} = [{}_0B]\{\delta^t\}_i$ , with  ${}_0B$  based on the initial 3 dimensional configuration, is used to calculate the in plane strain induced in each of the elements of the flattened panel mesh. The panel is now flat but the elements are the incorrect size.
9. An arbitrary isotropic plane stress material stiffness matrix is defined in order to calculate in plane stress induced by the strain derived in step 4 using equation (3.4),  $\{\sigma\} = [E]\{\varepsilon\}$ .
10. Equation (3.34),  $\{R\} = \sum_{i=1}^{i=m} \left[ \int_V [{}_tB]^T [\sigma] dV \right]_i - \{P\}$ , where the applied load,  $\{P\} = 0$  and the B matrix is based on the deformed Z=0 plane configuration, is then used to calculate the equivalent nodal loads form the stress calculated in step 5. All Z-direction equivalent nodal loads are set to zero.
11. A dynamic relaxation run is completed using the current Z=0 plane mesh and equivalent nodal loads calculated in step 6. At the kinetic energy peak the Z=0 plane mesh is updated using the resulting X and Y displacements.
12. A new applied displacement vector may then be found by subtracting the current Z=0 plane mesh geometry from the initial 3-dimensional panel mesh.
13. Step 4 to 8 is repeated until the equivalent nodal forces derived in 6 converge to close to zero (i.e.  $\{R\} < 0.00001$ ).

If the panel is fully developable, for example is curved in only one direction the resulting flat geometry will contain no 'locked in' stress. In the case of a non-developable doubly curved panel residual stresses will remain after the patterning procedure has converged. These stresses may be output from the analysis in order to assess the magnitude of the 'locked in' stresses and used to inform the need for adjustments to the panels. It should be noted that only an arbitrary isotropic plane stress material stiffness matrix is used in the calculations of the stresses and therefore the patterning output may not be treated as actual stress only as a indication of relative magnitude,

After initial development the flattened panels would be reduced in size using experimentally determined compensation factors to account for the non-recoverable strain induced during installation in order to induce the required uniform prestress. A reliable patterning procedure is vital as the development of a uniform prestress is imperative for the safety of the structure. If the compensation is over estimated the

prestress induced may exceed the acceptable stress range and result in damage to the fabric. Inversely if the compensation is under estimated initial strains induced during installation, which are unrecoverable, may cause the membrane to become slack leading to problems such as ponding and excessive vibration under wind loading. Finally a seam allowance is required to allow for the overlap required to form a seam which will be joined via welding or some other process.

To demonstrate the process the hypar mesh, Section 3.2.3, is patterned using the proposed procedure. The initial non form found mesh has been formed from 7 fabric panels, these panels are used in patterning. The mesh has been controlled during form-finding using geodesic string elements in-order to maintain straight panel edges and a straigh central seam. After the form finding is complete it is a simple task to divide the mesh into its constituent panels. These panel meshes provide the initial geometry for the patterning process, Figure 3-28.



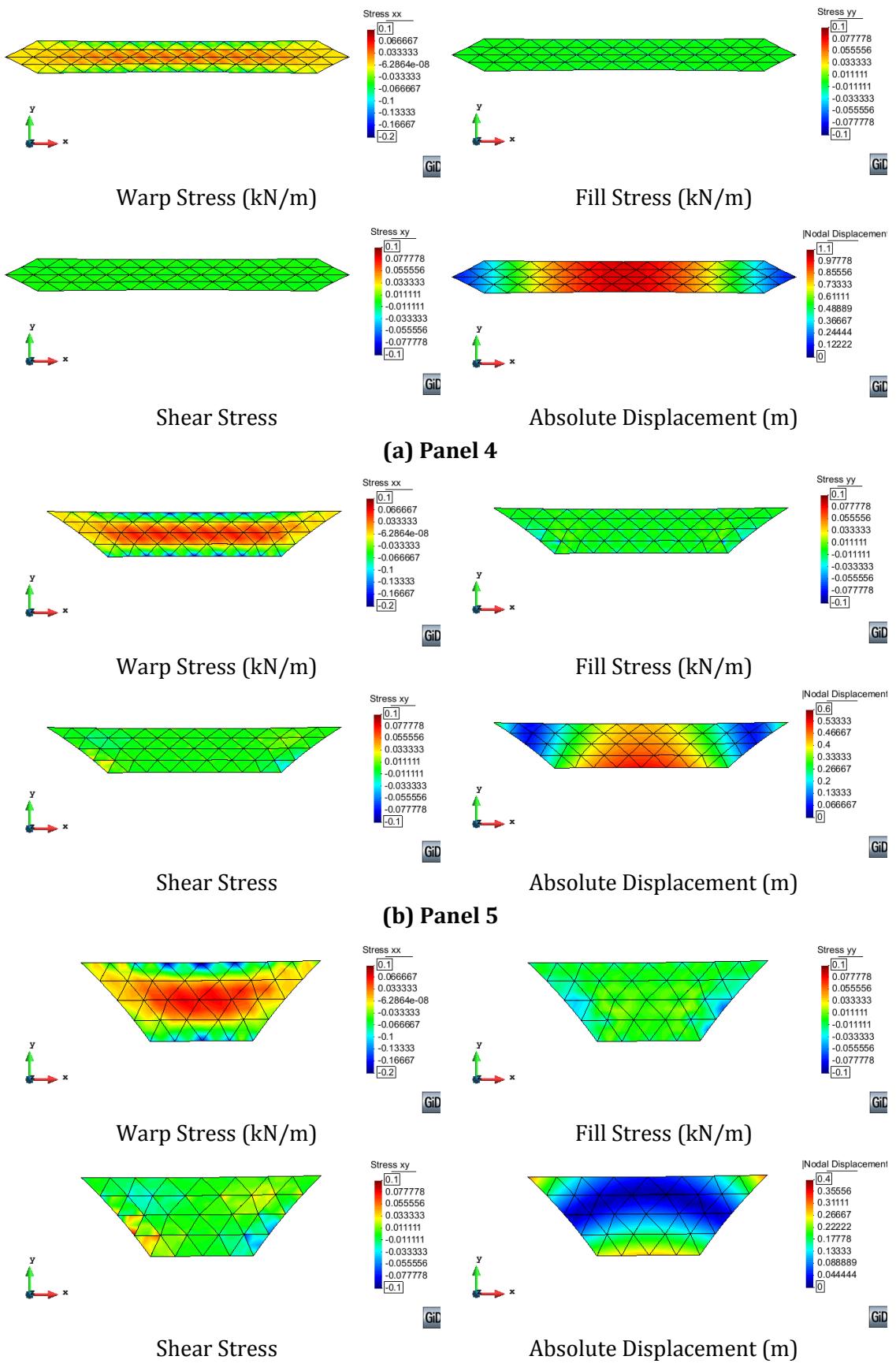


Figure 3-29: Patterning analysis output

The stress and displacement output for Panels 1 to 3 from the patterning analysis is graphically presented in Figure 3-29 in order to allow comparison. Panel 3 exhibits the highest level of residual stress while undergoing the lowest level of displacement. In contrast Panel 1 from the centre of the structure undergoes the largest absolute displacements but exhibits relatively low residual stress. This indicates that Panel 3 has greater double curvature than Panel 1. The 'locked in' stresses shown in Panel 3 may be reduced by patterning the structure with a narrower panel width in this region.

To assess the validity of the final patterned panels three metrics are selected namely panel area, perimeter length and seam compatibility. The initial and flattened areas and perimeters of the 7 panels are presented in Table 3-9 along with the percentage error between the two configurations. The seam compatibility for the 6 seams is presented in Table 3-10.

| <b>Panel</b> | <b>Configuration</b> | <b>Area<br/>(mm<sup>2</sup>)</b> | <b>Area<br/>(%error)</b> | <b>Perimeter<br/>(mm)</b> | <b>Perimeter<br/>(%error)</b> |
|--------------|----------------------|----------------------------------|--------------------------|---------------------------|-------------------------------|
| <b>1</b>     | <b>Initial</b>       | 1333751                          | 0.0000%                  | 5472.9                    | 0.0111%                       |
|              | <b>Flattened</b>     | 1333750                          |                          | 5472.3                    |                               |
| <b>2</b>     | <b>Initial</b>       | 4012297                          | 0.0001%                  | 9682.9                    | 0.0939%                       |
|              | <b>Flattened</b>     | 4012291                          |                          | 9673.8                    |                               |
| <b>3</b>     | <b>Initial</b>       | 5896534                          | 0.0017%                  | 15034.6                   | 0.0811%                       |
|              | <b>Flattened</b>     | 5896434                          |                          | 15022.4                   |                               |
| <b>4</b>     | <b>Initial</b>       | 6300090                          | 0.0041%                  | 17970.9                   | 0.0403%                       |
|              | <b>Flattened</b>     | 6299834                          |                          | 17963.6                   |                               |
| <b>5</b>     | <b>Initial</b>       | 5896521                          | 0.0015%                  | 15034.6                   | 0.0811%                       |
|              | <b>Flattened</b>     | 5896432                          |                          | 15022.4                   |                               |
| <b>6</b>     | <b>Initial</b>       | 4012278                          | 0.0002%                  | 9682.9                    | 0.0940%                       |
|              | <b>Flattened</b>     | 4012272                          |                          | 9673.7                    |                               |
| <b>7</b>     | <b>Initial</b>       | 1333772                          | 0.0001%                  | 5472.9                    | 0.0113%                       |
|              | <b>Flattened</b>     | 1333771                          |                          | 5472.3                    |                               |

**Table 3-9: Initial and patterned panel comparison**

Percentage error between initial panel area and flattened panel area is less than 0.005%. The maximum absolute error of 256.2mm<sup>2</sup> is observed in Panel 4 which is the largest panel. Panel 6 exhibits the maximum percentage error between initial panel perimeter length and flattened panel perimeter length of 0.094%. This equates to a maximum absolute change in perimeter length of 12.2mm. while the absolute values appear

significant when considered as a percentage they are deemed acceptable. Especially when the approximated compensation discussed above is to be applied to the patterned panel.

| <b>Seam</b> | <b>Panel</b> | <b>Seam Length (mm)</b> | <b>Absolute error (mm)</b> | <b>Required % length change</b> |
|-------------|--------------|-------------------------|----------------------------|---------------------------------|
| <b>1</b>    | 1            | 1805.6                  | 1.95                       | -0.0540%                        |
|             | 2            | 1803.6                  |                            | 0.0541%                         |
| <b>2</b>    | 2            | 4197.2                  | 0.98                       | -0.0117%                        |
|             | 3            | 4196.2                  |                            | 0.0117%                         |
| <b>3</b>    | 3            | 7143.0                  | 2.12                       | 0.0149%                         |
|             | 4            | 7145.1                  |                            | -0.0149%                        |
| <b>4</b>    | 4            | 7145.1                  | 2.11                       | -0.0148%                        |
|             | 5            | 7143.0                  |                            | 0.0148%                         |
| <b>5</b>    | 5            | 4196.2                  | 0.99                       | 0.0118%                         |
|             | 6            | 4197.2                  |                            | -0.0118%                        |
| <b>6</b>    | 6            | 1803.6                  | 1.95                       | 0.0542%                         |
|             | 7            | 1805.6                  |                            | -0.0541%                        |

**Table 3-10: Seam compatibility**

A more significant measure of validity is the compatibility of the panel edges which will joined at seams to form the final structure. A maximum incompatibility of 2.12mm is observed in the seam between panels 3 and 4. If the incompatibility is distributed evenly between the two panels edges Panel 4's edge length would need to be reduced by -0.022% and Panel 3's length increased by 0.015%. The largest required length change occurs in seam 1 and 6 and is equal to 0.054%. However, such incompatibilities may well be within the construction tolerance for the structure and may be accounted for in the seam allowance.

### 3.4 Summary and Conclusions

This chapter describes the derivation for the formulation of a six node Linear Strain Triangle (LST) element including large strain terms. The element is implemented within finite-element analysis code and is demonstrated using a simple patch along with two simple but realistic fabric structure types namely a square base conic and a hypar.

Alongside the LST membrane element a liner cable element and special geodesic control element are also developed and implemented.

The general element formulation is validated using a simple patch. Analysis results show that the displacement dependent terms of the B-matrix should include a factor of a half. The updated co-ordinate system is also required when calculating the nodal out-of balance forces from the current element stress.

The geodesic control elements are demonstrated using both the hypar and conic structures. The control elements are shown to have a limited impact on the final membrane shape however they are vital in controlling the distribution of elements within the form found mesh. It is important for the seams and centre lines of the mesh panels to remain straight in order to enable accurate definition of material direction for load analysis and also enable patterning of the individual mesh panels.

The conic results are of particular interest as the width of the panel had some effect on the final stress distribution given by the analysis. This is due to the changes in material direction at the panel seams. This gives a more accurate representation of stress within a real structure than defining the material direction using a continuous radial local coordinate system as used by other specialist fabric analysis software, for example Oasys GSA.

The conic undergoing wind uplift loading underwent unacceptable high negative strain close to the hoop indicating that this structure would fail under such loading conditions.

A wrinkling procedure is implemented where an element is deemed to have become wrinkled. A combined wrinkling criteria is used i.e. an element is deemed to be wrinkled when the minimum principle stress is negative and the maximum principle strain is positive. Where wrinkling occurs the elastic modulus is modified in the direction of the minimum principle stress in order to reduce the stiffness of the wrinkled element in the wrinkling direction. Where the element is deemed to be slack i.e. the maximum principle stress is negative the stiffness is reduced in all directions. The procedure is demonstrated

by the hypar undergoing a projected wind load of 1kN/m. Discontinuities in the stress field indicate that this procedure requires improvement.

A patterning procedure for a panelled hypar mesh has been presented. The validity of the flattened panels has been investigated through comparison panel area, perimeter length and seam compatibility before and after flattening. Incompatibility between panels at the seams may indicate that implementation of seam length control may improve the patterning procedure. Nevertheless, the relatively low levels of incompatibility between initial and flattened panel geometries demonstrates that this patterning procedure shows promise. The patterning analysis output offers a valuable insight into flattening of the proposed mesh. The results may be used to improve the design of the structural mesh in order to reduce the 'locked in' stresses induced in the flattened panels therefore improving the potential uniformity of the prestress induced in the final in situ fabric structure.



**Chapter 4.**  
**Neural Network Training and Validation**



## Contents

|   |     |
|---|-----|
| Chapter 4. Neural Network Training and Validation.....                  | 139 |
| 4.1 Neural Networks .....   | 146 |
| 4.1.1 Network architecture .....  | 146 |
| 4.1.2 Network training .....  | 149 |
| 4.2 Biaxial Testing.....  | 155 |
| 4.2.1 Cruciform sample preparation .....                                | 155 |
| 4.2.2 Test rig .....  | 156 |
| 4.3 Response Surface Neural Network Material Model.....                 | 157 |
| 4.3.1 Network performance criteria.....                                 | 157 |
| 4.3.2 Response surface training data collection and pre-processing..... | 158 |
| 4.3.3 Response Surface Neural Network Training and Validation.....      | 165 |
| 4.4 History Neural Network Material Model.....                          | 177 |
| 4.4.1 Network internal variables [114].....                             | 177 |
| 4.4.2 Cyclic uniaxial training data collection and pre-processing.....  | 178 |
| 4.4.3 Cyclic uniaxial network training and validation.....              | 180 |
| 4.4.4 Cyclic biaxial training data collection and pre-processing.....   | 188 |
| 4.4.5 Cyclic biaxial network training and validation .....              | 195 |
| 4.5 Summary and Conclusions.....  | 209 |

## Figures

|  |     |
|--|-----|
| Figure 4-1: Network Architecture .....                                 | 147 |
| Figure 4-2: A General Neuron .....                                     | 147 |
| Figure 4-3: Cruciform Sample Dimensions (Adapted from [59]).....       | 156 |
| Figure 4-4: Load ratio arms investigated during PVC biaxial test ..... | 159 |
| Figure 4-5: Response surface derived from PVC biaxial test data.....   | 159 |
| Figure 4-6: Load ratio arms investigated during PTFE biaxial test..... | 159 |

|  |     |
|--|-----|
| Figure 4-7: Response surface derived from PTFE biaxial test data.....  | 159 |
| Figure 4-8: Testing profile for PVC coated polyester fabric .....  | 160 |
| Figure 4-9: Testing profile for PTFE coated glass fibre fabric.....  | 161 |
| Figure 4-10: Generation of additional training and testing data set.....   | 162 |
| Figure 4-11: Stress to strain neural network architecture .....  | 163 |
| Figure 4-12: Response surface neural network figure key.....   | 163 |
| Figure 4-13: Network 7, Table 4.2, PVC (502S) stress to strain network<br>demonstrating best performance .....   | 164 |
| Figure 4-14: : Network 8, Table 4.3, PTFE (B18059) stress to strain network<br>demonstrating best performance .....  | 164 |
| Figure 4-15: Strain to stress neural network architecture .....  | 165 |
| Figure 4-16: Network 10, Table 4.4, PVC (502S) strain to stress network demonstrating<br>best performance with experimentally generated data used in training .....                | 166 |
| Figure 4-17: Network 5, Table 4.4, PVC (502S) strain to stress network demonstrating<br>best performance with 'unseen' network generated data used in testing .....                | 166 |
| Figure 4-18: Network 1, Table 4.5 , PVC (502S) strain to stress network demonstrating<br>best performance with both data sets .....  | 167 |
| Figure 4-19: Network 9, Table 4.6, PVC (502S) strain to stress network demonstrating<br>best performance with both data sets .....   | 168 |
| Figure 4-20: Network 4, Table 4.7, PTFE (B18059) strain to stress network demonstrating<br>best performance with experimentally generated data used in training .....              | 169 |
| Figure 4-21: Network 5, Table 4.7, PTFE (B18059) strain to stress network demonstrating<br>best performance with 'unseen' network generated data used in testing .....             | 169 |
| Figure 4-22: Network 3, Table 4.8, PTFE (B18059) strain to stress network<br>demonstrating best performance with experimentally generated data<br>partially used in training ..... | 171 |
| Figure 4-23: Network 1, Table 4.6, PTFE (B18059) strain to stress network<br>demonstrating best performance with 'unseen' experimentally generated<br>data used in testing.....    | 172 |
| Figure 4-24: 3 input strain to stress neural network architecture.....   | 173 |
| Figure 4-25: Network 4, Table 4.10, PTFE (B18059) 3 input strain to stress network<br>demonstrating best performance with 'unseen' network generated data<br>used in testing.....  | 174 |
| Figure 4-26: Network 2, Table 4.11, PTFE (B18059) 3 input strain to stress network<br>demonstrating best performance with both data sets.....                                      | 175 |

|  |     |
|--|-----|
| Figure 4-27: Network 5, Table 4.10, PTFE (B18059) 3 input strain to stress network demonstrating best performance with 'unseen' experimentally generated data used in testing..... | 176 |
| Figure 4-28: Variable sign combinations for strain controlled hysteresis loop.....   | 178 |
| Figure 4-29: Uniaxial cyclic load profiles 1,2 and 3, load is shown in terms of percentage of ultimate tensile strength for use with a range of fabrics .....                      | 179 |
| Figure 4-30: 4 input strain to stress uniaxial hysteresis neural network architecture .....  | 180 |
| Figure 4-31: The effect of hidden node number on performance.....  | 184 |
| Figure 4-32: Network output tested in recurrent mode using training data gathered using Profile 3.....   | 186 |
| Figure 4-33: Network output tested in recurrent mode using previously unseen data generated from profile 1.....  | 187 |
| Figure 4-34: Network output tested in recurrent mode using previously unseen data generated from profile 2.....  | 187 |
| Figure 4-35: Biaxial load ratio arms .....   | 189 |
| Figure 4-36: PVC Biaxial 1:1 stress profile and results for history network training and validation.....   | 190 |
| Figure 4-37: PTFE Biaxial 1:1 stress profile and results for history network training and validation.....  | 191 |
| Figure 4-38: PVC Biaxial 1:1 stress and resulting strain profiles divided into training and testing data respectively .....  | 193 |
| Figure 4-39: PTFE Biaxial 1:1 stress and resulting strain profiles divided into training and testing data respectively .....   | 194 |
| Figure 4-40: 8 input strain to stress biaxial hysteresis neural network architecture.....  | 195 |
| Figure 4-41: Box and whisker diagrams of PVC network model testing performance ( $R^2$ values) for warp material direction .....   | 196 |
| Figure 4-42: Box and whisker diagrams of PVC network model testing performance ( $R^2$ values) for warp material direction.....  | 197 |
| Figure 4-43: Box and whisker diagrams of PTFE network model testing performance ( $R^2$ values) for warp material direction.....   | 198 |
| Figure 4-44: Strain stress plots from PVC network tested with full profiles in recurrent mode .....  | 201 |
| Figure 4-45: PVC network tested with 1:1 profile in recurrent mode .....   | 203 |
| Figure 4-46: Strain stress plots from PTFE network tested with full profiles in recurrent mode .....   | 205 |
| Figure 4-47: PTFE network tested with 1:1 profile in recurrent mode.....   | 207 |

## Tables

|   |     |
|---|-----|
| Table 4.1: Maximum and minimum stress for biaxial profiles .....  | 158 |
| Table 4.2: Coefficients of determination for PVC coated polyester (502S) stress to strain neural network material model .....   | 164 |
| Table 4.3: Coefficients of determination for PTFE coated glass fibre (B18059) stress to strain neural network material model .....                                      | 164 |
| Table 4.4: Coefficients of determination for PVC coated polyester (502S) strain to stress neural network material model trained with 9 experimental load ratios.....    | 165 |
| Table 4.5: Coefficients of determination for PVC coated polyester (502S) strain to stress neural network material model, trained with 5 load ratios.....                | 167 |
| Table 4.6: Coefficients of determination for PVC coated polyester (502S) strain to stress neural network material model, trained with network generated data .....      | 168 |
| Table 4.7: Coefficients of determination for PTFE coated glass fibre (B18059) strain to stress neural network material model, trained with 9 load ratios.....           | 169 |
| Table 4.8: Coefficients of determination for PTFE coated glass fibre (B18059) strain to stress neural network material model, trained with 5 load ratios.....           | 171 |
| Table 4.9: Coefficients of determination for PTFE coated glass fibre (B18059) strain to stress neural network material model, trained with network generated data.....  | 172 |
| Table 4.10: Coefficients of determination for PTFE coated glass fibre (B18059) strain to stress neural network material model, trained with 9 load ratios.....          | 174 |
| Table 4.11: Coefficients of determination for PTFE coated glass fibre (B18059) strain to stress neural network material model, trained with 5 load ratios.....          | 175 |
| Table 4.12: Coefficients of determination for PTFE coated glass fibre (B18059) strain to stress neural network material model, trained with network generated data..... | 176 |
| Table 4.13: Variable sign combinations for strain controlled hysteresis loop .....  | 178 |
| Table 4.14: $R^2$ value matrix for 10 hidden node networks.....   | 181 |
| Table 4.15: $R^2$ value matrix for 7 hidden node networks .....   | 182 |
| Table 4.16: $R^2$ value matrix for 4 hidden node networks .....   | 182 |
| Table 4.17: Maximum and minimum stress for biaxial profiles .....   | 188 |

## Chapter 4. Neural Network Training and Validation

As discussed in Chapter 3 fitting a plane stress model to biaxial test data for typical architectural fabrics leads to inconsistencies between the physical and theoretical descriptions, with values of Poisson's ratio in excess of the compressibility limit of 0.5, and for some fabrics approaching 2.0 [124]. The allowable values of Poisson's ratio are also constrained by the requirement that the product of the two orthotropic Poisson's ratios is less than 1, equation (2.5), Section 2.2. When designed with appropriate architecture neural networks are capable of capturing highly non-linear response. With the addition of historical inputs and internal variables it has been demonstrated that neural network models are capable of representing complex history dependant behaviour [114], Section 2.3.

This chapter begins by introducing the feed forward neural network, its basic architecture and the chosen training algorithm. The process of creating a neural network material model is then presented in terms of 3 stages. The first stage involves the collection and processing of training and testing data. This requires the development of specialised uniaxial and biaxial testing profiles. The second stage involves training of the neural network; this process includes choice of network architecture, activation functions, and other parameters. The third stage involves validation of trained network's performance using 'unseen' data sets. Implementation of the trained and validated network model for simulation and prediction of structural response within a finite element analysis programme, is covered in depth in Chapter 5.

Initially the fabric response is modelled with residual strain removed. This form of model maps a surface to the response in each of the material directions (warp and fill). Therefore, it is referred to as a response-surface neural network model. This model is comparable with existing plane-stress architectural fabric material models and utilises similar experimental data in its development. The neural network model is then extended to include load history requiring the development of specialist testing. This form of model is referred to as a load-history neural network model and is used initially to model uniaxial response followed by biaxial response. All trained networks are validated using 'unseen' data not used in training.

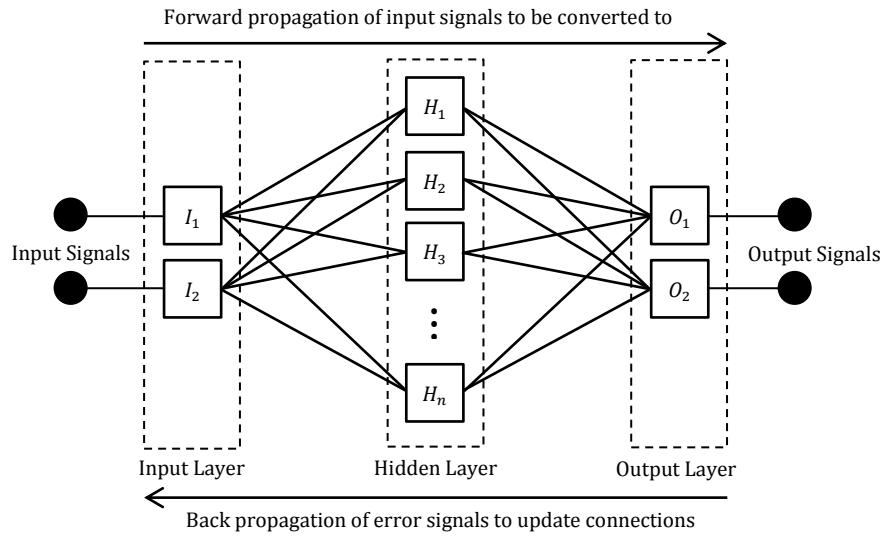
## 4.1 Neural Networks

Neural networks are an artificial intelligence concept; through a process of training they may become capable of capturing the relationship between sets of input and output data. The trained network may then be used to generate output from previously unseen inputs. It is the neural networks ability to generalise that makes them such a powerful tool, Section 2.3.

### 4.1.1 Network architecture

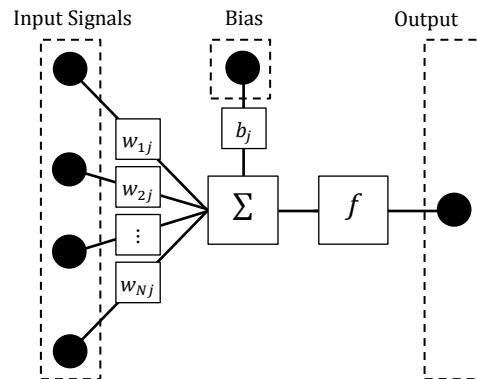
The architecture of a feed forward neural network takes on a layered form. Each layer contains a number of neurons which in turn are connected to the neurons of the next layer via weighted connections. Connections between neurons in the same layer are not permitted. In usual operation, information only travels forward through the network, hence the network is referred to as a feed-forward neural network. However, during training information is also passed backwards. The most commonly used training algorithm with feed-forward neural networks is the back-propagation training algorithm.

A graphical representation of a general multilayer feed-forward neural network is shown in Figure 1. This form of network contains an input layer made up of a neuron for each input value which is connected to a subsequent layer known as the hidden layer. This layer contains a user defined number of neurons, in this case 10. The network may also be built to contain any number of additional hidden layers although one is often sufficient. The final layer is the output layer. This layer contains a neuron for each network output. The selection of input data and output data is critical as a feed-forward neural network is incapable of capturing 1 to many mappings. The inputs to the network must be such that each set of inputs is uniquely mapped to a set of outputs. A single input set cannot be mapped to multiple output sets.



**Figure 4-1: Network Architecture**

Figure 4-2 depicts a single neuron within either the hidden or output layer. Each neuron sums the weighted output signals from each neuron of the previous layer, adds a bias signal and passes the result through an activation function. The activation function may be any differentiable function. In this study a tan sigmoid transfer function is used in the hidden layer and a linear transfer function is used in the output layer. The use of a non-linear transfer function in the hidden layer enables this kind of network to capture non-linear relationships between inputs and outputs.



**Figure 4-2: A General Neuron**

The feed forward neural network shown in Figure 4-1, containing a single hidden layer with a tan-sigmoid transfer function, may be represented by the following set of equations. The form of these equations is similar to that used by Hashash et al. [119].

The output of the  $i^{th}$  neuron of the input layer  $I_i^{NN}$ , is given by

$$I_i^{NN} = \frac{2(I_i - I_i^{\min})}{(I_i^{\max} - I_i^{\min})} - 1 \text{ such that } -1 < I_i^{NN} < 1 \quad (4-1)$$

where  $i$  is the index of the neuron in the input layer ( $i = 1 \rightarrow NI$ ),  $I_i$  is the un-scaled input signal and  $I_i^{max}$  and  $I_i^{min}$  are the maximum and minimum values of the inputs contained in the data set used for network training respectively. All input signals presented to the neural network are scaled to fall within a range between -1 and 1 to allow the use of inputs with different units. Scaling all inputs to the same range equalises the importance attributed to each input regardless of unit. This leads to weight and bias values which fall into a smaller more predictable range increasing the stability of the network training algorithm. After training all new inputs must be scaled using the same scaling process. This is why the scaling process is represented as part of the network and not as a separate process prior to implementation or training.

The output of the  $j^{th}$  neuron of the hidden layer,  $H_j$ , is given by

$$H_j = \tanh\left(\left[\sum_{i=1}^{NI} w_{ji}^{IH} I_i^{NN} + b_j^H\right]\right) \quad (4-2)$$

where  $j$  is the index of the neuron in the hidden layer ( $j = 1 \rightarrow NH$ ),  $w_{ji}^{IH}$  is the weight assigned to the connection between the  $i^{th}$  neuron of the input layer and the  $j^{th}$  neuron of the hidden layer,  $b_j$  is the neuron bias. In matrix form, the output of the hidden layer,  $[B]$ , may be expressed as

$$[H] = \tanh\left(\left[w^{IH}\right][I^{NN}] + [b^H]\right) \quad (4-3)$$

where  $[H]$  is a 1 by  $NH$  array of hidden layer outputs,  $[I^{NN}]$  is a 1 by  $NI$  array of scaled inputs from the input layer,  $[w^{IH}]$  is the  $NB$  by  $NI$  matrix of connection weights between the input and hidden layers and  $[b^H]$  is a 1 by  $NB$  array of hidden layer biases.

The output of the  $k^{th}$  neuron of the output layer,  $O_k^{NN}$ , is given by

$$O_k^{NN} = \left(\left[\sum_{j=1}^{NH} w_{kj}^{HO} H_j + b_k^O\right]\right) \quad (4-4)$$

where  $k$  is the index of the neuron in the output layer ( $k = 1 \rightarrow NO$ ),  $w_{jk}^{HO}$  is the weight assigned to the connection between the  $j^{th}$  neuron of the hidden layer and the  $k^{th}$  neuron of the output layer,  $b_k$  is the neuron bias. In matrix form the output of the output layer,  $[O^{NN}]$ , may be expressed as

$$[O^{NN}] = \left(\left[w^{BO}\right][B] + [b^O]\right) \quad (4-5)$$

where  $[O^{NN}]$  is a 1 by  $NO$  array of output layer outputs,  $[w^{BO}]$  is the  $NO$  by  $NB$  matrix of connection weights between the hidden and output layer and  $[b^O]$  is a 1 by  $NO$  array of output layer biases.

Prior to training all target outputs in the training data set are scaled in the same manner as the training inputs (4-1). Again this removes the effect of outputs with different units to each other and to the input values. Therefore, to return the real value with the same units as the original target outputs it is necessary to reverse the scaling on the network output. The final un-scaled output of the  $k^{th}$  neuron of the output layer,  $O_k$ , is given by the inverse of (4-1) yielding,

$$O_k = \left( \frac{O_k^{NN} + 1}{2} \right) \times (O_k^{max} - O_k^{min}) + O_k^{min} \text{ such that } -1 < O_k^{NN} < 1 \quad (4-6)$$

where  $O_k^{max}$  and  $O_k^{min}$  are the maximum and minimum values, respectively, of the target outputs contained in the data set used for network training.

## 4.1.2 Network training

A number of training algorithms have been developed, but the most commonly used in training feed-forward neural networks fall under the general term of back-propagation. The original back propagation learning algorithm came to the fore of the field in the 1980s [125]. The aim of network training is to find the combination of network weights and biases that reduces the error between network output and target outputs when the network is presented with a training data set of input and output pairs.

Back propagation uses a gradient descent method in order to search for the minimum of the networks total error function. The combination of weights and biases which minimise the error function is the solution to the learning problem. The gradient of the error function at each neuron within the network is required at each iteration step in order to update the weights in the most efficient direction. This is found via the partial derivative of total network error with respect to the network weights.

In this study the Matlab Neural Network Toolbox [126] is used for the development of the neural networks. The Levenberg-Marquardt back-propagation training method ‘trainlm’ is used; this method is an amalgamation of both quasi-newton and gradient decent methods. In order to describe the Levenberg-Marquardt algorithm the original back-propagation procedure as described in [125] is presented followed by the Levenberg-Marquardt method as described in [127].

### 4.1.2.1 Back-propagation training [125]

The back-propagation training procedure comprises a three step process. In the first step, input signals from the training data set, with corresponding known outputs, are propagated forward through the network to be converted to an output signal via the

neurons and weighted connections. During this phase all connection weights are held constant. The error between the network output and the target output may then be calculated. The second step involves the back propagation of error function derivatives in order to derive the sensitivity of the error function to the individual weights and biases. This is done to find the steepest gradient to the problem solution. The third and final step involves updating all network weights and biases according to the calculated gradient. This process is repeated iteratively until some convergence criterion is met. The initial values for the weights and biases may be any arbitrary value but are often randomly generated.

The first step is straight forward. However, the derivation of partial derivatives of the error function is more complex. The network may be viewed as a complex chain of functions. Therefore, the chain rule must be employed to find the required partial derivatives. The partial derivatives of the error function are propagated backwards from the output layer back to the input layer in order to determine the gradients required to update the weights and biases.

For a given set of input vectors with associated target outputs the total network error,  $E$ , is given by the following error function,

$$E = \frac{1}{2} \sum_c \sum_k (O_{k,c} - T_{k,c})^2 \quad (4-7)$$

where  $O_{k,c}$  denotes network output,  $T_{k,c}$  denotes target output. As above, the subscript  $k$  denotes the index of the output neuron and  $c$  the specific input-output pair of the training data set.

Taking a single input output pair, the backward pass of error derivatives begins with the partial derivative of the error function with respect to the network output. Taking the  $j^{th}$  neuron, differentiation of equation (4-7), temporarily disregarding the index  $c$ , yields

$$\frac{\partial E}{\partial O_k} = O_k - T_k \quad (4-8)$$

For the purposes of training it may be assumed that all input and output pairs have already been scaled. Therefore the scaling input and output layers may be disregarded for the purposes of training. Revisiting (4-4) and splitting the neuron processes into an input unit and an output unit containing the activation function, the total input to the  $k^{th}$  neuron of the output layer,  $x_k^0$ , is given by

$$x_k^0 = \sum_j w_{kj} H_j + b_k \quad (4-9)$$

The bias may be treated as a weight by including a neuron with a constant output of 1 in the preceding layer. In matrix form this yields

$$\begin{bmatrix} x_1^O \\ \vdots \\ x_{NO}^O \end{bmatrix} = \begin{bmatrix} w_{1,1} & \dots & w_{1,NH} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ w_{NO,1} & \dots & w_{NO,NH} & b_{NH} \end{bmatrix} \begin{bmatrix} B_1 \\ \vdots \\ B_{NH} \\ 1 \end{bmatrix} \quad (4-10)$$

The output of the  $k^{th}$  neuron in the output layer,  $O_k$ , is given by the output of the chosen activation function,  $f$ . In the case of the network shown in Figure 4-1 this is a linear function. Therefore the partial derivative of the output of the  $k^{th}$  neuron in the output layer with respect to the layers input,  $\frac{\partial O_k}{\partial x_k}$  is given by

$$\frac{\partial O_k}{\partial x_k} = 1 \quad (4-11)$$

The chain rule may then be employed to find the partial derivative of error with respect to the output layer input,  $x_k$ .

$$\frac{\partial E}{\partial x_k^O} = \frac{\partial E}{\partial O_k} \cdot \frac{dO_k}{dx_k^O} = \frac{\partial E}{\partial O_k} \quad (4-12)$$

This gives the sensitivity of the error function to the total input of the  $k^{th}$  neuron of the output layer.

By inspection of (4-9) it may be seen that the neuron input is a linear function of the previous neuron outputs and the connection weights. Therefore, the partial derivative of network error with respect to the connection weight,  $w_{ij}$ , between the  $k^{th}$  neuron of the current layer and the  $j^{th}$  neuron of the preceding hidden layer is given by

$$\frac{\partial E}{\partial w_{kj}^{BO}} = \frac{\partial E}{\partial x_k^O} \cdot \frac{dx_k^O}{\partial w_{kj}^{BO}} = \frac{\partial E}{\partial x_k^O} \cdot H_j \quad (4-13)$$

The sensitivity of the network error with respect to the output of the  $j^{th}$  node of the hidden layer,  $\frac{\partial E}{\partial H_j}$ , is then found by considering the effect of neuron  $j$  of the hidden layer on neuron  $k$  of the output layer.

$$\frac{\partial E}{\partial x_k^O} \cdot \frac{\partial x_k^O}{\partial B_j} = \frac{\partial E}{\partial x_k^O} \cdot w_{kj}^{HO} \quad (4-14)$$

Taking into account all the connections emanating from neuron  $k$  yields,

$$\frac{\partial E}{\partial H_j} = \sum_j \frac{\partial E}{\partial x_k^O} \cdot w_{kj}^{HO} \quad (4-15)$$

It is now possible to calculate  $\frac{\partial E}{\partial H}$  for any neuron in the hidden layer given  $\frac{\partial E}{\partial O}$  for each neuron in the output layer. This process is now repeated in order to find  $\frac{\partial E}{\partial w_{ji}^{IH}}$  for the weights between the hidden and input layers.

Again splitting the neuron process into an input unit and an output unit containing the activation function the total input to the  $j^{th}$  neuron of the hidden layer,  $x_j$ , is given by

$$x_j^H = \sum_i w_{ji}^{IH} I_i^{NN} + b_j^H \quad (4-16)$$

The output of the  $j^{th}$  neuron of the hidden layer,  $B_j$ , is given by the output of the chosen non-linear activation function,  $f$ . A tan-sigmoid activation function which is equivalent to hyperbolic tan is used throughout this thesis.

$$H_j = f(x_j^H) \approx \tanh(x_j^H) \approx \frac{2}{1+e^{-2x_j^H}} - 1 \quad (4-17)$$

The chain rule is then employed at the preceding layer to find the partial derivative of error with respect to the layer input,  $x_j$ .

$$\frac{\partial E}{\partial x_j^H} = \frac{\partial E}{\partial B_j} \cdot \frac{\partial B_j}{\partial x_j^H} \quad (4-18)$$

The derivative of the tan-sigmoid function in (4-17) is given by

$$\frac{\partial B_j}{\partial x_j^H} = B_j(1 - B_j^2) \quad (4-19)$$

Substitution of this into (4-18) yields,

$$\frac{\partial E}{\partial x_j^H} = \frac{\partial E}{\partial B_j} \cdot B_j(1 - B_j^2) \quad (4-20)$$

This gives the sensitivity of the error function to the total input of the  $j^{th}$  neuron of the hidden layer. By inspection of (4-16) it again may be seen that the neuron input is a linear function of the previous neuron outputs and the connection weights. Therefore, the partial derivative of network error with respect to the connection weight,  $w_{ji}^{IH}$ , between the  $j^{th}$  neuron of the current layer and the  $i^{th}$  neuron of the preceding input layer is given by

$$\frac{\partial E}{\partial w_{ji}^{IH}} = \frac{\partial E}{\partial x_j^H} \cdot \frac{\partial x_j}{\partial w_{ji}^{IH}} = \frac{\partial E}{\partial x_j^H} \cdot I_i \quad (4-21)$$

The partial derivatives of the error function with respect to all network weights may now be found. This process may be expanded to apply to any size network with any number of hidden layers containing any differentiable activation function.

The values of  $\frac{\partial E}{\partial w}$  may be used to alter the weights individually for each input-output pair, this is termed on-line training. Alternatively the values of  $\frac{\partial E}{\partial w}$  are accumulated over the entire training set before updating the weights, this is termed batch training. The simplest form of gradient descent changes each weight by an amount proportional to  $\frac{\partial E}{\partial w}$ .

$$\Delta w = -\epsilon \frac{\partial E}{\partial w} \quad (4-22)$$

where  $\epsilon$  is the learning rate. Convergence is improved by using an acceleration method where the current gradient is used to modify the velocity of the point in weight space instead of its position

$$\Delta w(t) = -\epsilon \frac{\partial E}{\partial w_{(t)}} + \alpha \Delta w_{(t-1)} \quad (4-23)$$

where  $t$  is incremented by 1 for each run through the whole set of input-output cases,  $\alpha$  is an exponential decay factor between 0 and 1 that determines the relative contribution of the current gradient and earlier gradients to the weight change.

#### 4.1.2.2 Levenberg-Marquardt algorithm [127]

The Levenberg-Marquardt algorithm is an approximation to Newton's method. Given a function  $f(\bar{w})$  which is to be minimised with respect to the vector  $\bar{w}$ , then Newton's method is given by

$$\Delta \bar{w} = -[\nabla^2 E(\bar{w})]^{-1} \nabla E(\bar{w}) \quad (4-24)$$

where  $\nabla^2 f(\bar{w})$  is the Hessian matrix and  $\nabla f(\bar{w})$  is the gradient. Assuming that  $f(\bar{w})$  is a sum of the squares function of the form

$$E(\bar{w}) = \sum_{i=1}^N e_i^2(\bar{w}) \quad (4-25)$$

it can be shown that

$$\nabla E(\bar{w}) = J^T(\bar{w})e(\bar{w}) \quad (4-26)$$

$$\nabla^2 E(\bar{w}) = J^T(\bar{w})J(\bar{w}) + S(\bar{w}) \quad (4-27)$$

where  $J(\bar{w})$  is the Jacobian matrix

$$J(\bar{w}) = \begin{bmatrix} \frac{\partial e_1(\bar{w})}{\partial w_1} & \frac{\partial e_1(\bar{w})}{\partial w_2} & \dots & \frac{\partial e_1(\bar{w})}{\partial w_n} \\ \frac{\partial e_2(\bar{w})}{\partial w_1} & \frac{\partial e_2(\bar{w})}{\partial w_2} & \ddots & \frac{\partial e_2(\bar{w})}{\partial w_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_N(\bar{w})}{\partial w_1} & \frac{\partial e_N(\bar{w})}{\partial w_2} & \dots & \frac{\partial e_N(\bar{w})}{\partial w_n} \end{bmatrix} \quad (4-28)$$

and

$$S(\bar{w}) = \sum_{i=1}^N e_i(\bar{w}) \nabla^2 e_i(\bar{w}) \quad (4-29)$$

For the Gauss-Newton method it is assumed that  $S(\bar{w}) \approx 0$ . Therefore (4-24) becomes

$$\Delta \bar{w} = -[J^T(\bar{w})J(\bar{w})]^{-1}J^T(\bar{w})e(\bar{w}) \quad (4-30)$$

The modification introduced to equation (4-24) by the Levenberg-Marquardt may be given as

$$\Delta \bar{w} = -[J^T(\bar{w})J(\bar{w}) + \mu I]^{-1}J^T(\bar{w})e(\bar{w}) \quad (4-31)$$

The parameter  $\mu$  is multiplied by some factor  $\beta$  whenever a step would result in an increase in  $E(\bar{w})$ . When a step reduces  $E(\bar{w})$ ,  $\mu$  is divided by  $\beta$ . When  $\mu$  is large the algorithm becomes steepest descent, for small  $\mu$  the algorithm becomes Guass-Newton.

The terms of the Jacobian, equation (4-28), are found from the back-propagation procedure described above, Section 4.1.2.1.

#### **4.1.2.3 Training data division and early stopping**

The Matlab Neural Network toolbox [126] employs a number of different criteria in order to stop training. Training is halted when the gradient reaches a defined minimum ( $10^{-5}$ ), the performance reaches a defined minimum or goal ( $10^{-10}$ ), training time exceeds an upper limit, or a maximum number of training epochs are completed ( $10^3$ ). The final stopping criterion involves validation checking leading to early stopping.

A key consideration when training neural networks is the issue of over fitting, where a network is trained to a point where it no longer possesses the ability to generalise, Section 2.3. It is vital that the network is tested using previously ‘unseen’ data in order to identify where over fitting has occurred. This problem has been the subject of numerous studies. The factors identified as key to over fitting are network architecture, training control and training data selection.

The default method of over fitting control employed by the Matlab Neural Network Toolbox is early stopping. Available training data is divided into three subsets a training set, a validation set and a testing set. The data is divided according to pre-defined ratios. The training set comprises the largest proportion, typically 70%. The rest of the data is divided equally between the validation and testing sets, 15% each. Allocation to the sets is done by either a regular or random index list.

The training set is used to calculate current network error and the gradient with which the weights and biases are updated. The validation data set is presented to the network after each weight and bias update and the current error calculated. The error on the validation data set is monitored throughout training. In the early stages of training both the training

and validation data error generally decreases. As training progresses the training data error will continue to decrease. However, as the network begins to over fit the training data the validation data error will increase. If the validation data error consistently increases for 6 iterations then training is stopped.

The testing data set is not used during training. It may be used after training to provide a comparison to the validation data set error to check the division of data. By plotting the mean square error at each training epoch the training progress may be investigated. If the testing data error minimum is at a significantly different epoch to the validation set this may indicate a poor division of data.

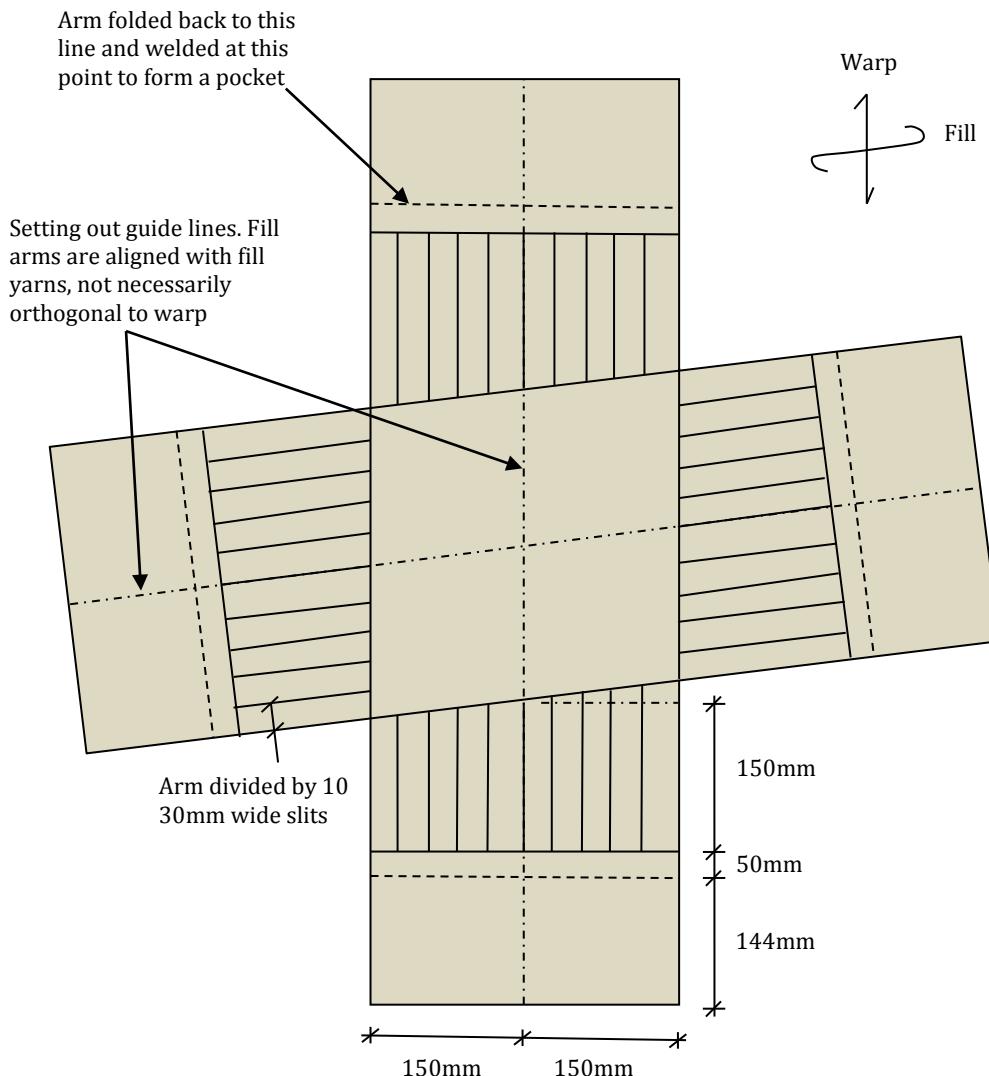
## 4.2 Biaxial Testing

The data required for network training and validation is collected by biaxial testing of cruciform samples of architectural fabric. The aim of biaxial testing is to apply a known, uniform stress field and accurately measure the resulting strain. All testing is done using a custom built 'floating' rig and test protocol initially developed at Newcastle University in the thesis of Bridgens [59].

### 4.2.1 Cruciform sample preparation

The cruciform sample (Figure 4-3) is designed to enable the uniform transfer of the maximum amount of applied load into the central region of the sample, thus generating a near uniform known stress field. The cruciform consists of a 300mm square central region surround by 150mm long arms with additional allowance to form a welded pocket for a clamping bar. The sample is laid out so that the warp and fill yarns are aligned with centre lines of the cruciform arms. This may result in a non-orthogonal layout as illustrated in Figure 4-3.

The cruciform arms contain slits set at regular intervals. Numerical simulations undertaken by Bridgens [59] and also in the development of the standard published by the MSAJ [43] indicate that the inclusion of slits significantly increases the total load transferred to the central region of the cruciform. Slits also decrease the effect of large shear deformations in the cruciform arms which result from unequal large strains in the approximately orthogonal material directions.



Note: Non-orthogonal configuration of warp and fill yarns exaggerated for clarity

**Figure 4-3: Cruciform Sample Dimensions (Adapted from [59])**

In order to convert the recorded applied load to an accurate stress value for the central region of the cruciform a correction factor or equivalent effective gauge length is applied. This is done to account for the loss of load due to the spreading of the cruciform arms. Through numerical simulation of the cruciform sample a correction factor of 0.95, or an equivalent effective gauge length of 315mm, has been determined by Bridgens in [59]. This factor has been deemed suitable for all load ratios.

#### 4.2.2 Test rig

The test rig consists of two reaction frames into which the cruciform sample is clamped using a bar set into machined circular hole. The upper frame is mounted on spherical bearings enabling free movement in the plane of the fabric. This ensures that the rig

remains aligned with the material warp and fill direction as deformation takes place. It also allows for the testing of non-orthogonal cruciform samples.

Biaxial load is applied in the warp and fill material directions via two hydraulic rams. Current load is measured by a load cell located between the ram and the sample clamp plate. The ram head position is measured by a LVDT displacement transducer. Both load and ram position are output to a control program in real time and either may be used to control the ram. Testing for this body of work is load controlled. Strain in the central region of the sample is measured at 2 second intervals concurrently with the load cell output via LVDT displacement transducers aligned with the warp and fill directions.

### 4.3 Response Surface Neural Network Material Model

Architectural fabric material models, particularly those based on the plane-stress framework and those based on response surfaces, are fitted to experimental data in the form of load ratio arms, Section 2.2. When a neural network is trained using data in the form of load ratio arms it may be used to interpolate between those arms thus creating a response surface style model. As this model is easy to visualise it is selected as a good starting point for the initial development of an architectural fabric neural network material model.

A constitutive material model endeavours to describe a material's response to external stimuli. In the case of architectural fabric modelling it is the relationship between stress and strain that is of interest. Within a typical finite element analysis the current strain state is used to determine the current level of stress which in turn is used to determine whether the forces within a structure are in equilibrium with the external forces applied to the structure. The current strain will therefore be used as input and the stress as output. This leads to a network comprising two inputs (warp and fill strain), a single hidden layer and two outputs (warp and fill stress).

#### 4.3.1 Network performance criteria

The non-uniqueness of neural network material models has been identified as a key feature [88]. Because neural networks are typically initialised using random numbers for all weights and biases, two networks presented with identical training data will produce different functional mappings. In order to find a network with the highest possible performance, multiple networks are trained, and from that group the network with the

best performance is selected. The performance criteria used for this purpose is the coefficient of determination,  $R^2$  given by,

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - f_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (4-32)$$

where  $y_i$  is the target value,  $f_i$  is the network output and  $\bar{y}$  is the mean value of the targets. The coefficient of determination is a relative measure of error and is therefore a clearer indication of error than mean-square error. The closer the coefficient of determination is to 1 the higher performance of the network. For each training data set 10 individual networks are trained. Each network effectively models two response surfaces representing warp and fill response separately. Therefore, the mean of the coefficients of determination for the two surfaces is used to identify the network with the best overall performance.

### 4.3.2 Response surface training data collection and pre-processing

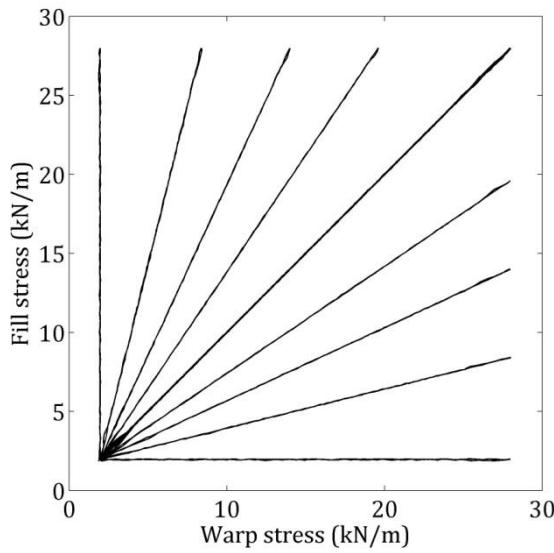
#### 4.3.2.1 Experimental training data collection and pre-processing

A biaxial testing profile is developed that includes additional load ratios between the standard 0:1, 1:2, 1:1, 2:1 and 1:0 ratios commonly used for the derivation of plane stress elastic constants [43]. These ratios offer the opportunity to further investigate the true shape of the interpolated response surface (Figure 4-5 and Figure 4-7) .The additional load ratios may also provide ‘unseen’ testing data for network validation.

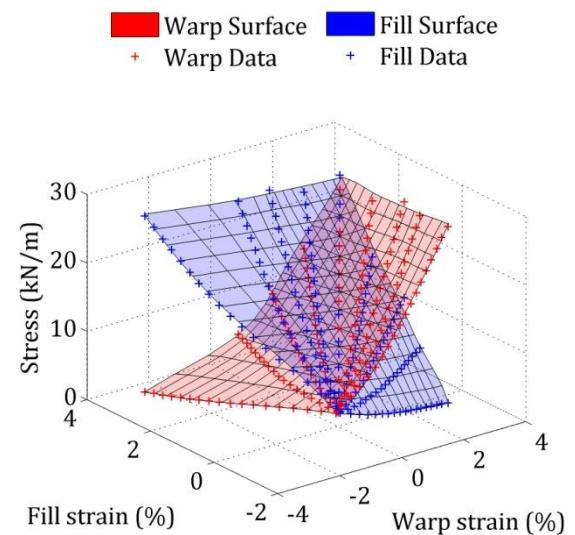
|               | Ultimate Tensile Strength (kN/m) |      | Maximum Stress (kN/m) |      | Minimum Stress (kN/m) |      |
|---------------|----------------------------------|------|-----------------------|------|-----------------------|------|
| Material      | Warp                             | Fill | Warp                  | Fill | Warp                  | Fill |
| PVC (502S)    | 112                              | 112  | 28                    | 28   | 2.8                   | 2.8  |
| PTFE (B18089) | 160                              | 140  | 40                    | 35   | 3.5                   | 3.5  |

Table 4.1: Maximum and minimum stress for biaxial profiles

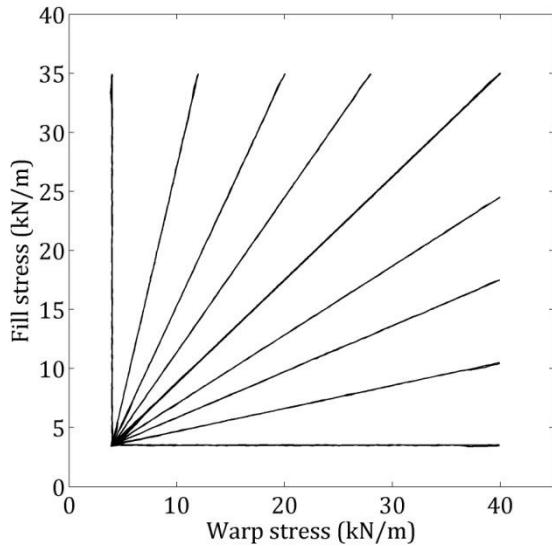
A minimum stress of 2.5% of ultimate fabric strength is used for pre-stress. The maximum stress is 25% of the ultimate tensile strength. Due to limitations in the computerised load control system of the rig it is not possible achieve load ratios with a true 0 load. Therefore, load ratios of 0:1 and 1:0 refer to load ration where the direction referred to by 0 is held at the minimum stress value. In the case of PTFE coated glass fabric the ultimate tensile strength is different between the warp and fills directions. The profile values for the PVC coated polyester and PTFE glass fibre fabrics tested are shown in Table 4.1.



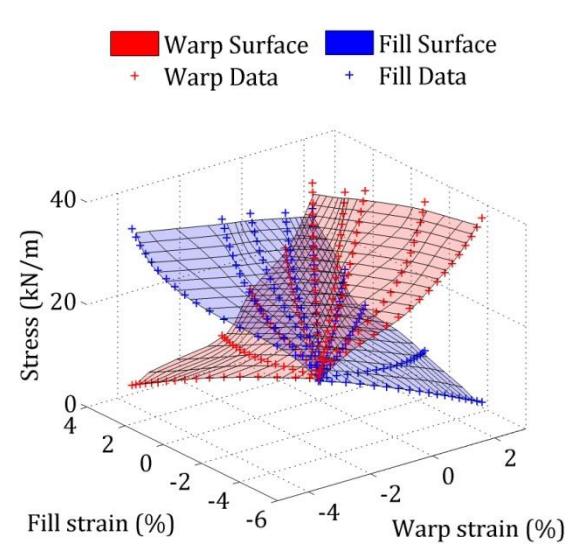
**Figure 4-4: Load ratio arms investigated during PVC biaxial test**



**Figure 4-5: Response surface derived from PVC biaxial test data.**

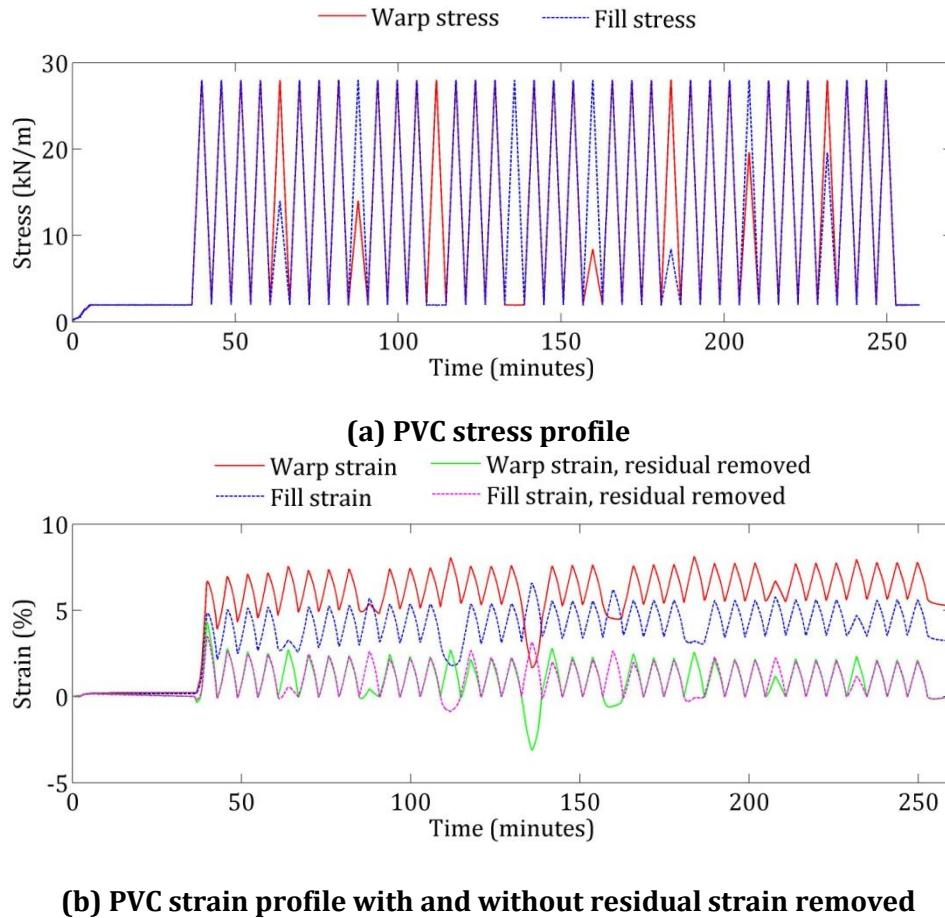


**Figure 4-6: Load ratio arms investigated during PTFE biaxial test**



**Figure 4-7: Response surface derived from PTFE biaxial test data.**

Biaxial testing and data processing is based on the protocol laid out in [60]. As shown in Figure 4-8 and Figure 4-9, the fabric is initially held at pre-stress for an arbitrary 30minutes followed by three 1:1 cycles. This initial combination of loading is applied to condition the fabric so as to capture in situ behaviour in the following cycles as oppose to the fabric's initial behaviour which is heavily affected by crimp interchange. The profile then continues through each ratio of interest with three 1:1 cycles between each. This minimises the effect of the previous load ratio on the subsequent one.



**Figure 4-8: Testing profile for PVC coated polyester fabric**

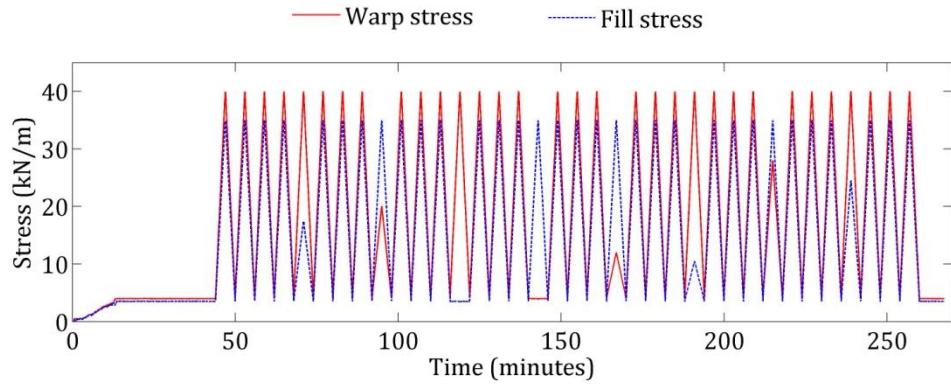
In post processing residual strain is removed from the experimental data (Figure 4-8 and Figure 4-9), in order to eliminate the effect of skew introduced by the accumulation of residual strain throughout the test. This enables the representation of the response by a single surface and removes the effect of load path dependency.

The procedure for residual strain removal begins by finding the index of each minimum stress turning point,  $i_{TP}$ , within the profile. The strain,  $\varepsilon_{i_{TP}}$ , at each turning point may then be found. As the profile returns to the same level of stress at the beginning and end of each complete cycle the residual strain at each turning point,  $\varepsilon_{i_{TP}}^{Res}$ , may be found from,

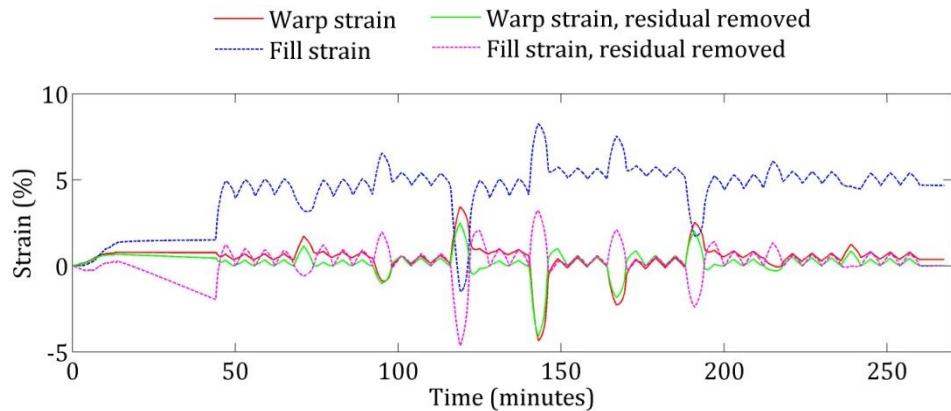
$$\varepsilon_{i_{TP}}^{Res} = \varepsilon_{i_{TP}} - \varepsilon_{i_{TP-1}} \quad (4-33)$$

where  $\varepsilon_{i_{TP}}$  is the total strain at the current turning point and  $\varepsilon_{i_{TP-1}}$  is total strain at the previous turning point. The residual strain is cumulatively subtracted from the total strain to find the strain at any time step within the profile with current residual strain removed. The strain with the cumulative residual strain removed,  $\varepsilon_i^{RR}$ , between the current and previous turning point index, ( $i = i_{TP-1} \rightarrow i_{TP}$ ) is given by

$$\varepsilon_{i_{TP-1} \rightarrow i_{TP}}^{RR} = \varepsilon_i - (i - i_{TP-1}) \left( \frac{\varepsilon_{i_{TP}}^{Res}}{(i_{TP} - i_{TP-1})} \right) - \sum_{i_{TP}=1}^{i_{TP}=i_{TP-1}} \varepsilon_{i_{TP}}^{Res}. \quad (4-34)$$



(a) PTFE stress profile



(b) PTFE strain profile with and without residual strain removed

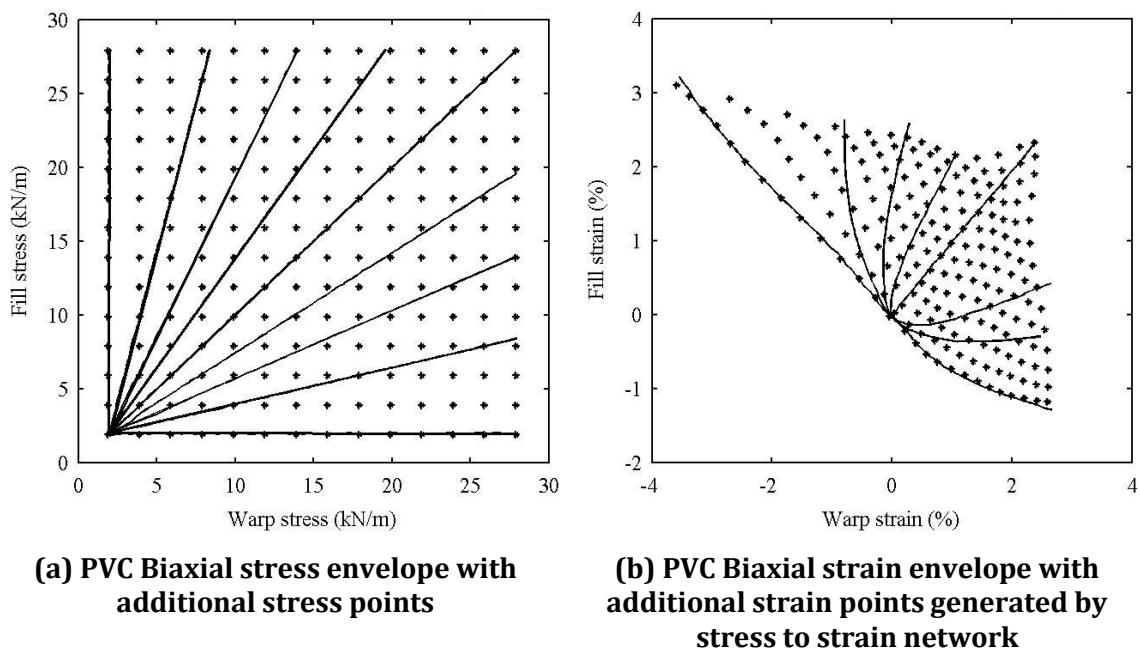
Figure 4-9: Testing profile for PTFE coated glass fibre fabric

Three individual sets of data are used to train and validate the networks. The first data set comprises the full set of 9 load ratios and is used as the training data set for Network 1. The second data set comprises only the 0:1, 1:2, 1:1, 2:1 and 1:0 load ratios and is used as the training data set for Network 2. The intermediate load ratios are used as 'unseen' testing data for Network 2. The third data set is derived from a network trained to take the warp and fill stress as input and output warp and fill strain. This network is also trained using the full set of 9 load ratios. During biaxial testing the stress and strain information is logged at 2 second intervals. For the purposes of network training the density of the experimental data is reduced by only taking every 5<sup>th</sup> data point. This data reduction is done to avoid over fitting and also enables the clearer plotting of the trained network output.

#### 4.3.2.2 Additional synthetic training data generation via stress to strain network

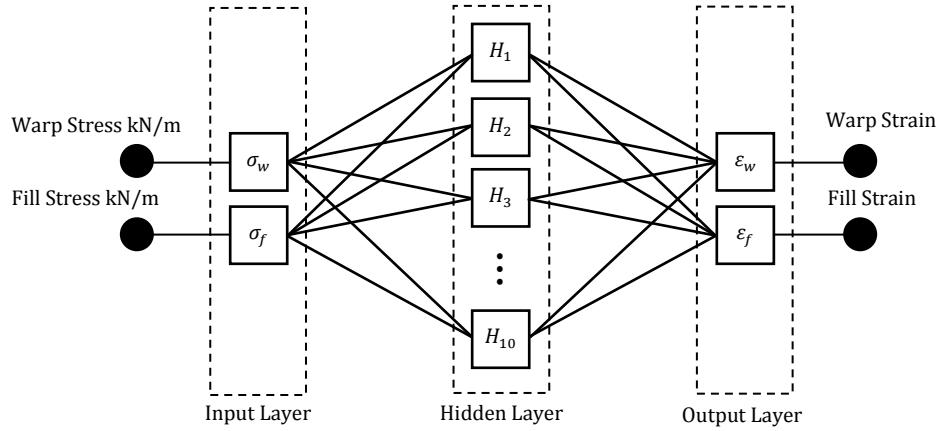
As the experimental data is collected using a load-controlled process the stress envelope is regular and the resulting strain envelope is non-regular. This irregularity makes it difficult to directly generate additional strain values to comprehensively explore the response surface.

A stress to strain network model directly represents the load controlled biaxial test and is therefore useful for producing a more comprehensive data set. Once trained this network is presented with a regular grid of stress points, Figure 4-10 (a), and the network output strains, Figure 4-10 (b), are taken to build an additional synthetic data set. This set is used both as a training data set for Network 3 and as a testing set for Network 1 and 2. In this way the gaps between loading arms may be investigated and cases of over-fitting may be identified.



**Figure 4-10: Generation of additional training and testing data set**

The stress to strain neural network shown in Figure 4-11 comprises an input layer containing two inputs, warp and fill stress, a hidden layer containing 10 neurons and an output layer containing 2 neurons for warp and fill strain. Tan-sigmoid transfer functions are used in the neurons of the hidden layer and a linear transfer function is used in the neurons of the output layer. For the purposes of early stopping 70% of the full training data set is used for training, 15% for validation and 15% for testing. This division will be used for all networks trained in this thesis.



**Figure 4-11: Stress to strain neural network architecture**

All trained stress to strain networks produce  $R^2$  values close to 1 indicating a high correlation between network output and the target outputs. All networks are trained in a relatively low number of training iterations (approx. 5 to 10) and do not trigger early stopping criteria. The stopping criteria triggered is the minimum gradient. Version 7 of the network trained and tested with the PVC (502S) data set produces the best performance. Version 8 of the network trained and tested with the PTFE (B18059) data set produces the best performance. Visually the interpolation between the loading arms shown in Figure 4-13 and Figure 4-14 appear reasonable. The network output from these networks will form the additional data set for training and testing strain to stress network.

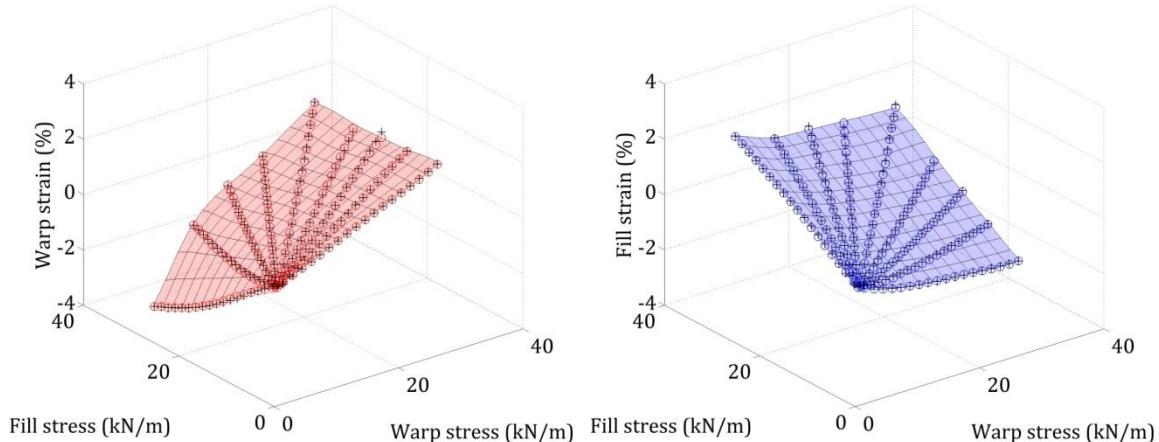
In order to conserve space a universal key (Figure 4-12) is used for all figures depicting network output. Each figure is linked with a specific network within the table of  $R^2$  values directly above it. The networks plotted generally demonstrate the highest performance with one or both of the data sets used for training and testing.

| Warp  | Fill   |   |
|---|--|---|
| <span style="background-color: red;">█</span> | <span style="background-color: blue;">█</span> | Network generated target data                     |
| +   | +  | Experimentally generated target data              |
| *   | *  | Network output from network generated data        |
| ○   | ○  | Network output from experimentally generated data |

**Figure 4-12: Response surface neural network figure key**

| <b>Network performance when presented 9 experimental load ratios (training)</b> |          |          |          |          |          |          |          |          |          |           |
|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|
| <b>Network</b>  | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> | <b>10</b> |
| <b>Warp</b>   | 0.9953   | 0.9964   | 0.9984   | 0.9970   | 0.9986   | 0.9973   | 0.9987   | 0.9943   | 0.9982   | 0.9977    |
| <b>Fill</b>   | 0.9955   | 0.9960   | 0.9973   | 0.9944   | 0.9972   | 0.9964   | 0.9978   | 0.9969   | 0.9975   | 0.9978    |
| <b>Mean</b>   | 0.9954   | 0.9962   | 0.9979   | 0.9957   | 0.9979   | 0.9968   | 0.9982   | 0.9956   | 0.9978   | 0.9978    |

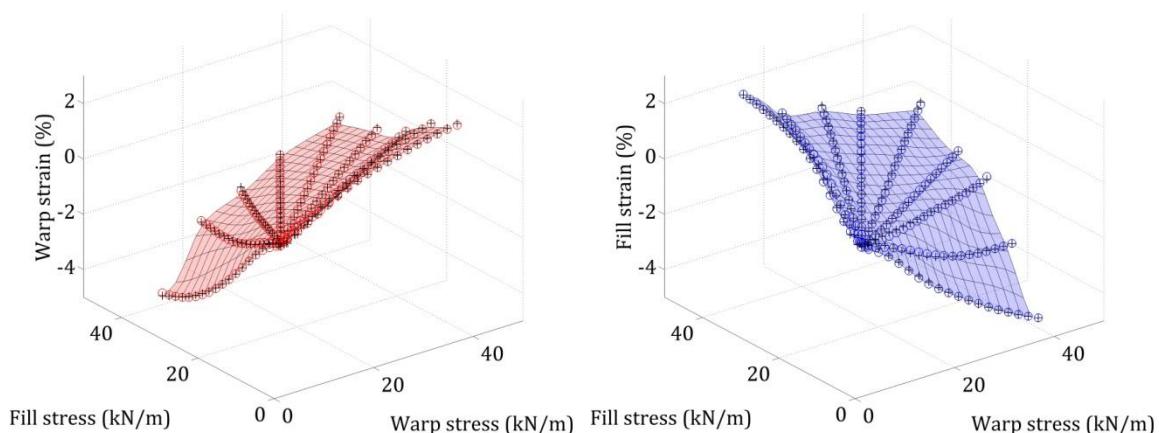
**Table 4.2: Coefficients of determination for PVC coated polyester (502S) stress to strain neural network material model**



**Figure 4-13: Network 7, Table 4.2, PVC (502S) stress to strain network demonstrating best performance**

| <b>Network performance when presented 9 experimental load ratios (training)</b> |          |          |          |          |          |          |          |          |          |           |
|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|
| <b>Network</b>  | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> | <b>10</b> |
| <b>Warp</b>   | 0.9963   | 0.9971   | 0.9984   | 0.9909   | 0.9963   | 0.9961   | 0.9864   | 0.9979   | 0.9806   | 0.9957    |
| <b>Fill</b>   | 0.9965   | 0.9961   | 0.9976   | 0.9934   | 0.9964   | 0.9961   | 0.9837   | 0.9982   | 0.9823   | 0.9954    |
| <b>Mean</b>   | 0.9964   | 0.9966   | 0.9980   | 0.9921   | 0.9963   | 0.9961   | 0.9851   | 0.9981   | 0.9815   | 0.9956    |

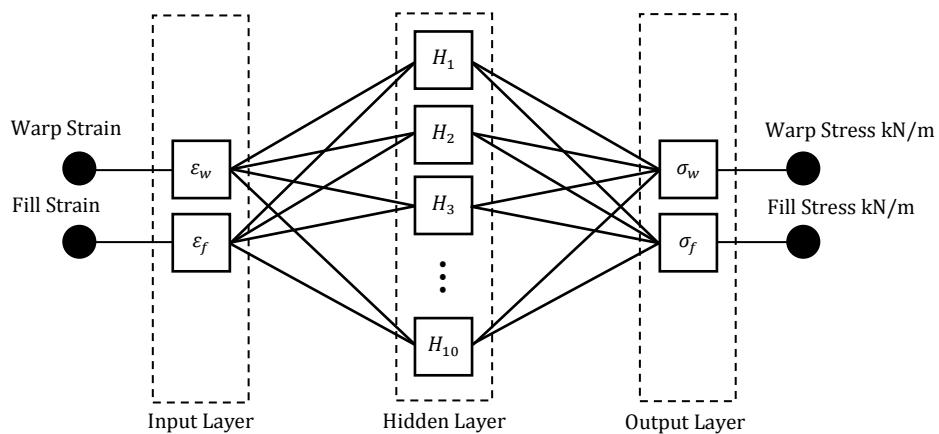
**Table 4.3: Coefficients of determination for PTFE coated glass fibre (B18059) stress to strain neural network material model**



**Figure 4-14: Network 8, Table 4.3, PTFE (B18059) stress to strain network demonstrating best performance**

### 4.3.3 Response Surface Neural Network Training and Validation

Similar to the stress to strain network used to generate the more comprehensive data set. The strain to stress neural network shown in Figure 4-15 comprises an input layer containing two inputs, warp and fill strain, a hidden layer containing 10 neurons and an output layer containing 2 neurons for warp and fill stress. Tan-sigmoid transfer functions are used in the neurons of the hidden layer and a linear transfer function is used in the neurons of the output layer.

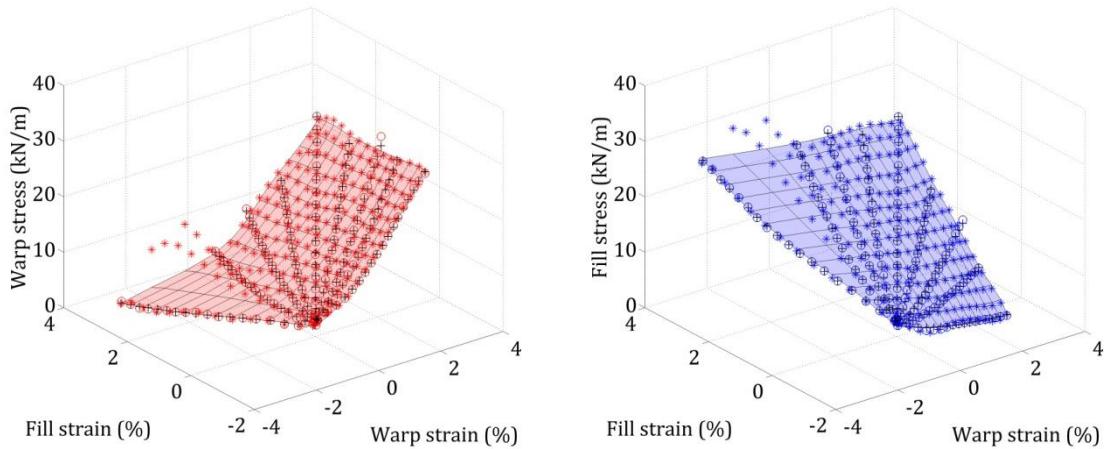


**Figure 4-15: Strain to stress neural network architecture**

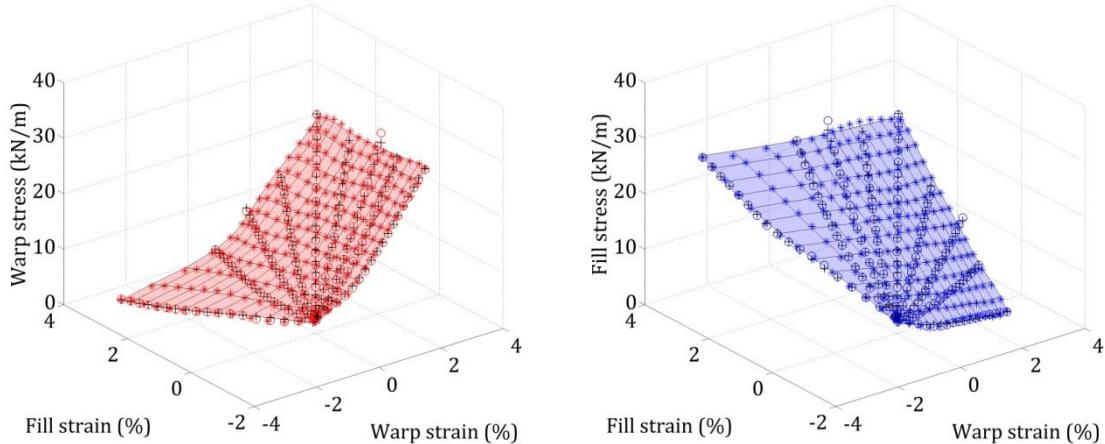
#### 4.3.3.1 PVC (502S) response surface neural network

| Network performance with 9 experimental load ratios (training data)     |        |        |        |        |        |        |        |        |        |        |
|---|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Network   | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | 10     |
| <b>Warp</b>   | 0.9977 | 0.9978 | 0.9984 | 0.9948 | 0.9973 | 0.9968 | 0.9973 | 0.9975 | 0.9976 | 0.9982 |
| <b>Fill</b>   | 0.9982 | 0.9986 | 0.9979 | 0.9945 | 0.9977 | 0.9958 | 0.9978 | 0.9979 | 0.9978 | 0.9984 |
| <b>Mean</b>   | 0.9980 | 0.9982 | 0.9981 | 0.9947 | 0.9975 | 0.9963 | 0.9976 | 0.9977 | 0.9977 | 0.9983 |
| Network performance with network generated data ('unseen' testing data) |        |        |        |        |        |        |        |        |        |        |
| Network   | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | 10     |
| <b>Warp</b>   | 0.9984 | 0.9880 | 0.9973 | 0.9928 | 0.9988 | 0.9937 | 0.9900 | 0.9991 | 0.9975 | 0.9656 |
| <b>Fill</b>   | 0.9979 | 0.9734 | 0.9926 | 0.9944 | 0.9987 | 0.9956 | 0.9961 | 0.9983 | 0.9968 | 0.9810 |
| <b>Mean</b>   | 0.9981 | 0.9807 | 0.9950 | 0.9936 | 0.9988 | 0.9946 | 0.9931 | 0.9987 | 0.9972 | 0.9733 |

**Table 4.4: Coefficients of determination for PVC coated polyester (502S) strain to stress neural network material model trained with 9 experimental load ratios**



**Figure 4-16: Network 10, Table 4.4, PVC (502S) strain to stress network demonstrating best performance with experimentally generated data used in training**

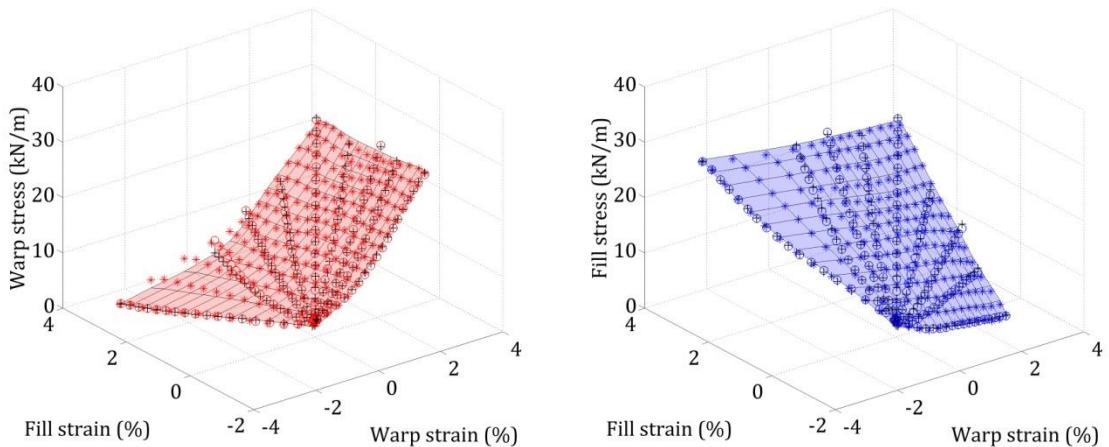


**Figure 4-17: Network 5, Table 4.4, PVC (502S) strain to stress network demonstrating best performance with 'unseen' network generated data used in testing**

Version 10 of the network trained using the 9 load ratios, Table 4.4, shows the best performance when tested with the full experimental load ratio set. However, when tested with the more comprehensive synthetic network generated data set a much lower performance is observed, this indicates over-fitting may have occurred. This is confirmed visually in Figure 4-16 where the effects of over fitting may be clearly seen between the 0:1 and 1:2 load ratio arms in both the warp and fill surfaces. It is noted that an  $R^2$  value lower than 0.990 indicates and unacceptable network error. Version 5 of the network trained with the same data (Figure 4-17) produces a slightly decreased  $R^2$  value when presented with the full experimental load ratio set. However, when presented with the network generated data set this network produces a much higher  $R^2$  value. The network has a greater ability to generalise within a reasonable range. This demonstrates the importance of network testing using unseen data.

| <b>Network performance with 9 experimental load ratios (partial training)</b> |          |          |          |          |          |          |          |          |          |           |
|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|
| <b>Network</b>  | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> | <b>10</b> |
| <b>Warp</b>   | 0.9943   | 0.9935   | 0.9903   | 0.9944   | 0.9945   | 0.9770   | 0.9866   | 0.9918   | 0.9937   | 0.9879    |
| <b>Fill</b>   | 0.9958   | 0.9957   | 0.9953   | 0.9946   | 0.9952   | 0.9959   | 0.9874   | 0.9943   | 0.9964   | 0.9928    |
| <b>Mean</b>   | 0.9951   | 0.9946   | 0.9928   | 0.9945   | 0.9949   | 0.9865   | 0.9870   | 0.9930   | 0.9950   | 0.9904    |
| <b>Network performance with network generated data (unseen)</b>               |          |          |          |          |          |          |          |          |          |           |
| <b>Network</b>  | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> | <b>10</b> |
| <b>Warp</b>   | 0.9925   | 0.9892   | 0.9858   | 0.9908   | 0.9896   | 0.9687   | 0.9776   | 0.9843   | 0.9919   | 0.9786    |
| <b>Fill</b>   | 0.9963   | 0.9947   | 0.9956   | 0.9939   | 0.9958   | 0.9961   | 0.9812   | 0.9917   | 0.9961   | 0.9812    |
| <b>Mean</b>   | 0.9944   | 0.9920   | 0.9907   | 0.9924   | 0.9927   | 0.9824   | 0.9794   | 0.9880   | 0.9940   | 0.9799    |

**Table 4.5: Coefficients of determination for PVC coated polyester (502S) strain to stress neural network material model, trained with 5 load ratios**

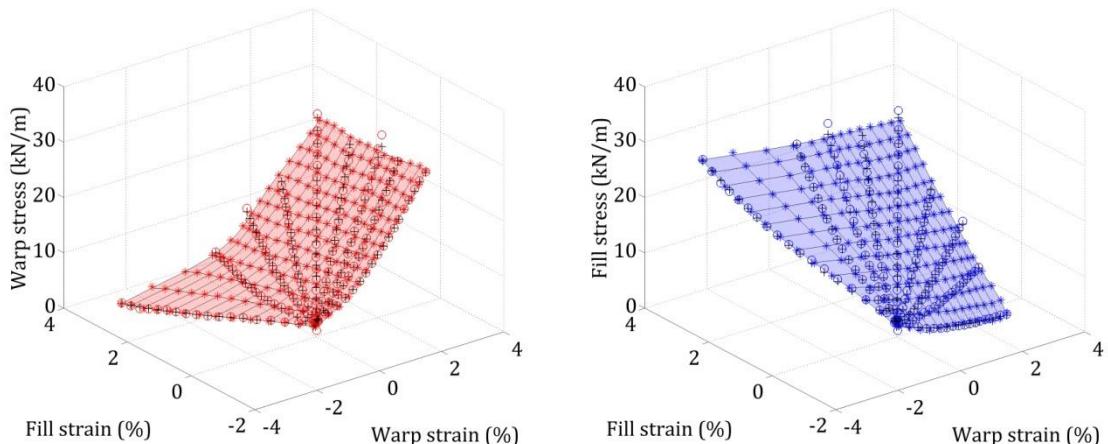


**Figure 4-18: Network 1, Table 4.5 , PVC (502S) strain to stress network demonstrating best performance with both data sets**

The networks trained using the partial data set, Table 4.5, achieve the lowest performance especially in the fill direction. This is expected as these networks have been trained with the least comprehensive data set. Version 1 of the network trained using the reduced experimental data set (Figure 4-18) exhibits the best performance when presented with both the full experimental data set and the network generated data set. Both of these data sets contain information relating to all of the 9 load ratios. Therefore, providing that over fitting has not occurred, both data sets should produce a similar level of performance.

| Network performance with 9 experimental load ratios (unseen) |        |        |        |        |        |        |        |        |        |        |
|--|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Network  | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | 10     |
| Warp   | 0.9970 | 0.9972 | 0.9971 | 0.9968 | 0.9969 | 0.9972 | 0.9973 | 0.9971 | 0.9968 | 0.9976 |
| Fill   | 0.9971 | 0.9975 | 0.9968 | 0.9967 | 0.9974 | 0.9972 | 0.9975 | 0.9970 | 0.9967 | 0.9969 |
| Mean   | 0.9970 | 0.9974 | 0.9970 | 0.9968 | 0.9972 | 0.9972 | 0.9974 | 0.9971 | 0.9968 | 0.9973 |
| Network performance with network generated data (training)   |        |        |        |        |        |        |        |        |        |        |
| Network  | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | 10     |
| Warp   | 0.9996 | 0.9994 | 0.9995 | 0.9991 | 0.9991 | 0.9995 | 0.9991 | 0.9997 | 0.9998 | 0.9995 |
| Fill   | 0.9996 | 0.9994 | 0.9996 | 0.9994 | 0.9993 | 0.9996 | 0.9993 | 0.9996 | 0.9998 | 0.9997 |
| Mean   | 0.9996 | 0.9994 | 0.9996 | 0.9993 | 0.9992 | 0.9995 | 0.9992 | 0.9997 | 0.9998 | 0.9996 |

**Table 4.6: Coefficients of determination for PVC coated polyester (502S) strain to stress neural network material model, trained with network generated data**



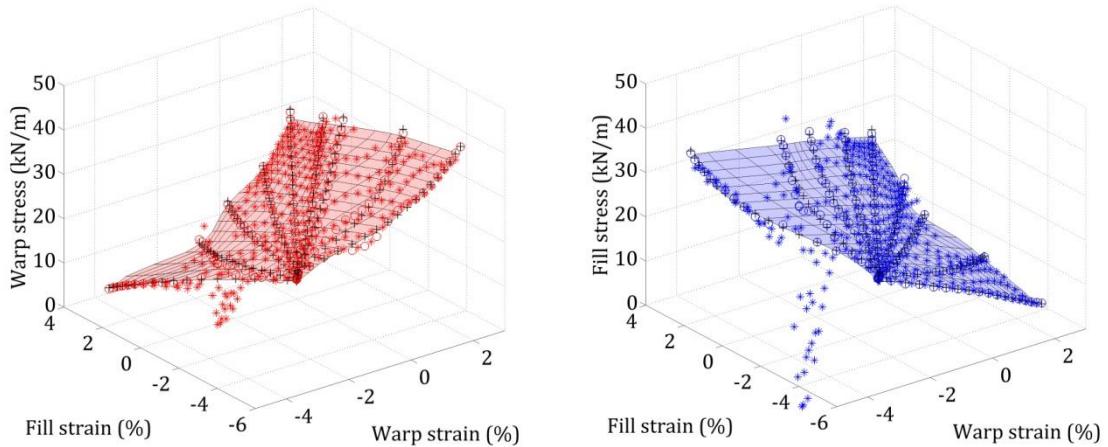
**Figure 4-19: Network 9, Table 4.6, PVC (502S) strain to stress network demonstrating best performance with both data sets**

All of the networks trained using the PVC (502S) data sets achieve  $R^2$  values close to 1 indicating a high correlation between network output and the target outputs. As would be expected the highest  $R^2$  values are observed when a network is tested using the same set of data it was trained with.

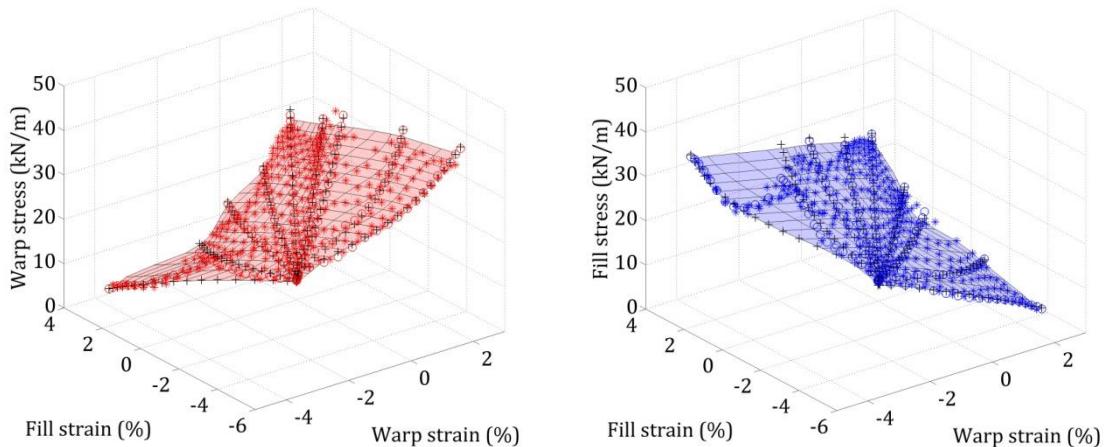
Version 9 of the network trained with the network generated data set, Table 4.6, demonstrates the best overall performance when presented with both the training data and the unseen data sets. This network may be selected as having the best ability to generalise the PVC fabric response. The use of synthetic training data, Section 4.3.2.2, reduces the risk of over-fitting without the need for additional expensive physical testing. It is however of extreme importance to gather sufficient data to thoroughly train and validate any network model to be used in structural analysis.

| <b>Network performance when presented 9 experimental load ratios (training)</b> |          |          |          |                      |          |          |          |          |          |           |
|---|----------|----------|----------|----------------------|----------|----------|----------|----------|----------|-----------|
| <b>Network</b>  | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b>             | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> | <b>10</b> |
| <b>Warp</b>   | 0.9946   | 0.9956   | 0.9956   | 0.9944               | 0.9936   | 0.9936   | 0.9888   | 0.9937   | 0.9955   | 0.9939    |
| <b>Fill</b>   | 0.9803   | 0.9861   | 0.9870   | 0.9916               | 0.9685   | 0.9825   | 0.9813   | 0.9818   | 0.9858   | 0.9817    |
| <b>Mean</b>   | 0.9874   | 0.9908   | 0.9913   | 0.9930               | 0.9810   | 0.9880   | 0.9851   | 0.9878   | 0.9907   | 0.9878    |
| <b>Network performance when network presented with network data (unseen)</b>    |          |          |          |                      |          |          |          |          |          |           |
| <b>Network</b>  | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b>             | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> | <b>10</b> |
| <b>Warp</b>   | 0.9684   | 0.9576   | 0.9576   | 0.8168               | 0.9657   | 0.9580   | 0.9697   | 0.9534   | 0.9589   | 0.9598    |
| <b>Fill</b>   | 0.9027   | 0.8640   | 0.8861   | -0.4581 <sup>1</sup> | 0.9385   | 0.8746   | 0.9210   | 0.8966   | 0.8910   | 0.8895    |
| <b>Mean</b>   | 0.9356   | 0.9108   | 0.9219   | 0.1794               | 0.9521   | 0.9163   | 0.9453   | 0.9250   | 0.9250   | 0.9246    |

**Table 4.7: Coefficients of determination for PTFE coated glass fibre (B18059) strain to stress neural network material model, trained with 9 load ratios**



**Figure 4-20: Network 4, Table 4.7, PTFE (B18059) strain to stress network demonstrating best performance with experimentally generated data used in training**



**Figure 4-21: Network 5, Table 4.7, PTFE (B18059) strain to stress network demonstrating best performance with 'unseen' network generated data used in testing**

<sup>1</sup> A negative  $R^2$  value indicates that the mean on the testing data provides a better fit to the target output than the output generated by the trained network.

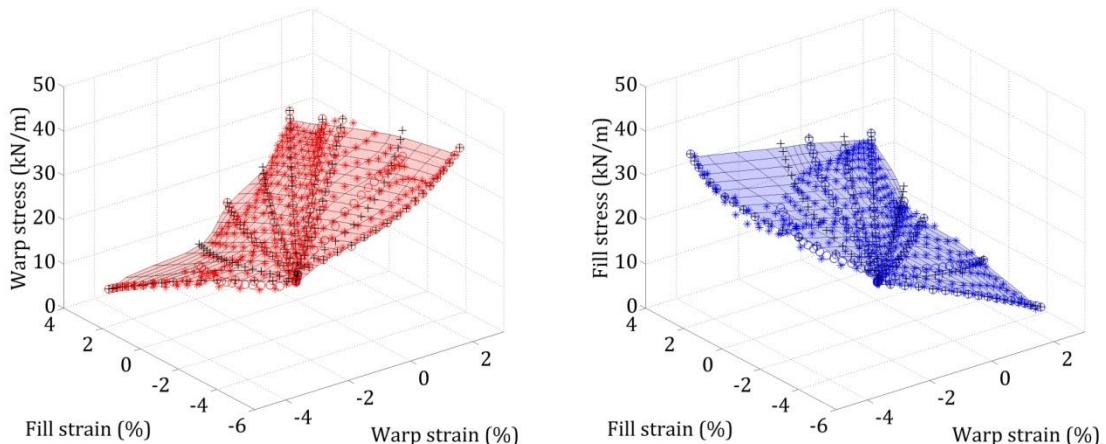
Version 4 of the network trained using the 9 load ratios shows the best performance, although still lower than 0.990, when tested with the full experimental load ratio set. As with the PVC networks the fill surface produces a considerably lower  $R^2$  value than the warp surface. This indicates that the material response of the fill direction yarns is more complex than the warp. This may be due to the high level of crimp in this material direction.

When tested with the more comprehensive network generated data set a much lower performance is observed in both surfaces, this indicates that over-fitting has occurred. This is confirmed visually in (Figure 4-20) where the poor correlation between network output and targets may be clearly seen, especially between the 0:1 and 1:1 load ratio arms where the fill direction carries a greater load than the warp direction. Version 5 of the network trained with the same data (Figure 4-21) produces a slightly decreased  $R^2$  value when presented with the full experimental load ratio set. When presented with the network generated data set this network produces a slightly higher  $R^2$  value indicating a decreased level of over fitting.

The networks trained using the partial data set achieve the lowest performance, this is the same as the PVC networks. Version 3 (Figure 4-22) produces the highest  $R^2$  value when tested with the full experimental load ratio set. However, this performance is poor especially for the load ratio arms not used in the training set. The 'unseen' load ratio arm between the 0:1 and 1:2 load ratios collapses onto the 0:1 load ratio arm. This is due to the high stiffness of the material leading to a steep response surface where one set of strain values leads to multiple values of stress. These results indicate that a 2 input neural network does not have sufficient inputs to avoid the detrimental effect of the one to many mappings present in the PTFE training data

| <b>Network performance when presented 9 experimental load ratios (partial training)</b> |          |          |          |          |          |          |          |          |          |           |
|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|
| <b>Network</b>  | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> | <b>10</b> |
| <b>Warp</b>   | 0.9691   | 0.9658   | 0.9705   | 0.9632   | 0.8411   | 0.9348   | 0.9697   | 0.9697   | 0.9439   | 0.8618    |
| <b>Fill</b>   | 0.8833   | 0.8948   | 0.8933   | 0.8825   | 0.8574   | 0.8841   | 0.8874   | 0.8937   | 0.8851   | 0.8705    |
| <b>Mean</b>   | 0.9262   | 0.9303   | 0.9319   | 0.9229   | 0.8492   | 0.9095   | 0.9286   | 0.9317   | 0.9145   | 0.8662    |
| <b>Network performance when network presented with network data (unseen)</b>            |          |          |          |          |          |          |          |          |          |           |
| <b>Network</b>  | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> | <b>10</b> |
| <b>Warp</b>   | 0.9511   | 0.9613   | 0.9597   | 0.9349   | 0.7278   | 0.9398   | 0.9659   | 0.9574   | 0.9475   | 0.8778    |
| <b>Fill</b>   | 0.8331   | 0.8624   | 0.8606   | 0.8216   | 0.7812   | 0.8470   | 0.8450   | 0.8579   | 0.8526   | 0.8349    |
| <b>Mean</b>   | 0.8921   | 0.9119   | 0.9101   | 0.8782   | 0.7545   | 0.8934   | 0.9055   | 0.9077   | 0.9001   | 0.8563    |

**Table 4.8: Coefficients of determination for PTFE coated glass fibre (B18059) strain to stress neural network material model, trained with 5 load ratios**

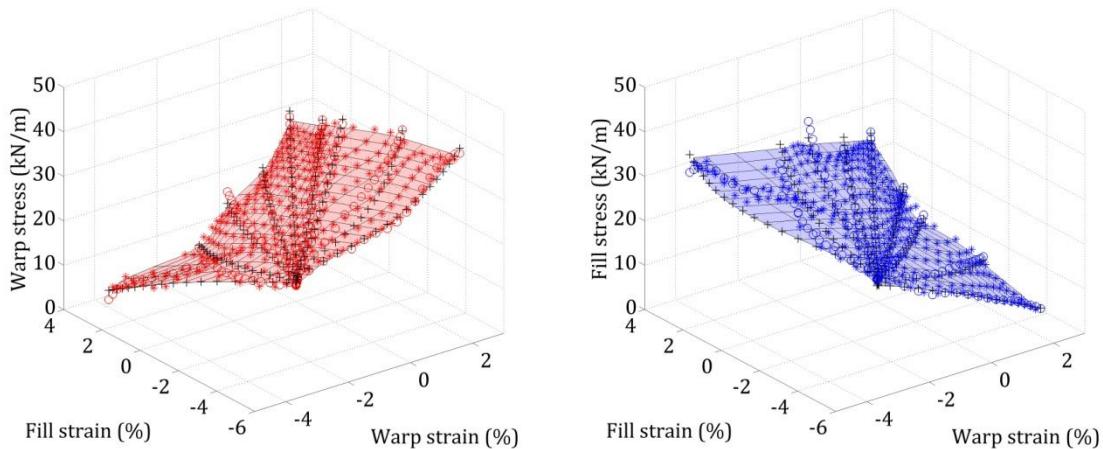


**Figure 4-22: Network 3, Table 4.8, PTFE (B18059) strain to stress network demonstrating best performance with experimentally generated data partially used in training**

The one to many mapping issue, leading to poor performance, is also demonstrated by the networks trained with the network generated data set (Figure 4-23). The warp surface shows the most improvement compared to the networks trained with only experimental data. However, the fill surface remains extremely poor. Although these networks show the best overall performance they are not sufficiently accurate for use in a structural simulation.

| Network performance when presented 9 experimental load ratios (unseen)  |        |        |        |        |        |        |        |        |        |        |
|---|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Network   | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | 10     |
| Warp  | 0.9769 | 0.9751 | 0.9778 | 0.9608 | 0.9773 | 0.9734 | 0.9734 | 0.9743 | 0.9749 | 0.9737 |
| Fill  | 0.9508 | 0.9520 | 0.9487 | 0.9315 | 0.9425 | 0.9382 | 0.9418 | 0.9443 | 0.9442 | 0.9275 |
| Mean  | 0.9639 | 0.9635 | 0.9633 | 0.9461 | 0.9599 | 0.9558 | 0.9576 | 0.9593 | 0.9596 | 0.9506 |
| Network performance when network presented with network data (training) |        |        |        |        |        |        |        |        |        |        |
| Network   | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | 10     |
| Warp  | 0.9886 | 0.9915 | 0.9902 | 0.9905 | 0.9886 | 0.9910 | 0.9905 | 0.9917 | 0.9914 | 0.9918 |
| Fill  | 0.9664 | 0.9698 | 0.9697 | 0.9644 | 0.9645 | 0.9693 | 0.9687 | 0.9697 | 0.9697 | 0.9702 |
| Mean  | 0.9775 | 0.9806 | 0.9799 | 0.9775 | 0.9765 | 0.9801 | 0.9796 | 0.9807 | 0.9805 | 0.9810 |

**Table 4.9: Coefficients of determination for PTFE coated glass fibre (B18059) strain to stress neural network material model, trained with network generated data**



**Figure 4-23: Network 1, Table 4.6, PTFE (B18059) strain to stress network demonstrating best performance with 'unseen' experimentally generated data used in testing**

All of the 2 input networks trained using the PTFE (B18059) data sets achieve unacceptably low  $R^2$  values (less than 0.990) indicating a poor correlation between network output and the target outputs. When presented with unseen data the PTFE networks exhibit a very poor ability to generalise the response.

#### 4.3.3.2 Additional Input

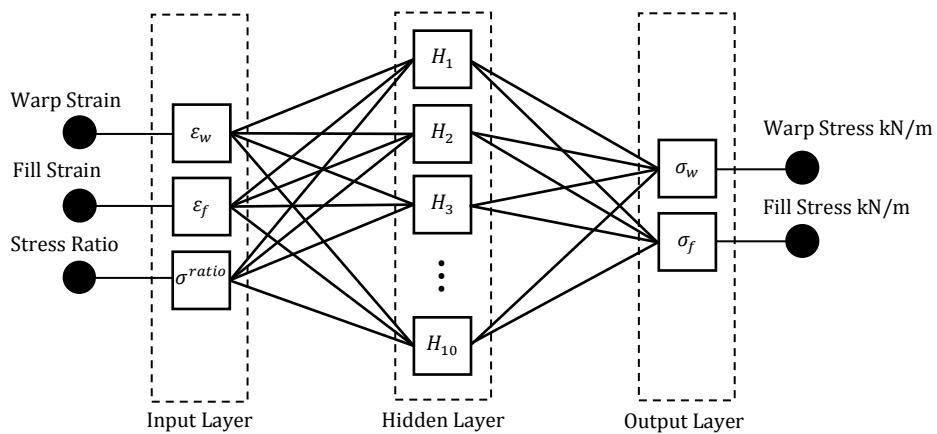
In order to capture the more complex material response of PTFE which includes one to many mappings, a third input is required. The state equations of the finite element analysis are solved iteratively via a dynamic relaxation algorithm, Section 3.1.2. The warp and fill stress from the previous dynamic relaxation energy peak is readily available. This information may be used to provide the approximate strain radial arm on which the target stress is positioned in the form of a stress ratio. As in the case of the activation functions within the hidden layer of the networks a tan sigmoid (or hyperbolic tan) function is used

to scale the resulting ratio between -1 and 1. Therefore, the additional stress ratio input,  $\sigma^{ratio}$ , is given by,

$$\sigma^{ratio} = \tanh\left(\frac{\sigma_w^{n-1}}{\sigma_f^{n-1}}\right) \quad (4-35)$$

where  $\sigma_w^{n-1}$  and  $\sigma_f^{n-1}$  are the warp and fill stresses from the previous dynamic relaxation energy peak. These values will become increasingly accurate as the analysis progresses.

In order to generate values of  $\sigma^{ratio}$  for the training data set random perturbations in the range of -1.0 to 1.0 are applied to the target stresses to be used in place of  $\sigma_w^{n-1}$  and  $\sigma_f^{n-1}$ . This range of perturbation is used to generate previous iteration stress values in a similar range to those generated during implementation within finite element analysis.



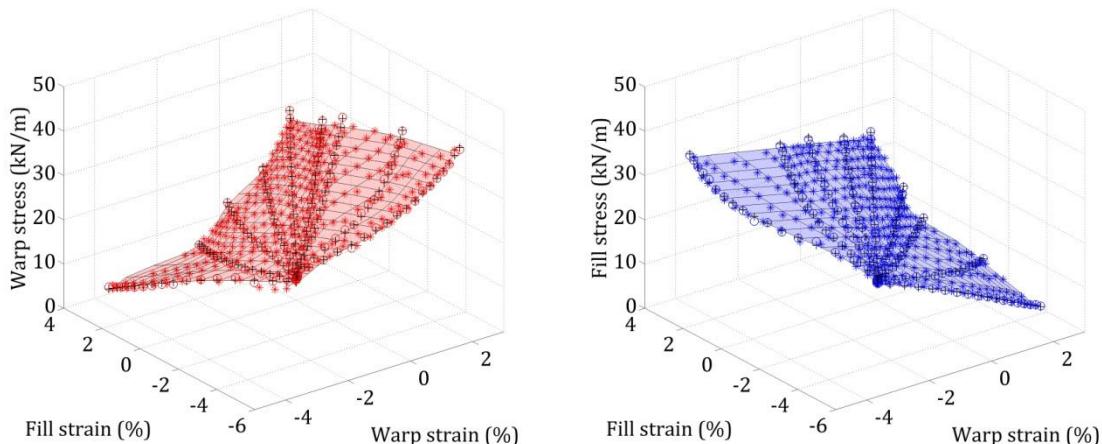
**Figure 4-24: 3 input strain to stress neural network architecture**

The resulting 3 input strain to stress neural network, shown in Figure 4-24, comprises an input layer containing three inputs, warp and fill strain and the stress ratio, a hidden layer containing 10 neurons and an output layer containing 2 neurons for warp and fill strain. Tan sigmoid transfer functions are used in the neurons of the hidden layer and a linear transfer function is used in the neurons of the output layer.

All of the networks trained using the PTFE (B18059) data sets with the addition stress ratio input show a considerable improvement in performance. Once again the highest  $R^2$  values are observed when a network is tested using the same set of data it was trained with. When presented with unseen data the 3 input PTFE networks exhibit a good ability to generalise the response. Version 4 of the network trained using the 9 load ratios (Figure 4-25) shows the best performance across all data sets producing  $R^2$  values greater than 0.990. The fill surface also consistently produces a greater  $R^2$  value than the warp when presented with the network generated unseen data set.

| <b>Network performance when presented 9 experimental load ratios (training)</b> |          |          |          |          |          |          |          |          |          |           |
|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|
| <b>Network</b>  | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> | <b>10</b> |
| <b>Warp</b>   | 0.9982   | 0.9995   | 0.9972   | 0.9988   | 0.9984   | 0.9995   | 0.9994   | 0.9939   | 0.9993   | 0.9988    |
| <b>Fill</b>   | 0.9988   | 0.9992   | 0.9986   | 0.9990   | 0.9991   | 0.9991   | 0.9990   | 0.9979   | 0.9992   | 0.9989    |
| <b>Mean</b>   | 0.9985   | 0.9993   | 0.9979   | 0.9989   | 0.9988   | 0.9993   | 0.9992   | 0.9959   | 0.9992   | 0.9989    |
| <b>Network performance when network presented with network data (unseen)</b>    |          |          |          |          |          |          |          |          |          |           |
| <b>Network</b>  | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> | <b>10</b> |
| <b>Warp</b>   | 0.9674   | 0.9839   | 0.9515   | 0.9903   | 0.9774   | 0.9880   | 0.9861   | 0.9795   | 0.9838   | 0.9609    |
| <b>Fill</b>   | 0.9976   | 0.9981   | 0.9970   | 0.9981   | 0.9986   | 0.9984   | 0.9984   | 0.9971   | 0.9983   | 0.9964    |
| <b>Mean</b>   | 0.9825   | 0.9910   | 0.9743   | 0.9942   | 0.9880   | 0.9932   | 0.9923   | 0.9883   | 0.9911   | 0.9786    |

**Table 4.10: Coefficients of determination for PTFE coated glass fibre (B18059) strain to stress neural network material model, trained with 9 load ratios**



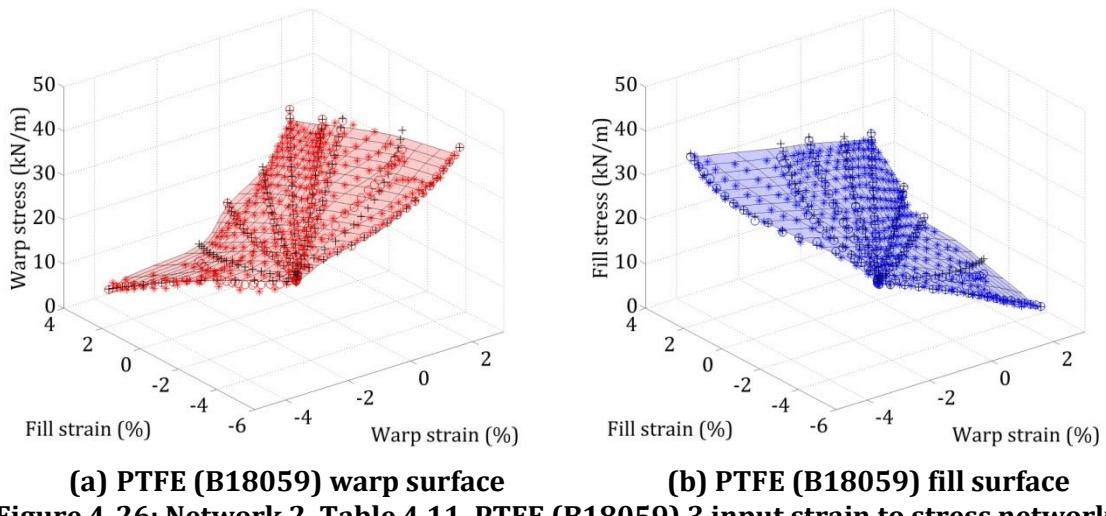
**Figure 4-25: Network 4, Table 4.10, PTFE (B18059) 3 input strain to stress network demonstrating best performance with 'unseen' network generated data used in testing**

The networks trained using the partial data set again achieve the lowest performance.

Version 2 (Figure 4-26) produces the highest  $R^2$  values when tested with both the full experimental load ratio set and the network generated data set. However, this performance is poor especially for the load ratio arms not used in the training set. The 'unseen' load ratio arms between the 0:1 and 1:2 and 2:1 and 1:0 load ratios collapse onto the 0:1 and 1:0 load ratio arms respectively. This demonstrates that the additional load ratio arms are required to capture the complex material response of PTFE coated glass fibre fabric. This may indicate that further load ratio arms should be added to the biaxial testing profile in order to provide unseen experimental testing data.

| <b>Network performance when presented 9 experimental load ratios (partial training)</b> |          |          |          |          |          |          |          |          |          |           |
|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|
| <b>Network</b>  | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> | <b>10</b> |
| <b>Warp</b>   | 0.9367   | 0.9772   | 0.9749   | 0.9743   | 0.9650   | 0.9789   | 0.9637   | 0.8864   | 0.8894   | 0.9765    |
| <b>Fill</b>   | 0.8871   | 0.9912   | 0.9366   | 0.9193   | 0.9289   | 0.8934   | 0.9519   | 0.8412   | 0.9260   | 0.9675    |
| <b>Mean</b>   | 0.9119   | 0.9842   | 0.9558   | 0.9468   | 0.9469   | 0.9361   | 0.9578   | 0.8638   | 0.9077   | 0.9720    |
| <b>Network performance when network presented with network data (unseen)</b>            |          |          |          |          |          |          |          |          |          |           |
| <b>Network</b>  | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> | <b>10</b> |
| <b>Warp</b>   | 0.9356   | 0.9640   | 0.9755   | 0.9649   | 0.9326   | 0.9583   | 0.9411   | 0.7176   | 0.8822   | 0.9498    |
| <b>Fill</b>   | 0.8347   | 0.9828   | 0.9068   | 0.8784   | 0.8946   | 0.8404   | 0.9194   | 0.7713   | 0.8826   | 0.9467    |
| <b>Mean</b>   | 0.8852   | 0.9734   | 0.9411   | 0.9217   | 0.9136   | 0.8994   | 0.9303   | 0.7444   | 0.8824   | 0.9482    |

**Table 4.11: Coefficients of determination for PTFE coated glass fibre (B18059) strain to stress neural network material model, trained with 5 load ratios**



**Figure 4-26: Network 2, Table 4.11, PTFE (B18059) 3 input strain to stress network demonstrating best performance with both data sets**

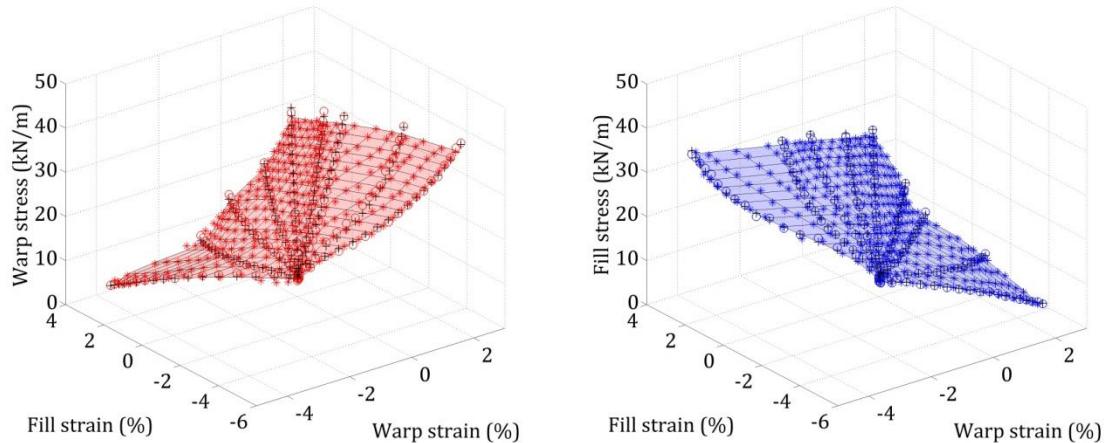
Version 5 of the network trained with the network generated data set (Figure 4-27) demonstrates the best overall performance when presented with both the training data and the unseen data sets. This network may be selected as having the best ability to generalise the PTFE fabric response. These results demonstrate that the additional input is necessary for the network to produce a good prediction of stress from strain for a PTFE coated glass fabric. This network requires further testing while implemented within a finite element analysis in order to ensure that the training load ratio inputs are within an appropriate range.

| <b>Network performance when presented 9 experimental load ratios (unseen)</b> |          |          |          |          |          |          |          |          |          |           |
|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|
| <b>Network</b>  | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> | <b>10</b> |
| <b>Warp</b>   | 0.9916   | 0.9896   | 0.9927   | 0.9910   | 0.9949   | 0.9896   | 0.9913   | 0.9926   | 0.9915   | 0.9909    |
| <b>Fill</b>   | 0.9985   | 0.9988   | 0.9986   | 0.9986   | 0.9986   | 0.9980   | 0.9986   | 0.9987   | 0.9975   | 0.9986    |
| <b>Mean</b>   | 0.9950   | 0.9942   | 0.9957   | 0.9948   | 0.9968   | 0.9938   | 0.9950   | 0.9956   | 0.9945   | 0.9947    |

| <b>Network performance when network presented with network data (training)</b> |          |          |          |          |          |          |          |          |          |           |
|--|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|
| <b>Network</b>   | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> | <b>8</b> | <b>9</b> | <b>10</b> |
| <b>Warp</b>  | 0.9984   | 0.9982   | 0.9983   | 0.9984   | 0.9981   | 0.9977   | 0.9988   | 0.9982   | 0.9946   | 0.9984    |
| <b>Fill</b>  | 0.9988   | 0.9989   | 0.9988   | 0.9988   | 0.9989   | 0.9985   | 0.9988   | 0.9989   | 0.9982   | 0.9988    |
| <b>Mean</b>  | 0.9986   | 0.9985   | 0.9986   | 0.9986   | 0.9985   | 0.9981   | 0.9988   | 0.9986   | 0.9964   | 0.9986    |

**Table 4.12: Coefficients of determination for PTFE coated glass fibre (B18059) strain to stress neural network material model, trained with network generated data**



**Figure 4-27: Network 5, Table 4.10, PTFE (B18059) 3 input strain to stress network demonstrating best performance with 'unseen' experimentally generated data used in testing**

## 4.4 History Neural Network Material Model

To date limited attempts have been made incorporate the effects of load history and residual strain within fabric material models. It has been shown in [114] that relatively simple neural network models are capable of representing hysteretic behaviour through the use of ‘internal variables’ similar to the load ratio used in Section 4.3.3.2. This method will be used to capture the hysteretic behaviour of architectural fabric.

The use of internal variables is proposed by Yun et al [114] as a solution to transform a one to many mapping to a single valued mapping in order to model materials which exhibit hysteretic behaviour. Initially these internal variables are adopted to capture the uniaxial hysteretic behaviour of PVC coated polyester architectural fabric. This will then be extended to biaxial hysteretic behaviour of both PVC coated polyester and PTFE coated glass.

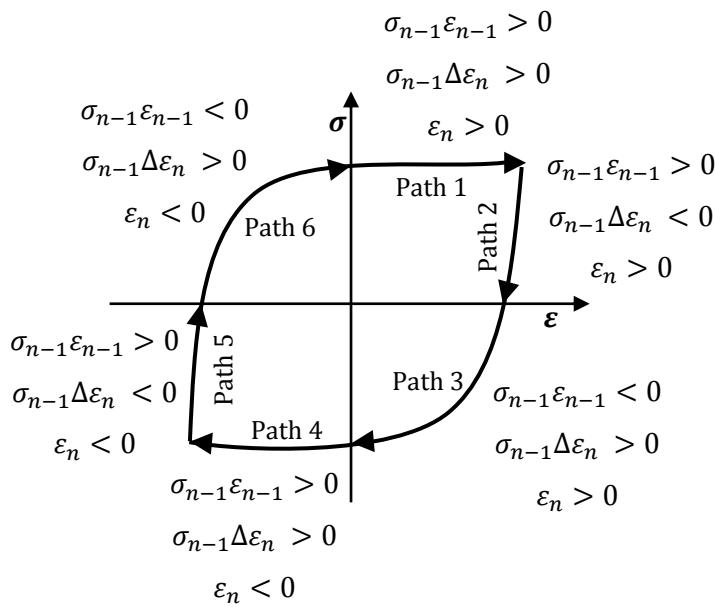
### 4.4.1 Network internal variables [114]

For a strain controlled material model the following two ‘phenomenological’ variables may be used to describe the material behaviour. The value may be either scalar, as is the case of a uniaxial model, or tensor, as in the case of a biaxial model [114]. The first internal variable,  $\xi_n$  is the product of the previous level of stress  $\sigma_{n-1}$  and the previous level of strain  $\varepsilon_{n-1}$ . The second internal variable is the product of the previous level of stress and the current strain increment. The subscript  $n$  denotes the  $n^{th}$  incremental step.

$$\xi_n = \sigma_{n-1} \varepsilon_{n-1} \quad (4-36)$$

$$\Delta\eta_n = \sigma_{n-1} (\varepsilon_n - \varepsilon_{n-1}) = \sigma_{n-1} \Delta\varepsilon_n \quad (4-37)$$

The following proof that the neural network functional mapping is single-valued is given in [114]. A closed hysteresis loop is subdivides in 6 separate paths as shown in Figure 4-28. The sign of each of the three independent network inputs,  $\varepsilon_n$ ,  $\xi_n$  and  $\Delta\eta_n$  are assessed along with the sign of the network output,  $\sigma_n$  and recorded in Table 4.13.



**Figure 4-28: Variable sign combinations for strain controlled hysteresis loop**

|               | $\xi_n$ | $\Delta\eta_n$ | $\varepsilon_n$ |   | $\sigma_n$ |
|---------------|---------|----------------|-----------------|---|------------|
| <b>Path 1</b> | +       | +              | +               | → | +          |
| <b>Path 2</b> | +       | -              | +               |   | +          |
| <b>Path 3</b> | -       | +              | +               |   | -          |
| <b>Path 4</b> | +       | +              | -               |   | -          |
| <b>Path 5</b> | +       | -              | -               |   | -          |
| <b>Path 6</b> | -       | +              | -               |   | +          |

**Table 4.13: Variable sign combinations for strain controlled hysteresis loop**

Due to the complexity of the biaxial response of architectural fabric this simple proof is not sufficient for the tensor case. However, extensive testing using ‘unseen’ data will be used to validate the model for biaxial representation. To reduce the number of inputs it has been demonstrated in [114] that the two internal variables may be combined into a single internal variable,  $\zeta_n$  with no reduction in performance.

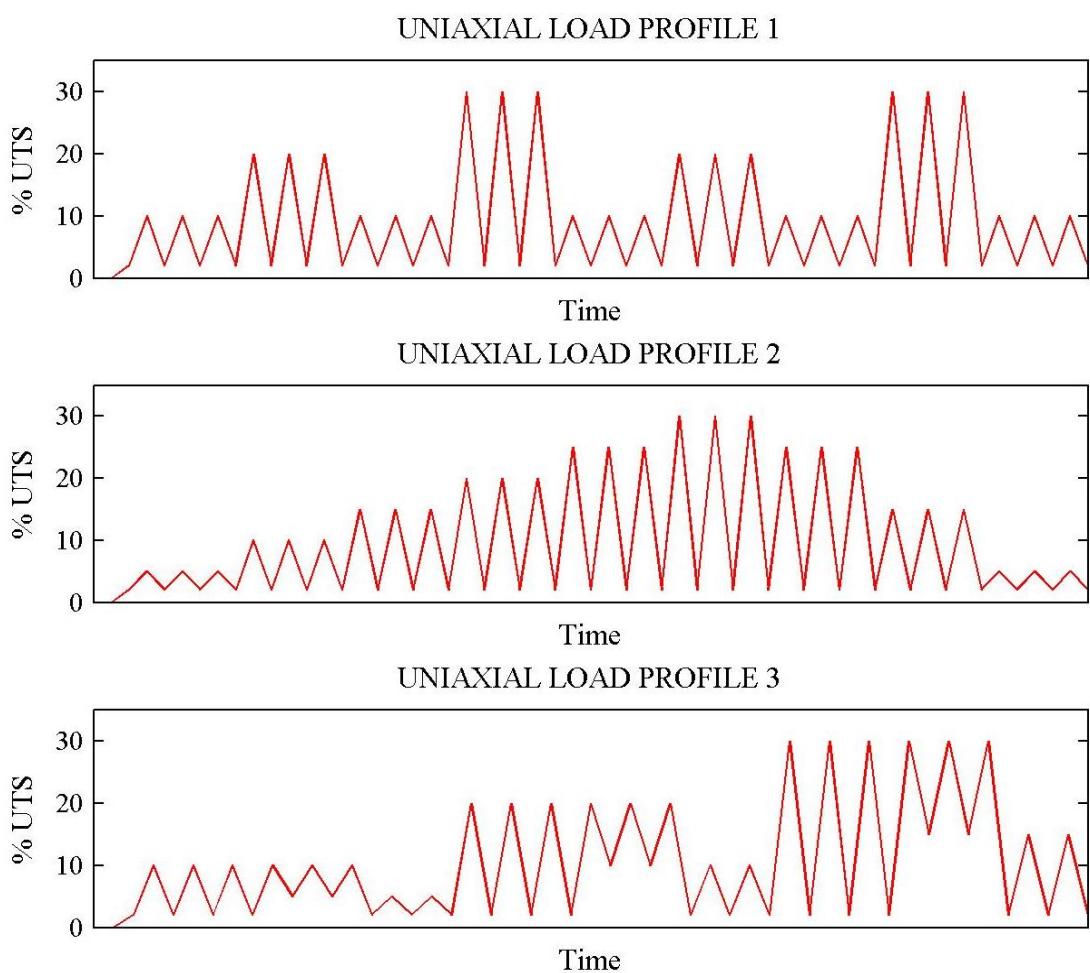
$$\zeta_n = \xi_n + \Delta\eta_n \quad (4-38)$$

#### 4.4.2 Cyclic uniaxial training data collection and pre-processing

The uniaxial testing protocol is based on the British Standard ‘Rubber or plastics coated fabrics - Determination of tensile strength and elongation at break’ (BS EN ISO1421:1998) [128]. Testing is completed using an Instron constant rate extension machine.

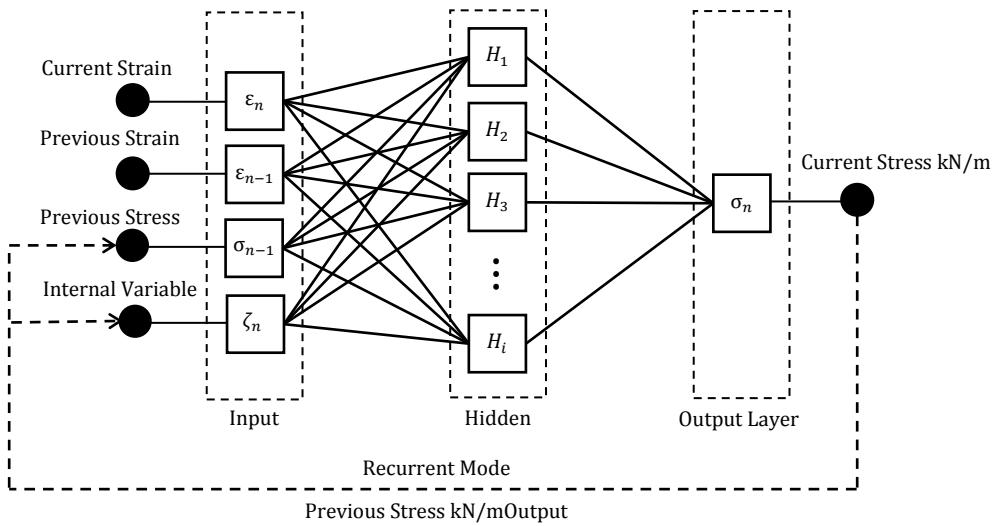
Each test piece of PVC coated polyester fabric was cut to be  $50 \text{ mm} \pm 0.5 \text{ mm}$  wide and of sufficient length to allow a distance of  $200 \text{ mm} \pm 1 \text{ mm}$  between the jaws of the test

machine. All samples were taken from the central region of the fabric roll and were aligned along the warp yarn direction. The samples were mounted in the Instron machine using serrated jaws tightened by hand using a strap wrench. The distance between the jaws is set to 200mm using vernier callipers and the position of the head re-set to zero. The load was not zeroed after the sample has been mounted in order to record any initial loading induced during mounting. Deformation was applied at a constant rate of 10mm/min and loading is controlled by a load profile. Three different profiles, shown in Figure 8, were used to provide full data sets for training and testing and to investigate the ideal training profile.



**Figure 4-29: Uniaxial cyclic load profiles 1,2 and 3, load is shown in terms of percentage of ultimate tensile strength for use with a range of fabrics**

#### 4.4.3 Cyclic uniaxial network training and validation



**Figure 4-30: 4 input strain to stress uniaxial hysteresis neural network architecture**

As in the previous study a Matlab fitting neural network is used. The resulting 4 input strain to stress neural network, shown in Figure 4-30, comprises an input layer containing four inputs, a single hidden layer containing a user defined number of nodes and an output layer containing a single output. The network model input comprises current strain, previous strain, previous stress, and the internal variables combined into a single input via addition. The output is the current level of stress. When run in recurrent mode the network output stress is used to produce the internal variable and previous stress input, Tan sigmoid transfer functions are used in the neurons of the hidden layer and a linear transfer function is used in the neurons of the output layer. For the purposes of early stopping 70% of the full training data set is used for training, 15% for validation and 15% for testing. This division will be used for all networks trained in this section.

An initial brief study into the effect of training profile, training data density and number of nodes in hidden layer was undertaken using the PVC data sets. This was done to inform the network architecture and training profile design. Different densities of training data were investigated. The experimental data is reduced according to a reduction factor. Each loading and unloading cycle is reduced to contain the number of data points specified by the reduction factor. Three training data sets were generated for each uniaxial profile the first with a reduction factor of 6, the second with a reduction factor of 10 and the third with a reduction factor of 14. Therefore, a total of 9 data sets were generated.

Three networks were trained for each training data set. The first with 4 nodes in the hidden layer, a second with 7 nodes and a third with 10 nodes. To account for the effect of

random node initiation, 10 networks were trained for each case and the network which exhibited the lowest error when tested using the full training data set in recurrent mode was selected. Each selected network is then tested using the remaining 8 'unseen' data sets. As in Section 4.3.2 the network performance was assessed using the  $R^2$  value, equation (4-32).

The full sets of  $R^2$  values are presented below in Table 4.14 to Table 4.16. The results are grouped according to the number of nodes of the hidden layer. The details of the trained networks are given in the column headers and the row titles give details of the testing data. The grey cells contain the  $R^2$  values relating to testing using the training data sets. The remaining values represent performance relating to testing using 'unseen' data. The  $R^2$  values are calculated in recurrent mode as this is the way in which a network would function once implemented.

It can be seen in all tables that generalisation across the three profiles is typically best when the network is trained using Profile 3. This is because profile 3 is the only profile which contains features from across all 3 of the profiles being investigated. For this reason, Profile 3 also yields the lowest regression values when used as an 'unseen' testing set for networks which have been trained with Profile 1 or Profile2. This highlights the importance of developing a comprehensive training data set.

|                  |           | Training Data |       |       |           |       |       |           |       |       |       |       |
|------------------|-----------|---------------|-------|-------|-----------|-------|-------|-----------|-------|-------|-------|-------|
| Reduction Factor |           | Profile 1     |       |       | Profile 2 |       |       | Profile 3 |       |       | Mean  |       |
|                  |           | 6             | 10    | 14    | 6         | 10    | 14    | 6         | 10    | 14    |       |       |
| Testing Data     | Profile 1 | 6             | 0.995 | 0.943 | 0.249     | 0.990 | 0.933 | 0.633     | 0.990 | 0.869 | 0.911 | 0.835 |
|                  |           | 10            | 0.972 | 0.997 | 0.936     | 0.987 | 0.992 | 0.964     | 0.989 | 0.991 | 0.987 | 0.979 |
|                  |           | 14            | 0.952 | 0.991 | 0.995     | 0.973 | 0.989 | 0.992     | 0.984 | 0.992 | 0.993 | 0.985 |
|                  | Profile 2 | 6             | 0.988 | 0.965 | 0.602     | 0.995 | 0.952 | 0.788     | 0.978 | 0.912 | 0.932 | 0.901 |
|                  |           | 10            | 0.956 | 0.982 | 0.970     | 0.981 | 0.997 | 0.981     | 0.967 | 0.979 | 0.972 | 0.976 |
|                  |           | 14            | 0.934 | 0.967 | 0.986     | 0.963 | 0.995 | 0.997     | 0.956 | 0.972 | 0.977 | 0.972 |
|                  | Profile 3 | 6             | 0.947 | 0.785 | 0.474     | 0.878 | 0.913 | 0.672     | 0.994 | 0.916 | 0.938 | 0.835 |
|                  |           | 10            | 0.929 | 0.779 | 0.914     | 0.835 | 0.947 | 0.899     | 0.993 | 0.997 | 0.990 | 0.920 |
|                  |           | 14            | 0.914 | 0.779 | 0.942     | 0.806 | 0.945 | 0.889     | 0.992 | 0.997 | 0.997 | 0.918 |
|                  | Mean      | 0.954         | 0.910 | 0.785 | 0.934     | 0.963 | 0.868 | 0.983     | 0.958 | 0.966 |       |       |

Table 4.14:  $R^2$  value matrix for 10 hidden node networks

|                  |           | Training Data |       |       |           |       |       |           |       |       |       |       |
|------------------|-----------|---------------|-------|-------|-----------|-------|-------|-----------|-------|-------|-------|-------|
| Reduction Factor |           | Profile 1     |       |       | Profile 2 |       |       | Profile 3 |       |       | Mean  |       |
|                  |           | 6             | 10    | 14    | 6         | 10    | 14    | 6         | 10    | 14    |       |       |
| Testing Data     | Profile 1 | 6             | 0.995 | 0.935 | 0.852     | 0.991 | 0.972 | 0.704     | 0.992 | 0.716 | 0.835 | 0.888 |
|                  |           | 10            | 0.975 | 0.996 | 0.985     | 0.991 | 0.993 | 0.972     | 0.989 | 0.993 | 0.984 | 0.986 |
|                  |           | 14            | 0.941 | 0.993 | 0.997     | 0.975 | 0.991 | 0.993     | 0.984 | 0.993 | 0.995 | 0.985 |
|                  | Profile 2 | 6             | 0.983 | 0.955 | 0.897     | 0.995 | 0.981 | 0.843     | 0.985 | 0.959 | 0.896 | 0.944 |
|                  |           | 10            | 0.944 | 0.994 | 0.99      | 0.985 | 0.996 | 0.988     | 0.980 | 0.968 | 0.977 | 0.980 |
|                  |           | 14            | 0.865 | 0.988 | 0.994     | 0.967 | 0.989 | 0.997     | 0.974 | 0.962 | 0.948 | 0.965 |
|                  | Profile 3 | 6             | 0.709 | 0.874 | 0.797     | 0.878 | 0.887 | 0.774     | 0.996 | 0.234 | 0.868 | 0.780 |
|                  |           | 10            | 0.57  | 0.941 | 0.933     | 0.836 | 0.878 | 0.932     | 0.992 | 0.992 | 0.986 | 0.896 |
|                  |           | 14            | 0.504 | 0.939 | 0.945     | 0.805 | 0.862 | 0.931     | 0.988 | 0.993 | 0.997 | 0.885 |
|                  |           | Mean          | 0.832 | 0.957 | 0.932     | 0.936 | 0.950 | 0.904     | 0.987 | 0.868 | 0.943 |       |

Table 4.15:  $R^2$  value matrix for 7 hidden node networks

|                  |           | Training Data |       |       |           |       |       |           |       |       |       |       |
|------------------|-----------|---------------|-------|-------|-----------|-------|-------|-----------|-------|-------|-------|-------|
| Reduction Factor |           | Profile 1     |       |       | Profile 2 |       |       | Profile 3 |       |       | Mean  |       |
|                  |           | 6             | 10    | 14    | 6         | 10    | 14    | 6         | 10    | 14    |       |       |
| Testing Data     | Profile 1 | 6             | 0.995 | 0.976 | 0.922     | 0.959 | 0.885 | 0.294     | 0.986 | 0.910 | 0.964 | 0.877 |
|                  |           | 10            | 0.982 | 0.995 | 0.989     | 0.951 | 0.994 | 0.946     | 0.983 | 0.668 | 0.984 | 0.944 |
|                  |           | 14            | 0.965 | 0.987 | 0.995     | 0.936 | 0.990 | 0.995     | 0.978 | 0.130 | 0.980 | 0.884 |
|                  | Profile 2 | 6             | 0.994 | 0.98  | 0.941     | 0.994 | 0.949 | 0.609     | 0.988 | 0.954 | 0.978 | 0.932 |
|                  |           | 10            | 0.981 | 0.992 | 0.987     | 0.977 | 0.996 | 0.979     | 0.979 | 0.607 | 0.991 | 0.943 |
|                  |           | 14            | 0.965 | 0.978 | 0.988     | 0.948 | 0.988 | 0.997     | 0.970 | 0.451 | 0.988 | 0.919 |
|                  | Profile 3 | 6             | 0.893 | 0.911 | 0.906     | 0.927 | 0.853 | 0.470     | 0.988 | 0.961 | 0.968 | 0.875 |
|                  |           | 10            | 0.846 | 0.862 | 0.963     | 0.925 | 0.840 | 0.887     | 0.984 | 0.993 | 0.991 | 0.921 |
|                  |           | 14            | 0.82  | 0.825 | 0.969     | 0.933 | 0.792 | 0.870     | 0.977 | 0.951 | 0.991 | 0.903 |
|                  |           | Mean          | 0.938 | 0.945 | 0.962     | 0.950 | 0.921 | 0.783     | 0.981 | 0.736 | 0.982 |       |

Table 4.16:  $R^2$  value matrix for 4 hidden node networks

The size of the reduction factor effects the quantity of training data available but also effects the size of the load step between iterations. When testing using corresponding reduction factors, network generalisation typically improves with reducing reduction factor. This indicates that a more sparse data set or higher load step achieves better results. However, changes in load step between training and testing sets will also affect network performance. From the tables above it can be seen that network performance

reduces as the difference between the testing reduction factor and the training reduction factor increases. Inspecting the mean  $R^2$  value across all training sets testing with a reduction factor of 10 usually produces the highest performance.

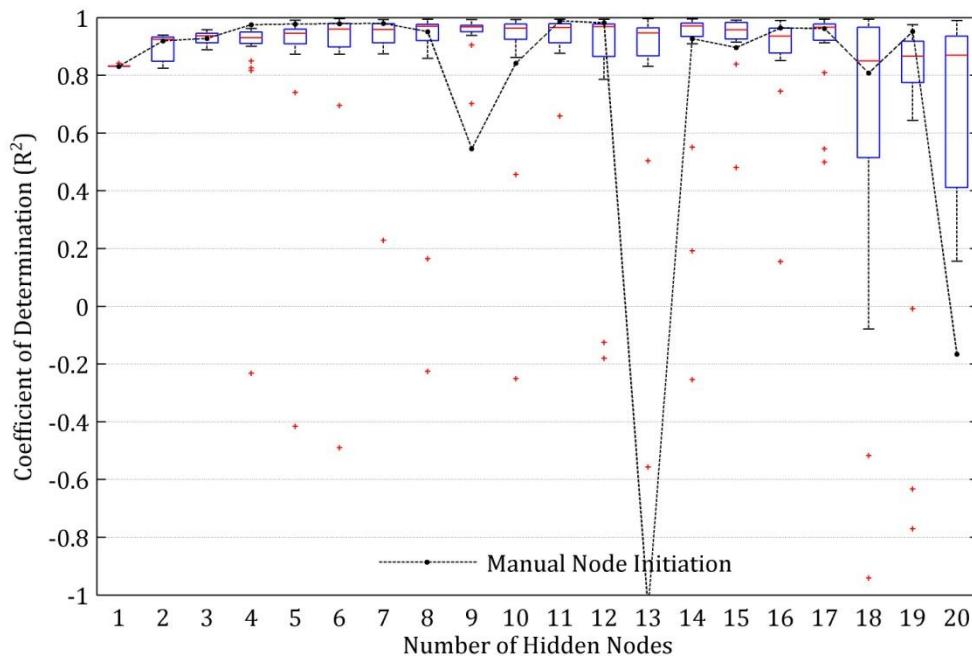
It is difficult to identify the most effective reduction factor, and resulting load step, for training from Table 4.14 to Table 4.16. However, it is likely that a training load step in the middle of the expected load step range is most suitable as it will be on average closest to all load steps within the testing range. It is also probable that the load step during analysis will be large and variable compared with data gathered through testing. The reduction factor for training should be chosen with this in mind. Therefore, network generalisation across varying load steps may be increased by creating a training data set containing non-uniform load steps. This may be done by reducing the training data using a varying, random reduction factor within a suitable range.

The effect of number of nodes in the hidden layer is difficult to identify. However, the number of nodes required in the hidden layer is a contributing factor to the ability of a network to generalise. The lowest  $R^2$  values, indicating poorest performance, occur more frequently in networks with fewer hidden nodes (Table 4.16). This may indicate that the 4 node network does not have sufficient complexity to accurately capture the fabric response. When the mean  $R^2$  value across all testing data sets is inspected the network with the highest ability to generalise has 7 hidden nodes and is trained using profile 3 reduced by a reduction factor of 6.

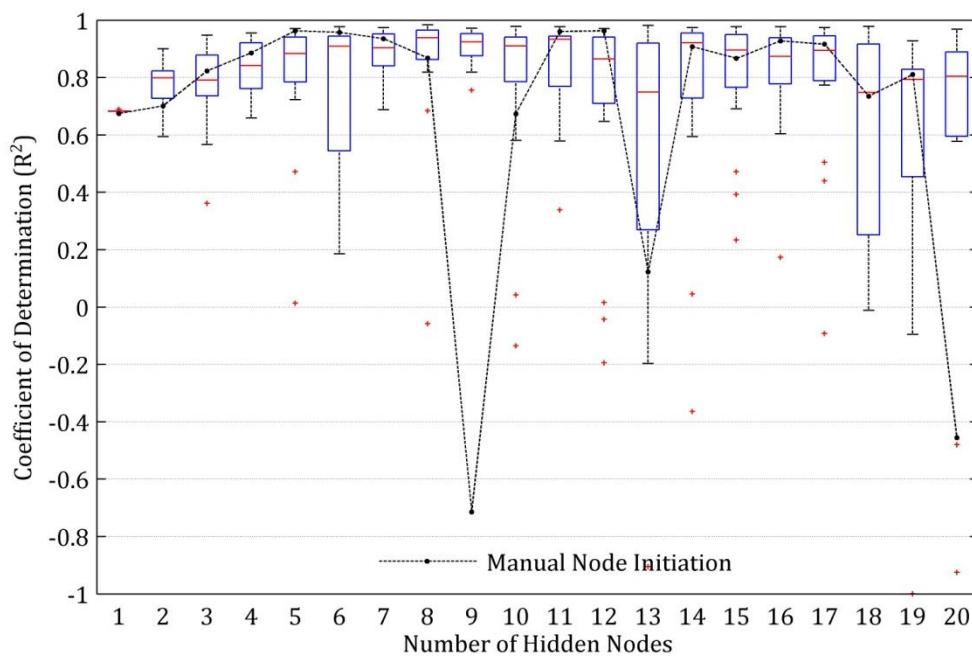
Various methods for the selection of an optimum number of nodes have been proposed. Methods of optimisation implemented during training exist and often involve the removal or ‘pruning’ of unnecessary neurons. Ghaboussi and Sidartra developed an adaptive training regime whereby additional nodes are progressively added to the network layers as existing nodes become saturated [99, 129]. Other widely used more simplistic methods for selecting the number of neurons in a network are based on either the ratio between number of training patterns and network weights or the number of input and output nodes.

The effect of the number of nodes in the hidden layer is investigated further by training networks with a number of hidden nodes from 1 to 20. The training data set with the best overall performance from the previous investigation (Profile 3 reduced by a reduction factor of 10) is used as the training set. 20 randomly initiated networks are trained for each number of hidden nodes. In addition a single network manually initiated with weights and biases set to 1 is also trained for each material. The performance of this

network is plotted alongside the boxplots. Each trained network is tested in recurrent mode using both the training data set and an 'unseen' testing data set generated from Profile 1 also reduced with a reduction factor of 10. The  $R^2$  values generated from recurrent testing using the training and testing data sets are plotted in the form of box-plots in Figure 4-31a and Figure 4-31b respectively.



**(a) Testing performance with training data set in recurrent mode**



**(b) Testing performance with 'unseen' testing data set in recurrent mode**

Figure 4-31: The effect of hidden node number on performance.

The worst  $R^2$  value range in Figure 4-31 is produced by the 1 node networks. All other networks produce a range of  $R^2$  values. Outliers outside the maximum  $R^2$  value range, defined by red points in Figure 4-31, are also produced by all networks with more than 2 nodes in the hidden layer. The maximum range, or whisker length, is given by,

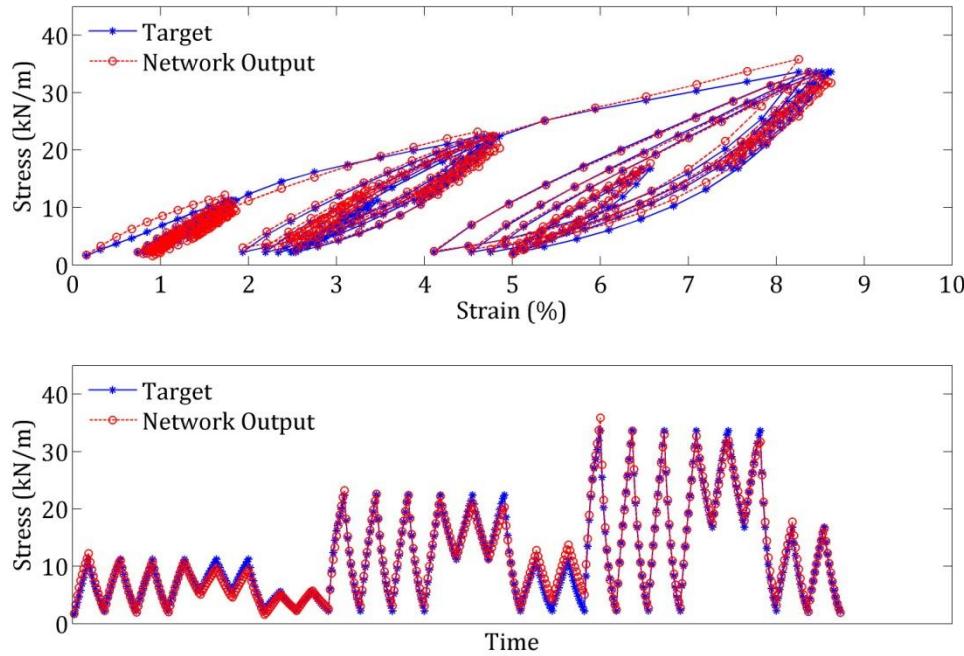
$$Q_3 + w(Q_3 - Q_1) < \text{Outlier} < Q_1 - w(Q_3 - Q_1) \quad (4-39)$$

where  $Q_1$  and  $Q_3$  are the 25th and 75th percentiles, respectively, and  $w$  is a constant equal to 1.5 which corresponds to approximately 99.3 coverage if the data points are normally distributed. The whisker length is defined by the most extreme value which are not considered outliers.

The differing levels of performance between networks with the same architecture is a product of the complexity of the error function minimised during training and the starting point defined by the randomised initial values of the weights and biases. The number of local minima in the error surface searched during training increase with increasing numbers of hidden nodes. It is therefore more likely that a local minimum as oppose to a global minimum will be found during training. Early over fitting will also increase with increasing hidden node number. This will lead to early stopping of training prior to the network fully 'learning' the material response. This again highlights the necessity of training and testing multiple randomly initiated networks. Testing with both training data and 'unseen' testing data should be considered when identifying the best network.

In both Figure 4-31a and b the median  $R^2$  value consistently increases with increasing hidden nodes up to 9 nodes. When testing using the training data set (Figure 4-31a) median performance and  $R^2$  value range remains approximately the same between networks with from 9 to 17 hidden nodes. However, in Figure 4-31b, median performance generally drops and the  $R^2$  value range shows a marked increase in networks with more than 9 nodes in the hidden layer. This indicates that networks with more than 9 nodes are more likely to over fit the training data and lose the ability to generalise between different profiles. In both figures there is a considerable drop in median performance as well as a marked increase in the range in networks with more than 17 nodes. This indicates that in networks with more than 17 hidden nodes over fitting, to a point where results produced when testing the network in recurrent mode leads to unacceptable errors, is more likely to occur. This is the case when tested with both training and 'unseen' testing data sets. These networks have effectively lost all ability to generalise and small deviations from the training data presented to the network lead to disproportionately large errors.

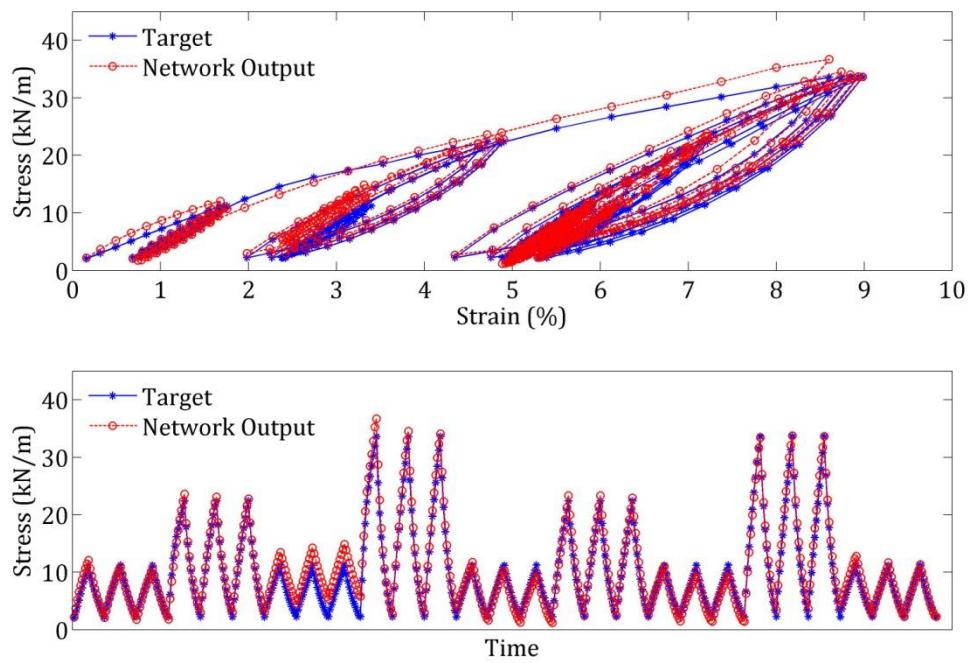
From the investigations above a network architecture including 9 hidden nodes in the hidden layer is selected to capture the fabric response to cyclic uniaxial loading. Profile 3 reduced using a reduction factor of 10 is used for network training and Profile 1 and 2 reduced with the same factor are used as 'unseen' test data. The network is tested in recurrent mode.



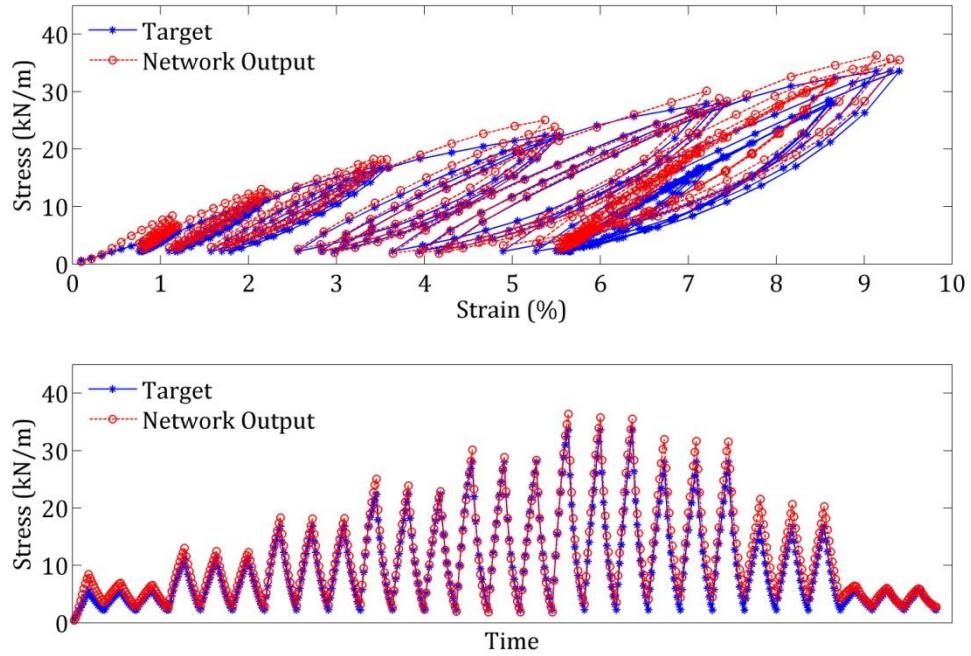
**Figure 4-32: Network output tested in recurrent mode using training data gathered using Profile 3.**

Figure 4-32 shows the network output generated in recurrent mode using the training profile. The network model successfully captures the complex hysteretic fabric response. Increasing error between the experimental target data and network output in the central region of the profile occurs but is recovered later in the profile. This indicates that the network is resistant to error accumulation occurring during network simulation in recurrent mode.

The powerful generalisation capability of the network is demonstrated in Figure 4-33 and Figure 4-34 where it is tested with 'unseen' data gathered using profiles 1 and 2 respectively. A similar pattern of error is observed in Figure 4-33 in the third cycle set to the error shown in the sixth cycle set in Figure 4-32 this may indicate over fitting of this specific profile feature. In all testing cases the network generates the greatest error at maximum stress levels. This highlights the importance of capturing data beyond the bounds of stress range anticipated in analysis in order to stay within the bounds of the training data set.



**Figure 4-33:** Network output tested in recurrent mode using previously unseen data generated from profile 1.



**Figure 4-34:** Network output tested in recurrent mode using previously unseen data generated from profile 2.

A key factor that has not been investigated here is the effect of differing load steps contained within training data and data presented to the network during analysis. As the internal variable given in (4-37) contains the strain step it is anticipated that the network will be sensitive to the size of the strain step taken. Therefore, the load and strain steps required in the analysis should be taken into account when gathering and processing the training data for the network model. It is also vital that the network undergoes a vigorous validation procedure. This will be further investigated through demonstrations of the network material models implemented in finite element analysis, Section 5.3.1.

#### **4.4.4 Cyclic biaxial training data collection and pre-processing**

Biaxial data is gathered using the same procedure described in Section 4.2 except the loading profile is designed to include additional features based on those investigated in the uniaxial profile. In contrast to the response surface style network the residual strain is not removed prior to training. This leads to the need to generate multiple stress strain profiles to train a single network each with a different initial load ratio and ratio order.

Each profile is made up of 6 sets of loading and unloading cycles each containing 5 load ratios. The resulting profiles are named according to the first load ratio applied to the fabric in each set of loading and unloading cycles. The sets are distinguished by the maximum and minimum loads applied in each material direction. In this way 5 unique profiles are generated which aim to explore as much of the fabric response as possible (Figure 4-35). The ultimate tensile strength shown in Table 4.17 are used in the generation of the load profiles. For demonstration purposes the 1:1 profiles with strain results for PVC (Figure 4-36) and PTFE (Figure 4-37) are shown below. The full sets of profiles for each material have been provided in Appendix B.

|                      | <b>Ultimate Tensile Strength (kN/m)</b> |             | <b>Maximum Stress (kN/m)</b> |             | <b>Minimum Stress (kN/m)</b> |             |
|----------------------|---|-------------|------------------------------|-------------|------------------------------|-------------|
| <b>Material</b>      | <b>Warp</b>                             | <b>Fill</b> | <b>Warp</b>                  | <b>Fill</b> | <b>Warp</b>                  | <b>Fill</b> |
| <b>PVC (1202T2)</b>  | 112                                     | 112         | 28                           | 28          | 2.8                          | 2.8         |
| <b>PTFE (B18059)</b> | 140                                     | 120         | 35                           | 30          | 3.5                          | 3           |

**Table 4.17: Maximum and minimum stress for biaxial profiles**

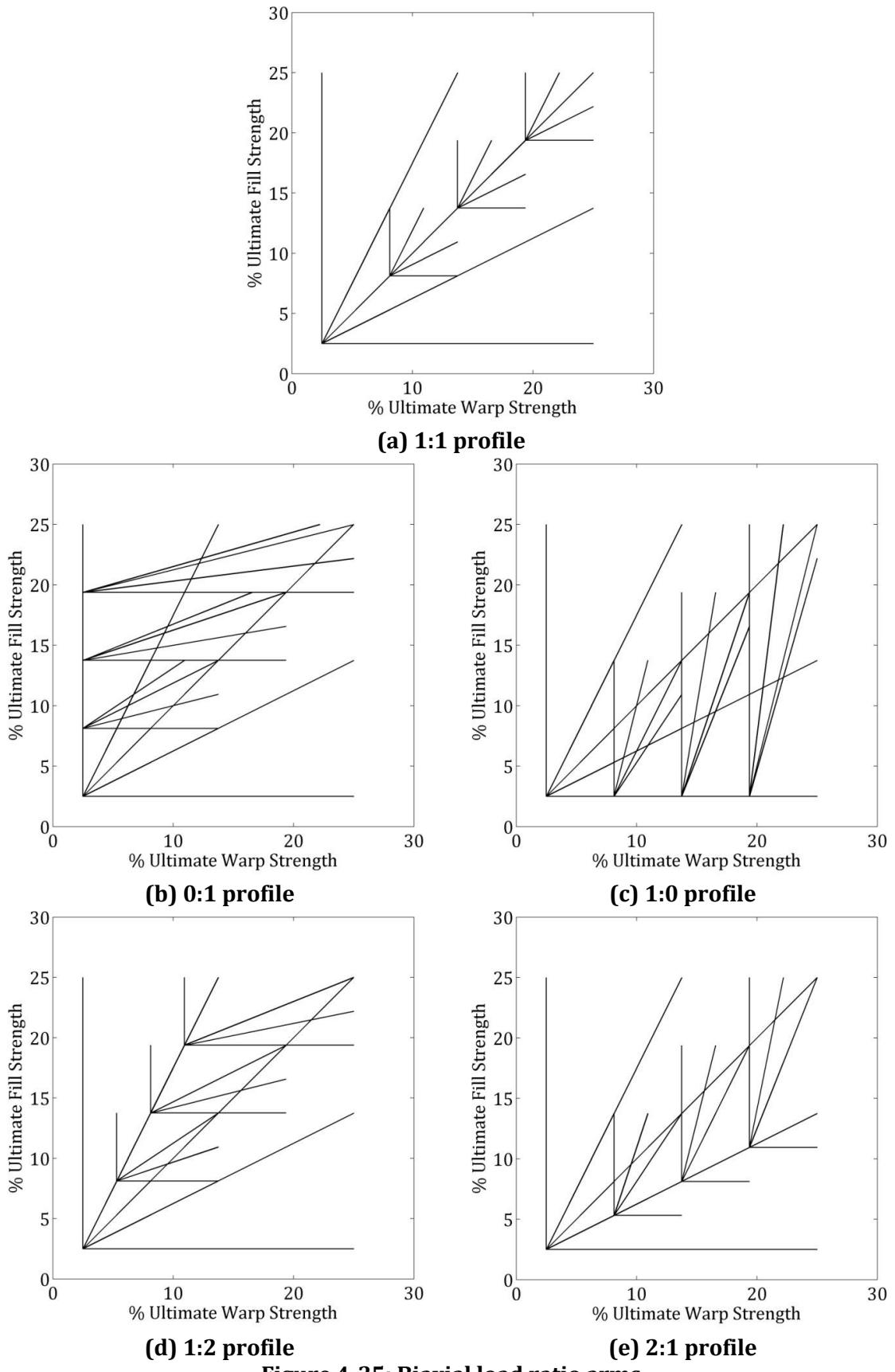


Figure 4-35: Biaxial load ratio arms

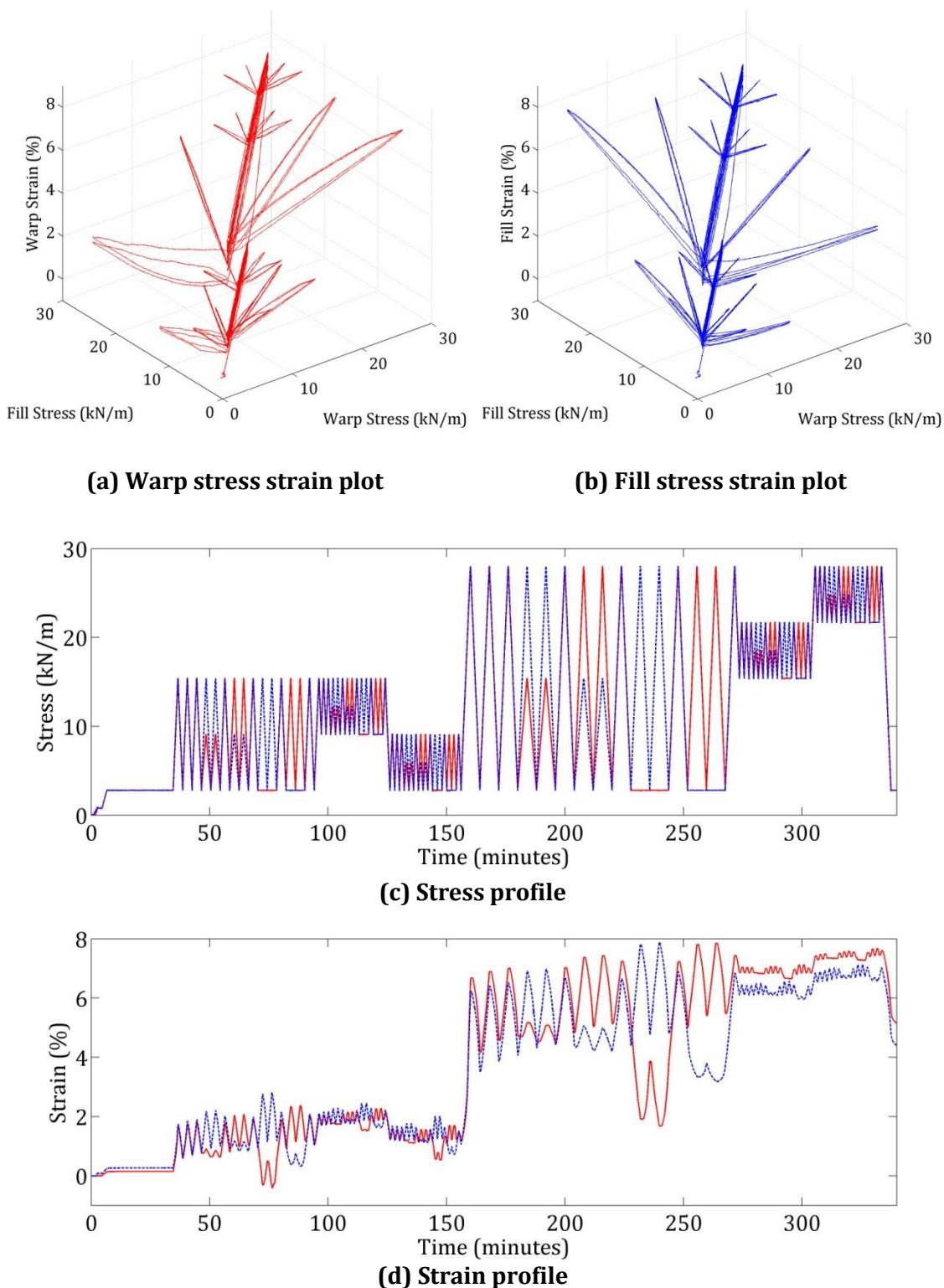
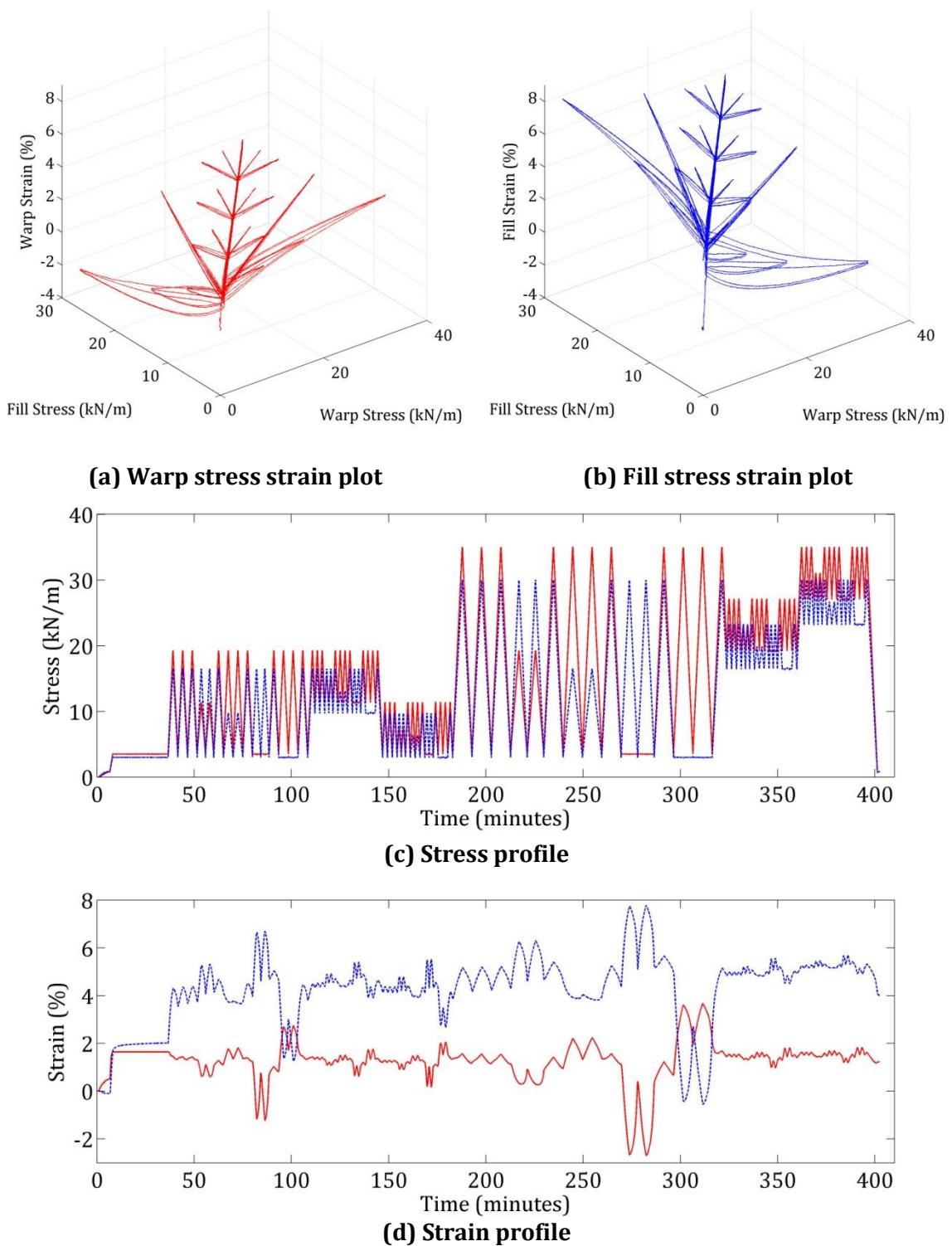


Figure 4-36: PVC Biaxial 1:1 stress profile and results for history network training and validation (— warp, .... fill)



The fabric stress strain responses of the two materials are very distinct. The PVC strain response is similar in both material directions, Figure 4-36 (d). This is due the 'precontrant' method of fabric production where tension is applied in both material directions during fabrication. The approximately equal tension applied to the fill direction during coating results in a similar level of crimp in both material directions. This reduces the effect of crimp interchange upon initial loading. In contrast the PTFE coated glass fabric is not tensioned in the fill direction during production. This leads to a higher degree of crimp in the fill direction compared to the warp. Therefore, the PTFE fabric shows a much greater initial strain in the fill direction compared to the warp direction. As the fill direction yarns straighten, thus inducing positive strain in the fill direction, the warp direction yarns become more crimped, inducing negative strain in the warp direction. This initial fill direction strain is never recovered despite the lower minimum and maximum applied stresses, Figure 4-37(d).

Despite the reduction of initial crimp interchange the PVC coated polyester fabric still exhibits a significantly different response to initial loading compared to the conditioned response. The initial behaviour of the PVC fabric is largely dependent on the elastic material properties of the polyester yarns as oppose to the mechanical interaction of the yarns. Initial strain behaviour is also observed each time the profile reaches a new maximum stress for example of the third set of cycles leading to a jump in residual strain. The PVC coated polyester fabric also exhibits an accumulation of residual strain in both directions. After the initial loading cycle the PTFE coated glass fabric becomes similarly stiff in both material directions and does not exhibit further initial behaviour into the profile. The PTFE coated glass fabric also exhibits limited build-up of residual strain in either direction. The large "Poisson's effect" exhibited by both fabrics throughout the applied profile may largely be attributed to the mechanical interaction of the yarns which in turn is effected by the preceding stress ratio. This behaviour further highlights the need for multiple profiles for testing.

In order to generate comprehensive training and 'unseen' testing data sets which contain as many features as possible, half of the cycles of each profile are taken as training data (Figure 4-38a and Figure 4-39a) and half as testing data (Figure 4-38b and Figure 4-39b). The data points within the selected cycles are reduced by a factor of 0.1 and then combined to form the final training and testing sets, the data points are denoted by black points in the figures. The full sets of divided profiles for each material have been provided in Appendix B.

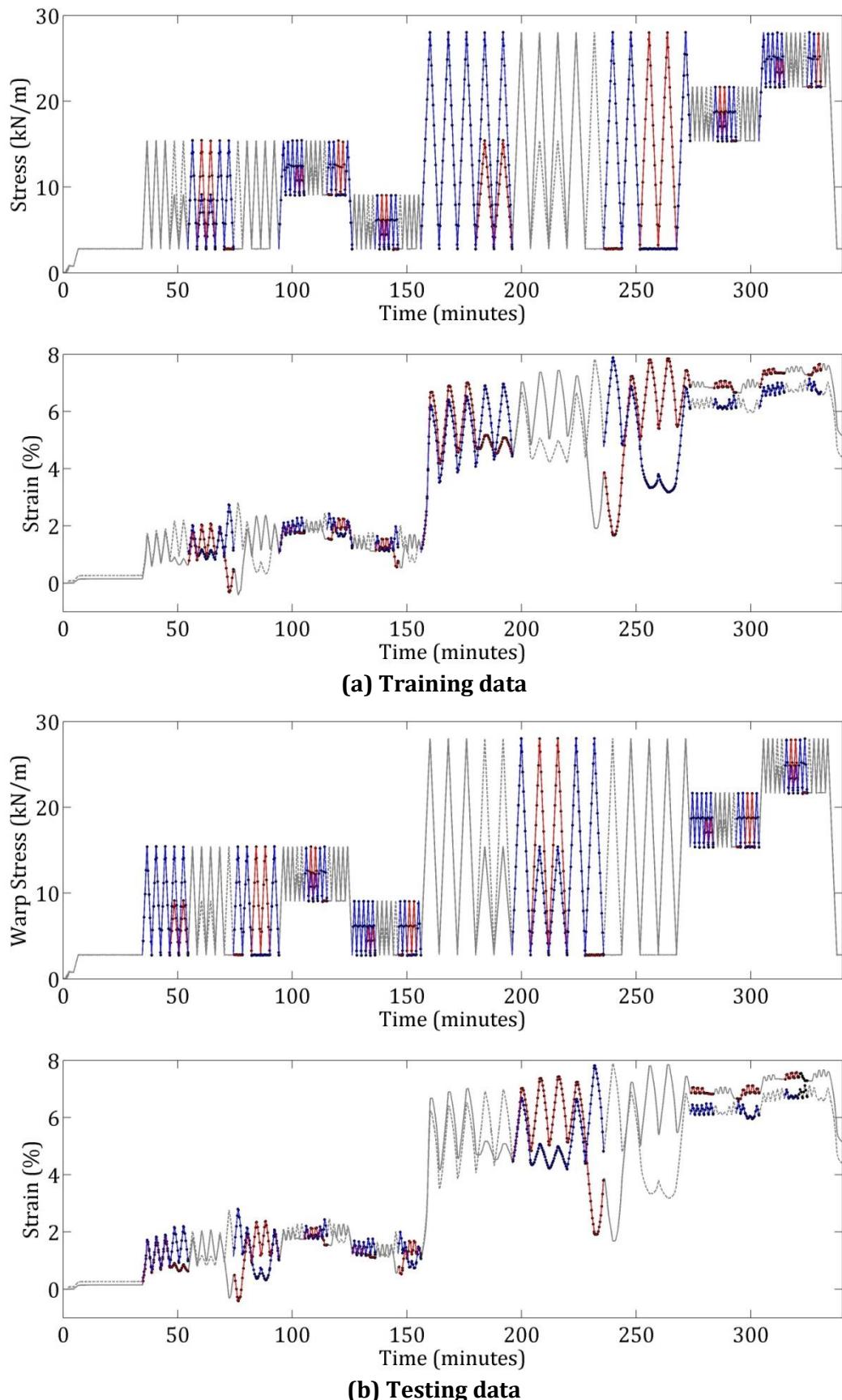
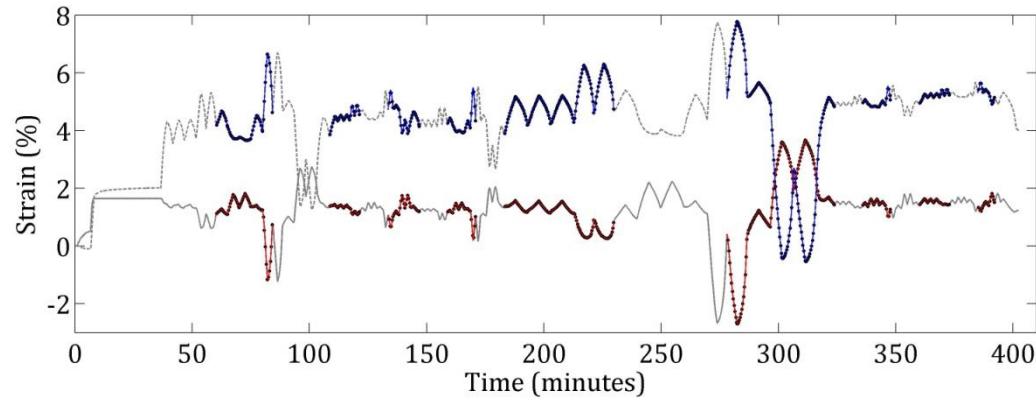
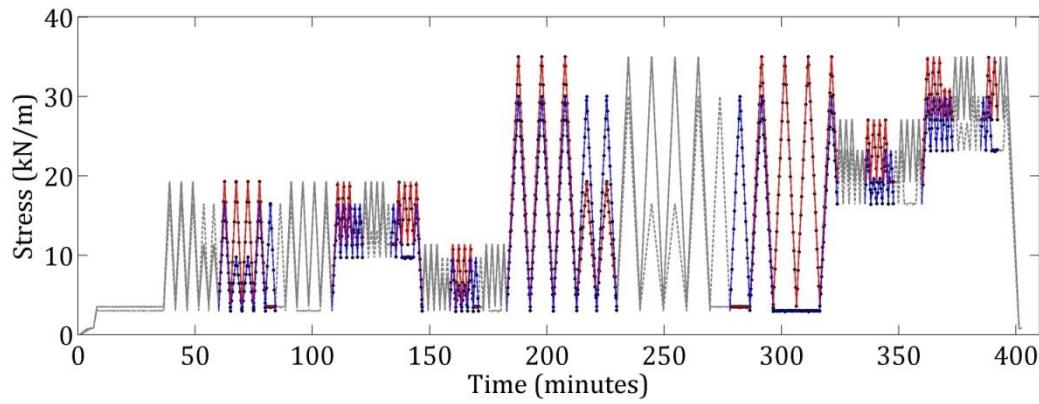
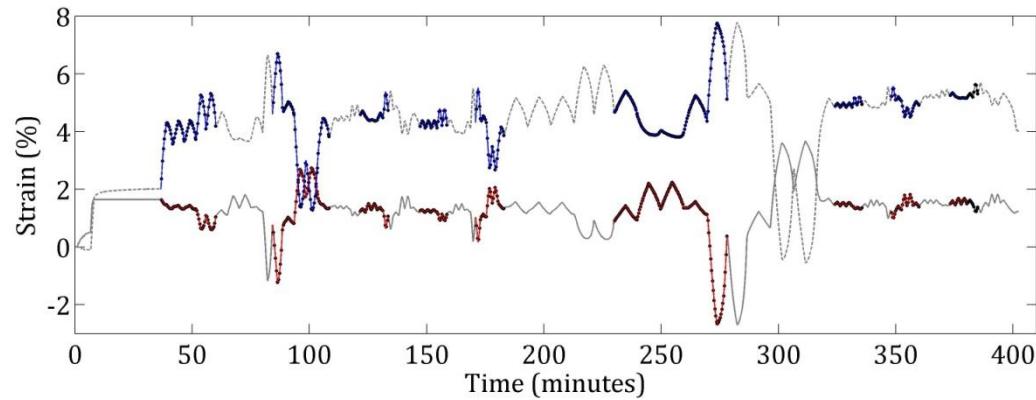
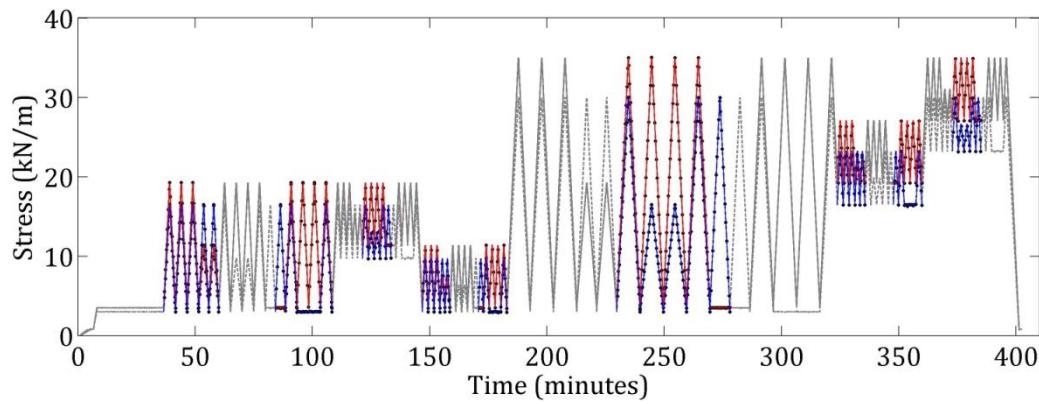


Figure 4-38: PVC Biaxial 1:1 stress and resulting strain profiles divided into training and testing data respectively (— warp, .... fill, • data point)



(a) Training data



(b) Testing data

**Figure 4-39: PTFE Biaxial 1:1 stress and resulting strain profiles divided into training and testing data respectively (— warp, .... fill, • data point )**

#### 4.4.5 Cyclic biaxial network training and validation

When modelling a biaxial response the internal variables (4-36)and (4-37)become,

$$\xi_{w,n} = \sigma_{w,n-1}\varepsilon_{w,n-1} \quad (4-40)$$

$$\xi_{f,n} = \sigma_{f,n-1}\varepsilon_{f,n-1} \quad (4-41)$$

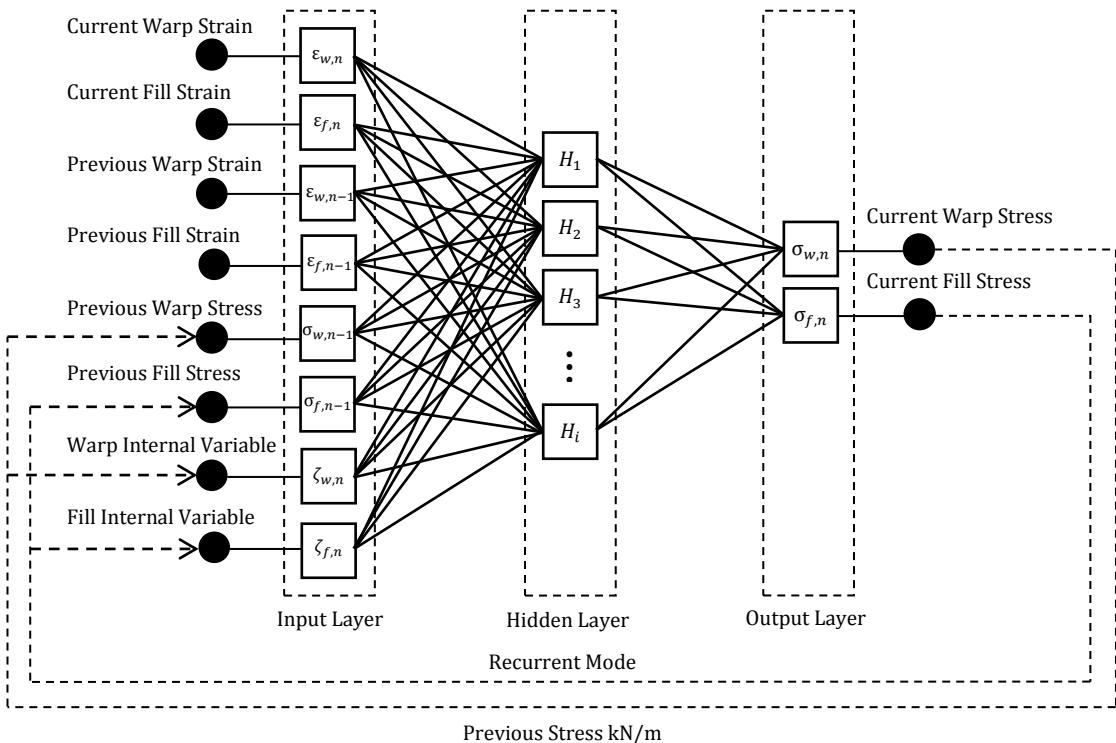
$$\Delta\eta_{w,n} = \sigma_{w,n-1}(\varepsilon_{w,n} - \varepsilon_{w,n-1}) = \sigma_{w,n-1}\Delta\varepsilon_{w,n} \quad (4-42)$$

$$\Delta\eta_{f,n} = \sigma_{f,n-1}(\varepsilon_{f,n} - \varepsilon_{f,n-1}) = \sigma_{f,n-1}\Delta\varepsilon_{f,n} \quad (4-43)$$

where the additional subscripts  $w$  and  $f$  denote warp and fill directions respectively. As in the case of the uniaxial internal variables the two variables may be combined into a single variable for each material direction.

$$\zeta_{w,n} = \xi_{w,n} + \Delta\eta_{w,n} \quad (4-44)$$

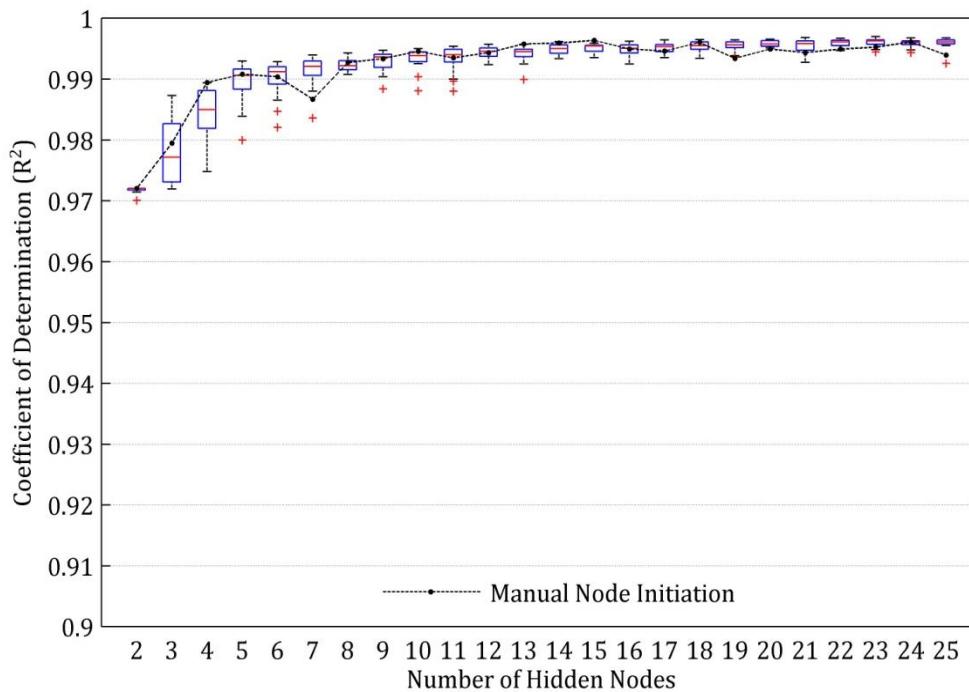
$$\zeta_{f,n} = \xi_{f,n} + \Delta\eta_{f,n} \quad (4-45)$$



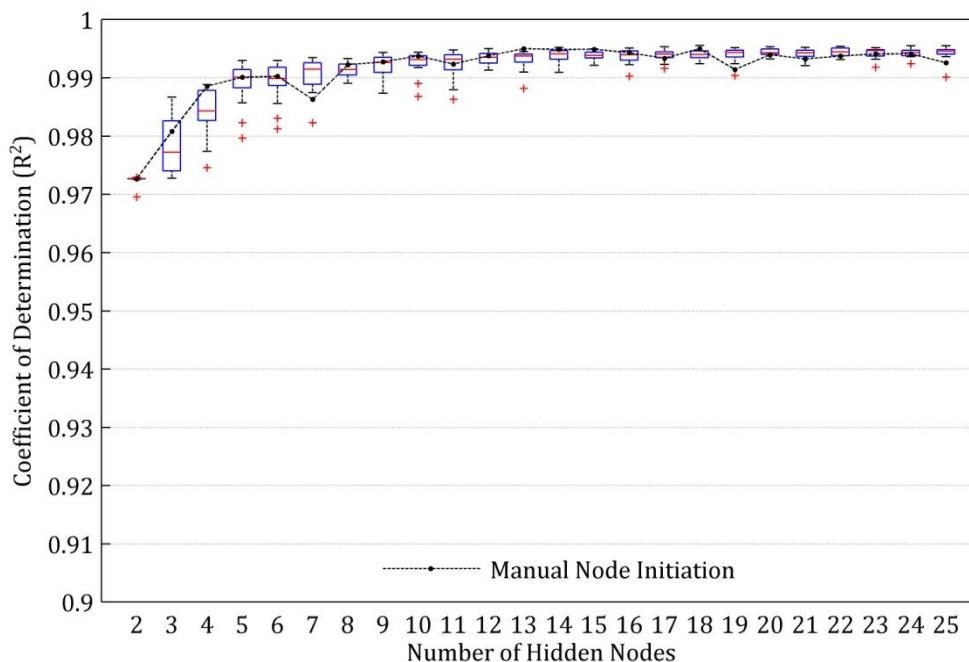
**Figure 4-40: 8 input strain to stress biaxial hysteresis neural network architecture**

As with the uniaxial history network the effect of the number of nodes in the hidden layer is investigated by training multiple networks with 1 to 40 hidden nodes. The training data sets described in Section 4.4.2 are used for each material. 20 randomly initiated networks are trained for each number of hidden nodes along with an additional network manually initiated with weights and biases set to 1. Each trained network is tested in recurrent mode using both the training data set and the 'unseen' testing data set. The  $R^2$  values generated from recurrent testing using the training (Figure 4-42a and Figure 4-43a) and

testing data sets (Figure 4-42b and Figure 4-43b) are plotted in the form of box-plots for both materials. The performance of the manually initiated network is plotted alongside the boxplots. As the network shows similar performance in both material directions only the warp direction performance is shown. For comparison the non- recurrent mode coefficients of determination for the PVC network model are plotted in Figure 4-41.

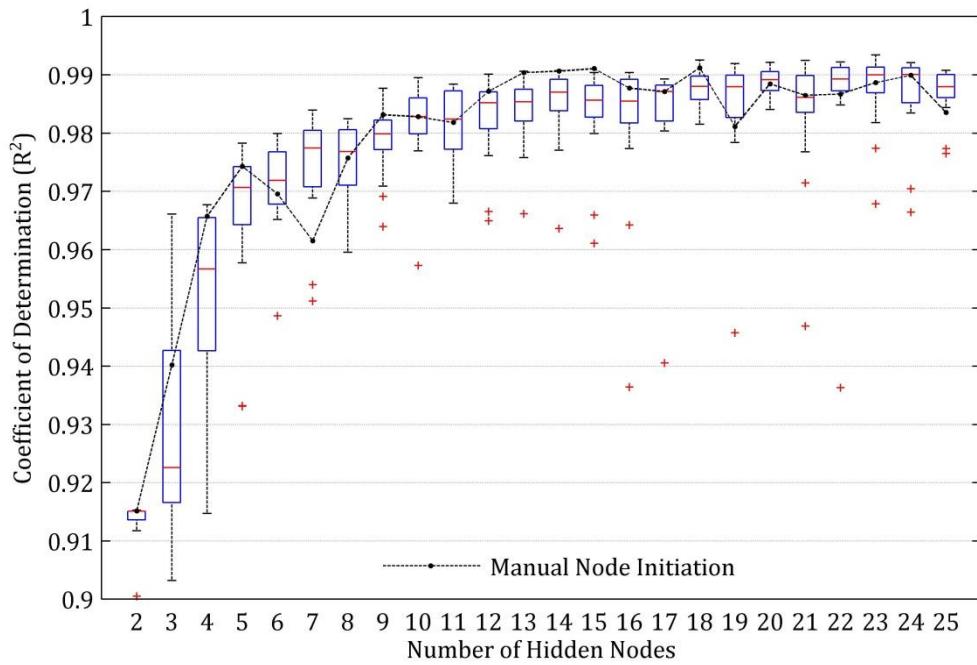


(a) Testing using training data set in non-recurrent mode

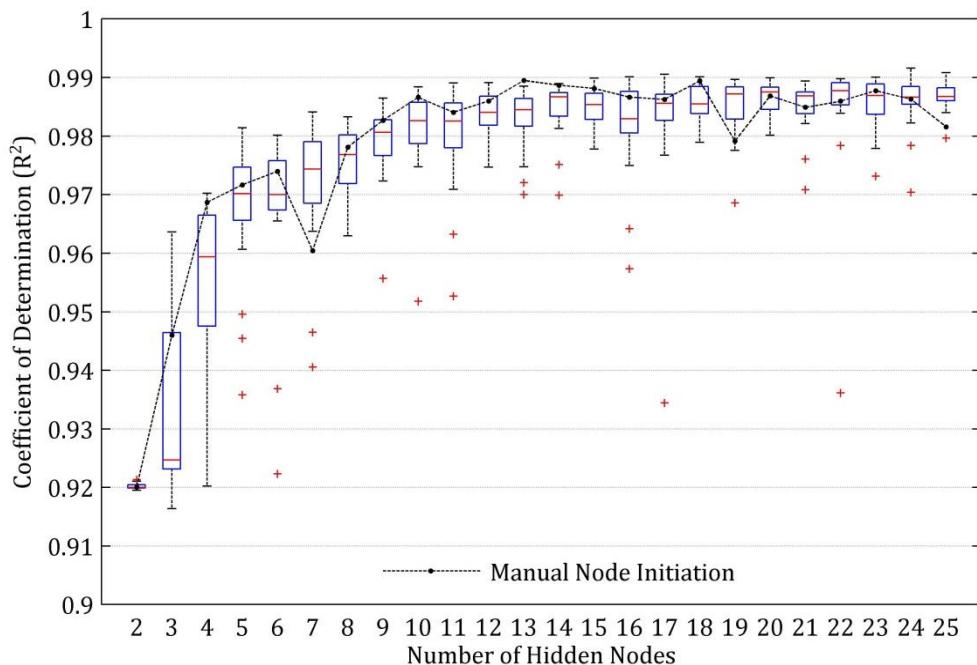


(b) Testing using 'unseen' testing data set in non-recurrent mode

Figure 4-41: Box and whisker diagrams of PVC network model testing performance ( $R^2$  values) for warp material direction



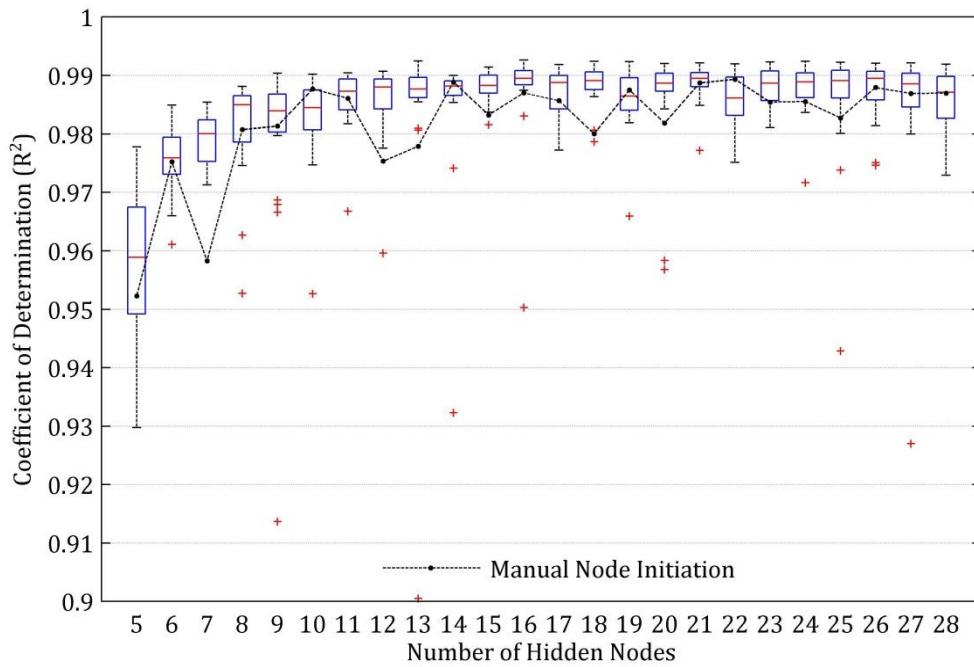
(a) Testing using training data set in recurrent mode



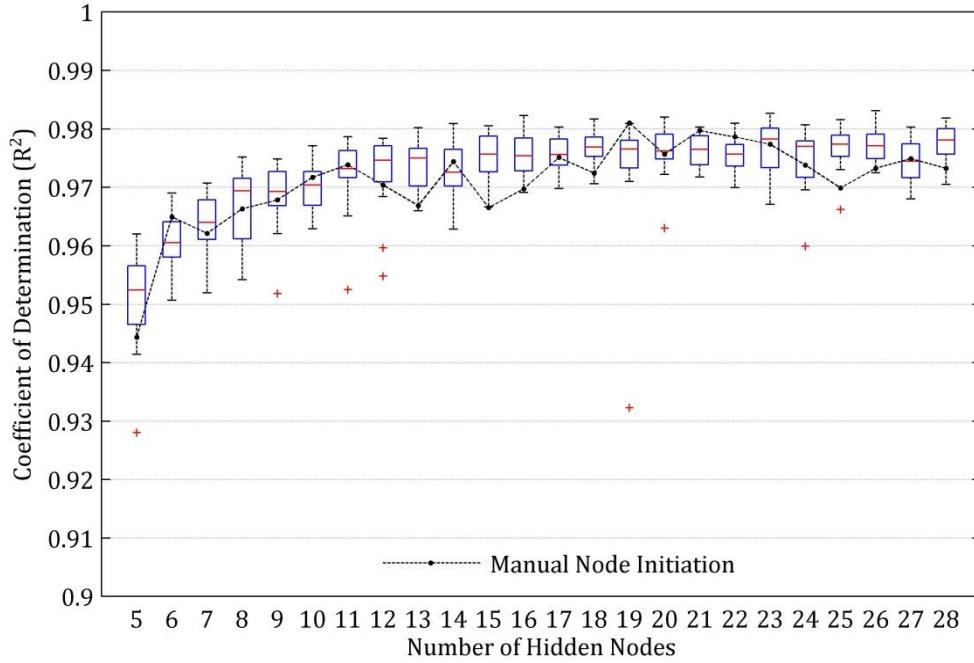
(b) Testing using 'unseen' testing data set in recurrent mode

**Figure 4-42: Box and whisker diagrams of PVC network model testing performance ( $R^2$  values) for warp material direction.**

As can be seen with sufficient nodes, i.e. greater than 10, the PVC networks achieve coefficients of determination in excess of 0.99 for both the training data set and the 'unseen' testing data set. It is also of note that the recurrent performance, Figure 4-42, follows the same trend as the non-recurrent performance, Figure 4-41.



(a) Testing using training data set in recurrent mode



(b) Testing using 'unseen' testing data set in recurrent mode

**Figure 4-43: Box and whisker diagrams of PTFE network model testing performance ( $R^2$  values) for warp material direction.**

Both the PVC coated polyester and PTFE coated glass fabric networks with more than 10 hidden nodes show good overall performance,  $R^2$  values greater than 0.95, for all trained networks contained within the box plots. Outliers denoted by red points are generated for most hidden layer sizes and are omitted from the boxplots. For both fabrics, median

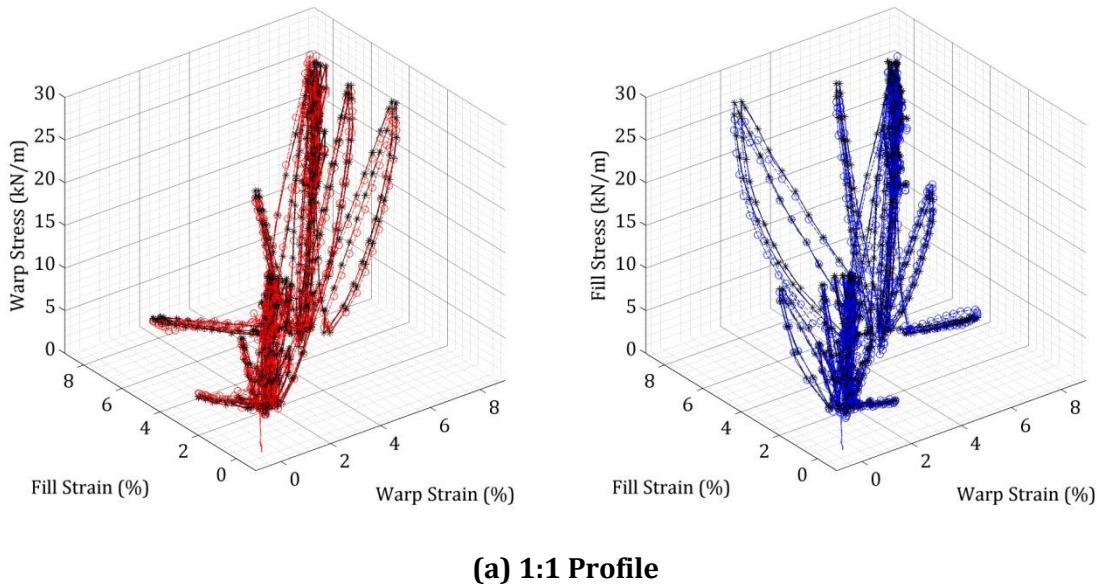
network performance, denoted by the red line, improves with increasing hidden node number up to approximately 15 hidden nodes. Increasing the number of hidden nodes from 15 up to 40 does not consistently increase or decrease median performance. The performance of networks with numbers of hidden nodes greater than 25 for PVC and 28 for PTFE are omitted in order to increase the clarity of the plots. The inter-quartile range denoted by the box or the total range denoted by the whiskers, excluding outliers, also does not demonstrate a consistent increase or decrease with increasing hidden node number. This indicates that the trained networks are resistant to over training. As in the case of the response surface style network, the PVC response is generalised more effectively than the PTFE response.

The same general pattern is followed by the manually initialised networks, Figure 4-41 to Figure 4-43. Variations in the performance of the network are attributed to the local minima in the error surface. In only a few cases the manually initiated network achieves a greater  $R^2$  value than the best randomly initiated network in a set. In the case of the PTFE network the manually initiated network rarely demonstrates a performance greater than the median of the randomly initiated networks. This indicates that the most effective way to generate networks with the high performance is to train multiple randomly initialised networks and select the one demonstrating the best performance when tested with 'unseen' testing data.

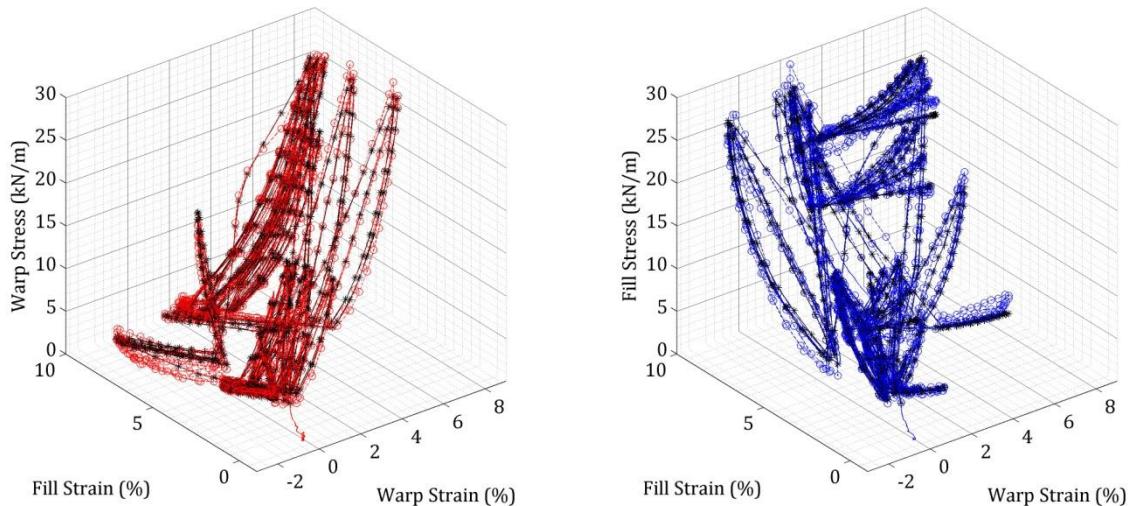
Based on this information the final biaxial history network will have 15 nodes in the hidden layer. A further 20 randomly initiated networks are trained and tested and the network with the greatest  $R^2$  value when tested with 'unseen' testing data are graphically presented for PVC (Figure 4-44 and Figure 4-45) and for PTFE (Figure 4-46, and Figure 4-47). Each full profile containing both the training and testing data are presented to the trained network in recurrent mode. The resulting network output is plotted in a 3 dimensional strain-strain-stress plot for each material direction and for each profile (Figure 4-44 and Figure 4-46). In order to show network performance more clearly stress is plotted against time in 20 cycle sections for the 1:1 profile (Figure 4-45 and Figure 4-47). Additional graphical results for all 5 profiles may be found in Appendix B.

In Figure 4-44 it may be visually confirmed that the stress strain response of PVC coated polyester fabric, including load history, has been captured across all profiles. Some error is visible for certain cycles. However this error is recovered in later cycles. In Figure 4-45a it may be seen that the first 20 stress cycles of the profile show generally good agreement between network output and targets. There is however some small underestimation in the

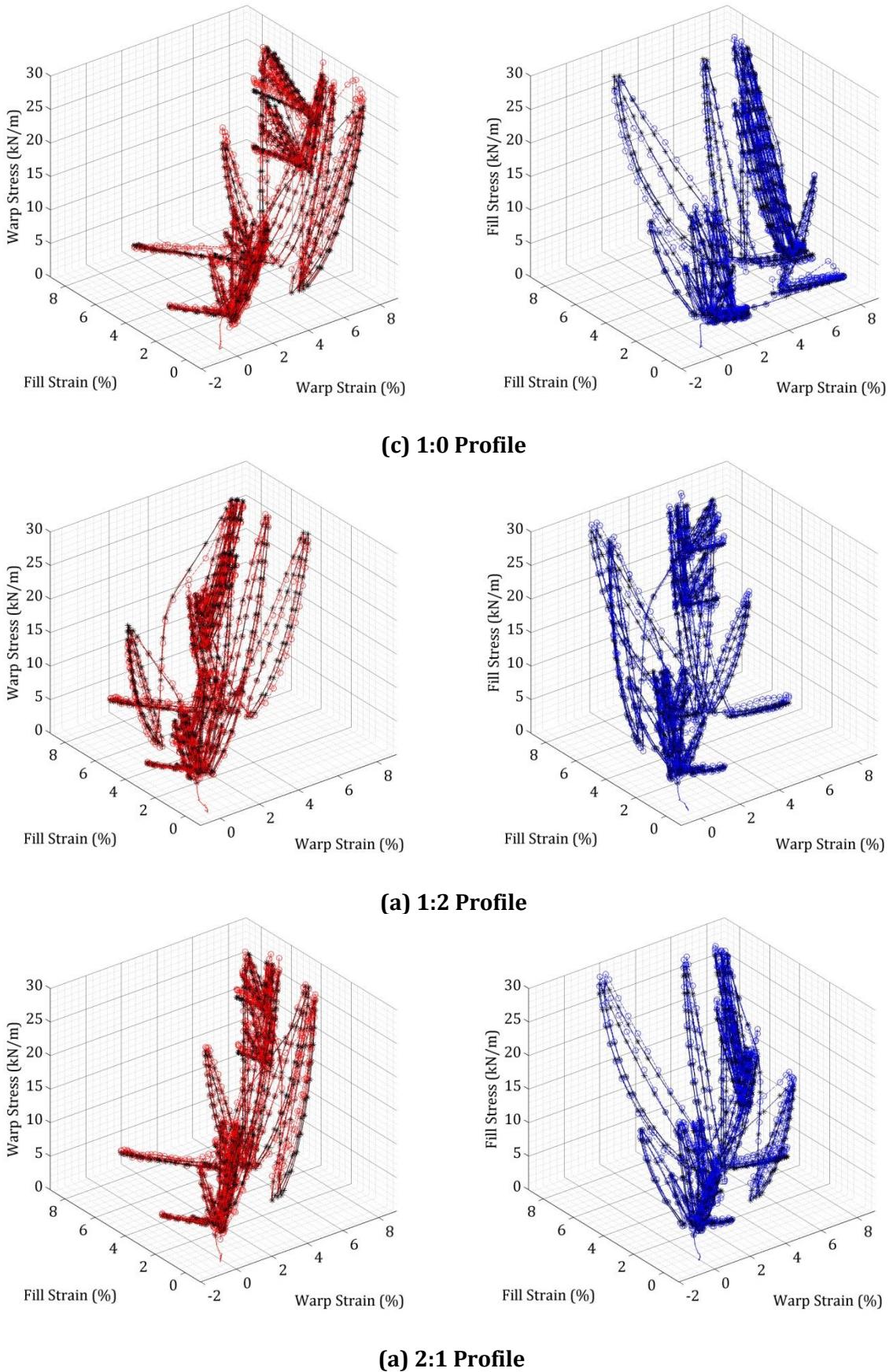
cycles after the first cycle in each set. In the following 60 cycles (Figure 4-45b to d) some significant underestimation occurs in the loading curves and over estimation in the unloading curves particularly in the warp direction. This error does not appear to accumulate (Figure 4-45c). The greatest error occurs in the cycles where one material direction is held at constant stress while the other is varied. This leads to considerable fluctuations in stress in the constant stress material direction. However, the error in the non-constant material direction remains low. Similar features appear in all the profiles tested (Appendix A). There is no clear visible distinction between data used in training and unseen data used in testing. This demonstrates the neural network material models strong ability to generalise.



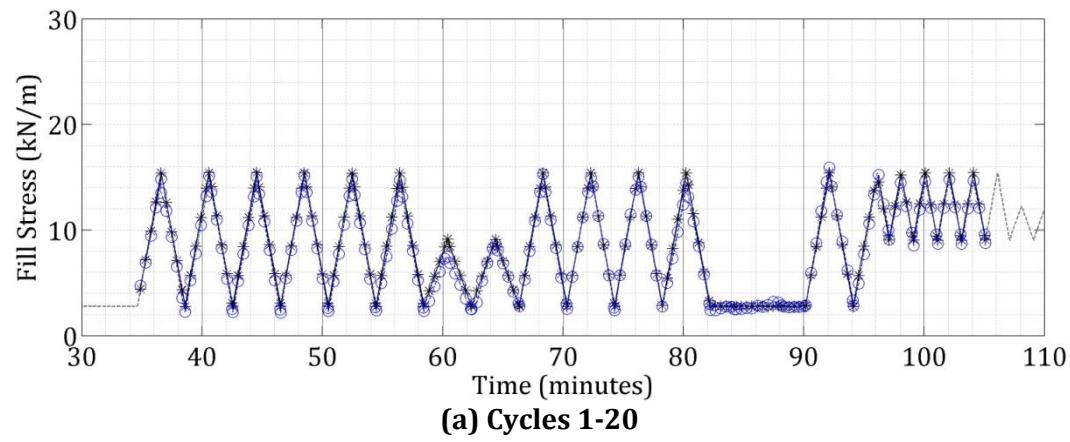
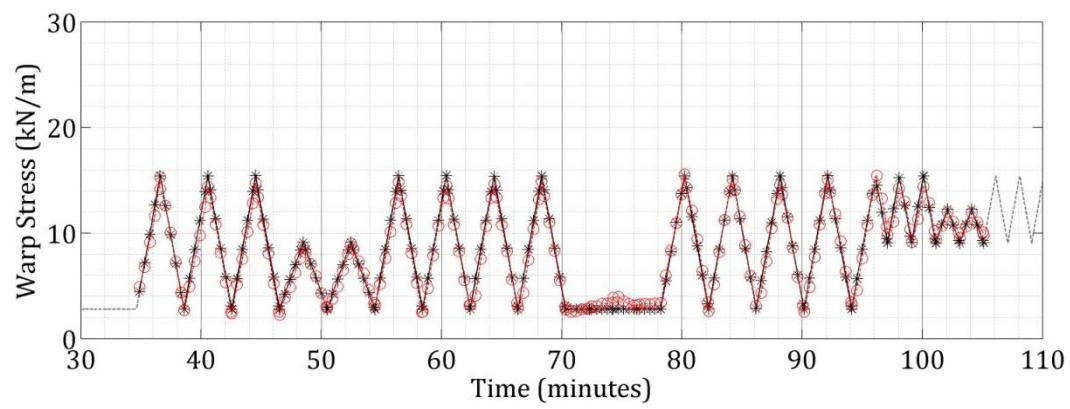
(a) 1:1 Profile



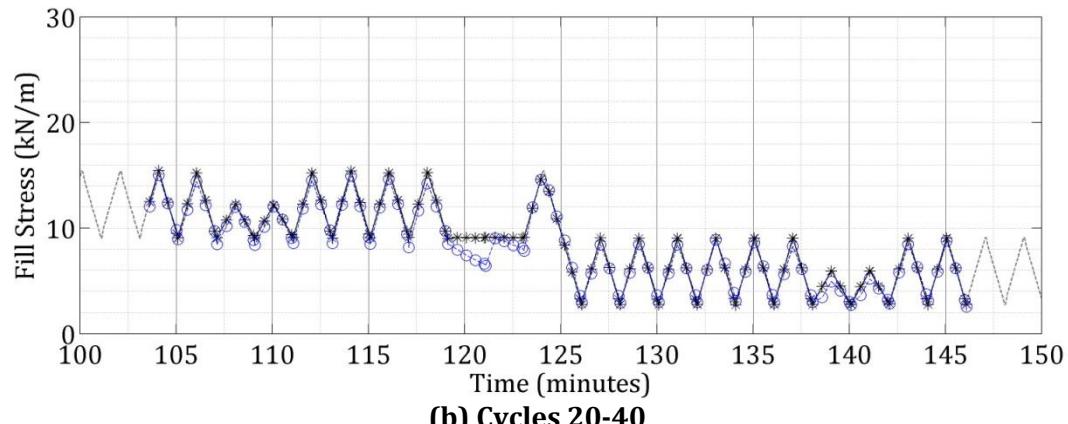
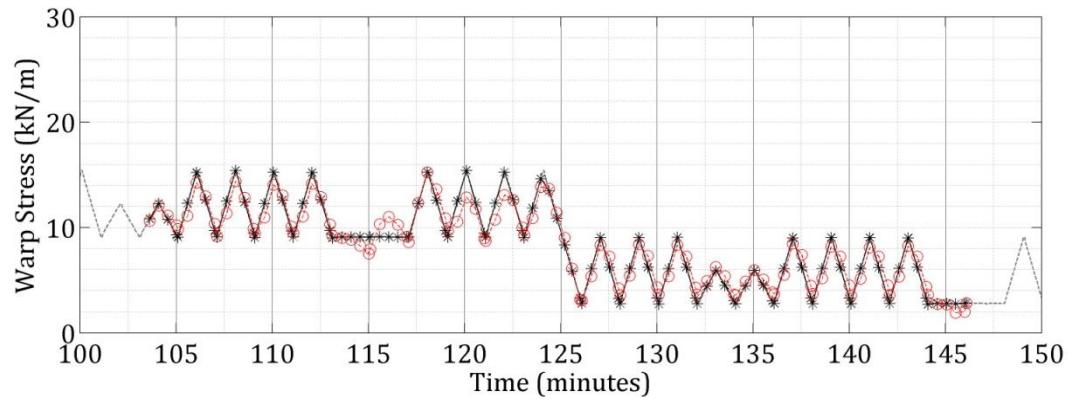
(b) 0:1 Profile



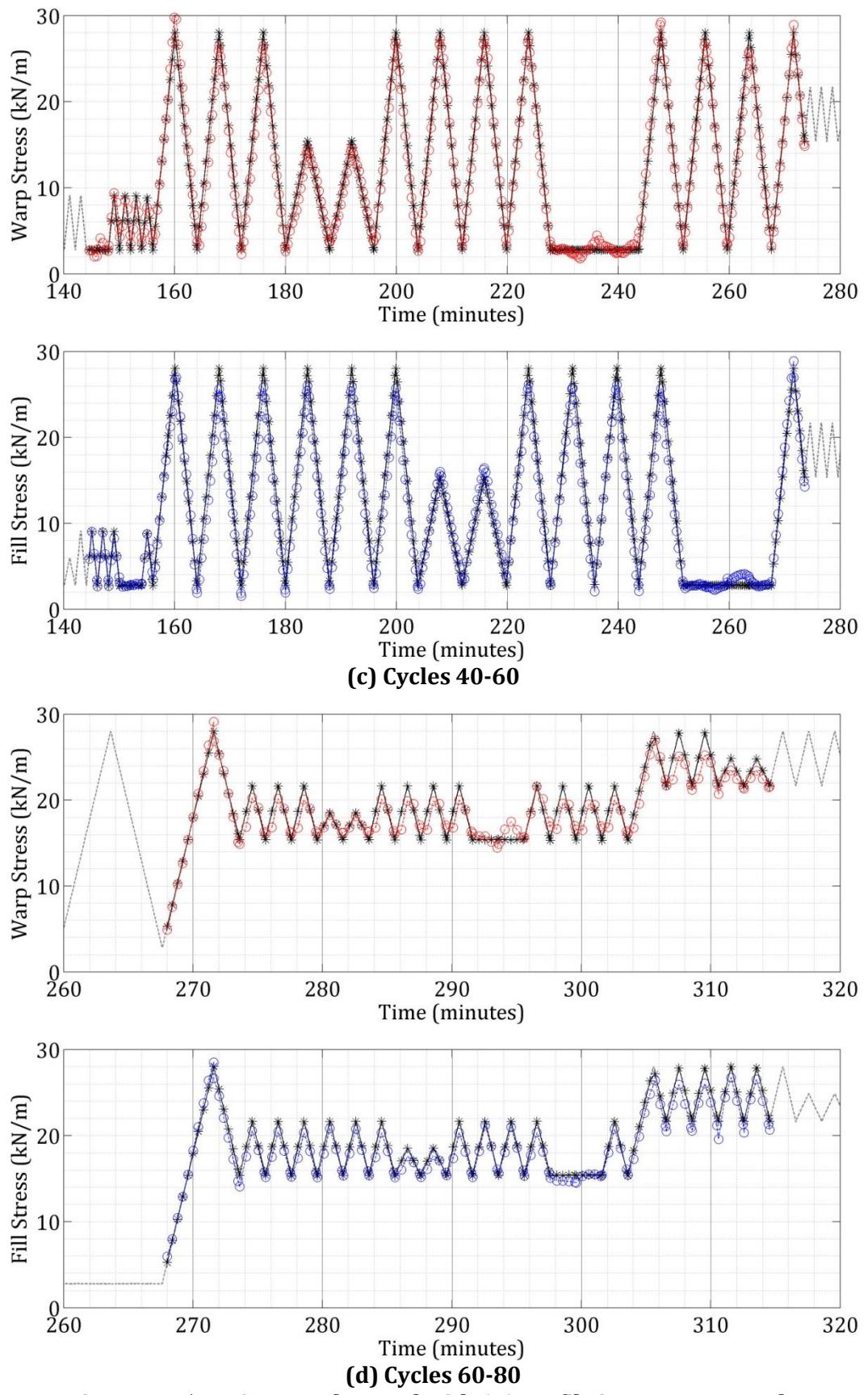
**Figure 4-44: Strain stress plots from PVC network tested with full profiles in recurrent mode (— warp, — fill, \* target, ° network output)**



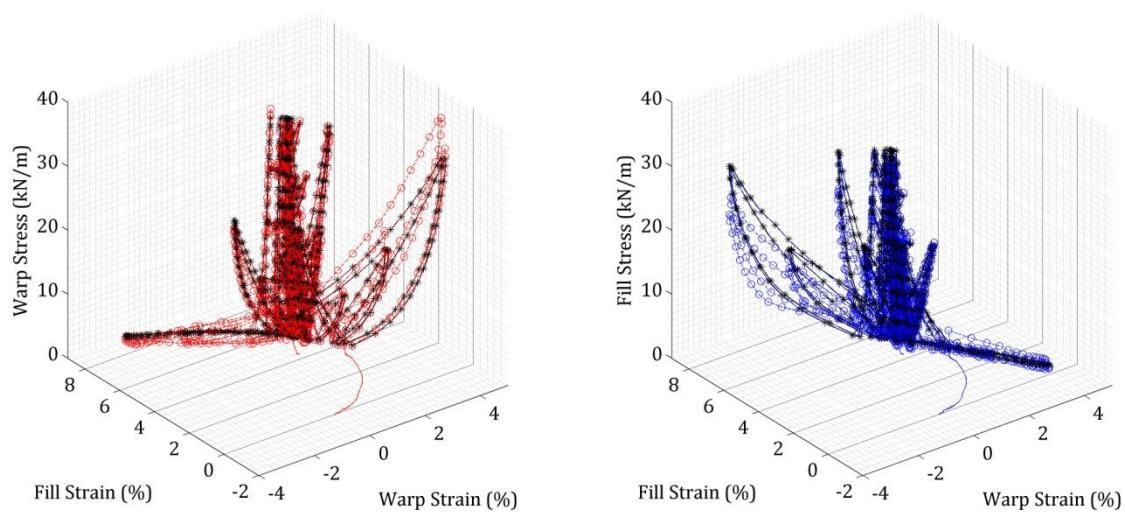
(a) Cycles 1-20



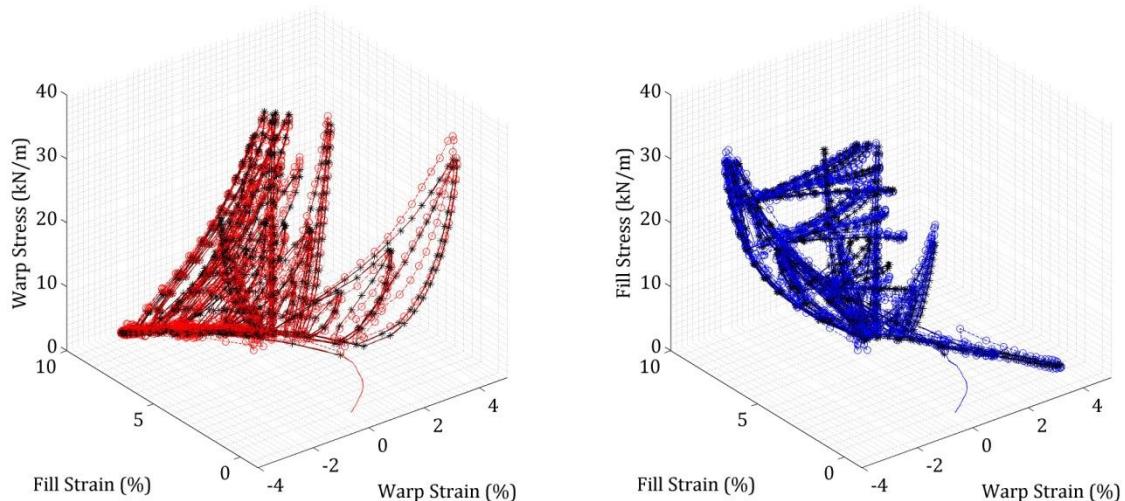
(b) Cycles 20-40



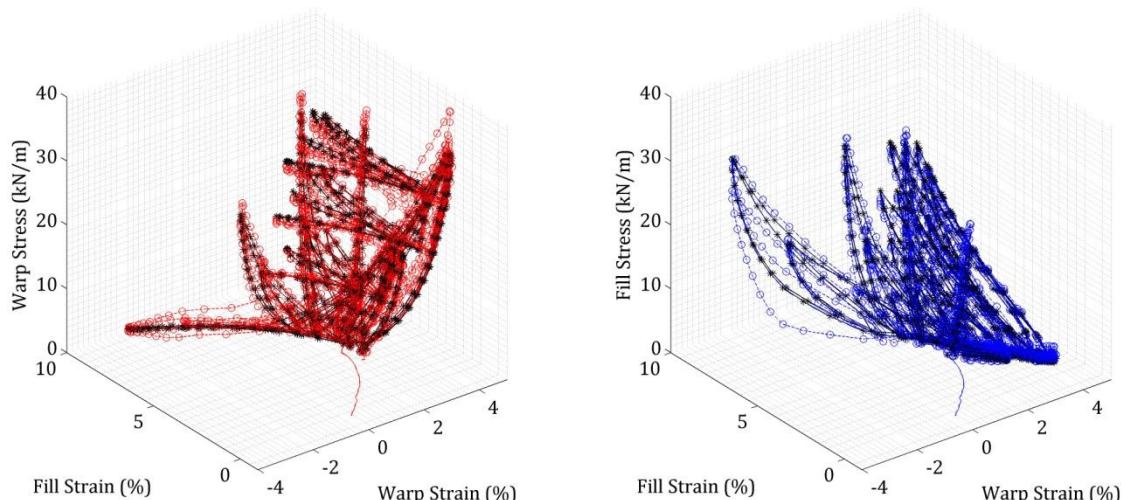
**Figure 4-45: PVC network tested with 1:1 profile in recurrent mode**  
**(— warp, — fill, \* target, ° network output)**



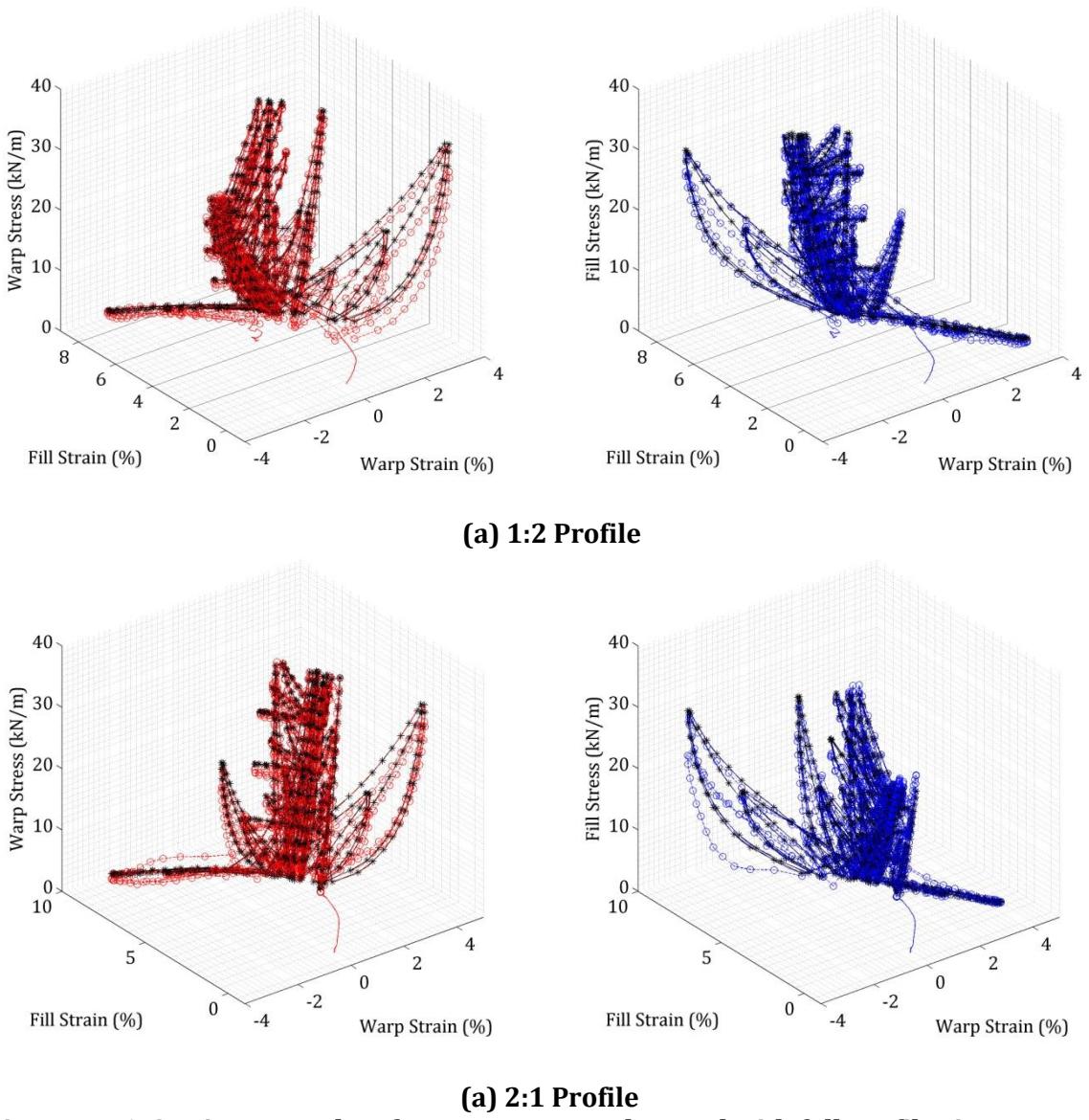
**(a) 1:1 Profile**



**(b) 0:1 Profile**



**(c) 1:0 Profile**

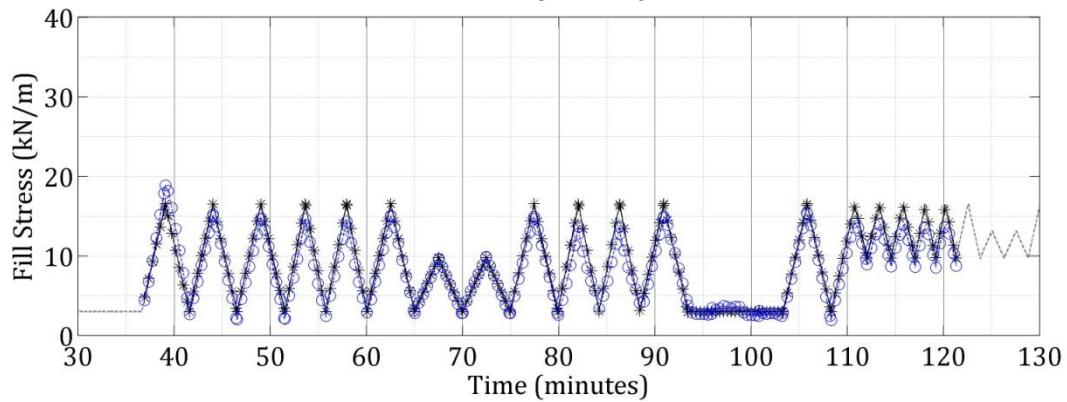
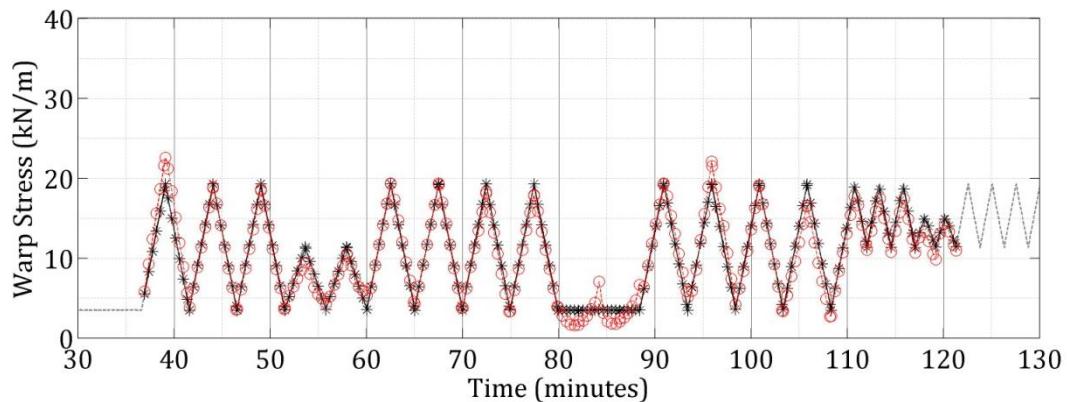


**Figure 4-46: Strain stress plots from PTFE network tested with full profiles in recurrent mode (— warp, — fill, \* target, ◦ network output)**

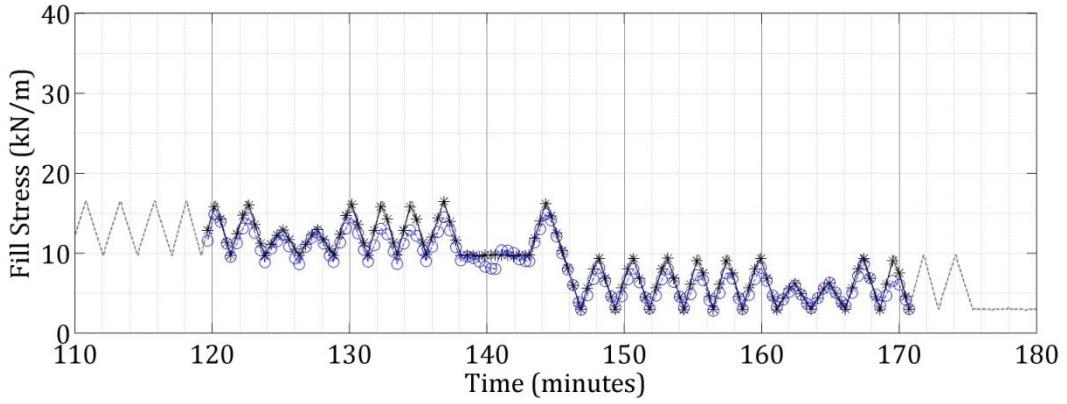
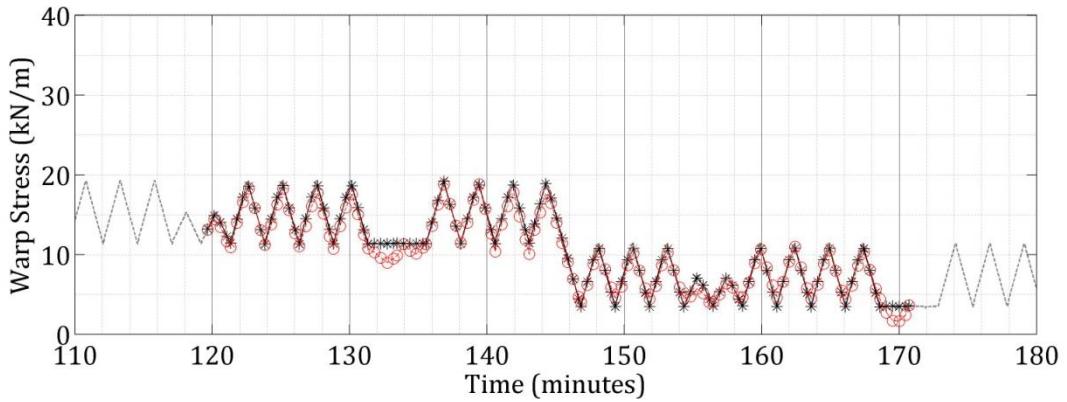
In Figure 4-46 it may be visually confirmed that as with the PVC coated polyester the stress strain response of PTFE coated glass fabric, including load history, has been captured across all profiles. Similarities between the PVC and PTFE network performance are observed. The greatest error occurs in the cycles where one material direction is held at constant stress while the other is varied. This loading ratio leads to error in both material directions. There is also no clear visible distinction between data used in training and unseen data used in testing for the PTFE network.

Unique to the PTFE network there is some over estimation in the first cycle of the profile and also in the warp direction in the first cycle where the fill direction is held constant. This occurs across all similar cycles in the rest of the profiles (Appendix A). In the following 60 cycles (Figure 4-45b to d) some significant underestimation occurs in the

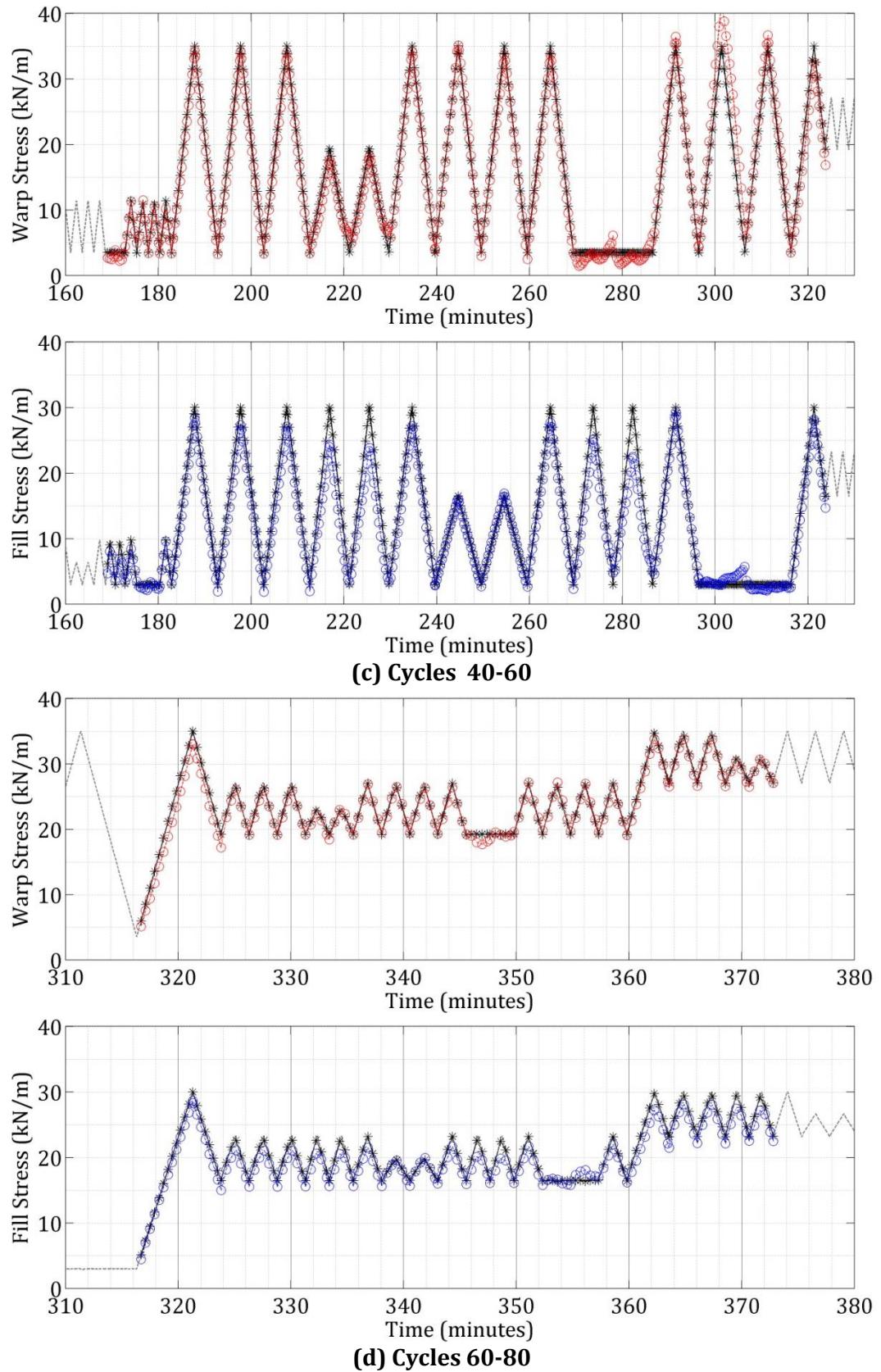
loading and unloading curves particularly in the fill direction. This error does not appear to accumulate and in fact recovers (Figure 4-45c).



(a) Cycles 1-20



(b) Cycles 20-40



**Figure 4-47: PTFE network tested with 1:1 profile in recurrent mode**  
**(— warp, — fill, \* target, ° network output)**

Despite the areas of error identified in the neural network models above, the models provide a good representation of the complex material response. Compared to the capability of a plane stress model to represent material response as a single flat surface this form of material model enables the representation of non-linear history dependant behaviour. This form of network also does not require the removal of residual strain from the training data. The output of the network approximates the stress resulting from total strain applied to the fabric.

## 4.5 Summary and Conclusions

It has been demonstrated by the networks described above that neural network material models are capable of capturing the complex non-linear stress-strain response of architectural fabrics. The response surface style network captures the strain stress relationship in a similar manner to the plane stress framework, but without the need for plane stress assumptions. Due to the removal of residual strain, the network is not trained using data directly captured from biaxial testing, and, therefore, the full response is not captured. This network also has limitations when employed to capture the much steeper response surface of PTFE coated glass fabric. However, this is overcome by the introduction of the approximate load ratio as an additional input.

The uniaxial network demonstrates the capability of neural networks to model the effects of load history. This initial study into uniaxial history neural networks offers a proof of concept which leads to a biaxial response network which includes the effects of load history. This form of network requires carefully designed experimental load profiles that explore the full response envelope to provide comprehensive testing and training data sets. The effects of load step have been investigated along with the number of hidden nodes for both the uniaxial and biaxial material networks.

The effect of hidden node number is investigated by training multiple networks with increasing numbers of hidden nodes. Box plots of the coefficients of determination, generated from comprehensive network testing, indicate that uniaxial history networks with 4 inputs require more than approximately 6 nodes but lose the ability to generalise with more than 17 nodes. In the case of a biaxial history network with 8 inputs a number of hidden nodes greater than 15 is found to provide good performance. However, there is no consistent fall in median performance with increasing hidden nodes up to 40 nodes. This brief study indicates that a good rule of thumb is that the number of hidden nodes required is approximately equal to twice the number of network inputs.

Graphical representation of the final trained history networks for PVC and PTFE fabrics demonstrate that it is possible to use appropriate neural network models in order to capture and reproduce fabric response. The next step is the implementation of these neural network material models for architectural fabrics within a custom fabric analysis finite element programme.



**Chapter 5.**

**Neural Network Material Model Implementation**



## Contents

|   |     |
|---|-----|
| Chapter 5. Neural Network Material Model Implementation .....       | 211 |
| 5.1 Implied Stiffness Matrix .....                                  | 218 |
| 5.1.1 Response surface style neural network implied stiffness ..... | 219 |
| 5.1.2 History neural network implied stiffness.....                 | 225 |
| 5.2 Response Surface Style Network Implementation.....              | 231 |
| 5.2.1 Additional areas, wrinkling criteria.....                     | 232 |
| 5.2.2 Patch test.....   | 237 |
| 5.2.3 Hypar .....   | 242 |
| 5.3 History neural network implementation .....                     | 249 |
| 5.3.1 Biaxial test.....   | 251 |
| 5.3.2 Hypar simulation.....   | 260 |
| 5.4 Summary and Conclusions.....                                    | 273 |

## Figures

|  |     |
|--|-----|
| Figure 5-1: Strain to stress neural network architecture.....  | 219 |
| Figure 5-2: Response surface neural network figure key.....  | 222 |
| Figure 5-3: Plane stress representation of PTFE response.....  | 222 |
| Figure 5-4: Plane stress strain to stress network output plotted with target data sets ....                  | 223 |
| Figure 5-5: 8 input strain to stress biaxial hysteresis neural network architecture.....                     | 226 |
| Figure 5-6: Network output plotted alongside target data from Profile 0:1, cycle 50. ....                    | 228 |
| Figure 5-7: Implied stiffness matrix output plotted alongside target data from Profile 0:1,<br>cycle 50..... | 229 |
| Figure 5-8: Finite Element analysis procedure with response surface neural network<br>material model.....    | 232 |
| Figure 5-9: Training data with additional fictitious plane stress data zones .....                           | 233 |
| Figure 5-10: Response surface with additional wrinkling criteria data points<br>figure key .....             | 234 |
| Figure 5-11: Training data with additional wrinkling criteria data points.....                               | 235 |

|   |     |
|---|-----|
| Figure 5-12: Response surface neural network figure key.....  | 236 |
| Figure 5-13: PVC (502S) strain to stress network demonstrating best performance<br>with 'unseen' testing data set ( $R^2 = 0.9978$ ) .....                        | 236 |
| Figure 5-14: 32 element patch for validation.....   | 237 |
| Figure 5-15: PVC (502S) strain to stress network with simulation results.....   | 238 |
| Figure 5-16: PTFE strain to stress network with simulation results. ....  | 240 |
| Figure 5-17: Hypar structure. ....  | 242 |
| Figure 5-17: Hypar prestress results .....  | 243 |
| Figure 5-18: Hypar snow loading PVC neural network results .....  | 244 |
| Figure 5-19: Hypar snow loading PVC plane stress results.....   | 245 |
| Figure 5-20: Hypar wind loading PVC neural network results .....  | 246 |
| Figure 5-21: Hypar wind loading PVC plane stress results .....  | 247 |
| Figure 5-22: Iterative finite element analysis procedure with history neural network<br>material model.....   | 251 |
| Figure 5-23: Biaxial test specimen, applied load 4kN/m.....   | 252 |
| Figure 5-24: Patch Simulation Results of Cycle 6 of 1:1 Profile Generated by a Load<br>History Network Trained Using 60 Cycles and a Reduction Factor of 0.1..... | 255 |
| Figure 5-25: Patch Simulation Results of Cycle 6 of 1:1 Profile Generated by a Load<br>History Network Trained Using 30 Cycles and a Reduction Factor of 0.1..... | 256 |
| Figure 5-26: Patch Simulation Results of Cycle 6 of 1:1 Profile Generated by a Load<br>History Network Trained Using 60 Cycles and a Reduction Factor of 0.15.... | 256 |
| Figure 5-27: Patch Simulation Results of Cycle 6 of 1:1 Profile Generated by a Load<br>History Network Trained Using 30 Cycles and a Reduction Factor of 0.15.... | 257 |
| Figure 5-28: Patch Simulation Results of Cycle 6 of 1:1 Profile Generated by a Load<br>History Network Trained Using 60 Cycles and a Reduction Factor of 0.2..... | 257 |
| Figure 5-29: Patch Simulation Results of Cycle 6 of 1:1 Profile Generated by a Load<br>History Network Trained Using 30 Cycles and a Reduction Factor of 0.2..... | 258 |
| Figure 5-31: Patch Simulation Results of Cycle 6 of 1:1 Profile Generated by a Plane<br>stress elastic stiffness matrix fit to Cycles 1 to 15 of 1:1 Profile..... | 259 |
| Figure 5-30: Form-found mesh with selected nodes for result plotting.....   | 261 |
| Figure 5-31: Hypar snow loading PVC history network maximum load<br>(Iteration 1: 0.0kN/m <sup>2</sup> ).....   | 263 |
| Figure 5-32: Hypar snow loading PVC history network maximum load<br>(Iteration 6: 0.6kN/m <sup>2</sup> ).....   | 264 |
| Figure 5-33: Hypar snow loading PVC history network maximum load<br>(Iteration 11: 0.0kN/m <sup>2</sup> ) .....   | 265 |

|  |     |
|--|-----|
| Figure 5-34: Hypar snow loading PVC history network nodal results.....   | 266 |
| Figure 5-35: Hypar wind loading PVC history network maximum load<br>(Iteration 1: 0.0kN/m <sup>2</sup> ).....  | 267 |
| Figure 5-36: Hypar wind loading PVC history network maximum load<br>(Iteration 6: 1.0kN/m <sup>2</sup> ).....  | 268 |
| Figure 5-37: Hypar snow loading PVC history network maximum load<br>(Iteration 11: 0.0kN/m <sup>2</sup> )..... | 269 |
| Figure 5-38: Hypar wind loading PVC history nodal results.....   | 270 |

## Tables

|  |     |
|--|-----|
| Table 5-1: PTFE elastic constants.....   | 222 |
| Table 5-2: PTFE plane-stress network variables .....   | 223 |
| Table 5-3: Plane stress network implied stiffness validation.....  | 224 |
| Table 5-4: PVC history network implied stiffness validation using finite difference<br>method ( $h = 0.0000001$ ) .....                                | 230 |
| Table 5-5: PVC plane stress elastic constants.....   | 233 |
| Table 5-6: PVC neural network patch simulation results.....  | 238 |
| Table 5-7: PTFE neural network patch simulation results .....  | 240 |
| Table 5-8: Hypar analysis results summary.....   | 248 |
| Table 5-9: Mean absolute percentage error between experimental results and patch<br>simulation of cycle 6 of 1:1 biaxial load profile (Chapter 4)..... | 254 |
| Table 5-10: PVC plane stress elastic constants.....  | 259 |
| Table 5-10: Hypar analysis results summary.....  | 271 |



## Chapter 5. Neural Network Material Model Implementation

In this chapter the history independent response surface and history dependant neural network material models derived in Chapter 4 are implemented in place of the plane stress framework within the finite element formulation described and demonstrated in Chapter 3. Derivations of an implied stiffness matrix for both the response surface style network and history network models are described, followed by a detailed description of neural network implementation including consideration of wrinkling criteria for the response surface network and iterative loading for the history network. Implementation studies for both networks are presented using patch simulations of experimental data and simulations of a more realistic hypar structure.

In a typical finite element analysis, Section 3.1, current stress is derived from the product a plane stress elastic compliance matrix and current strain, equation (3.1).

$$\{\sigma\} = [E]\{\varepsilon\} = \begin{bmatrix} \frac{E_w}{(1-v_{wf}v_{fw})} & \frac{E_w v_{fw}}{(1-v_{wf}v_{fw})} & 0 \\ \frac{E_f v_{wf}}{(1-v_{wf}v_{fw})} & \frac{E_f}{(1-v_{wf}v_{fw})} & 0 \\ 0 & 0 & G_{wf} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}. \quad (3.1)$$

This is the matrix equivalent of performing the following three calculations,

$$\sigma_x = \frac{E_w}{(1-v_{wf}v_{fw})} \varepsilon_x + \frac{E_w v_{fw}}{(1-v_{wf}v_{fw})} \varepsilon_y \quad (5.1)$$

$$\sigma_f = \frac{E_f v_{wf}}{(1-v_{wf}v_{fw})} \varepsilon_x + \frac{E_f}{(1-v_{wf}v_{fw})} \varepsilon_y \quad (5.2)$$

$$\tau_{xy} = G_{wf} \gamma_{xy} \quad (5.3)$$

It is clear that there is no interaction between the in plane strain and the shear strain, equations (5.2) to (5.4). Due to the complex interactions exhibited by a typical fabric material response this may not in fact be the case, Section 2.2. As the neural network material model is implemented within the finite element formulation in place of the plane stress elastic compliance matrix, equation(5.1), the network would ideally capture both direct strain stress behaviour and shear behaviour. However, the complexities involved in gathering sufficient training data with combined shear and biaxial stress response shear stress and strain is not included within the network model at this stage of development . It is deemed outside the scope of this thesis. Therefore, calculation of shear stress is performed separately via the shear modulus,  $G_{wf}$  (equation (5.4)).

The response surface style network, which takes 2 inputs comprising current strain in the warp and fill directions respectively works in a very similar manor to equations (5.2) and (5.3) and may be implemented in a plane stress based finite element formulation within its current form. The history network has an added level of complexity as it also takes previous stress and strain as inputs in order to find the current strain. This is accomplished through the use of history terms within the expression of equilibrium for the system. Therefore, implementation of this form of network model requires loading to be applied iteratively as opposed to a single load step.

The finite element formulation, Section 3.1, utilises a dynamic relaxation solver to iteratively solve the state equation for equilibrium. The current stress state at each Gauss point in an element may be derived directly from the network material models. However, the diagonal terms of the system's elastic stiffness matrix,  $K_E$ , equation (3.15), are required in order to provide damping to the dynamic relaxation algorithm, Section 3.1.2. In order to derive these terms the material model is required in the form of a 3 by 3 matrix. All 5 non-zero terms are required when the B matrix is not sparse.

$$[K_E] = \sum_{i=1}^{i=m} \left[ \int_V [B]^T [E][B] dV \right]_i \quad (3.15)$$

## 5.1 Implied Stiffness Matrix

Assuming  $G_{wf}$  is assumed to be a given constant, the numerical implementation of a neural network material model in finite element analysis is achieved by derivation of an ‘implied’ stiffness matrix,  $E^{implied}$ , similar to that proposed by Hashash et al. [119]. This involves the calculation of partial derivatives of the network output with respect to the input to give,

$$E = \begin{bmatrix} [E^{implied}] & 0 \\ 0 & G_{wf} \end{bmatrix}, \quad (5.4)$$

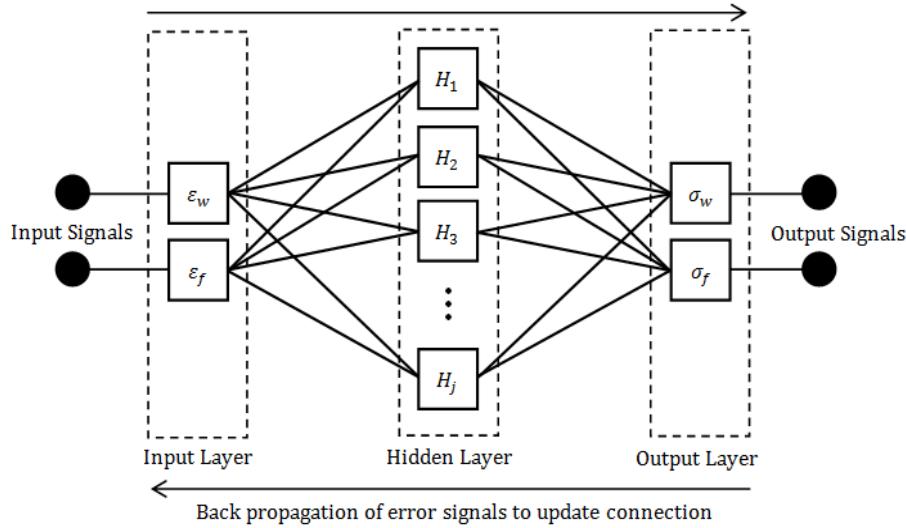
where,

$$E^{implied} = \begin{bmatrix} \frac{\partial \sigma_{k=1}}{\partial \varepsilon_{i=1}} & \frac{\partial \sigma_{k=1}}{\partial \varepsilon_{i=2}} \\ \frac{\partial \sigma_{k=2}}{\partial \varepsilon_{i=1}} & \frac{\partial \sigma_{k=2}}{\partial \varepsilon_{i=2}} \end{bmatrix}. \quad (5.5)$$

The network proposed in [119] has an input layer, two hidden layers and an output layer. Both hidden layers and the output layer have a tan-sigmoid (tanh) activation function. However, all of the networks used in this thesis consist of an input layer a single hidden layer with a tan-sigmoid transfer function and an output layer with a linear transfer

function, Figure 5-1. This form of network has been demonstrated as being sufficient for the representation of a number of engineering materials, Section 2.1.

### 5.1.1 Response surface style neural network implied stiffness



**Figure 5-1: Strain to stress neural network architecture**

Revisiting the network equations given in Section 4.1.1, equations (4.1) to (4.6), and rewriting them for the response surface style neural network shown in Figure 5-1 yields the following set of equations. The network inputs are the current warp and fill strains denoted by  $\varepsilon_i$  where,

$$\varepsilon_{i=1} = \varepsilon_w, \quad (5.6)$$

$$\varepsilon_{i=2} = \varepsilon_f. \quad (5.7)$$

The network outputs are the current warp and fill stresses denoted by  $\sigma_k$  where,

$$\sigma_{k=1} = \sigma_w, \quad (5.8)$$

$$\sigma_{k=2} = \sigma_f. \quad (5.9)$$

All network inputs are scaled to a range -1 to 1 via equation (4.1). The inputs defined in equation (5.7) and (5.6) are scaled according to the maximum,  $\varepsilon_i^{max}$ , and minimum,  $\varepsilon_i^{min}$ , strains of the training data,

$$\varepsilon_i^{NN} = \frac{2(\varepsilon_i - \varepsilon_i^{min})}{(\varepsilon_i^{max} - \varepsilon_i^{min})} - 1. \quad (5.10)$$

The output of the  $j^{th}$  hidden node with a tan sigmoid transfer function remains the same,

$$H_j = \tanh([\sum_{i=1}^{N\varepsilon} w_{ji}^{\varepsilon H} \varepsilon_i^{NN} + b_j^H]). \quad (4.2)$$

The scaled stress output of the  $k^{th}$  output node, equation (4.4), is given by,

$$\sigma_k^{NN} = \left( \left[ \sum_{j=1}^{NH} w_{kj}^{H\sigma} B_j + b_k^\sigma \right] \right). \quad (5.11)$$

The network output defined in equations (5.9) to (5.10), is finally scaled back to a value with a physical meaning via equation (4.4) and the training data maximum and minimum stress values,  $\sigma_k^{max}$  and  $\sigma_k^{min}$  respectively,

$$\sigma_k = \left( \frac{\sigma_k^{NN} + 1}{2} \right) \times (\sigma_k^{max} - \sigma_k^{min}). \quad (5.12)$$

The terms of the implied stiffness matrix may be found via partial differentiation of these network equations. This process is very similar to that used in the derivation of the derivatives used in back propagation training, Section 4.1.2.1. Employing the chain rule, the partial derivative of the network outputs (warp and fill stress) with respect to the network input, (warp and fill strain) is given by

$$\frac{\partial \sigma_k}{\partial \varepsilon_i} = \frac{\partial \sigma_k}{\partial \sigma_k^{NN}} \frac{\partial \sigma_k^{NN}}{\partial \varepsilon_i}. \quad (5.13)$$

From equation (5.12) the partial derivative of the final un-scaled network output,  $\sigma_k$ , with respect to the scaled output layer output,  $\sigma_k^{NN}$ , is given by

$$\frac{\partial \sigma_k}{\partial \sigma_k^{NN}} = \frac{(\sigma_k^{max} - \sigma_k^{min})}{2}. \quad (5.14)$$

The partial derivative of the output layer output,  $\sigma_k^{NN}$ , with respect to the un-scaled input,  $\varepsilon_i$ , requires further use of the chain rule leading to

$$\frac{\partial \sigma_k^{NN}}{\partial \varepsilon_i} = \sum_{j=1}^{NH} \left( \frac{\partial \sigma_k^{NN}}{\partial H_j} \frac{\partial H_j}{\partial \varepsilon_i} \right). \quad (5.15)$$

From equation (5.11) the partial derivative of the output of the  $k$ th node of the output layer,  $\sigma_k^{NN}$ , with respect to the output of the  $j^{th}$  node of the hidden layer,  $H_j$ , is given by

$$\frac{\partial \sigma_k^{NN}}{\partial H_j} = \frac{\partial (\beta [\sum_{j=1}^{NB} w_{kj}^{H\sigma} (B_j) + b_j])}{\partial H_j} = w_{jk}^{H\sigma}. \quad (5.16)$$

Substitution of equation (5.10) into equation (4.2) and expansion of the summation leads to the output of the  $j^{th}$  node of the hidden layer in terms of the un-scaled network inputs, as,

$$H_j = \tanh \left( w_{j1}^H \left( \frac{2(\varepsilon_1 - \varepsilon_1^{min})}{(\varepsilon_1^{max} - \varepsilon_1^{min})} - 1 \right) + w_{j2}^H \left( \frac{2(\varepsilon_2 - \varepsilon_2^{min})}{(\varepsilon_2^{max} - \varepsilon_2^{min})} - 1 \right) + b_j^H \right). \quad (5.17)$$

The partial derivative of a tan-sigmoid function is given by,

$$\frac{\partial \tanh(f(\varepsilon_i))}{\partial \varepsilon_i} = \left( 1 - (\tanh(f(\varepsilon_i)))^2 \right) \frac{\partial f(\varepsilon_i)}{\partial \varepsilon_i}. \quad (5.18)$$

The partial derivative of the input to the tan-sigmoid activation function,  $f(\varepsilon_i)$ , of the  $j^{th}$  node of the hidden layer with respect to the un-scaled input to the first node of the input layer is given by,

$$\frac{\partial \left[ w_{j1}^{\varepsilon H} \left( \frac{2(\varepsilon_{i=1} - \varepsilon_{i=1}^{\min})}{(\varepsilon_{i=1}^{\max} - \varepsilon_{i=1}^{\min})} - 1 \right) + w_{j2}^{\varepsilon H} \left( \frac{2(\varepsilon_{i=2} - \varepsilon_{i=2}^{\min})}{(\varepsilon_{i=2}^{\max} - \varepsilon_{i=2}^{\min})} - 1 \right) + b_j^H \right]}{\partial \varepsilon_{i=1}} = \frac{2w_{j1}^{\varepsilon H}}{(\varepsilon_{i=1}^{\max} - \varepsilon_{i=1}^{\min})}, \quad (5.19)$$

and with respect to the second node of the input layer by,

$$\frac{\partial \left[ w_{j1}^{\varepsilon H} \left( \frac{2(\varepsilon_{i=1} - \varepsilon_{i=1}^{\min})}{(\varepsilon_{i=1}^{\max} - \varepsilon_{i=1}^{\min})} - 1 \right) + w_{j2}^{\varepsilon H} \left( \frac{2(\varepsilon_{i=2} - \varepsilon_{i=2}^{\min})}{(\varepsilon_{i=2}^{\max} - \varepsilon_{i=2}^{\min})} - 1 \right) + b_j^H \right]}{\partial \varepsilon_{i=2}} = \frac{2w_{j2}^{\varepsilon H}}{(\varepsilon_{i=2}^{\max} - \varepsilon_{i=2}^{\min})}. \quad (5.20)$$

Therefore, substitution of a generalise version of equations (5.19) and (5.20) into equation (5.18) leads to the partial derivative of output of the  $j^{th}$  node of the hidden layer with respect to the un-scaled input of the  $i^{th}$  node of the input layer,

$$\frac{\partial H_j}{\partial \varepsilon_i} = \left( 1 - (H_j)^2 \right) \frac{2w_{ji}^{\varepsilon H}}{(\varepsilon_i^{\max} - \varepsilon_i^{\min})}. \quad (5.21)$$

Substitution of equations (5.21) and (5.16) into equation (5.15) and substitution of that along with equation (5.14) into equation (5.13) leads to the following set of equations for the terms of the implied elastic stiffness matrix,

$$\frac{\partial \sigma_{k=1}}{\partial \varepsilon_{i=1}} = \frac{(\sigma_{k=1}^{\max} - \sigma_{k=1}^{\min})}{2} \sum_{j=1}^{NH} \left( \{W_{1j}^{\sigma H}\} \times \left\{ \left( 1 - (H_j)^2 \right) \left( \frac{2w_{j1}^{\varepsilon H}}{(\varepsilon_{i=1}^{\max} - \varepsilon_{i=1}^{\min})} \right) \right\} \right), \quad (5.22)$$

$$\frac{\partial \sigma_{k=1}}{\partial \varepsilon_{i=2}} = \frac{(\sigma_{k=1}^{\max} - \sigma_{k=1}^{\min})}{2} \sum_{j=1}^{NH} \left( \{W_{1j}^{\sigma H}\} \times \left\{ \left( 1 - (H_j)^2 \right) \left( \frac{2w_{j2}^{\varepsilon H}}{(\varepsilon_{i=2}^{\max} - \varepsilon_{i=2}^{\min})} \right) \right\} \right), \quad (5.23)$$

$$\frac{\partial \sigma_{k=2}}{\partial \varepsilon_{i=1}} = \frac{(\sigma_{k=2}^{\max} - \sigma_{k=2}^{\min})}{2} \sum_{j=1}^{NH} \left( \{W_{2j}^{\sigma H}\} \times \left\{ \left( 1 - (H_j)^2 \right) \left( \frac{2w_{j1}^{\varepsilon H}}{(\varepsilon_{i=1}^{\max} - \varepsilon_{i=1}^{\min})} \right) \right\} \right), \quad (5.24)$$

$$\frac{\partial \sigma_{k=2}}{\partial \varepsilon_{i=2}} = \frac{(\sigma_{k=2}^{\max} - \sigma_{k=2}^{\min})}{2} \sum_{j=1}^{NH} \left( \{W_{2j}^{\sigma H}\} \times \left\{ \left( 1 - (H_j)^2 \right) \left( \frac{2w_{j2}^{\varepsilon H}}{(\varepsilon_{i=2}^{\max} - \varepsilon_{i=2}^{\min})} \right) \right\} \right). \quad (5.25)$$

### 5.1.1.1 Implied stiffness validation with plane stress neural network model

It is not the intention that a neural network material model should be used to represent a plane stress material response. However, it is useful to demonstrate effectiveness of the implied stiffness matrix using a network trained with plane stress data, e.g. a flat surface in  $\sigma, \varepsilon$  space. The PTFE response surface style test data from Section 4.3.1, Figure 4.9, is used to generate a plane stress training data set. Differential minimisation is used to fit a plane stress material model to 5 unique cycles of the experimental test data with residual strain removed. Error between strain generated from stress via a plane stress stiffness matrix, equation (5.26), and experimental strain is minimised. Un-constrained independent Poisson's ratios are used [68].

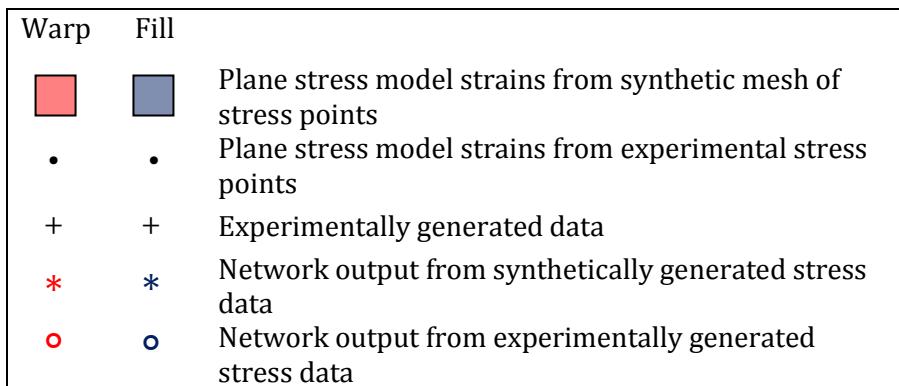
$$\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \end{bmatrix} = \begin{bmatrix} 1/E_x & -v_{yx}/E_y \\ -v_{xy}/E_x & 1/E_y \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \end{bmatrix} \quad (5.26)$$

The resulting elastic constants are shown in Table 5-1 along with the  $R^2$  value achieved by the plane stress material model. As an aside, it is notable that the performance of the plane stress model is significantly lower than the equivalent neural network material models assessed in Section 4.3.3 which consistently achieve  $R^2$  values in excess of 0.98.

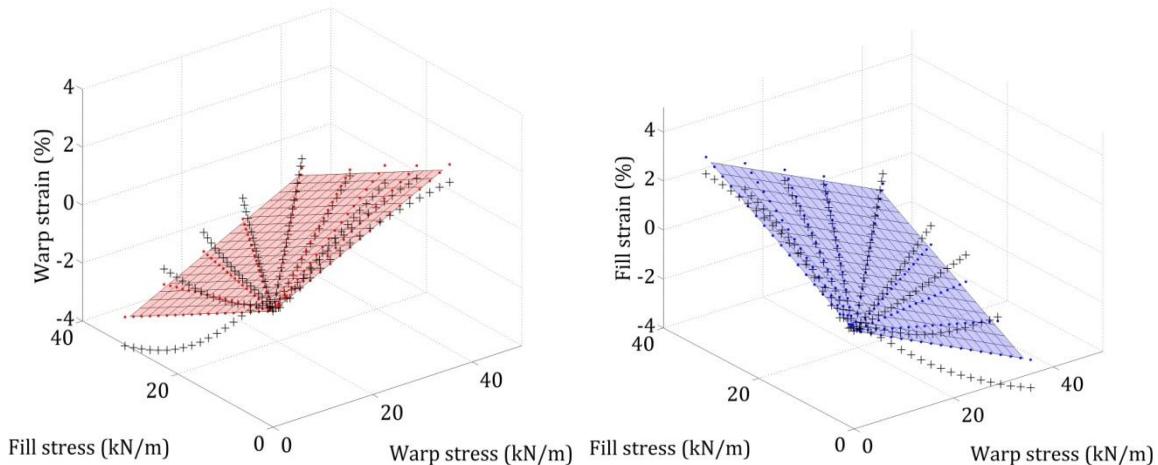
|             | $E_x$   | $E_y$  | $v_{xy}$ | $v_{yx}$ | $R^2$  |
|-------------|---------|--------|----------|----------|--------|
| PTFE B18089 | 1151.94 | 794.11 | 1.11     | 0.78     | 0.9105 |

**Table 5-1 PTFE elastic constants**

Two sets of data are generated using the plane-stress material model, equation (5.26). The first uses the experimental stresses to generate plane stress strains. The second set comprehensively explores the response area using a grid of stress points to generate further plane stress strains. The resulting plane stress data sets along with the original experimental data are plotted in Figure 5-3.



**Figure 5-2: Response surface neural network figure key**



**Figure 5-3: Plane stress representation of PTFE response**

A strain to stress neural network is trained using the plane stress data set derived from experimental stresses. The network comprises an input layer containing two inputs, warp and fill strain, a hidden layer containing 4 neurons and an output layer containing 2 neurons for warp and fill stress. The network variables, including weight and bias values along with the maximum and minimum values used in scaling the network input and output, are shown in Table 5-2.

| $w_{ji}^{\varepsilon H}$ |         | $b_j^H$ |         |
|--------------------------|---------|---------|---------|
| $i = 1$                  | $i = 2$ |         |         |
| $j = 1$                  | -1.8109 | -1.2209 | 0.0217  |
| $j = 2$                  | -1.7464 | -1.9173 | 0.5431  |
| $j = 3$                  | 0.4099  | 0.3738  | 0.0014  |
| $j = 4$                  | -2.2860 | -2.5098 | -0.7868 |

| $w_{kj}^{H\sigma}$ |         |         |         | $b_k^\sigma$ |        |
|--------------------|---------|---------|---------|--------------|--------|
| $j = 1$            | $j = 2$ | $j = 3$ | $j = 4$ |              |        |
| $k = 1$            | 0.0063  | -0.6703 | 31.1350 | -0.3894      | 0.0174 |
| $k = 2$            | 0.0005  | -5.3687 | 2.9609  | -3.1338      | 0.5831 |

| $\varepsilon_i^{max}$ |         | $\varepsilon_i^{min}$ |         | $\sigma_k^{max}$ |         | $\sigma_k^{min}$ |         |
|-----------------------|---------|-----------------------|---------|------------------|---------|------------------|---------|
| $i = 1$               | $i = 2$ | $i = 1$               | $i = 2$ | $k = 1$          | $k = 2$ | $k = 1$          | $k = 2$ |
| 0.0308                | 0.0391  | -0.0303               | -0.0341 | 35.8970          | 31.4160 | -0.0982          | -0.0633 |

Table 5-2: PTFE plane-stress network variables

When tested with both the training data set and the additional mesh of data points the network achieves  $R^2$  values greater than 0.999. The resulting network output is plotted alongside the target data in Figure 5-4.

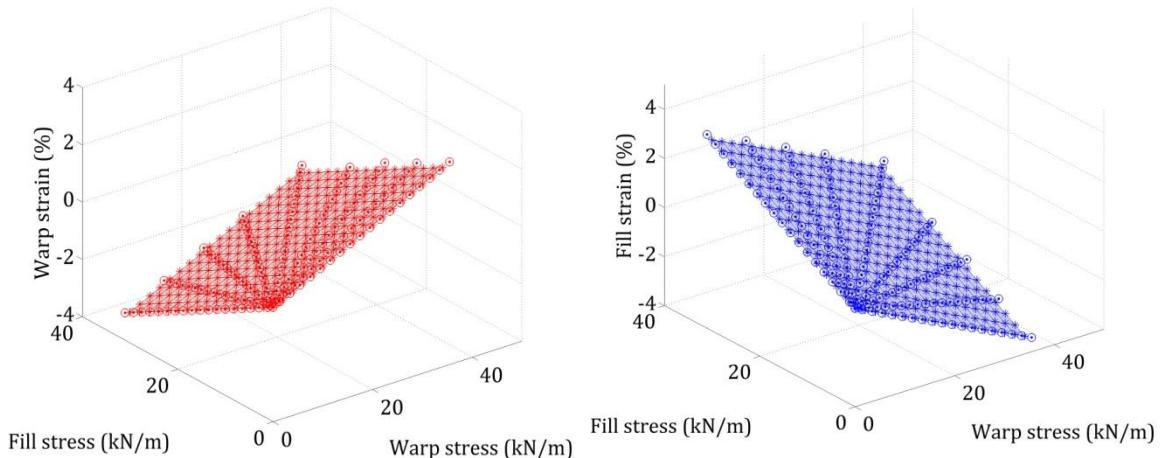


Figure 5-4: Plane stress strain to stress network output plotted with target data sets

In order to validate the implied stiffness matrix it is calculated at 3 points within the response surface using equations (5.6) to **Error! Reference source not found.** and the network variables, Table 5-2. The network is trained with strain as input and stress as

output. Therefore, the inverse of the stiffness matrix, equation (5.26) is used for validation. Substitution of the elastic constants in Table 5-1 leads to,

$$\begin{bmatrix} \sigma_w \\ \sigma_f \end{bmatrix} = \begin{bmatrix} E_w/(1 - v_{yx}v_{xy}) & E_wv_{fw}/(1 - v_{fw}v_{wf}) \\ E_fv_{wf}/(1 - v_{yx}v_{xy}) & E_f/(1 - v_{fw}v_{wf}) \end{bmatrix} \begin{bmatrix} \varepsilon_w \\ \varepsilon_f \end{bmatrix} = \begin{bmatrix} 8329.6 & 6470.3 \\ 6370.0 & 5742.2 \end{bmatrix} \begin{bmatrix} \varepsilon_w \\ \varepsilon_f \end{bmatrix}. \quad (5.27)$$

|  | Data Index: 7      |         | Data Index: 50     |         | Data Index: 120    |         |
|--|--------------------|---------|--------------------|---------|--------------------|---------|
|  | Value              | % Error | Value              | % Error | Value              | % Error |
| $\varepsilon_w$  | 0.0001             | -       | -0.0120            | -       | 0.0145             | -       |
| $\varepsilon_f$  | 0.0017             | -       | 0.0171             | -       | -0.0154            | -       |
| $\sigma_w$   | 12.179             | -0.0051 | 11.251             | -0.0048 | 21.366             | 0.0038  |
| $\sigma_f$   | 10.659             | 0.0004  | 22.208             | -0.0007 | 4.116              | 0.0008  |
| $\frac{\partial \sigma_{k=1}}{\partial \varepsilon_{i=1}}$ | 8330.6<br>(8329.6) | 0.0121  | 8330.7<br>(8329.6) | 0.0128  | 8330.3<br>(8329.6) | 0.0079  |
| $\frac{\partial \sigma_{k=1}}{\partial \varepsilon_{i=2}}$ | 6471.1<br>(6470.3) | 0.0125  | 6471.1<br>(6470.3) | 0.0127  | 6470.7<br>(6470.3) | 0.0070  |
| $\frac{\partial \sigma_{k=2}}{\partial \varepsilon_{i=1}}$ | 6370.0<br>(6370.0) | -0.0007 | 6370.3<br>(6370.0) | 0.0045  | 6370.3<br>(6370.0) | 0.0041  |
| $\frac{\partial \sigma_{k=2}}{\partial \varepsilon_{i=2}}$ | 5742.2<br>(5742.2) | -0.0009 | 5742.5<br>(5742.2) | 0.0044  | 5742.4<br>(5742.2) | 0.0040  |

**Table 5-3: Plane stress network implied stiffness validation**

In Table 5-3 the low magnitude of the error between target stress and network stress demonstrates that the implied stiffness matrix is equivalent to the compliance matrix used in the generation of the training and testing data. This demonstrates that the implied stiffness matrix is an effective equivalent to the plane stress stiffness matrix. When implementing a non-uniform neural network, i.e. one trained directly from experimental data as demonstrated in Section 4.3.2, each data point will generate a unique implied stiffness matrix. The implied stiffness matrix effectively maps a plane tangential to the network response surface at the current location. With a sufficiently small displacement step this matrix may be used to approximate surrounding points on the surface according to,

$$\begin{bmatrix} \sigma_{w,n} \\ \sigma_{f,n} \end{bmatrix} = \begin{bmatrix} \sigma_{w,n-1} \\ \sigma_{f,n-1} \end{bmatrix} + \begin{bmatrix} \Delta\sigma_w \\ \Delta\sigma_f \end{bmatrix} = \begin{bmatrix} \sigma_{w,n-1} \\ \sigma_{f,n-1} \end{bmatrix} + \begin{bmatrix} \frac{\partial \sigma_{k=1}}{\partial \varepsilon_{i=1}} & \frac{\partial \sigma_{k=1}}{\partial \varepsilon_{i=2}} \\ \frac{\partial \sigma_{k=2}}{\partial \varepsilon_{i=1}} & \frac{\partial \sigma_{k=2}}{\partial \varepsilon_{i=2}} \end{bmatrix} \begin{bmatrix} \Delta\varepsilon_w \\ \Delta\varepsilon_f \end{bmatrix}. \quad (5.28)$$

However, as previously discussed, when used in conjunction with a dynamic relaxation solution algorithm the elastic stiffness matrix is used only to calculate the diagonal terms of the system stiffness matrix for kinetic damping to control the rate of convergence.

### 5.1.2 History neural network implied stiffness

In order to capture material response with historical loading effects additional network inputs are required (Section 4.4.4). The network equations for the biaxial hysteresis neural network (Figure 5-5) are as follows. The network inputs are the current warp and fill strains denoted by  $\varepsilon_i$  ( $i = 1 \rightarrow 2$ ), where,

$$\varepsilon_{i=1} = \varepsilon_{w,n}, \quad (5.29)$$

$$\varepsilon_{i=2} = \varepsilon_{f,n}, \quad (5.30)$$

and the historical stresses and strains from the previous loading iteration denoted by  $SV_i$  ( $i = 3 \rightarrow 6$ ), where,

$$SV_{i=3} = \varepsilon_{w,n-1}, \quad (5.31)$$

$$SV_{i=4} = \varepsilon_{f,n-1}, \quad (5.32)$$

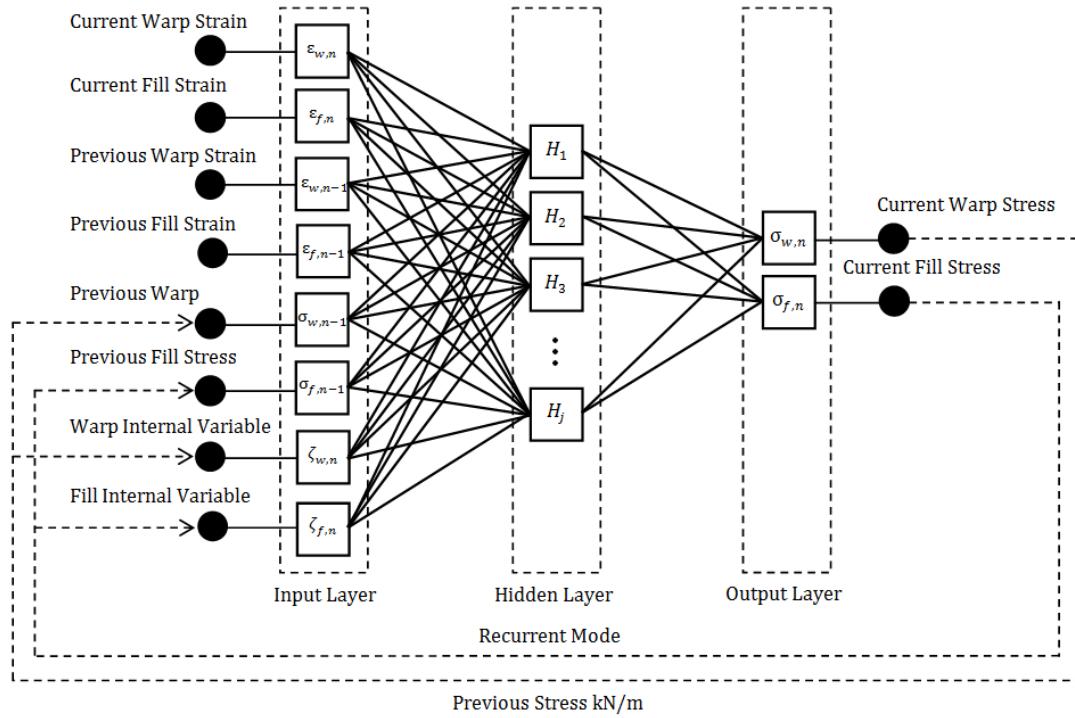
$$SV_{i=5} = \sigma_{w,n-1}, \quad (5.33)$$

$$SV_{i=6} = \sigma_{f,n-1}. \quad (5.34)$$

The subscript  $n$  indicates the current load step and  $n - 1$  the previous load step. The internal variables,  $IV_i$  ( $i = 7 \rightarrow 8$ ), are derived from historical stresses and historical and current strains, and are given by

$$IV_{i=7} = SV_{i=5} \times SV_{i=3} + SV_{i=5}(\varepsilon_{i=1} - SV_{i=3}), \quad (5.35)$$

$$IV_{i=8} = SV_{i=6} \times SV_{i=4} + SV_{i=6}(\varepsilon_{i=2} - SV_{i=4}). \quad (5.36)$$



**Figure 5-5: 8 input strain to stress biaxial hysteresis neural network architecture**

As in the case of the response surface style network all network inputs are scaled to a range between -1 and 1 as follows,

$$\varepsilon_i^{NN} = \frac{2(\varepsilon_i - \varepsilon_i^{\min})}{(\varepsilon_i^{\max} - \varepsilon_i^{\min})} - 1, \quad (5.37)$$

$$SV_i^{NN} = \frac{2(SV_i - SV_i^{\min})}{(SV_i^{\max} - SV_i^{\min})} - 1, \quad (5.38)$$

$$IV_i^{NN} = \frac{2(IV_i - IV_i^{\min})}{(IV_i^{\max} - IV_i^{\min})} - 1. \quad (5.39)$$

With the additional inputs, the output of the  $j^{\text{th}}$  hidden node,  $B_j$ , with a tan sigmoid transfer function is given by,

$$H_j = \tanh\left(\left[\sum_{i=1}^{N\varepsilon} w_{ji}^{\varepsilon H} \varepsilon_i^{NN} + \sum_{i=3}^{NSV} w_{ji}^{SVH} SV_i^{NN} + \sum_{i=7}^{NIV} w_{ji}^{IVH} IV_i^{NN} + b_j^H\right]\right). \quad (5.40)$$

The output of the  $k^{\text{th}}$  output node is given by equation (5.11) and the un-scaled output is given by equation (5.12). The derivatives of equations (5.13) to (5.16) also remain the same. Substitution of equations (5.37) to (5.39) into equation (5.40) and expansion of the summation leads to the output of the  $j^{\text{th}}$  node of the hidden layer in terms of the un-scaled network inputs, gives,

$$H_j = \tanh\left(w_{j1}^{\varepsilon H} \left(\frac{2(\varepsilon_{i=1} - \varepsilon_{i=1}^{\min})}{(\varepsilon_{i=1}^{\max} - \varepsilon_{i=1}^{\min})} - 1\right) + w_{j2}^{\varepsilon H} \left(\frac{2(\varepsilon_{i=2} - \varepsilon_{i=2}^{\min})}{(\varepsilon_{i=2}^{\max} - \varepsilon_{i=2}^{\min})} - 1\right) + w_{j3}^{SVH} \left(\frac{2(SV_{i=3} - SV_{i=3}^{\min})}{(SV_{i=3}^{\max} - SV_{i=3}^{\min})} - 1\right) + \dots + w_{jN}^{IVH} \left(\frac{2(IV_{i=N} - IV_{i=N}^{\min})}{(IV_{i=N}^{\max} - IV_{i=N}^{\min})} - 1\right) + b_j^H\right) \quad (5.41)$$

$$1 + w_{j4}^{SVH} \left( \frac{2(SV_{i=4} - SV_{i=4}^{\min})}{(SV_{i=4}^{\max} - SV_{i=4}^{\min})} - 1 \right) + w_{j5}^{SVH} \left( \frac{2(SV_{i=5} - SV_{i=5}^{\min})}{(SV_{i=5}^{\max} - SV_{i=5}^{\min})} - 1 \right) + \\ w_{j6}^{SVH} \left( \frac{2(SV_{i=6} - SV_{i=6}^{\min})}{(SV_{i=6}^{\max} - SV_{i=6}^{\min})} - 1 \right) + w_{j7}^{IVH} \left( \frac{2((SV_{i=5} \times SV_{i=3}) + SV_{i=5}(\varepsilon_{i=1} - SV_{i=3})) - IV_{i=7}^{\min}}{(IV_{i=7}^{\max} - IV_{i=7}^{\min})} - 1 \right) + \\ w_{j8}^{IVH} \left( \frac{2((SV_{i=6} \times SV_{i=4}) + SV_{i=6}(\varepsilon_{i=2} - SV_{i=4})) - IV_{i=8}^{\min}}{(IV_{i=8}^{\max} - IV_{i=8}^{\min})} - 1 \right) + b_j^H.$$

Where, the internal variables are a function of the current strain. Therefore, the partial derivative of the input to the tan-sigmoid activation function,  $f(\varepsilon_i)$ , of the  $j^{th}$  node of the hidden layer with respect to the un-scaled input to the first node of the input layer is given by,

$$\frac{\partial (\beta [\sum_{i=1}^{N\varepsilon} w_{ji}^{\varepsilon H} \varepsilon_i^{NN} + \sum_{i=3}^{NSV} w_{ji}^{SVH} SV_i^{NN} + \sum_{i=7}^{NIV} w_{ji}^{IVH} IV_i^{NN} + b_j^H])}{\partial \varepsilon_{i=1}} = \frac{2w_{j1}^{\varepsilon H}}{(\varepsilon_{i=1}^{\max} - \varepsilon_{i=1}^{\min})} + \frac{2w_{j7}^{IVH} SV_{i=5}}{(IV_{i=7}^{\max} - IV_{i=7}^{\min})}. \quad (5.42)$$

$$\frac{\partial (\beta [\sum_{i=1}^{N\varepsilon} w_{ij}^{\varepsilon H} \varepsilon_i^{NN} + \sum_{i=3}^{NSV} w_{ij}^{SVH} SV_i^{NN} + \sum_{i=7}^{NIV} w_{ij}^{IVH} IV_i^{NN} + b_j^H])}{\partial \varepsilon_{i=2}} = \frac{2w_{j2}^{\varepsilon H}}{(\varepsilon_{i=2}^{\max} - \varepsilon_{i=2}^{\min})} + \frac{2w_{j8}^{IVH} SV_{i=6}}{(IV_{i=8}^{\max} - IV_{i=8}^{\min})}. \quad (5.43)$$

Substitution of a generalised version of equations (5.42) and (5.43) into the standard derivative of a tan-sigmoid function, equation (5.18), leads to the partial derivative of the output of the  $j^{th}$  node of the hidden layer with respect to the un-scaled input of the  $i^{th}$  node of the input layer, as in,

$$\frac{\partial H_j}{\partial \varepsilon_i} = \left(1 - (H_j)^2\right) \frac{2w_{ji}^{\varepsilon H}}{(\varepsilon_i^{\max} - \varepsilon_i^{\min})}. \quad (5.44)$$

Substitution of equations (5.44) and (5.16) into equation (5.15) and substitution of the resulting expression along with equation (5.14) into equation (5.13) leads to the following general equation for the terms of the implied elastic stiffness matrix in equation,

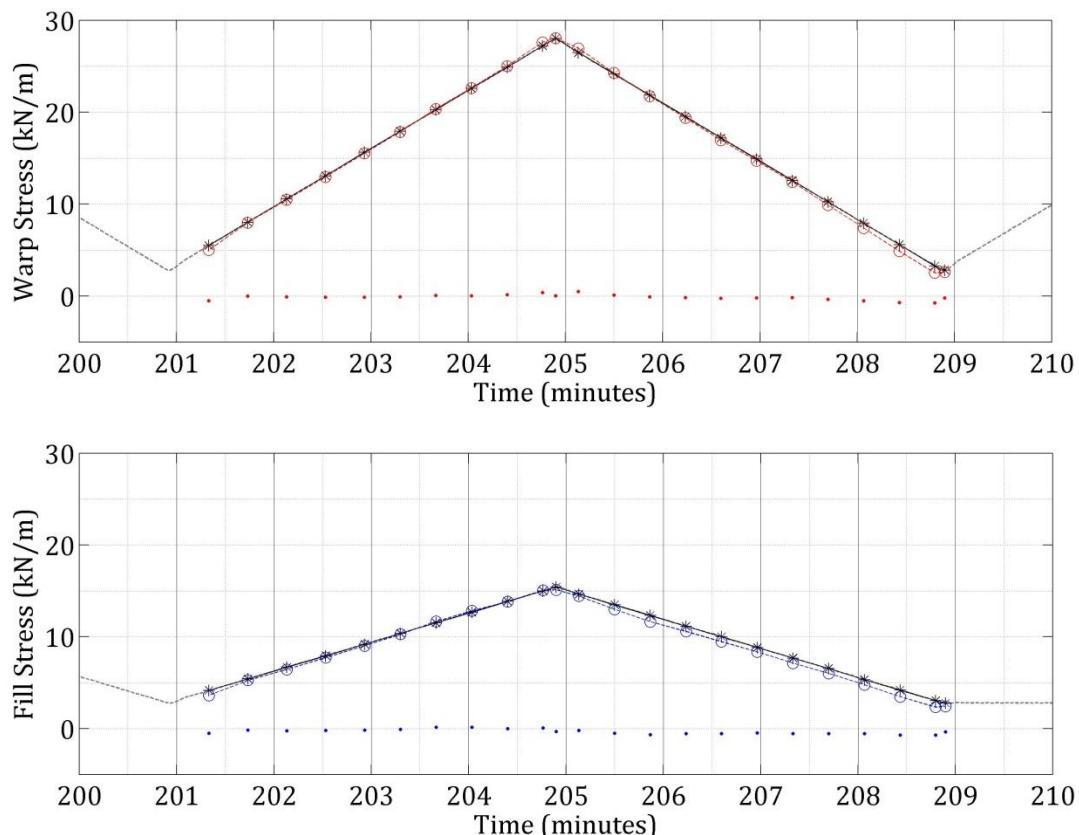
$$\frac{\partial \sigma_{k=1,2}}{\partial \varepsilon_{i=1,2}} = \frac{(\sigma_k^{\max} - \sigma_k^{\min})}{2} \sum_{j=1}^{NH} \left( \{W_{kj}^{H\sigma}\} \times \left\{ \left(1 - (B_j)^2\right) \left( \frac{2w_{j1}^{\varepsilon H}}{(\varepsilon_i^{\max} - \varepsilon_i^{\min})} + \frac{2w_{j(i+6)}^{IVH} SV_{(i+4)}}{(IV_{(i+6)}^{\max} - IV_{(i+6)}^{\min})} \right) \right\} \right). \quad (5.45)$$

Written explicitly this equation becomes,

$$\begin{aligned} \frac{\partial \sigma_{k=1}}{\partial \varepsilon_{i=1}} &= \frac{(\sigma_1^{\max} - \sigma_1^{\min})}{2} \sum_{j=1}^{NH} \left( \{W_{1j}^{H\sigma}\} \times \left\{ \left(1 - (H_j)^2\right) \left( \frac{2w_{j1}^{\varepsilon H}}{(\varepsilon_1^{\max} - \varepsilon_1^{\min})} + \frac{2w_{j(7)}^{IVH} SV_{(5)}}{(IV_{(7)}^{\max} - IV_{(7)}^{\min})} \right) \right\} \right), \\ \frac{\partial \sigma_{k=1}}{\partial \varepsilon_{i=2}} &= \frac{(\sigma_1^{\max} - \sigma_1^{\min})}{2} \sum_{j=1}^{NH} \left( \{W_{1j}^{H\sigma}\} \times \left\{ \left(1 - (H_j)^2\right) \left( \frac{2w_{j2}^{\varepsilon H}}{(\varepsilon_2^{\max} - \varepsilon_2^{\min})} + \frac{2w_{j(8)}^{IVH} SV_{(6)}}{(IV_{(8)}^{\max} - IV_{(8)}^{\min})} \right) \right\} \right), \\ \frac{\partial \sigma_{k=2}}{\partial \varepsilon_{i=1}} &= \frac{(\sigma_2^{\max} - \sigma_2^{\min})}{2} \sum_{j=1}^{NH} \left( \{W_{2j}^{H\sigma}\} \times \left\{ \left(1 - (H_j)^2\right) \left( \frac{2w_{j1}^{\varepsilon H}}{(\varepsilon_1^{\max} - \varepsilon_1^{\min})} + \frac{2w_{j(7)}^{IVH} SV_{(5)}}{(IV_{(7)}^{\max} - IV_{(7)}^{\min})} \right) \right\} \right), \\ \frac{\partial \sigma_{k=2}}{\partial \varepsilon_{i=2}} &= \frac{(\sigma_2^{\max} - \sigma_2^{\min})}{2} \sum_{j=1}^{NH} \left( \{W_{2j}^{H\sigma}\} \times \left\{ \left(1 - (H_j)^2\right) \left( \frac{2w_{j2}^{\varepsilon H}}{(\varepsilon_2^{\max} - \varepsilon_2^{\min})} + \frac{2w_{j(8)}^{IVH} SV_{(6)}}{(IV_{(8)}^{\max} - IV_{(8)}^{\min})} \right) \right\} \right). \end{aligned} \quad (5.46)$$

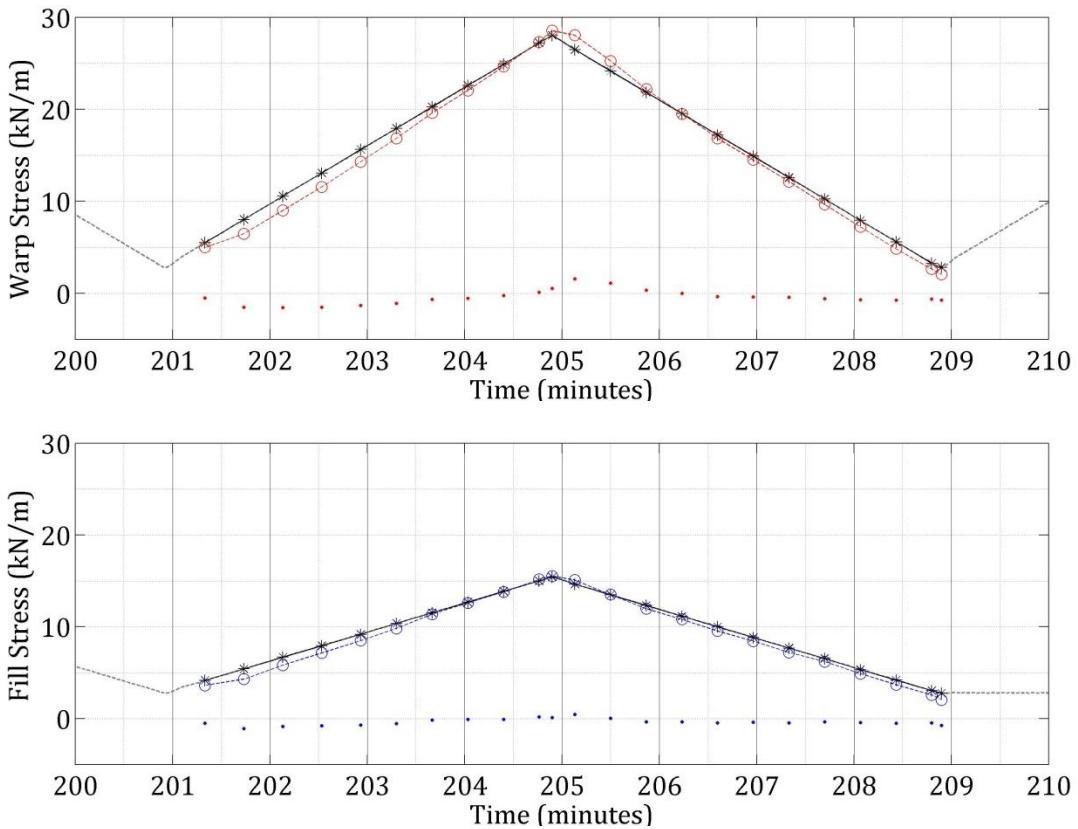
### 5.1.2.1 Implied stiffness validation with history neural network model

The 8 input PVC history network trained and tested in Section 4.4.4 is used to validate the implied stiffness matrix. Taking a single cycle of the 0:1 profile, the implied stiffness is calculated at each data point along with the network output. The network output in recurrent mode along with the target data is shown in Figure 5-6 and the implied stiffness results along with the target data is shown in Figure 5-7. It can clearly be seen that the network output closely fits the target data.



**Figure 5-6: Network output plotted alongside target data from Profile 0:1, cycle 50.**  
 ( — warp, — fill, \* target, ° network output)

The fit of the implied stiffness results to the target data is less good but offers a reasonable approximation. The error between the implied stiffness stress and target stress is due to the failure of the linear implied stiffness matrix approximation to capture the changing gradient of the non-linear fabric response. The next data point in the experimental response surface may either lay above or below the flat tangent plane defined at the current data point by the implied stiffness matrix. This leads to either an under or over estimation of next stress step. This is particularly clear in the warp stress plot of Figure 5-7 where stress is underestimated in the initial loading section and overestimated immediately after the turning point between loading unloading. This indicate that the gradient of the response surface is particularly variable in these areas.



**Figure 5-7: Implied stiffness matrix output plotted alongside target data from Profile 0:1, cycle 50. (— warp, — fill, \* target, o network output)**

To further validate the implied stiffness definition the analytical derivatives are compared with an equivalent finite difference method derivative in Table 5-4. A perturbation of  $10^{-7}$ ,  $h$ , is added to each of the current strain inputs in turn and presented to the network. The original network output,  $f(a)$ , and perturbed output,  $f(a + h)$ , may then be used to find the finite difference derivative.

$$\frac{\partial f(a)}{\partial a} \approx \frac{f(a+h) - f(a)}{h} \quad (5.47)$$

The error between the analytical and numerical derivatives is generally less than 1%. The only significant error occurs where the implied stiffness term is significantly lower than the other 3 terms, as in the case of the  $\frac{\partial \sigma_{k=2}}{\partial \varepsilon_{i=1}}$  term for the 10<sup>th</sup> data point which produces a 4.34% error. In addition this data point is positioned close to the loading to unloading turning point of the load profile. As discussed above it is likely that in this region the network warp response surface gradient varies significantly in agreement with the variation of the experimental response surface gradient. Although the perturbation applied to the strain is very small the perturbed input-output set represents a location on the response surface with a marginally different gradient to the unperturbed input-output set. The inconsistency between the strain to stress relationship at the perturbed and

unperturbed response surface locations leads to inconsistencies between the finite difference and analytical implied stiffness terms associated with the warp direction.

| Data Index: 10   |            | Data Index: 20    |         |
|--|------------|-------------------|---------|
| Inputs   |            | Inputs            |         |
| $\varepsilon_{w,n}$  | 0.0765     | 0.0615            |         |
| $\varepsilon_{f,n}$  | 0.0173     | 0.0111            |         |
| $\sigma_{w,n-1}$   | 24.8773    | 10.2473           |         |
| $\sigma_{f,n-1}$   | 13.8637    | 6.5550            |         |
| $\varepsilon_{w,n-1}$                                      | 0.0735     | 0.0646            |         |
| $\varepsilon_{f,n-1}$                                      | 0.0160     | 0.0119            |         |
| $\zeta_{w,n}$  | 1.9030     | 0.6304            |         |
| $\zeta_{f,n}$  | 0.2404     | 0.0727            |         |
| Outputs  |            | Outputs           |         |
| $\sigma_{w,n}$   | 27.6031    | 7.4002            |         |
| $\sigma_{f,n}$   | 15.0650    | 4.7717            |         |
|  | Analytical | Finite Difference | % error |
| $\frac{\partial \sigma_{k=1}}{\partial \varepsilon_{i=1}}$ | 790.4      | 785.1             | -0.6756 |
| $\frac{\partial \sigma_{k=1}}{\partial \varepsilon_{i=2}}$ | 113.9      | 113.8             | -0.1556 |
| $\frac{\partial \sigma_{k=2}}{\partial \varepsilon_{i=1}}$ | 53.0       | 50.7              | -4.3406 |
| $\frac{\partial \sigma_{k=2}}{\partial \varepsilon_{i=2}}$ | 808.2      | 808.2             | -0.0006 |
|  | Analytical | Finite Difference | % error |

Table 5-4: PVC history network implied stiffness validation using finite difference method ( $h = 0.0000001$ )

## 5.2 Response Surface Style Network Implementation

The expression of equilibrium for the system (Section 3.1) is given by,

$$\begin{aligned} \frac{\partial \Pi_{ps}}{\partial D_{1-n}} = \{R\} &= \sum_{i=1}^{i=m} \left[ \int_V [B]^T [E][B] dV \right]_i \{D\} - \sum_{i=1}^{i=m} \left[ \int_V [B]^T [E][\varepsilon_0] dV \right]_i + \\ &\quad \sum_{i=1}^{i=m} \left[ \int_V [B]^T [\sigma_0] dV \right]_i + \{P\} = \sum_{i=1}^{i=m} \left[ \int_V [B]^T [\sigma] dV \right]_i - \{P\} = 0. \end{aligned} \quad (3.11)$$

Where

$$[\sigma] = [E]([\varepsilon] - [\varepsilon_0]) + [\sigma_0] = [E]([B]\{D\} - [\varepsilon_0]) + [\sigma_0] \quad (3.12)$$

In the case of the response surface network material model this equation is replaced by,

$$[\sigma] = N([\varepsilon] + [\varepsilon_0]), \quad (5.48)$$

where  $N([\varepsilon] + [\varepsilon_0])$  denotes the network output which is a function of the total current strain. The initial strain applied to the fabric is denoted by  $\varepsilon_0$  and the current strain derived from the nodal displacement via the element B matrix by  $\varepsilon$ . The expression of equilibrium for the system becomes,

$$\frac{\partial \Pi_{ps}}{\partial D_{1-n}} = \{R\} = \sum_{i=1}^{i=m} \left[ \int_V [B]^T [N([\varepsilon] + [\varepsilon_0])] dV \right]_i - \{P\} = 0. \quad (5.49)$$

The initial strain is a constant value which is the strain applied in order to induce the desired pre-stress defined during form-finding. This initial pre-stress is represented by  $\sigma_0$  in equation (3.12) and initial strain is typically 0 unless applied displacement boundary conditions have been defined. In engineering practice, detailed numerical simulation of fabric installation is generally not undertaken. This is due to the complexity of modelling the fabrics initial response. However, an initial strain value that corresponds to the correct initial stress level may be estimated using a stress to strain neural network, Section 4.3.2.2, or through a patch test.

The current strain is given by the product of element B-matrix and nodal displacements with the addition of initial strain, such as,

$$\{\varepsilon\} = [B]\{\delta\}_i + [\varepsilon_0]. \quad (5.50)$$

The elastic stiffness matrix,  $K_E$ , is approximated by

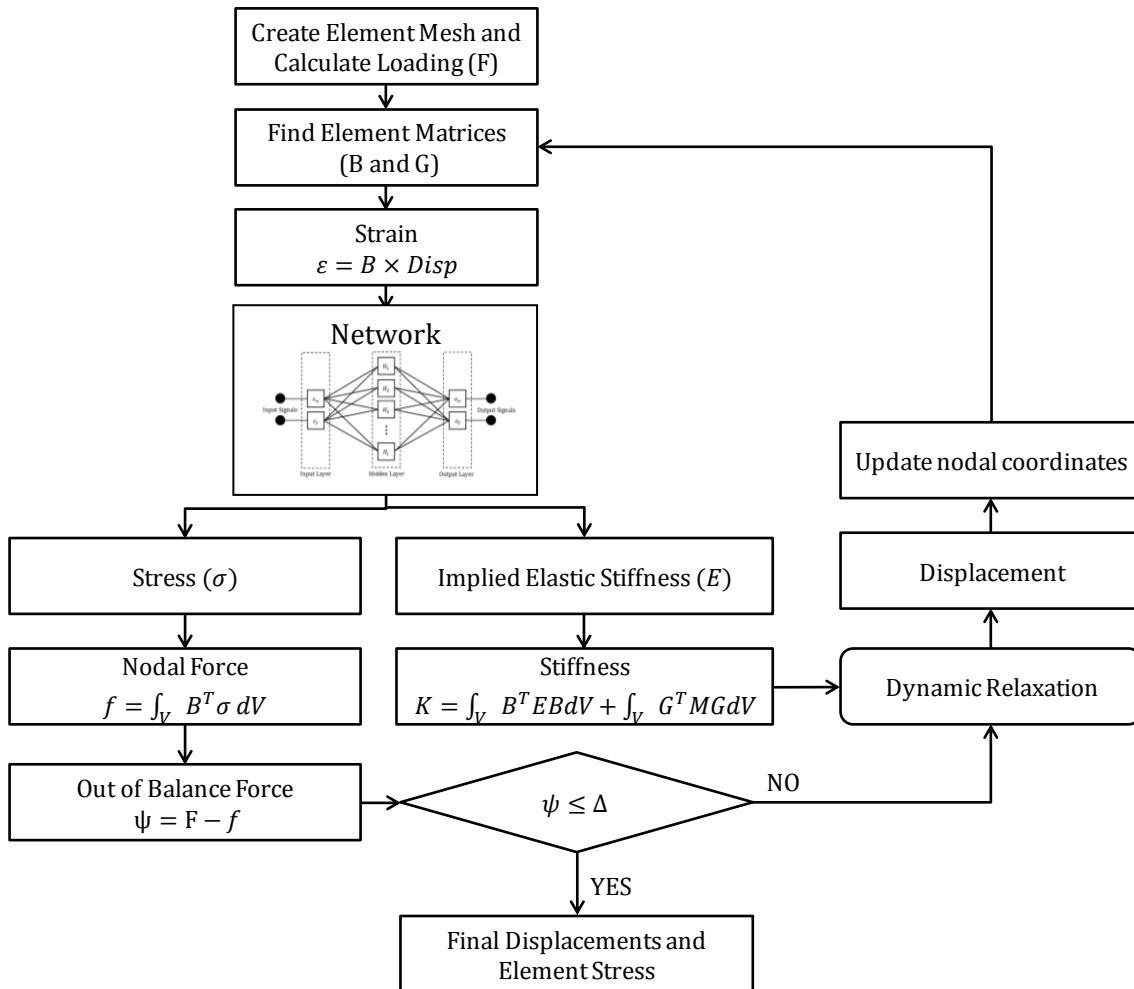
$$[K_E] = \sum_{i=1}^{i=m} \left[ \int_V [B]^T [E^{implied}] [B] dV \right]_i. \quad (5.51)$$

The geometric stiffness matrix,  $K_\sigma$ , given by

$$[K_\sigma] = \sum_{i=1}^{i=m} \left[ \int_V d[B]^T [\sigma] dV \right]_i, \quad (5.52)$$

where the network definition for stress,  $\sigma$ , equation (5.48), is used.

The implementation of the response surface network within the finite element analysis procedure is summarised in Figure 5-8.



**Figure 5-8: Finite Element analysis procedure with response surface neural network material model**

### 5.2.1 Additional areas, wrinkling criteria

Neural network generalisation is unreliable outside of the bounds of the training data.

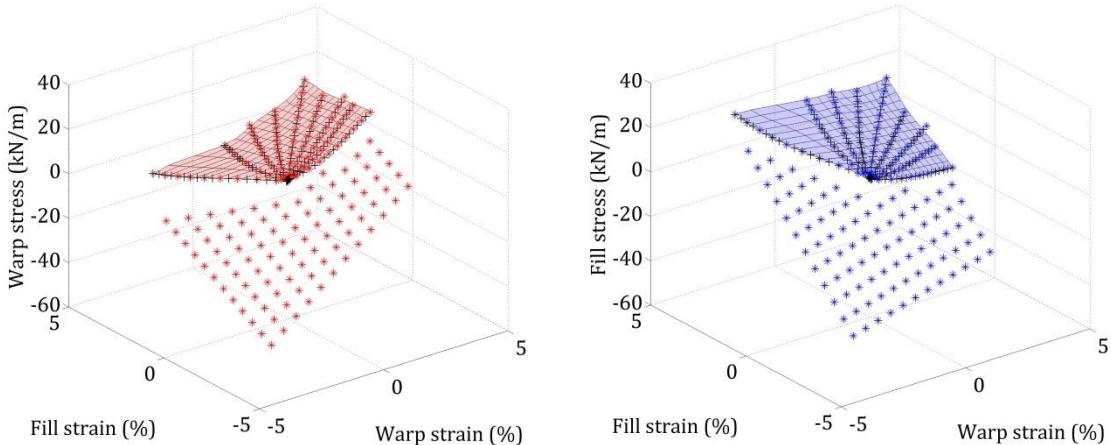
Therefore, it is necessary to include some additional data to enable the simulation of loading conditions outside the tested load ratios. In future work this is likely to involve the completion of more comprehensive testing profiles which explore stress response below pre-stress and into the wrinkled regions of the stress-strain response surface. However, it is reasonable to use a plane stress approximation (Table 5-5) with wrinkling criteria applied in order to generate additional data points to train the network in these areas of the response surface. Especially given that engineering design practice would not

normally permit such stress strain scenarios but data is required to enable the analysis to continue to a possible feasible or acceptable solution at equilibrium.

| $E_w$     | $E_f$     | $v_{wf}$ | $v_{fw}$ | $G_{wf}$ |
|-----------|-----------|----------|----------|----------|
| 769.3kN/m | 856.4kN/m | 0.32     | 0.59     | 30kN/m   |

**Table 5-5: PVC plane stress elastic constants**

Initially a regular grid of warp and fill strain pairs is generated. A minimum warp strain equal to minus 1 times the minimum absolute value of warp strain and a maximum equal to the maximum warp strain and the equivalent fill data is used to define the limits of the grid. Further data points are defined at intervals of 0.5% strain. All strain points that fall within the experimental response envelope are removed from the data set and the resulting points are used with plane stress elastic constants to generate equivalent stress data. The additional plane stress data points are shown in Figure 5-9.



**Figure 5-9: Training data with additional fictitious plane stress data zones**

In the current work shear stress is treated as being uncoupled from the direct stress, as in plane stress theory, and is calculated from shear strain using a shear modulus. For the purposes of applying wrinkling criteria the shear strain is assumed to be zero as is the case in the biaxial testing methodology used in the generation of the experimental training data. As discussed in Section 3.1.1.3 wrinkling is identified using one of 3 different sets of criterion, the principle stresses, the principle strains or a combination of both. The principle stress are given by,

$$\sigma_{max,min}^p = \frac{1}{2}(\sigma_f + \sigma_w) \pm \sqrt{\left(\frac{\sigma_w - \sigma_f}{2}\right)^2 + (\tau_{xy})^2}, \quad (3.132)$$

and the principle strains by,

$$\varepsilon_{max,min}^p = \frac{1}{2}(\varepsilon_f + \varepsilon_w) \pm \sqrt{\left(\frac{\varepsilon_w - \varepsilon_f}{2}\right)^2 + (\gamma_{xy})^2}. \quad (3.133)$$

1. Stress criterion based solely on principle stress,

$$\sigma_{min}^p > 0: \quad \text{'taut state'} \quad (3.135)$$

$$\sigma_{min}^p \leq 0 \text{ and } \sigma_{max}^p > 0: \quad \text{'wrinkled state'} \quad (3.136)$$

$$\text{otherwise:} \quad \text{'slack state'} \quad (3.137)$$

2. Strain criterion based solely on principle strain,

$$\varepsilon_{min}^p > 0: \quad \text{'taut state'} \quad (3.138)$$

$$\varepsilon_{min}^p \leq 0 \text{ and } \varepsilon_{max}^p > 0: \quad \text{'wrinkled state'} \quad (3.139)$$

$$\text{otherwise:} \quad \text{'slack state'} \quad (3.140)$$

3. Mixed criterion based on a combination of principle stress and strain,

$$\sigma_{min}^p > 0: \quad \text{'taut state'} \quad (3.141)$$

$$\sigma_{min}^p \leq 0 \text{ and } \varepsilon_{max}^p > 0: \quad \text{'wrinkled state'} \quad (3.142)$$

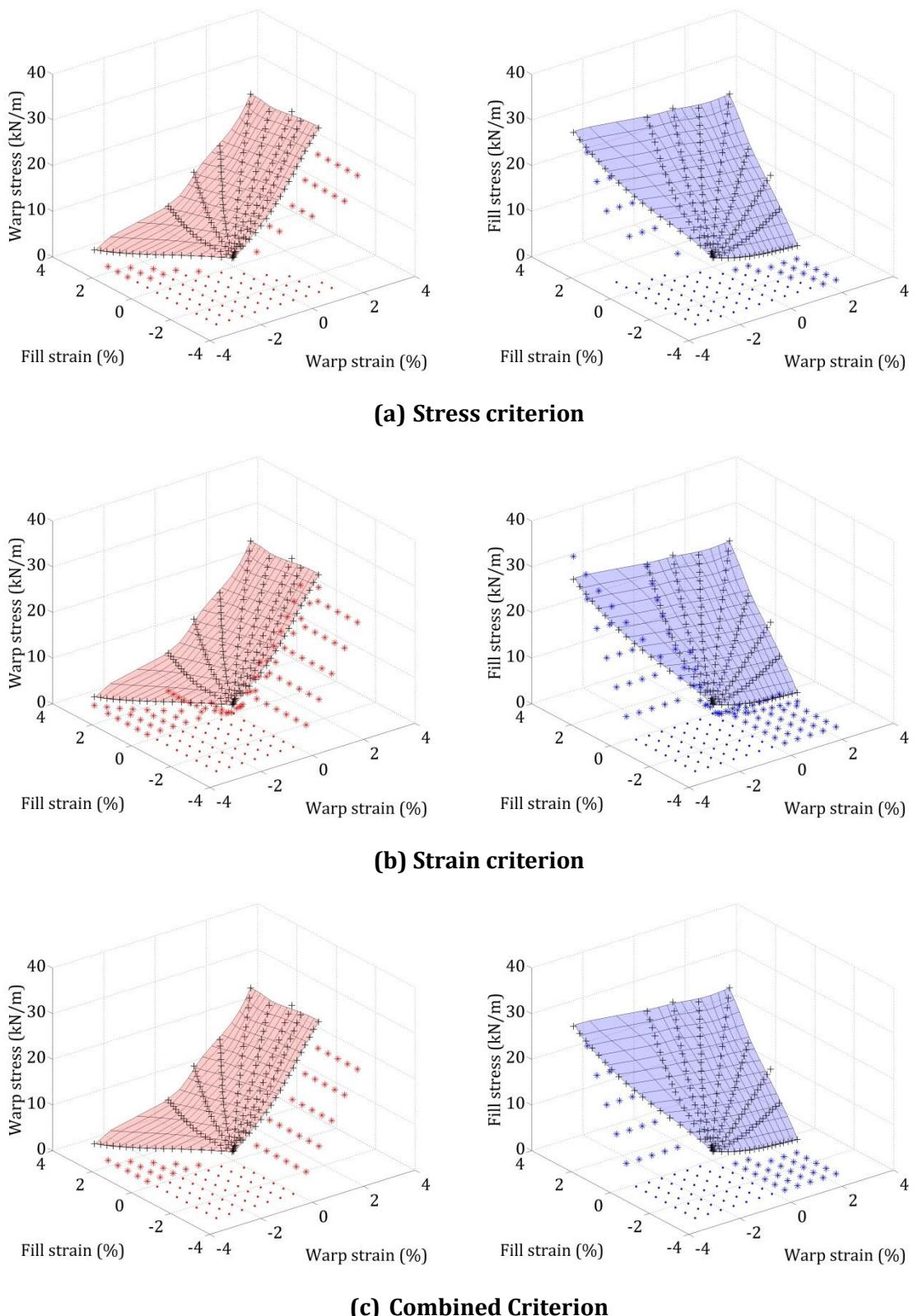
$$\text{otherwise:} \quad \text{'slack state'} \quad (3.143)$$

Depending upon the state identified a penalisation factor is applied to the elastic stiffness matrix. Where the fabric is identified as wrinkled, penalisation occurs in the direction of the wrinkle. In cases with zero shear strain the angle between maximum stress and the warp direction will either be  $0^\circ$  where warp stress is greater than fill or  $90^\circ$  where fill is greater than warp. Where the membrane is deemed to be slack the material stiffness matrix is penalised in all directions.

The removal of shear stress from the wrinkling procedure leads to a simplification as the wrinkle direction will not be used or in fact calculated during analysis. However, it is not likely that a simulated fabric structure with areas of wrinkled or slack material would be deemed acceptable. Therefore, the additional data only serves to stabilise the analysis and allow the identification of problem areas in the design. It should also be noted that all simulations provide only an approximation to the stress strain state of a real structure. Therefore uncertainty should be considered within the design procedure.

| Warp | Fill |  |
|------|------|--|
|      |      | Network generated strains from synthetic mesh of stress points       |
| +    | +    | Experimentally generated data  |
| •    | •    | Additional plane stress data points with slack wrinkling criteria    |
| *    | *    | Additional plane stress data points with wrinkled wrinkling criteria |

**Figure 5-10: Response surface with additional wrinkling criteria data points figure key**



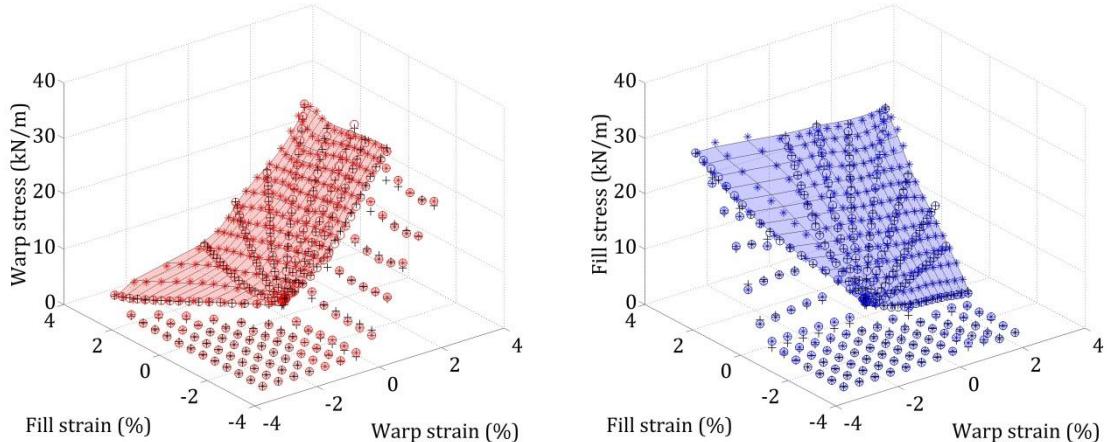
**Figure 5-11: Training data with additional wrinkling criteria data points**

When using the stress criterion (Figure 5-11a), negative maximum stress occurs in combination with a positive minimum strain and a wrinkled state is incorrectly defined as slack, this lead to a significant discontinuity in the surfaces. When using the strain

criterion (Figure 5-11b), due to the effect of Poisson's ratio, a taught state is misjudged as wrinkled where a negative minimum strain may coincide with positive minimum stress. In this context, the combined criterion (Figure 5-11c) is arguably the best method of identifying the state of the fabric and generates a continuous surface of additional training data points.

| Warp | Fill |   |
|------|------|---|
|      |      | Network generated target data                     |
| +    | +    | Experimentally generated target data              |
| *    | *    | Network output from network generated data        |
| ○    | ○    | Network output from experimentally generated data |

**Figure 5-12: Response surface neural network figure key**

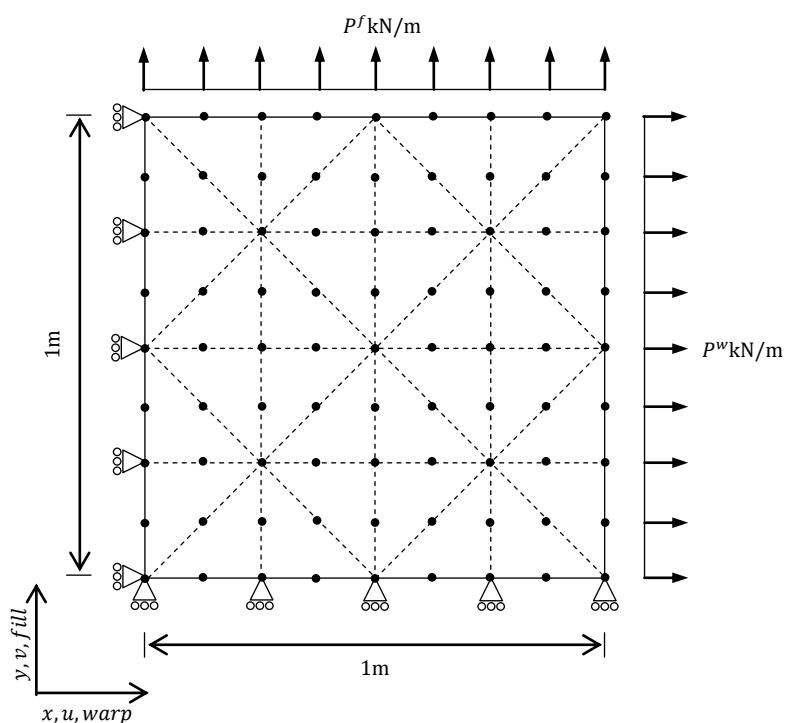


**Figure 5-13: PVC (502S) strain to stress network demonstrating best performance with 'unseen' testing data set ( $R^2 = 0.9978$ )**

Ten networks with 10 nodes in the hidden layer are trained with experimentally generated data and additional wrinkling criteria data. Each network is then tested using the 'unseen' network generated data set (Figure 5-13). The network demonstrating the highest  $R^2$  value when presented with the 'unseen' testing data is selected for use in simulations presented in the following sections. The same process is used to train a 3 input PTFE neural network with additional wrinkling criteria data. The tool developed in Matlab for training response surface style networks is demonstrated in Appendix C. The uncertainty introduced by the variation in response between multiple networks trained with the same training data is explored further in Section 6.2.

### 5.2.2 Patch test

The 1 metre square patch of fabric, discretised into 32 elements with 81 nodes and restrained along two edges by rolling restraints (Section 3.2.1, Figure 5-14), is used to test the model trained above (Section 5.2.1). Uniformly distributed loads (UDL) of varying magnitudes (Table 5-6) are applied along the free edges. Load is applied in the warp material direction,  $P^w$ , aligned with the global  $x$ -direction and in the fill material direction,  $P^f$ , aligned with the global  $y$ -direction.



**Figure 5-14: 32 element patch for validation**

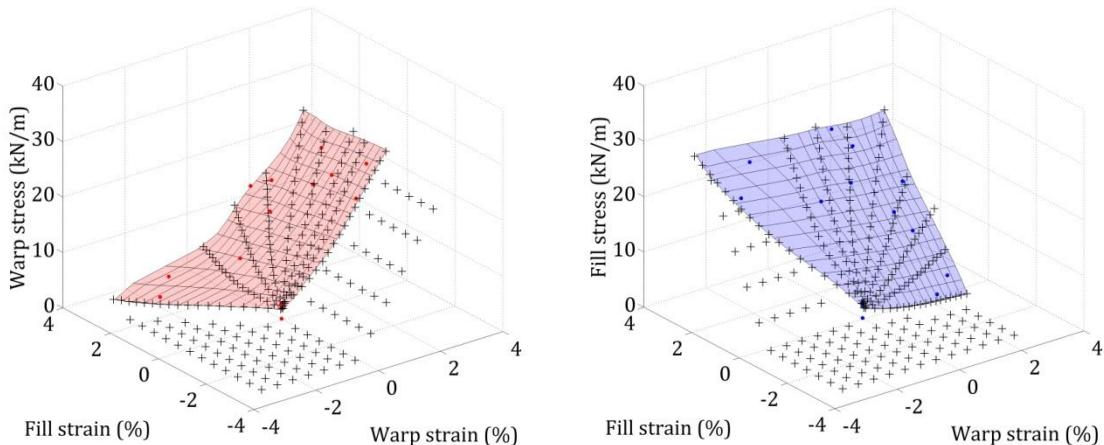
When using a plane stress model, the relationship between stress and strain is linear. Prestress is a constant added to the current stress increment given by the product of the elastic stiffness matrix and the current strain. As discussed above when using the neural network model the prestress is defined using the current strain only.

In pre-processing prior to training the residual strain is removed from the data set at each point where the profile returns to prestress, equation (4.34) (Section 4.3.1). Therefore, an input of zero strain returns stress equal to test prestress. Therefore a simulation with an applied load equivalent to the prestress will lead to zero strain. An applied load below the test prestress will lead to a negative displacement and equivalent negative strain. As described in Section 5.2, in order to apply a prestress level greater than that applied in the training data an initial uniform strain,  $\varepsilon_0$ , is applied to the structure. This strain may eventually be determined through simulation of installation. However, in this study the

stress to strain network, used to generate additional training data, Section 4.3.2.2, is used to find the strain that returns stress equal to desired prestress.

| Applied Load<br>(kN/m) |      | Strain<br>(%) |        | Displacement<br>(m) |         | Stress<br>(kN/m) |        |
|------------------------|------|---------------|--------|---------------------|---------|------------------|--------|
| Warp                   | Fill | Warp          | Fill   | Warp                | Fill    | Warp             | Fill   |
| 2.8                    | 2.8  | -0.041        | -0.027 | -0.0004             | -0.0003 | 2.801            | 2.801  |
| 2.8                    | 20   | -2.202        | 2.272  | -0.022              | 0.022   | 2.739            | 20.456 |
| 20                     | 2.8  | 1.836         | -0.664 | 0.018               | 0.007   | 20.134           | 2.750  |
| 20                     | 15   | 1.563         | 0.784  | 0.016               | 0.008   | 19.845           | 14.771 |
| 15                     | 20   | 0.733         | 1.503  | 0.007               | 0.015   | 14.780           | 19.855 |
| 25                     | 19   | 2.096         | 1.120  | 0.021               | 0.011   | 24.725           | 18.614 |
| 19                     | 25   | 1.140         | 1.962  | 0.011               | 0.019   | 18.638           | 24.720 |
| 17.5                   | 27.5 | 0.821         | 2.429  | 0.008               | 0.024   | 17.090           | 27.278 |
| 8                      | 18   | -0.195        | 1.537  | -0.002              | 0.015   | 7.880            | 18.035 |
| 22                     | 12   | 1.863         | 0.389  | 0.018               | 0.004   | 21.915           | 11.783 |
| 25                     | 5    | 2.314         | -0.471 | 0.023               | -0.005  | 25.119           | 4.888  |
| 5                      | 25   | -1.701        | 2.564  | -0.017              | 0.025   | 4.876            | 25.437 |
| 1                      | 1    | -0.203        | -0.205 | -0.002              | -0.002  | 1.002            | 1.002  |

**Table 5-6: PVC neural network patch simulation results**



**Figure 5-15: PVC (502S) strain to stress network with simulation results.**

The simulation results demonstrate a good fit with the response surfaces constructed from the data used for training, Figure 5-15. From the results, Table 5-6, it is observed that the applied load induces the expected equivalent stress. The small deviations occur due to the change in dimensions of the patch. For example, taking the results for the applied load

of 17.5kN/m in the warp direction and 27.5kN/m. The updated dimensions of the patch at the end of the simulation is given by,

$$x_{dimension} = 1 + \delta_w = 1 - 0.008 = 1.008 \text{ m}$$

$$y_{dimension} = 1 + \delta_f = 1 - 0.024 = 1.024 \text{ m}$$

The analysis stress is agrees with the applied load divided by the updated patch dimensions,

$$\sigma_w = \frac{\text{Warp Applied Load}}{y_{dimension}} = \frac{17.5}{1.024} = 17.090 ,$$

$$\sigma_f = \frac{\text{Fill Applied Load}}{x_{dimension}} = \frac{27.5}{1.008} = 27.282 .$$

The discrepancy between fill stress calculated above and analysis fill stress may be attributed to rounding error as demonstrated by

$$\sigma_f = \frac{\text{Fill Applied Load}}{x_{dimension}} = \frac{27.5}{1.0084} = 27.271.$$

The PTFE network (Figure 5-16, Table 5-7) is much more unstable than the PVC Network. This is due to the extremely high influence of crimp at the more extreme load ratios generating a very steep non-linear surface. It is also extremely difficult to generate meaningful additional data to represent wrinkling behaviour. This is due to discrepancies between the plane stress approximation used to generate the additional data points and the experimental data, particularly at the boundaries of response surface generated by the 0:1 and 1:0 load ratios. Because of this the interface between experimental and additional input data, i.e. strain points which generate stress values below the pre-stress applied to the test specimen, is discontinuous. Nevertheless the results located inside the boundaries of the experimental training data still demonstrate a good fit with the response surfaces generated from the PTFE experimental training data, Figure 5-16.

| Applied Load<br>(kN/m) |      | Strain<br>(%) |        | Displacement<br>(m) |         | Stress<br>(kN/m) |        |
|------------------------|------|---------------|--------|---------------------|---------|------------------|--------|
| Warp                   | Fill | Warp          | Fill   | Warp                | Fill    | Warp             | Fill   |
| 3.5                    | 3.5  | -0.053        | -0.020 | -0.0005             | -0.0002 | 3.501            | 3.502  |
| 3.5                    | 20   | -2.359        | 2.112  | -0.0239             | 0.0209  | 3.428            | 20.490 |
| 20                     | 3.5  | 1.611         | -2.707 | 0.0160              | -0.0274 | 20.560           | 3.445  |
| 20                     | 15   | 0.339         | 0.093  | 0.0034              | 0.0009  | 19.980           | 14.950 |
| 15                     | 20   | -0.375        | 0.914  | -0.0038             | 0.0091  | 14.860           | 20.080 |
| 25                     | 19   | 0.391         | 0.190  | 0.0039              | 0.0019  | 24.950           | 18.930 |
| 19                     | 25   | -0.400        | 1.096  | -0.0040             | 0.0109  | 18.800           | 25.100 |
| 17.5                   | 27.5 | -0.689        | 1.375  | -0.0069             | 0.0137  | 17.260           | 27.690 |
| 8                      | 18   | -0.872        | 1.244  | -0.0088             | 0.0124  | 7.903            | 18.160 |
| 22                     | 12   | 0.690         | -0.385 | 0.0069              | -0.0039 | 22.090           | 11.920 |
| 25                     | 5    | 1.768         | -2.699 | 0.0175              | -0.0274 | 25.700           | 4.914  |
| 5                      | 25   | -2.726        | 2.458  | -0.0276             | 0.0243  | 4.882            | 25.710 |
| 1                      | 1    | -0.107        | -0.210 | -0.0011             | -0.0021 | 1.002            | 1.001  |

Table 5-7: PTFE neural network patch simulation results

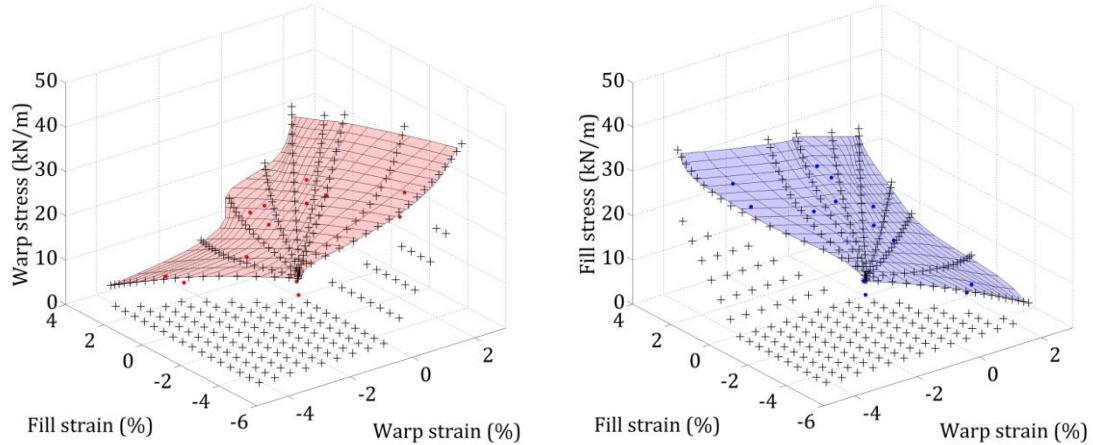


Figure 5-16: PTFE strain to stress network with simulation results.



### 5.2.3 Hypar

The PVC response surface style network model (Section 5.2.2) is used to simulate a realistic structure. The hypar structure, Figure 5-17, under both wind and snow loading, Section 3.2.3, is simulated. The simulation is also undertaken using the equivalent plane stress material model defined in Table 5-5. The results from both sets of simulations are summarised in Table 5-8.

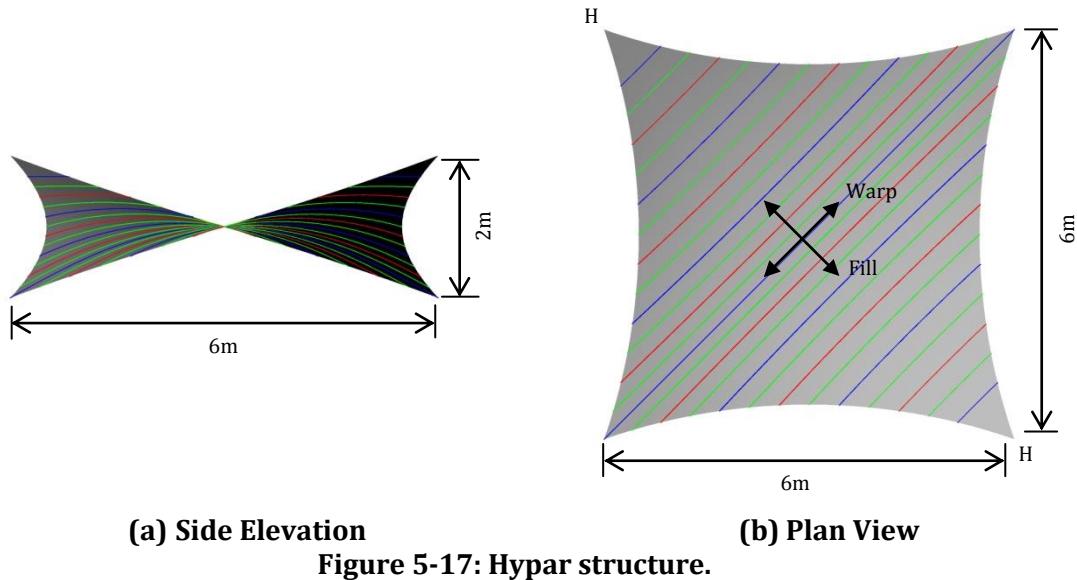


Figure 5-17: Hypar structure.

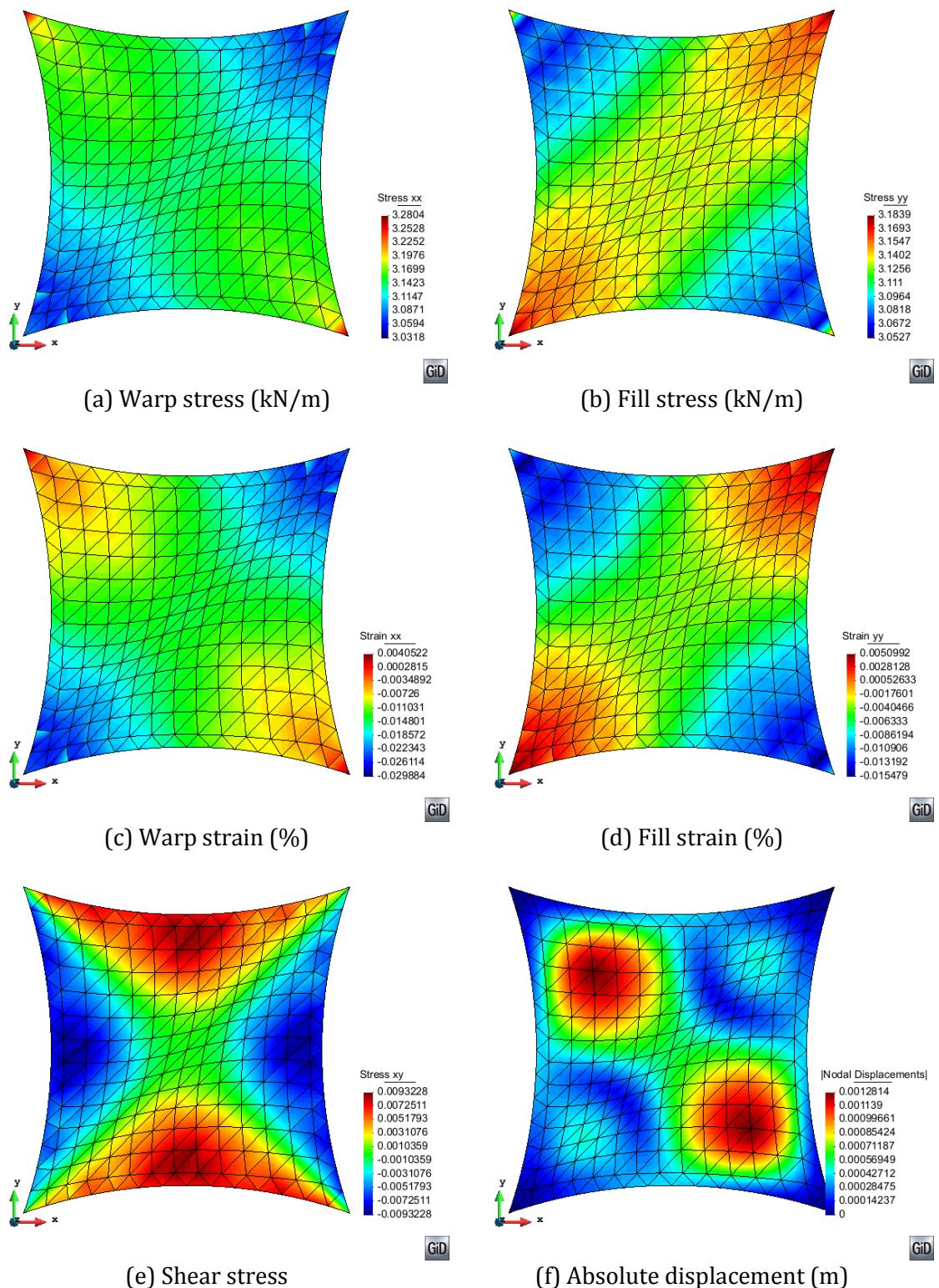
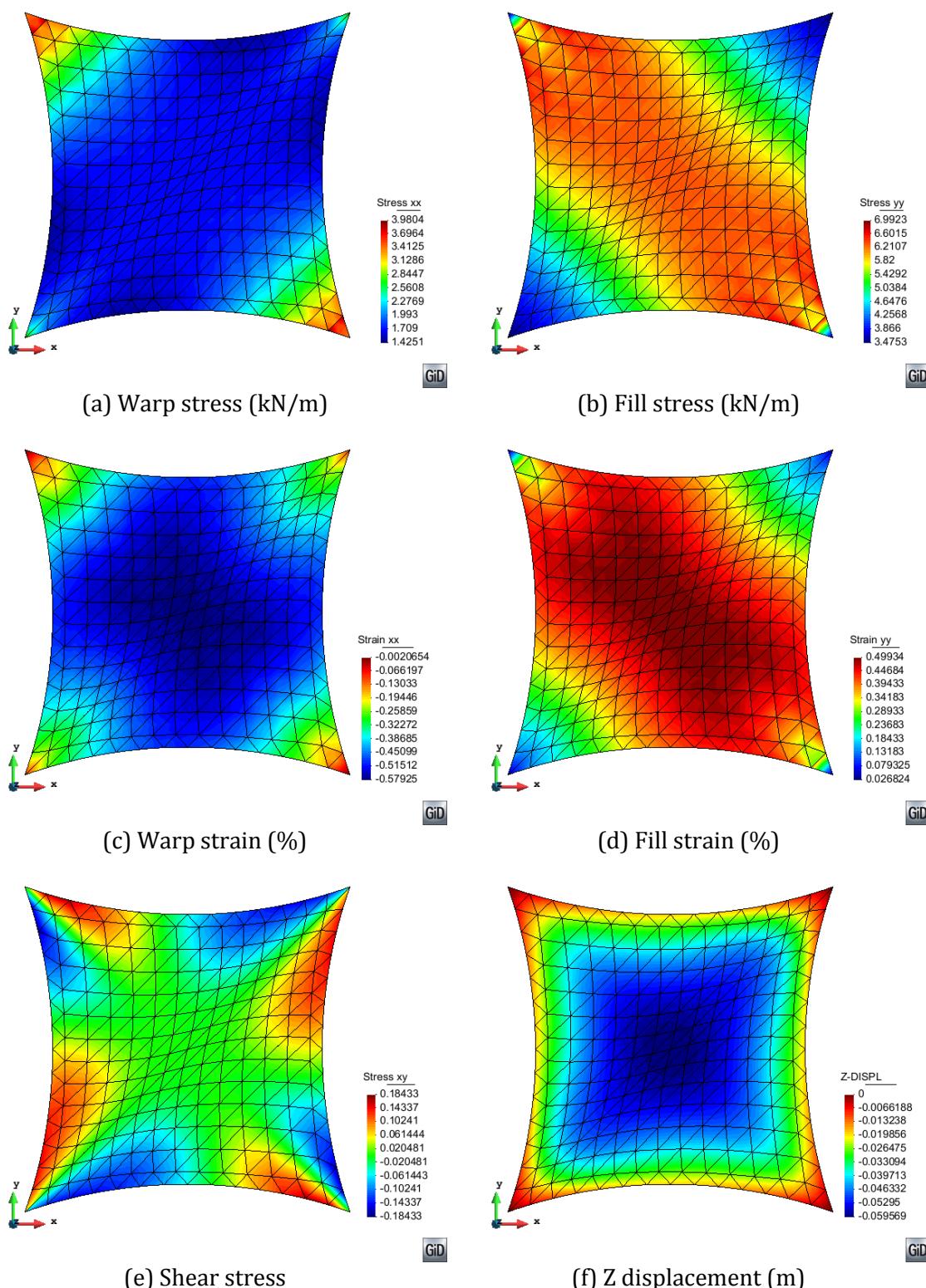
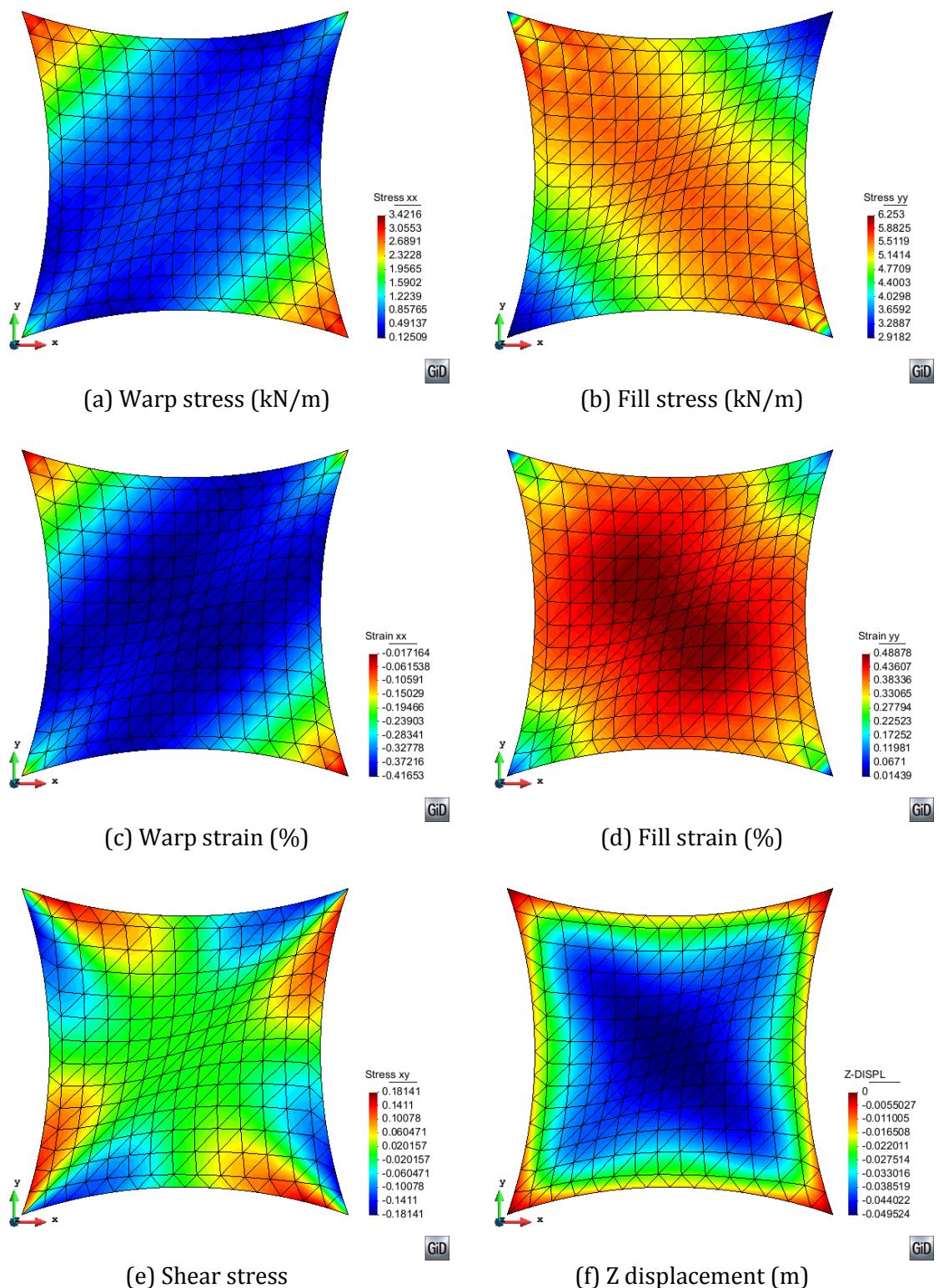


Figure 5-18: Hypar prestress results



**Figure 5-19: Hypar snow loading PVC neural network results**



**Figure 5-20: Hypar snow loading PVC plane stress results**

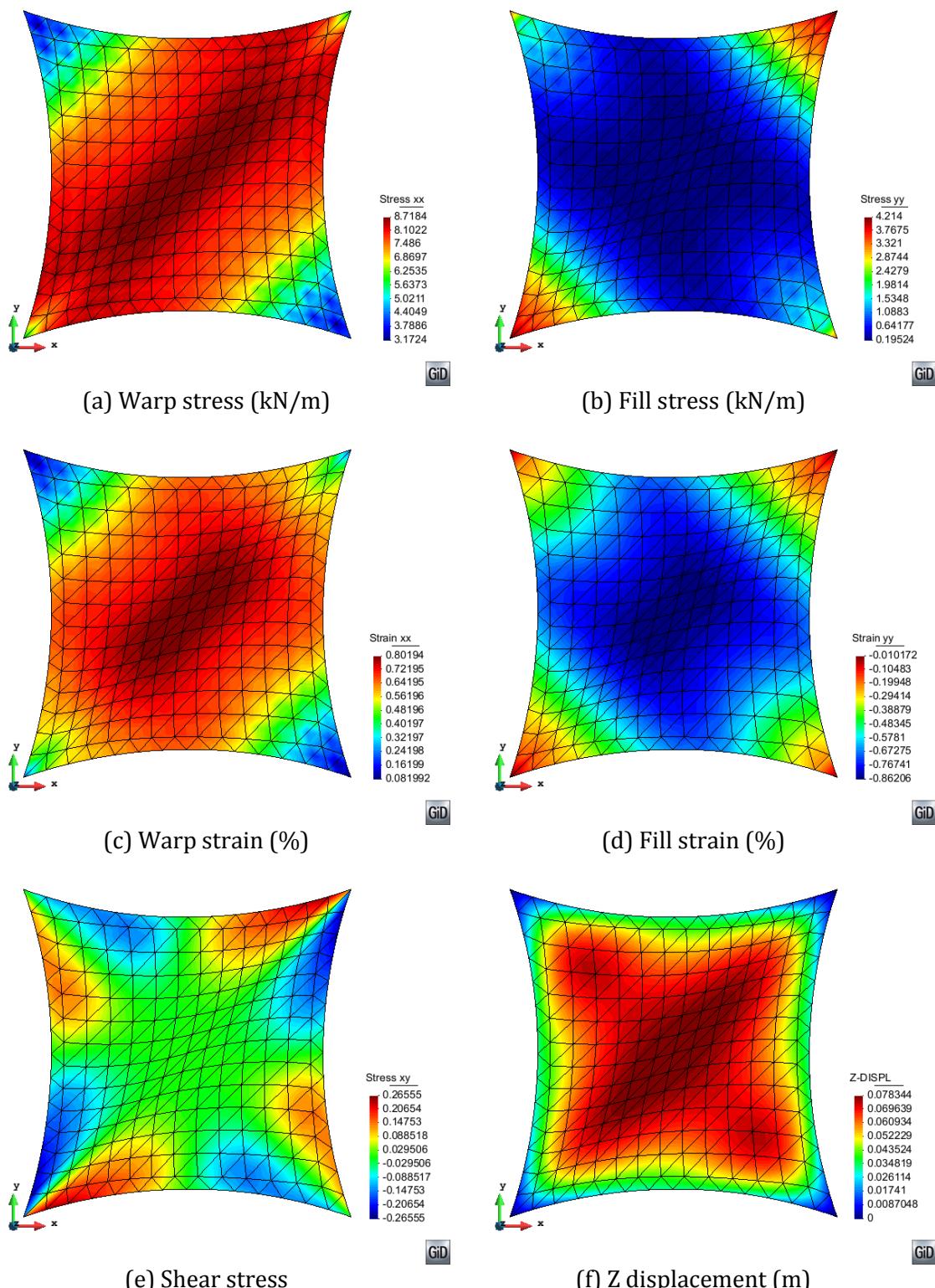
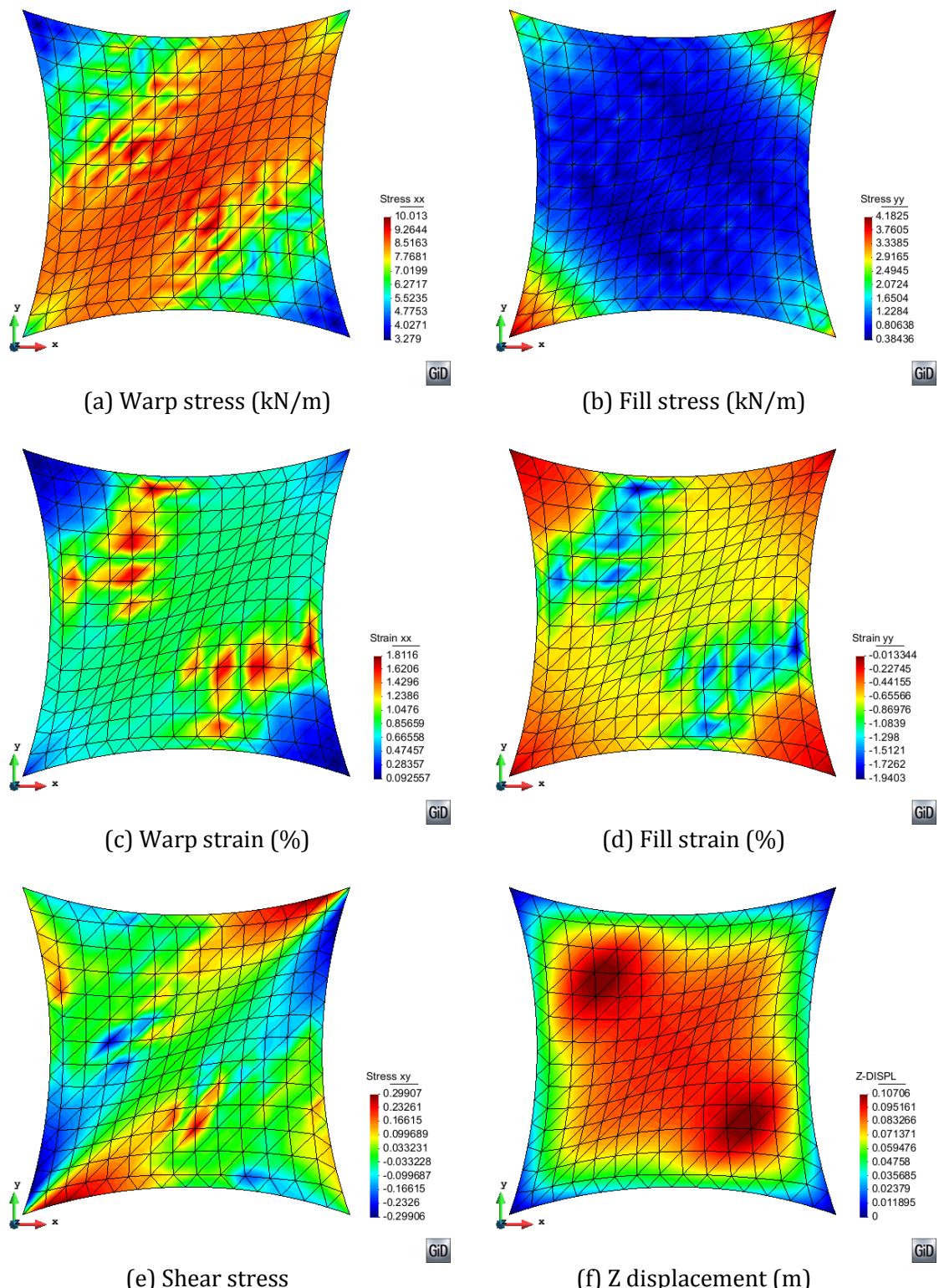


Figure 5-21: Hypar wind loading PVC neural network results



**Figure 5-22: Hypar wind loading PVC plane stress results**

|                             | Warp Stress (kN/m) |      | Fill Stress (kN/m) |       | Displacement (mm) |       |          |
|-----------------------------|--------------------|------|--------------------|-------|-------------------|-------|----------|
|                             | Max                | Min  | Max                | Min   | z                 |       | Absolute |
|                             |                    |      |                    |       | Max               | Min   | Max      |
| Prestress<br>(Plane Stress) | 3.00               | 3.00 | 3.00               | 3.00  | 0                 | 0     | 0        |
| Prestress<br>(Network)      | 3.28               | 3.03 | 3.18               | 3.05  | 1.2               | -0.4  | 1.3      |
| Snow Load<br>(Network)      | 3.98               | 1.43 | 6.99               | 3.48  | 0                 | -59.6 | 59.6     |
| Snow Load<br>(Plane Stress) | 3.42               | 0.13 | 6.25               | 2.92  | 0                 | -49.5 | 49.5     |
| Wind Load<br>(Network)      | 8.72               | 3.17 | 4.21               | 0.20  | 78.3              | 0.0   | 78.4     |
| Wind Load<br>(Plane Stress) | 10.12              | 3.23 | 4.18               | -0.32 | 107.1             | 0.0   | 110.0    |

**Table 5-8: Hypar analysis results summary**

The prestress simulation using the network material model generates stress results that deviate away from the applied prestress. The discrepancy is caused by smalls discrepancies between the network model output when presented with the initial strain and the pre-stress value used in the form-finding process. However, the deviations are small and the resulting absolute displacement is 1.3mm.

Under snow loading the stress distributions in both material directions follow a similar pattern. The maximum stress given by the neural network model is slightly greater than that given by the plane stress material model. The minimum stress given by the neural network is over 1kN/m greater in the warp direction than the plane stress material model. The maximum absolute displacement is approximately 10mm greater in the neural network simulation.

There are similarities between the stress pattern generated by both material models. The minimum warp stress and maximum fill stress produced by the simulations are within 0.1kN/m. However, the neural network material model produces a maximum warp stress that is 1.4kN/m greater than the plane stress material model results. The maximum absolute displacement is approximately 30mm greater.

During the plane stress simulation of the hypar with applied wind loading the wrinkling procedure was triggered. However, the negative stress remains after convergence. The resulting stress distribution is uneven suggesting that the simulation was unstable. The

neural network material model appears to generate a more stable analysis. When using the neural network material model the final response of the structure does not fall within the wrinkled or slack zone of the material response surface. However, the fill stress does reach levels lower than the applied pre-stress.

The stability of the analysis is potentially due to the continuous response surface, represented by the neural network material model that includes modified behaviour in the wrinkling-slack zone, Section 5.2.1. During the finite element analysis it is probable that areas of the structure undergo stresses and strains close to or within the wrinkled or slack regions of the material response surface. In the neural network material model the stiffness in the wrinkled-slack region is reduced avoiding the build up of significant compressive stress. As the analysis progresses the response may move back out of the wrinkled or slack region eventually converging to a smooth final result.

In comparison the plane stress analysis the wrinkling procedure is implemented in order to remove erroneous compressive stresses, Section 3.1.13. The structure is assessed using mixed wrinkling criterion, equation (3.141) to (3.143), after equilibrium has been reached. Regions of compressive stress have developed at this stage. Where wrinkling or slack regions are deemed to exist the elastic modulus is reduced and the analysis is re-run, this results in potentially large discontinuities in the material response between gauss points that are assessed as wrinkle and unwrinkled. In the case of the wind loading analysis, Figure 5-22, these discontinuities lead to an unstable analysis and eventual convergence to a uneven stress distribution.

### 5.3 History neural network implementation

In order to implement the history neural network historical stress and strain information is required. Load is applied to the model in steps and equilibrium is found at each step.

The expression for equilibrium of the system becomes,

$$\frac{\partial \Pi_{ps}}{\partial D_{1 \rightarrow n}} = \{R\} = \sum_{i=1}^{i=m} \left[ \int_V [B]^T [\sigma^n] dV \right]_i - \sum_{i=1}^{i=m} \left[ \int_V [B]^T [\sigma^{n-1}] dV \right]_i + \{P^n\} = 0. \quad (5.53)$$

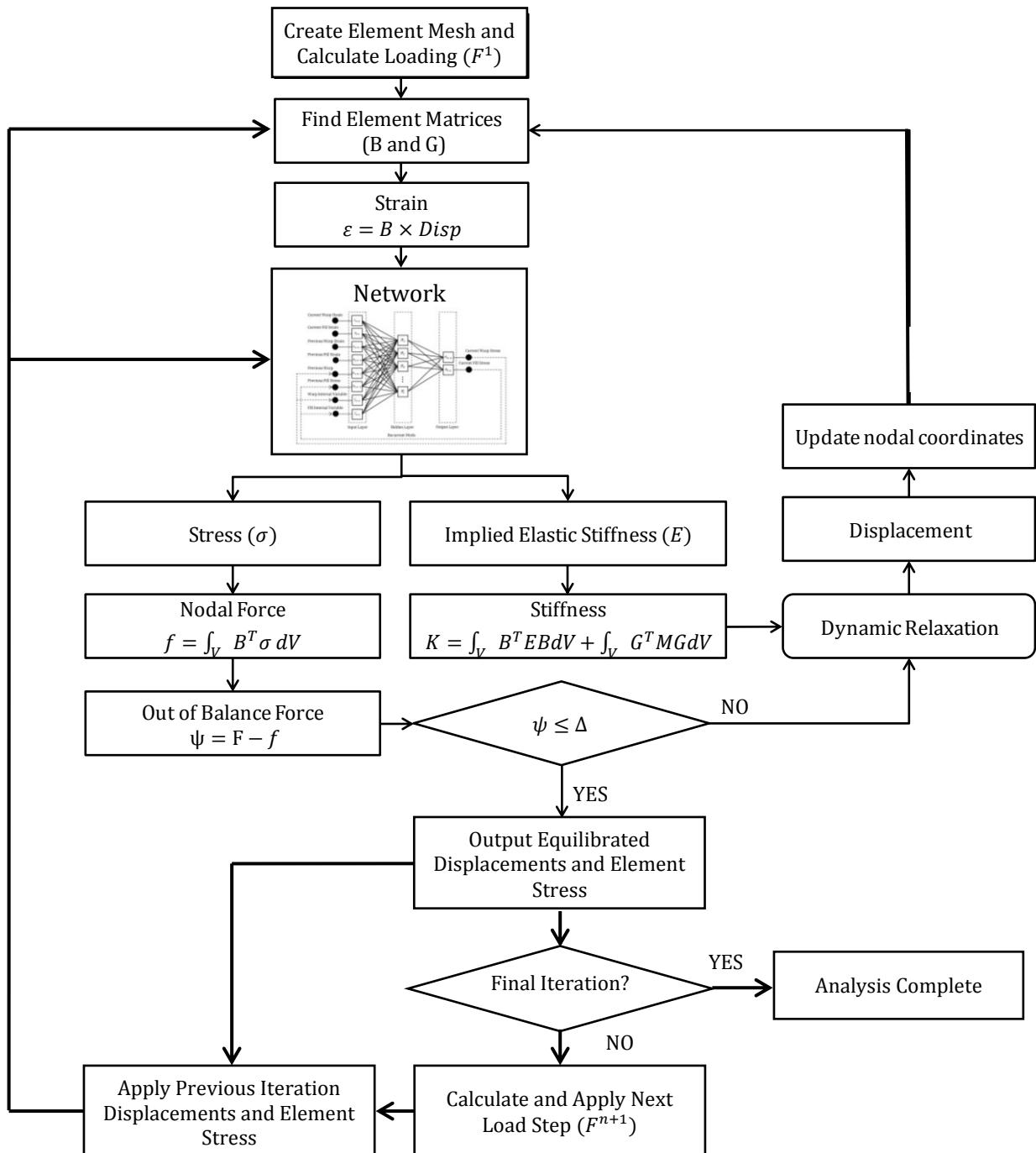
where  $P^n$  is the current load increment applied to each node,  $\sigma^n$  is the current stress at each gauss point in an element and  $\sigma^{n-1}$  is the stress at each gauss point from the previous load step. The history network material model equation **Error! Reference source not found.** for stress may be represented by,

$$[\sigma^n] = N([\varepsilon^n] + [\varepsilon_0], [\varepsilon^{n-1}] + [\varepsilon_0], [\sigma^{n-1}], \zeta^n). \quad (5.54)$$

Current strain is represented by  $\varepsilon^n$ , strain from the previous load step by  $\varepsilon^{n-1}$ , stress from the previous load step by  $\sigma^{n-1}$ , the current internal variables by  $\zeta^n$  and initial strain by  $\varepsilon_0$ . As with the response surface network a constant initial strain,  $\varepsilon_0$ , is applied to the fabric in order induce required prestress. Initial previous stress and strain values are also required for the first load step. A patch test with iterative loading may be used to determine an initial strain value with corresponding previous stress and strain values that corresponds to the correct initial stress level.

Equations (5.50) to (5.52) are used to calculate current strain, the elastic stiffness matrix and geometric stiffness matrices, respectively. Historical stress and strain values are updated at the end of each load step once equilibrium has been reached. The historical stress and strain values are not updated between kinetic energy peaks in the dynamic relaxation algorithm. During the iterative process the model undergoes pseudo oscillations about the equilibrium state. This would generate a false material response and potentially cause residual strain build up.

The implementation of the history network within the iterative finite element analysis procedure is summarised in Figure 5-23.

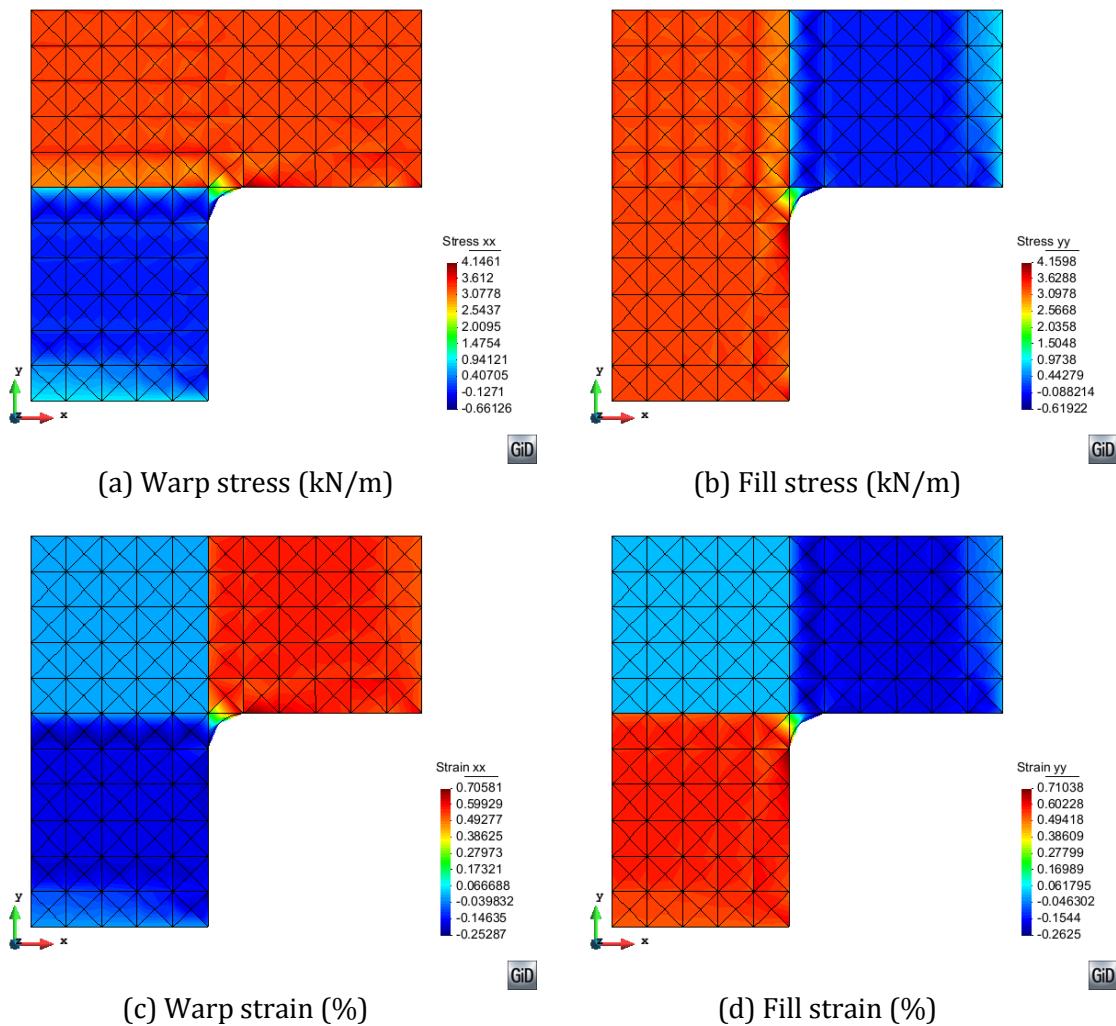


**Figure 5-23: Iterative finite element analysis procedure with history neural network material model**

### 5.3.1 Biaxial test

An attempt was made to validate the history neural network via simulation of the biaxial test used to generate the training data (Figure 5-14). In order to accurately model the biaxial test specimen arm slits that have unrestrained edges need to be included in the mesh. The neural network material model becomes unstable if used to simulate the

unrestrained slits. The use of a plane stress model for simulation of the arms and a network model for the central region has been explored. However, discontinuities between the two models leads to invalid results. The stress is reasonably consistent but the strain values are not inconsistent. Loading from zero is also impossible due to the constraints of the available testing data. Therefore an initial stress is required, for example 2.8kN/m for the PVC fabric. In order to apply this stress to the plane stress arms a prescribed pre stress may be applied. However, due to the unrestrained arm edges this leads to an erroneous out of balance force across the arm.



**Figure 5-24: Biaxial test specimen, applied load 4kN/m**

Due to the prohibitive complexities of simulating the biaxial cruciform with unrestrained edges, the 1m square patch used to investigate the response surface network, (Section 5.2.2), is employed to investigate the performance of the network in simulation of fabric response with load history effects. Iterative loading is applied to the unrestrained edges in order to induce stress equal to that in the biaxial profile. The simulation stress and strain

results may then be directly compared to the experimental results in order to assess to accuracy of the simulation.

The effect of load step and training data selection is investigated. A total of six network training data sets are generated, three use 30 cycles from each of the 5 unique experimental profiles and three use 60 cycles form each of the experimental PVC profiles, Chapter 4. The first of three data sets uses data reduced by a factor of 0.1, i.e. 10% of the data is used, the second a factor 0.15 and the third 0.2. Each training data set is used to train 20 randomly initiated networks. Each trained network is used in three simulations of cycle 6 of the 1:1 PVC experimental profile. Loading for each simulation is derived from stress points taken from the experimental profile reduced using reduction factors of 0.1, 0.15 and 0.2 respectively.

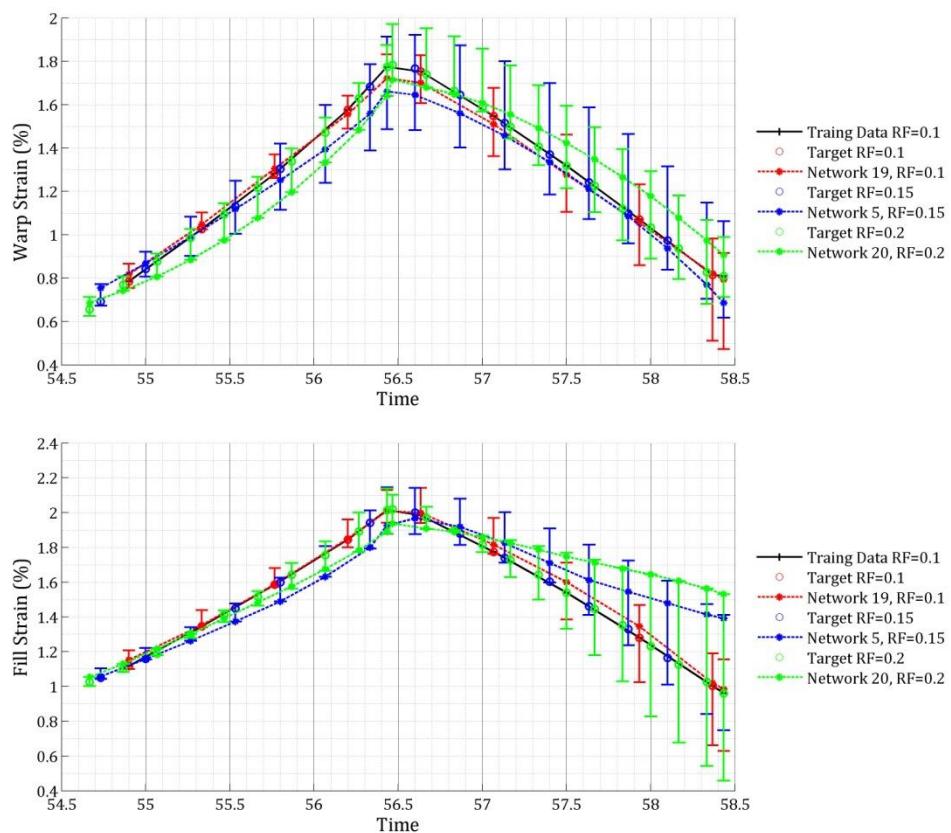
Mean absolute error between network simulation output and experimental results is calculated for each network simulation. The lowest mean absolute error, obtained from both warp and fill direction results, from each network and analysis set is reported in Table 5-9. Mean absolute percentage error (MAPE) is used as opposed to mean squared error as the effect of small values of strain is negated and a clearer impression of the level of accuracy can be gained. The number of network simulations that failed is also reported. A network simulation is deemed to have failed when a mean absolute percentage error above 15% is calculated in either material direction.

|                                 | Reduction Factor 0.1  |             |             |             |             |             |
|---------------------------------|-----------------------|-------------|-------------|-------------|-------------|-------------|
|                                 | 60 Cycles             |             |             | 30 Cycles   |             |             |
|                                 | 0.1                   | 0.5         | 0.2         | 0.1         | 0.15        | 0.2         |
| <b>Testing Reduction Factor</b> | <b>0.1</b>            | <b>0.5</b>  | <b>0.2</b>  | <b>0.1</b>  | <b>0.15</b> | <b>0.2</b>  |
| <b>No. Failed Networks</b>      | 10                    | 9           | 18          | 13          | 10          | 17          |
| <b>Best Network</b>             | 19                    | 5           | 20          | 3           | 9           | 1           |
| <b>MAPE Strain Warp (%)</b>     | <b>2.27</b>           | 5.01        | 8.43        | <b>2.38</b> | 5.41        | 10.61       |
| <b>MAPE Strain Fill (%)</b>     | <b>1.88</b>           | 10.95       | 13.04       | <b>6.15</b> | 9.18        | 11.18       |
| <b>MAPE Stress Warp (%)</b>     | <b>0.44</b>           | 0.51        | 0.55        | <b>0.46</b> | 0.52        | 0.58        |
| <b>MAPE Stress Fill (%)</b>     | <b>0.47</b>           | 0.49        | 0.56        | <b>0.48</b> | 0.50        | 0.60        |
|                                 | Reduction Factor 0.15 |             |             |             |             |             |
|                                 | 60 Cycles             |             |             | 30 Cycles   |             |             |
|                                 | 0.1                   | 0.15        | 0.2         | 0.1         | 0.15        | 0.2         |
| <b>No. Failed Networks</b>      | 19                    | 9           | 1           | 18          | 10          | 3           |
| <b>Best Network</b>             | 4                     | 3           | 13          | 7           | 10          | 10          |
| <b>MAPE Strain Warp (%)</b>     | 10.98                 | <b>5.71</b> | 4.13        | 7.52        | <b>3.54</b> | 3.58        |
| <b>MAPE Strain Fill (%)</b>     | 3.86                  | <b>3.60</b> | 6.79        | 7.04        | <b>3.79</b> | 8.24        |
| <b>MAPE Stress Warp (%)</b>     | 0.43                  | <b>0.51</b> | 0.51        | 0.48        | <b>0.51</b> | 0.60        |
| <b>MAPE Stress Fill (%)</b>     | 0.42                  | <b>0.61</b> | 0.63        | 0.54        | <b>0.55</b> | 0.61        |
|                                 | Reduction Factor 0.2  |             |             |             |             |             |
|                                 | 60 Cycles             |             |             | 30 Cycles   |             |             |
|                                 | 0.1                   | 0.15        | 0.2         | 0.1         | 0.15        | 0.2         |
| <b>No. Failed Networks</b>      | 20                    | 19          | 11          | 20          | 18          | 9           |
| <b>Best Network</b>             | -                     | 17          | 14          | -           | 3           | 13          |
| <b>MAPE Strain Warp (%)</b>     | -                     | 5.71        | <b>3.95</b> | -           | 5.88        | <b>6.43</b> |
| <b>MAPE Strain Fill (%)</b>     | -                     | 5.51        | <b>1.99</b> | -           | 2.43        | <b>3.28</b> |
| <b>MAPE Stress Warp (%)</b>     | -                     | 0.55        | <b>0.50</b> | -           | 0.49        | <b>0.54</b> |
| <b>MAPE Stress Fill (%)</b>     | -                     | 0.65        | <b>0.61</b> | -           | 0.56        | <b>0.68</b> |

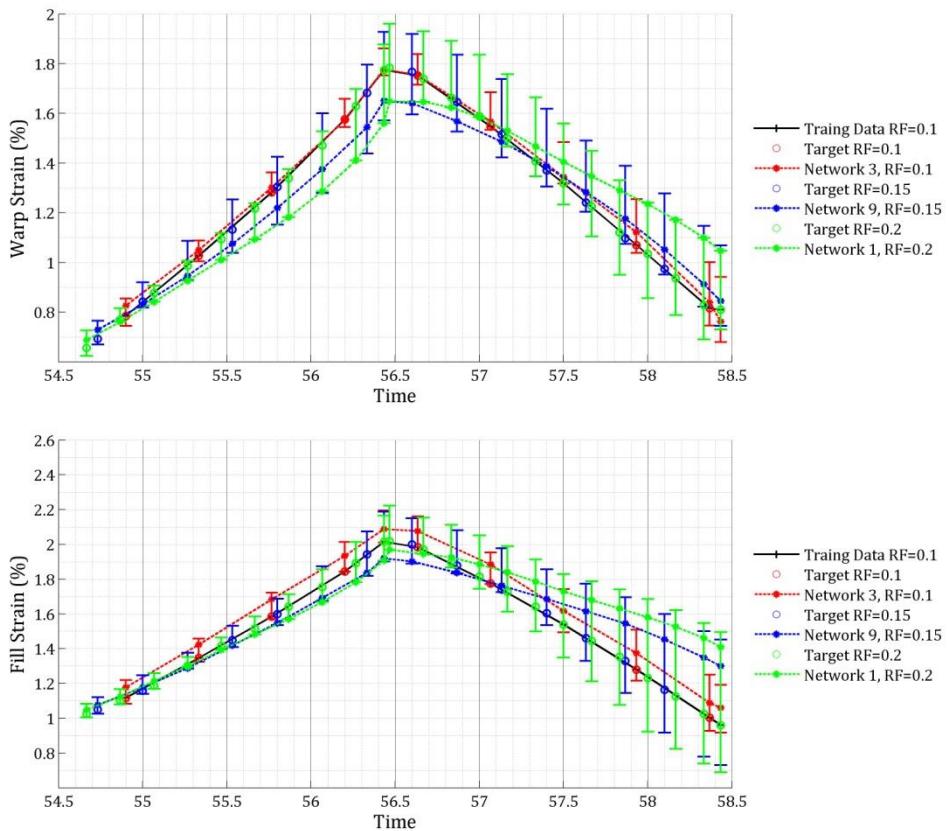
**Table 5-9: Mean absolute percentage error between experimental results and patch simulation of cycle 6 of 1:1 biaxial load profile (Chapter 4)**

Failure rates across each network group indicate that the training data load step has a significant effect on network performance. The minimum failure rate across a network group is 1.5% and the maximum 100%. Lower failure rates are observed when the load step simulated is the same or lower than that used in training. The highest failure rates occur when a network is used to simulate a greater load step. This is highlighted by the 100% failure rate of networks trained with data reduced by a factor of 0.2 when used to simulate experimental data reduced by a factor of 0.1.

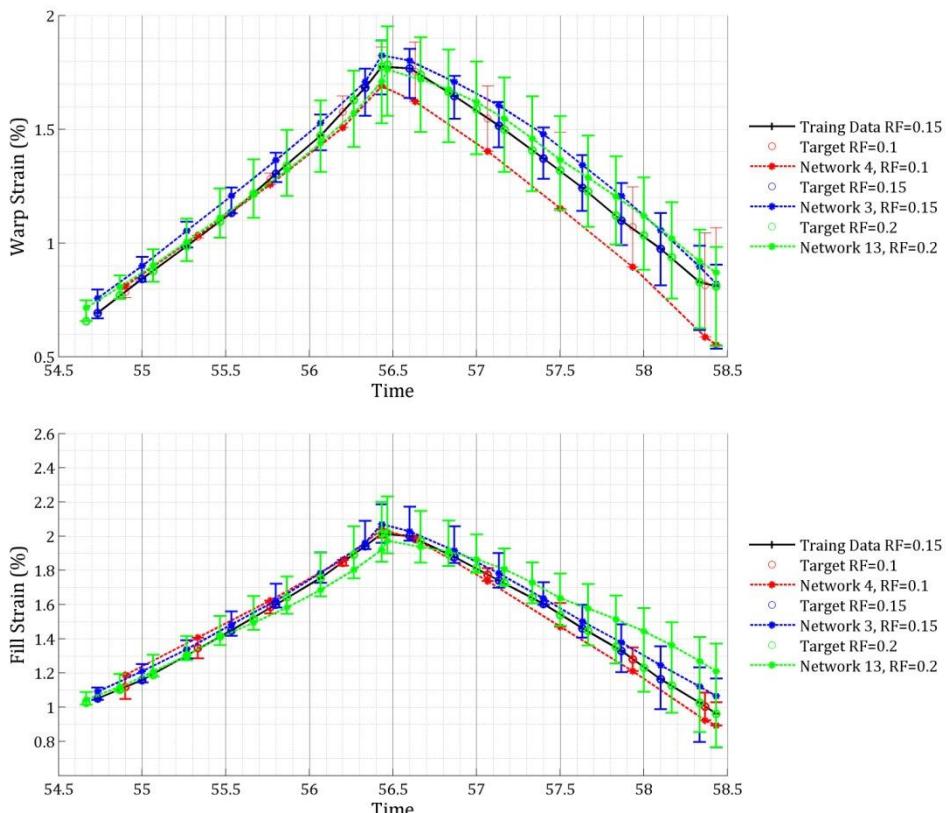
Graphical representations of the simulation set results are shown in Figure 5-25 to Figure 5-30. The training data is shown as a solid black line with data points indicated by +. Simulation of experimental data with a 0.1 reduction factor is shown in red, with a 0.15 reduction factor in blue and with a 0.2 reduction factor in red. Maximum and minimum error across all acceptable networks in an analysis set is depicted by error bars. Where only one network has been deemed acceptable the upper error bar is given by the absolute error and the lower error bar by negative absolute error. The strain results of the network simulation achieving the lowest combined absolute percentage error across both material directions (Table 5-9) is plotted using a dashed line with data points indicated by \*.



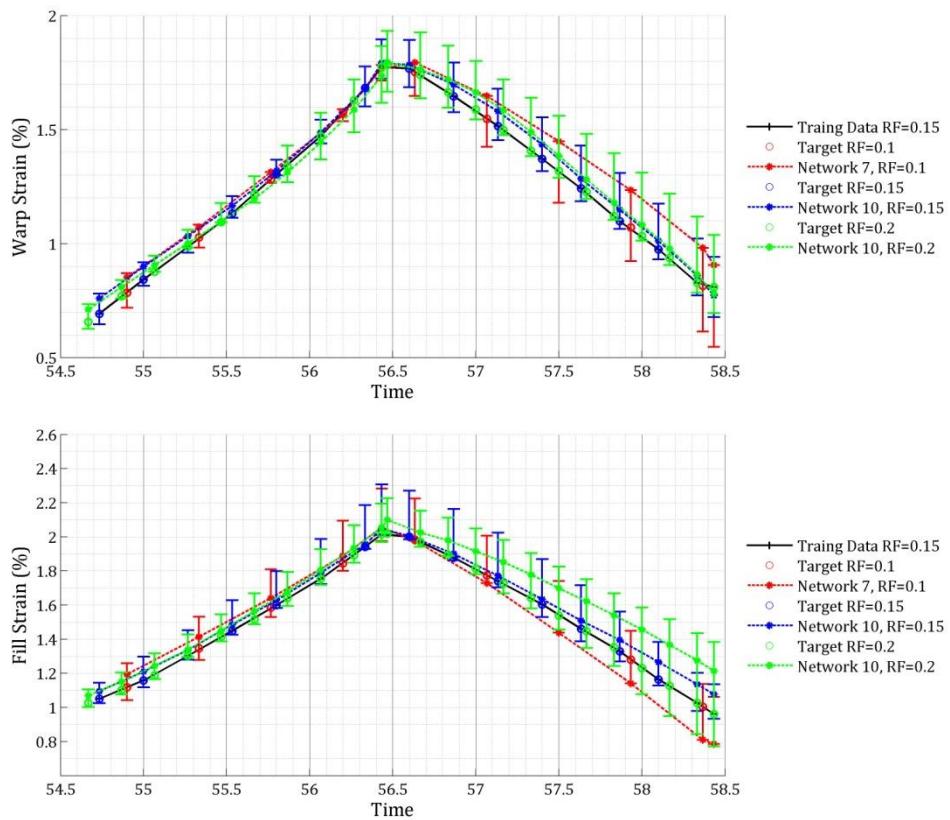
**Figure 5-25: Patch Simulation Results of Cycle 6 of 1:1 Profile Generated by a Load History Network Trained Using 60 Cycles and a Reduction Factor of 0.1.**



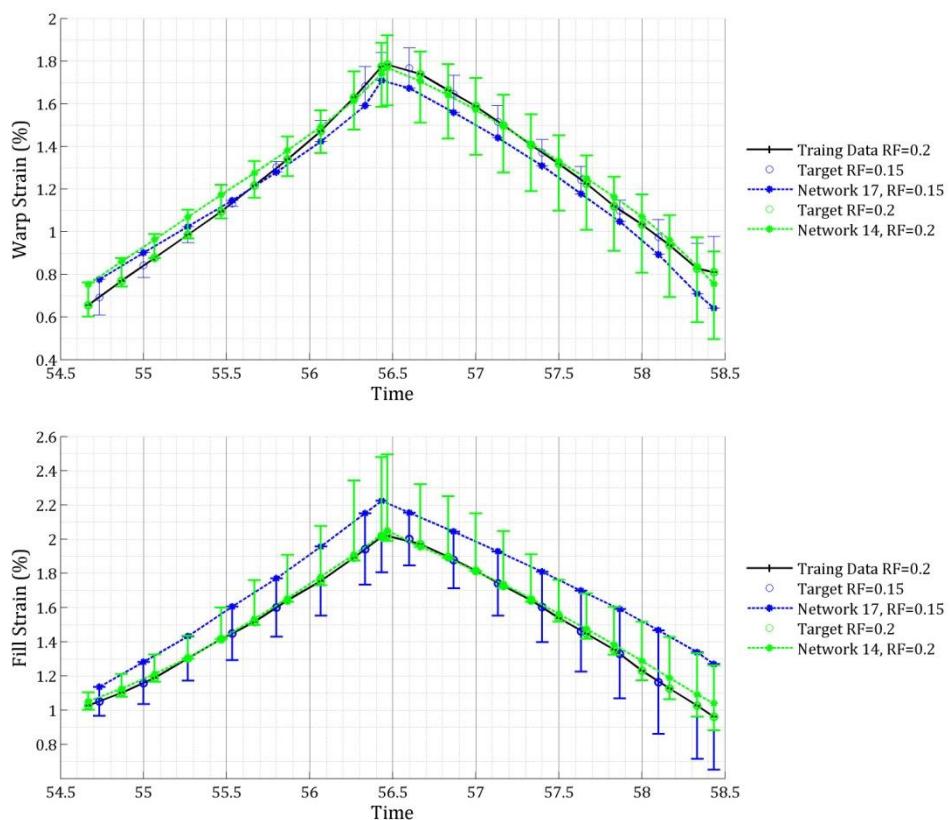
**Figure 5-26: Patch Simulation Results of Cycle 6 of 1:1 Profile Generated by a Load History Network Trained Using 30 Cycles and a Reduction Factor of 0.1.**



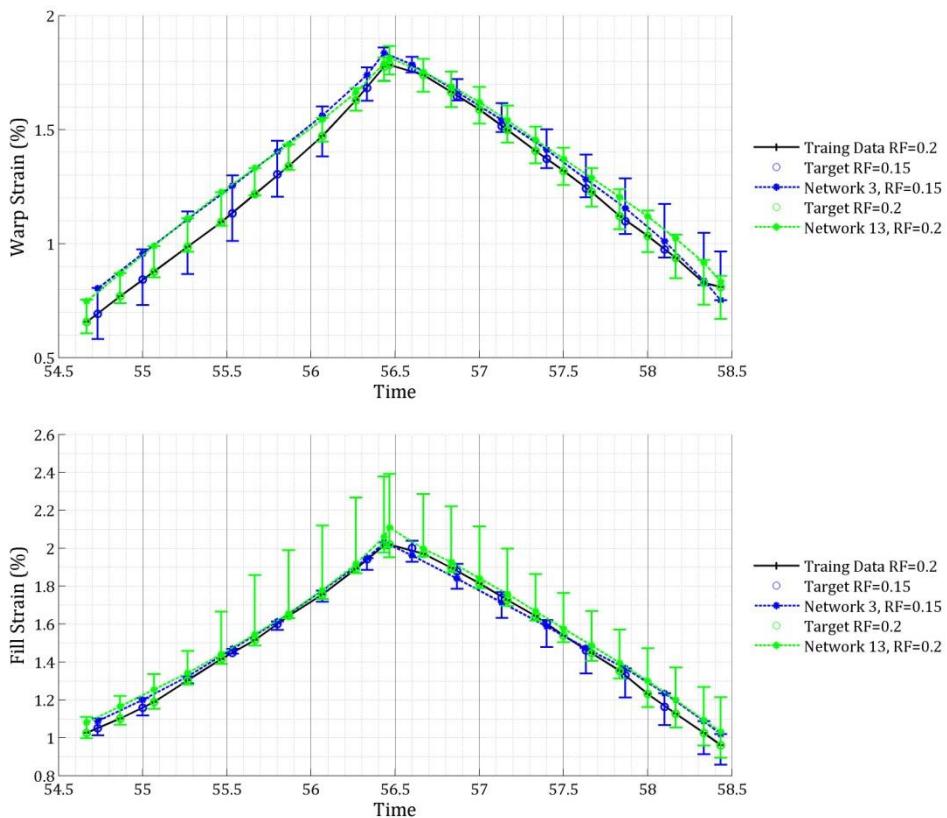
**Figure 5-27: Patch Simulation Results of Cycle 6 of 1:1 Profile Generated by a Load History Network Trained Using 60 Cycles and a Reduction Factor of 0.15.**



**Figure 5-28: Patch Simulation Results of Cycle 6 of 1:1 Profile Generated by a Load History Network Trained Using 30 Cycles and a Reduction Factor of 0.15.**



**Figure 5-29: Patch Simulation Results of Cycle 6 of 1:1 Profile Generated by a Load History Network Trained Using 60 Cycles and a Reduction Factor of 0.2.**



**Figure 5-30: Patch Simulation Results of Cycle 6 of 1:1 Profile Generated by a Load History Network Trained Using 30 Cycles and a Reduction Factor of 0.2.**

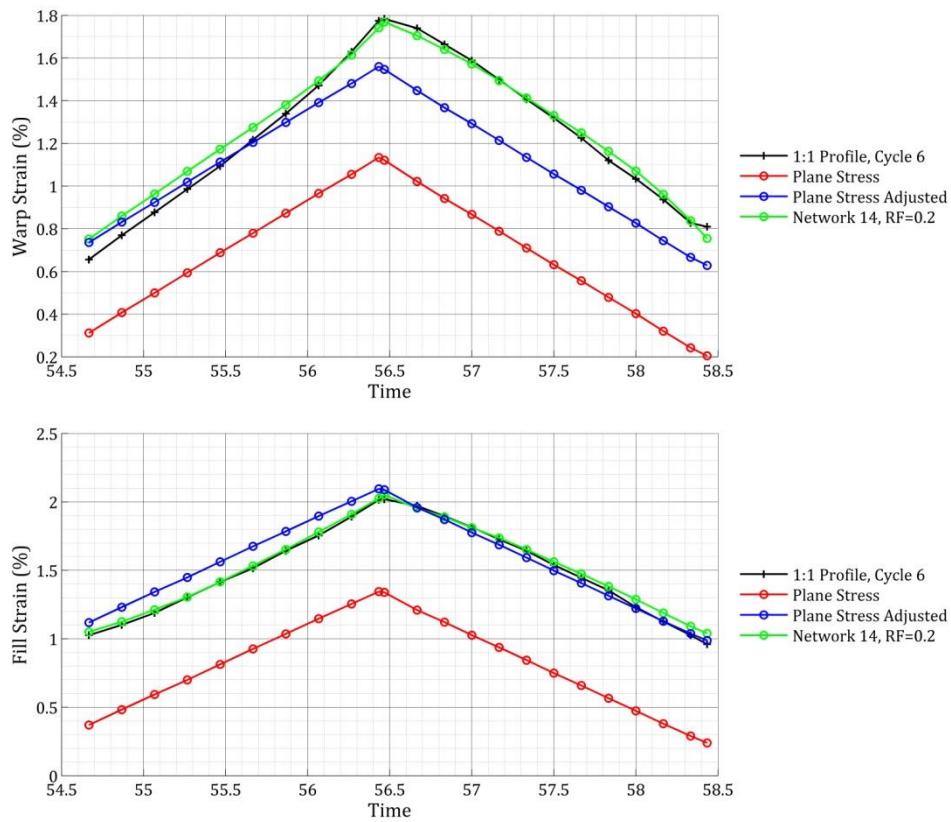
A high degree of variation between Network simulation results is seen. Initially looking at only the networks with the lowest mean percentage error it can be seen that absolute error generally increases throughout the iterative analysis. This leads to actual error greater than 15% at the final stages of the analysis. This trend is also confirmed across the full network sets with widening error bars as the analysis progresses, particularly in the case of the networks trained with data reduced by a factor of 0.1 or 0.15. The networks trained with data reduced by a factor of 0.2 are by far the least successful. However, this is largely due to the effect of greater load step on Network failure. It is likely that were the network to be used to simulate a smaller load step than that of the training data they would out-perform the networks trained with larger load steps. This indicates the development of the training data set is key in generating a successful Neural Network material model. Extensive testing is required to confirm that the trained network is suitable for the analysis task for which it will be employed.

For comparison a plane stress material model is fit to both loading and unloading portions of cycles 1 to 15 of the 1:1 profile by minimising error between experimental strain and model strain. The resulting elastic constants are given in Table 5-10 and were used in the simulation of cycle 6 of the 1:1 profile. As in Figure 5-25 to Figure 5-30 the plane stress

simulation is plotted with the experimental data. The results from the simulation using Network 14 from the network set trained using 60 cycles and a reduction factor of 0.2 is also plotted. The raw unadjusted simulation strain results, plotted in red, do not incorporate an applied prestress as input. The strain output is the result of the full applied load but cannot account for residual strain. The adjusted simulation result, plotted in blue, incorporated the prestress applied to the fabric as input. Therefore, the strain output is a result of the applied load minus the prestress, only the loading above prestress induces strain. The strain result is then adjusted by adding the initial strain associated with prestress taken from the first experimental data point of the cycle.

| $E_w$     | $E_f$     | $v_{wf}$ | $v_{fw}$ | $G_{wf}$ |
|-----------|-----------|----------|----------|----------|
| 804.6kN/m | 742.6kN/m | 0.37     | 0.37     | 30kN/m   |

**Table 5-10: PVC plane stress elastic constants**



**Figure 5-31: Patch Simulation Results of Cycle 6 of 1:1 Profile Generated by a Plane stress elastic stiffness matrix fit to Cycles 1 to 15 of 1:1 Profile.**

Clearly the adjusted plane stress simulation out performs the unadjusted plane stress simulation and shows a particularly good fit with the fill direction experimental strains. This is due to the reasonably linear response of the fabric in this material direction. In the warp direction the adjusted plane stress model demonstrates a similar fit to the

experimental data as the network model in the early stages of the loading portion of the cycle. However, the plane stress simulation significantly under-estimates the peak strain value and the unloading portion of the cycle. The network model is capable of replicating the non-linear material response in the warp direction whereas the linear plane stress model is not. The adjusted plane stress simulation result is made possible by the availability of the initial strain data. In a typical plane stress simulation of a more complex realistic structure this information would not be unavailable. The unadjusted result is more representative of the typical results generated by a plane stress simulation.

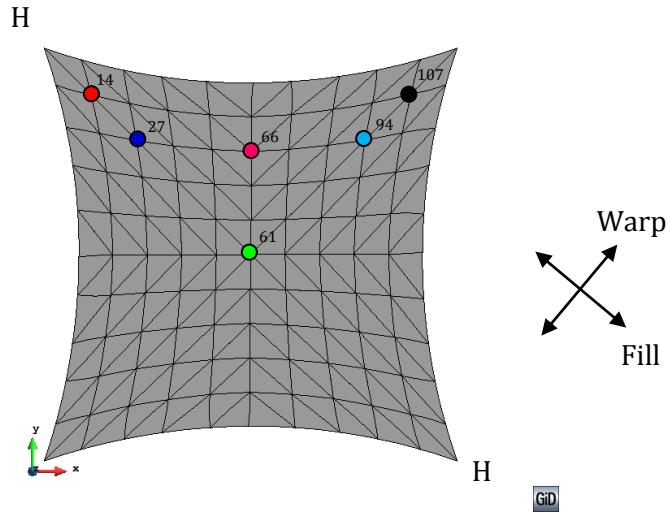
It is noted that the stress mean percentage error remains below 1% across all analysis sets, Table 5-9. This is due to the fact that the stress is prescribed by the loading and therefore in order for the simulation to reach convergence the stress in the patch must be in equilibrium with the applied load. The error is caused by the changing dimensions of the patch, hence the accuracy of the strain result has as an effect on the accuracy of the stress result. However, a degree of error will always remain.

### 5.3.2 Hypar simulation

The network with the lowest mean absolute percentage error in the network set trained with 60cycles of each profile reduced by a factor of 0.1 (Network 19) is used in the simulation of a hypar structure. The dimensions of the hypar are the same as those described in Section 3.2.3 and used in Section 5.2.3 (Figure 3.21). However, the fabric prestress is increased from 3kN/m to 5kN/m and the cable prestress from 30kN to 50kN. This is done in order to increase the likelihood that the simulation stresses will fall within the range of the network training data, i.e. above 2.8kN/m. However, the increase in prestress causes instability during the soap film form finding process with geodesic strings on the original initial mesh (Figure 3.22). Therefore, a more coarse symmetrical mesh comprising 200 elements is used. This mesh is more stable and reaches an equilibrium state with a uniform stress of 5kN/m.

Wind and snow load cases are simulated (Section 3.2.3 and 5.2.3). The loading is applied iteratively in 11 equal steps starting at 0kN/m<sup>2</sup> up to a maximum load of 0.6kN/m<sup>2</sup> at step 6 of for the snow load case and 1kN/m<sup>2</sup> for the wind before returning to 0kN/m<sup>2</sup> at step 11. GiD graphical output for iterations 1, 6 and 11 are shown in Figure 5-33 to Figure 5-35 for the snow case and Figure 5-37 to Figure 5-39 for the wind case. Iteration 1 is shown in order to provide a clear comparison for iteration 11. The result contours are standardised across all of the GiD figures.

The form found mesh is shown in Figure 5-32 with selected nodes highlighted by coloured points. The colours coincide with the colours used in the nodal result plots Figure 5-36 and Figure 5-40.



**Figure 5-32: Form-found mesh with selected nodes for result plotting.**

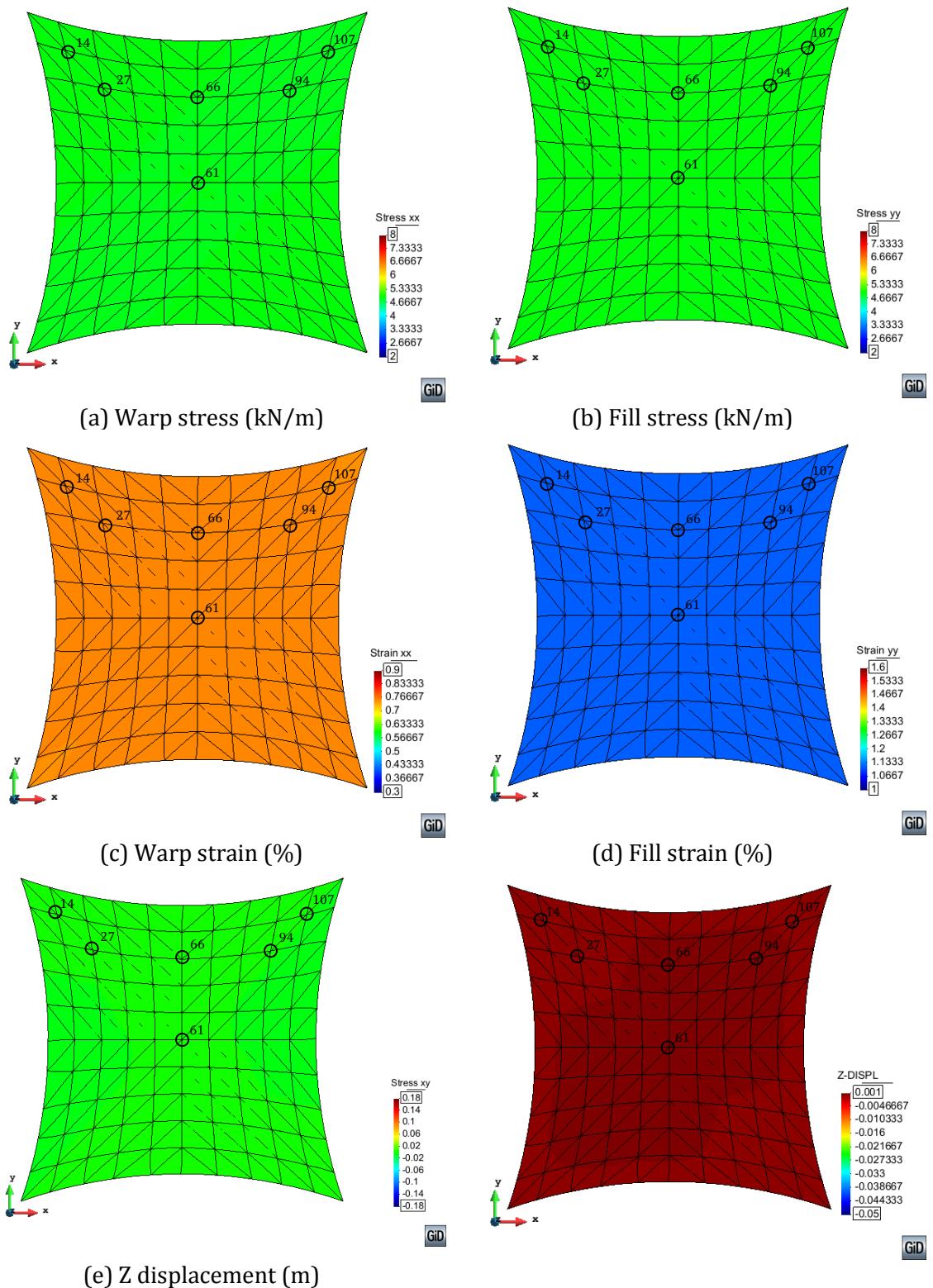
Initial strain and stress conditions are determined using a patch analysis where a stress of 5kN/m is maintained over three iterations. According to the patch test an initial strain of 0.75% is applied in the warp direction and 1.11% in the fill in order to induce a 5kN/m prestress.

At peak snow load (-0.6kN/m<sup>2</sup>) the stress distribution is as expected. Maximum stress occurs in the fill direction between the high points. Minimum stress, lower than initial prestress, occurs in the warp direction between the low points. At peak wind load (1.0kN/m<sup>2</sup>) the stress distribution is again as expected. Maximum stress occurs in the warp direction between the low points while minimum stress occurs in the fill direction between the high points.

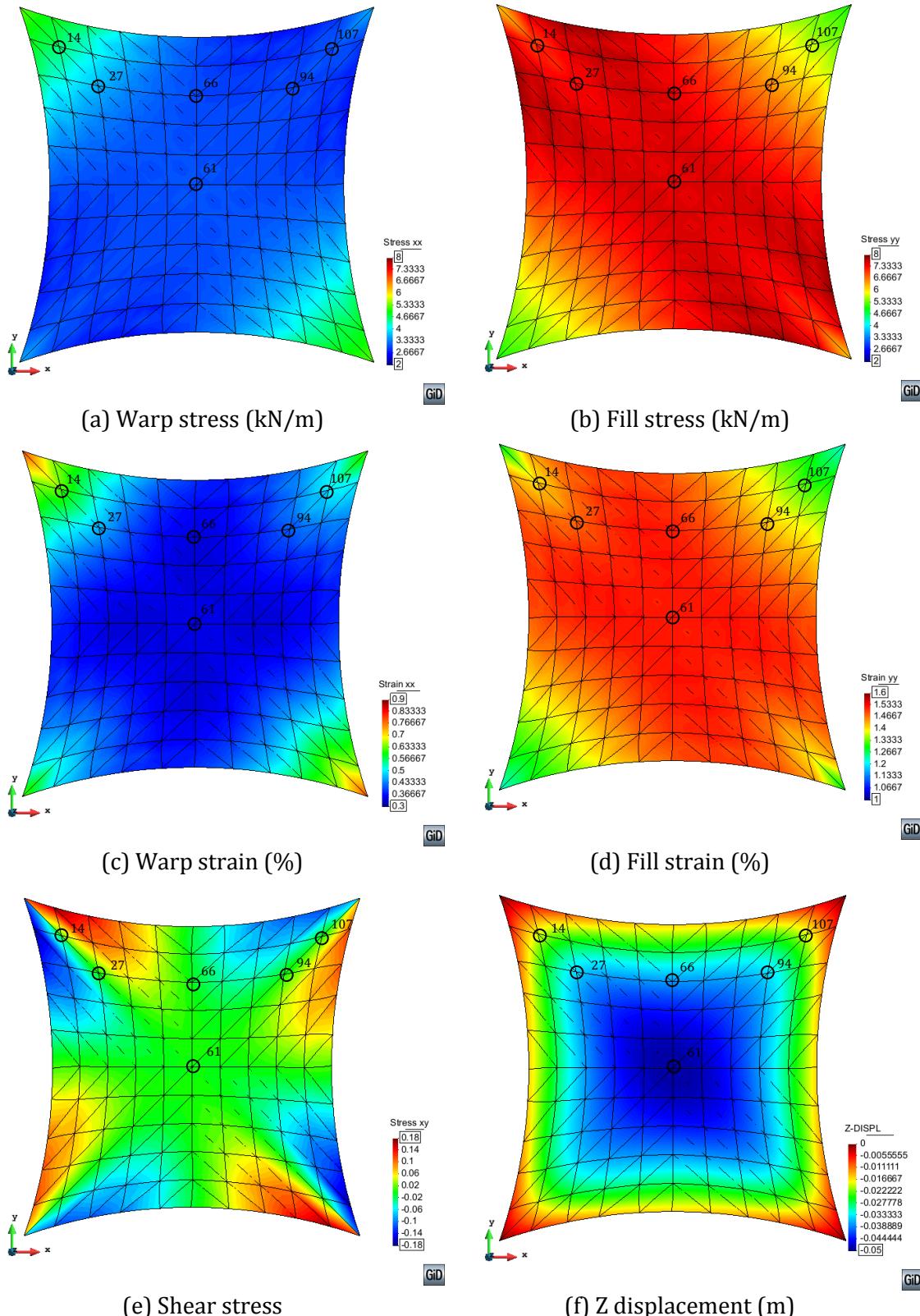
At iteration 11 (load = 0kN/m<sup>2</sup>) of the snow load case it can clearly seen that the hypar has not returned to the same stress strain state as iteration 1 (Figure 5-35, Figure 5-36). Residual deformation in the negative z direction has occurred. Stress is reduced in both material directions below initial prestress. Residual strain in the warp direction has reduced, i.e. been recovered, whereas residual strain in the fill direction has increased.

Residual deformation is also evident in the wind load case (Figure 5-39, Figure 5-40). However, in this case the residual deformation is in the positive z-direction. The stress in both directions is reduced. In the upward loading case the strain in the warp direction is increased but in the fill direction recovered. The change in residual strain in the fill direction, between the high points of the hypar, has a greater influence on residual

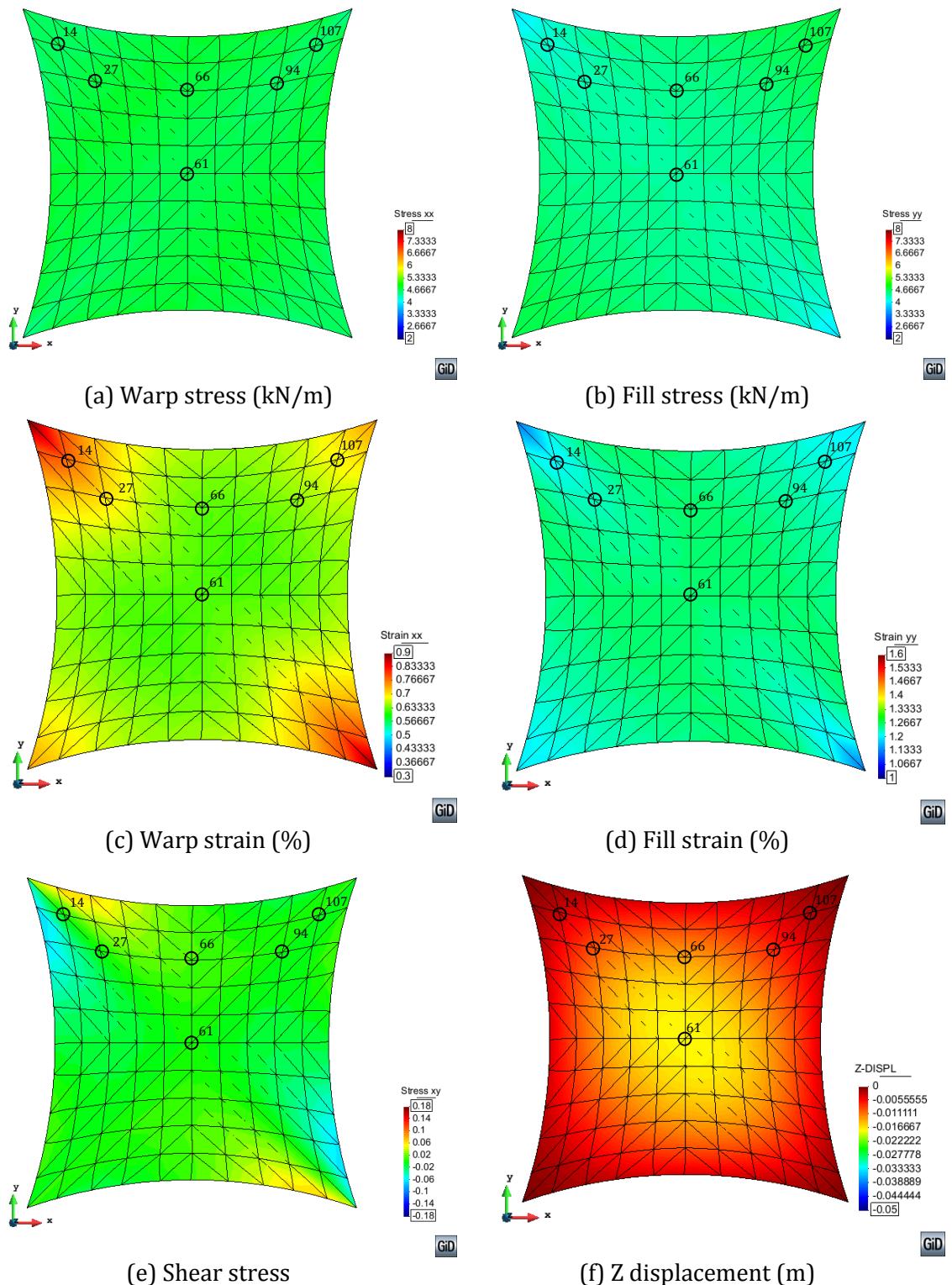
displacement than change in residual strain in the warp direction between the low points. Were the simulation to be continued through another load cycle in either loading direction residual strain would continue to build and stress would reduce until the fabric would eventually become slack.



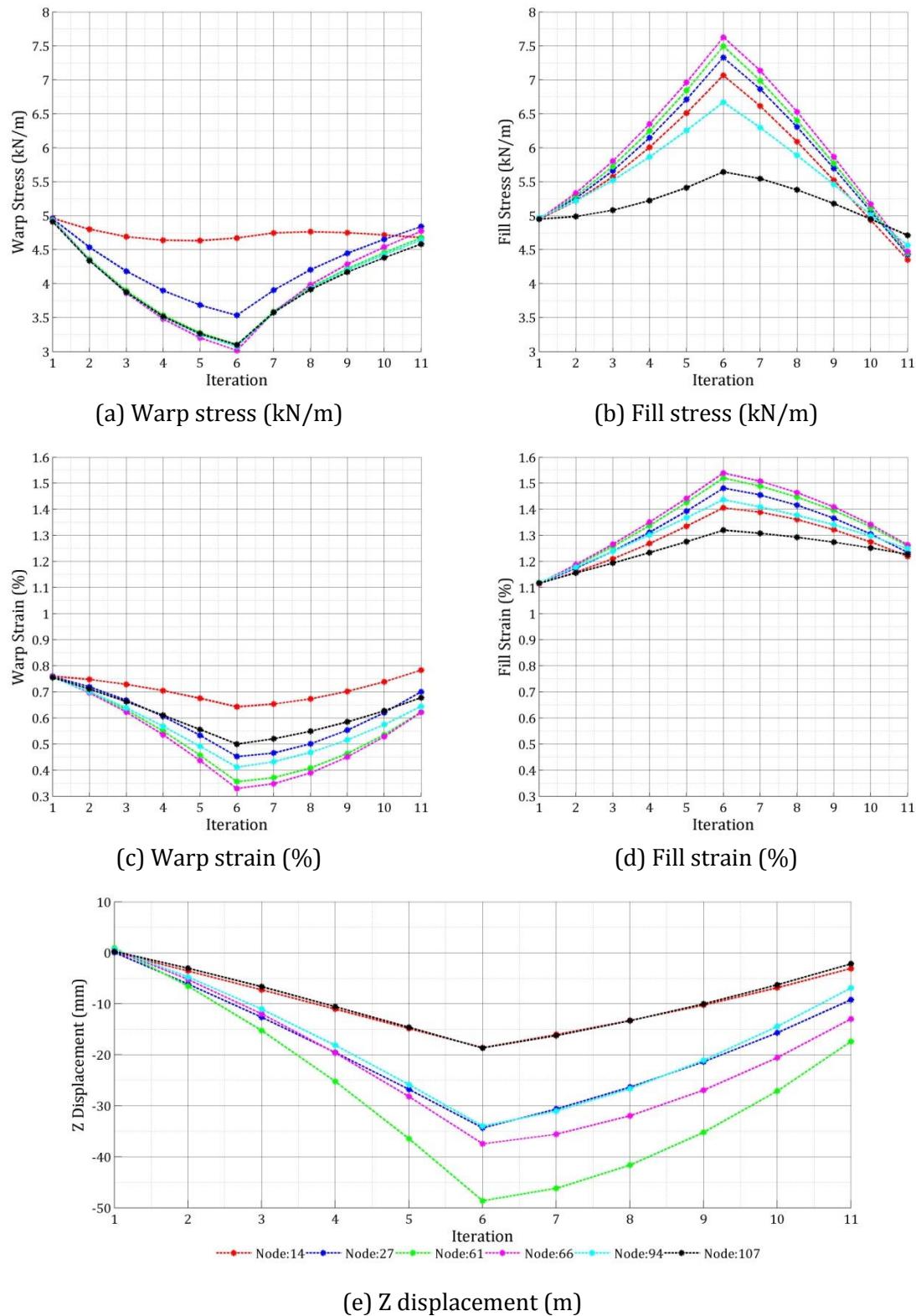
**Figure 5-33: Hypar snow loading PVC history network maximum load  
(Iteration 1: 0.0kN/m<sup>2</sup>)**



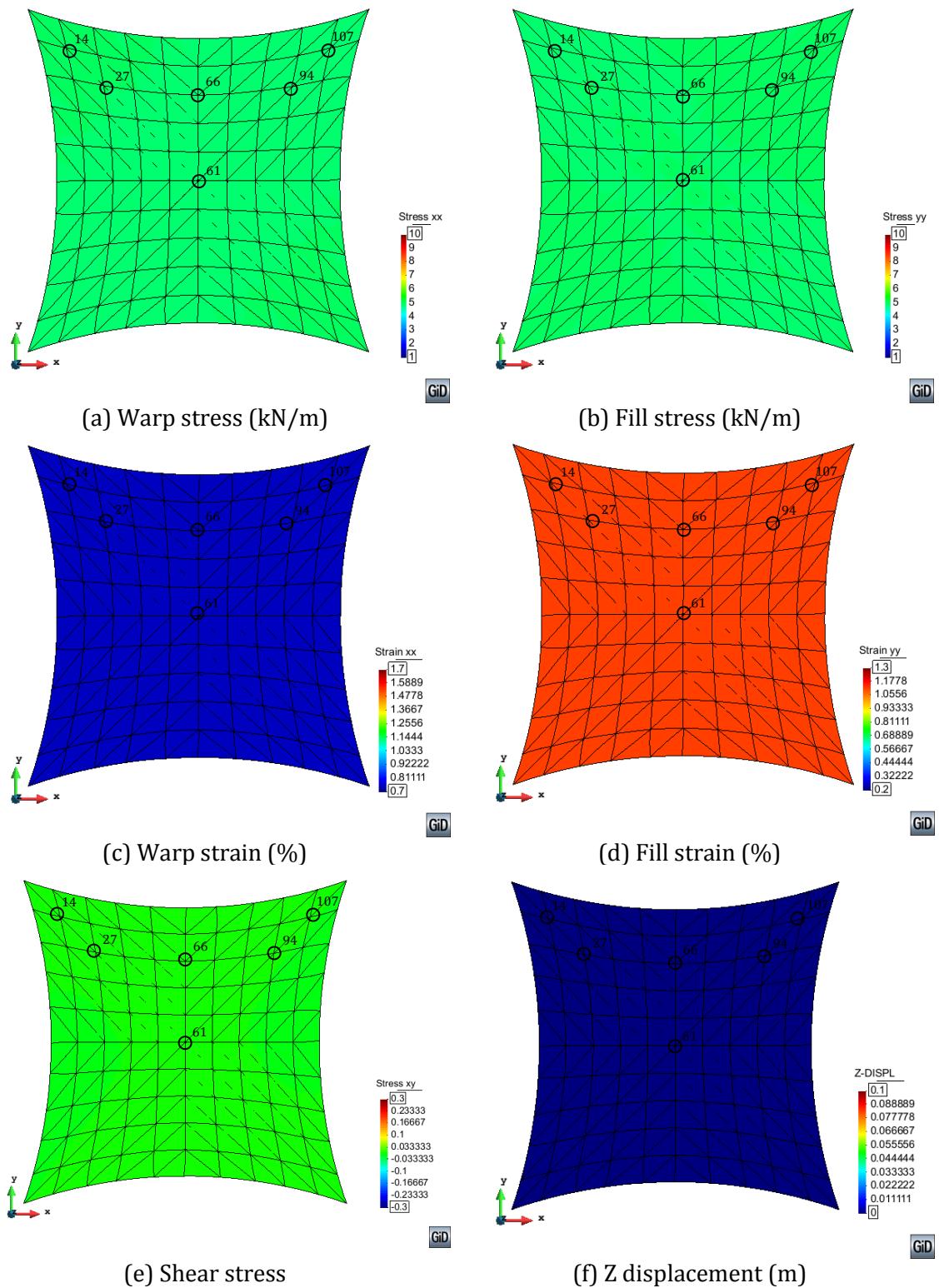
**Figure 5-34: Hypar snow loading PVC history network maximum load  
(Iteration 6:  $0.6\text{kN/m}^2$ )**



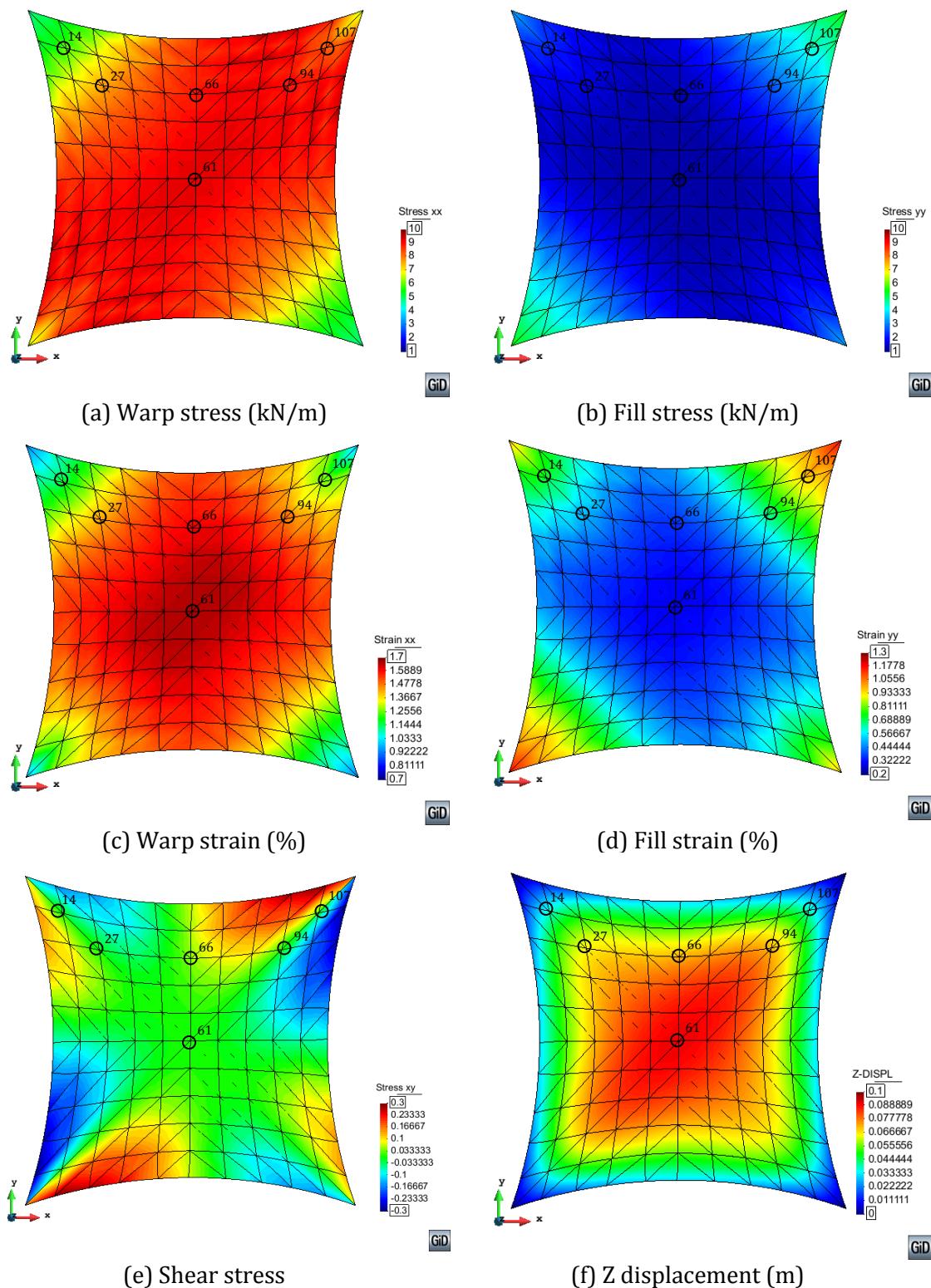
**Figure 5-35: Hypar snow loading PVC history network maximum load  
(Iteration 11: 0.0kN/m<sup>2</sup>)**



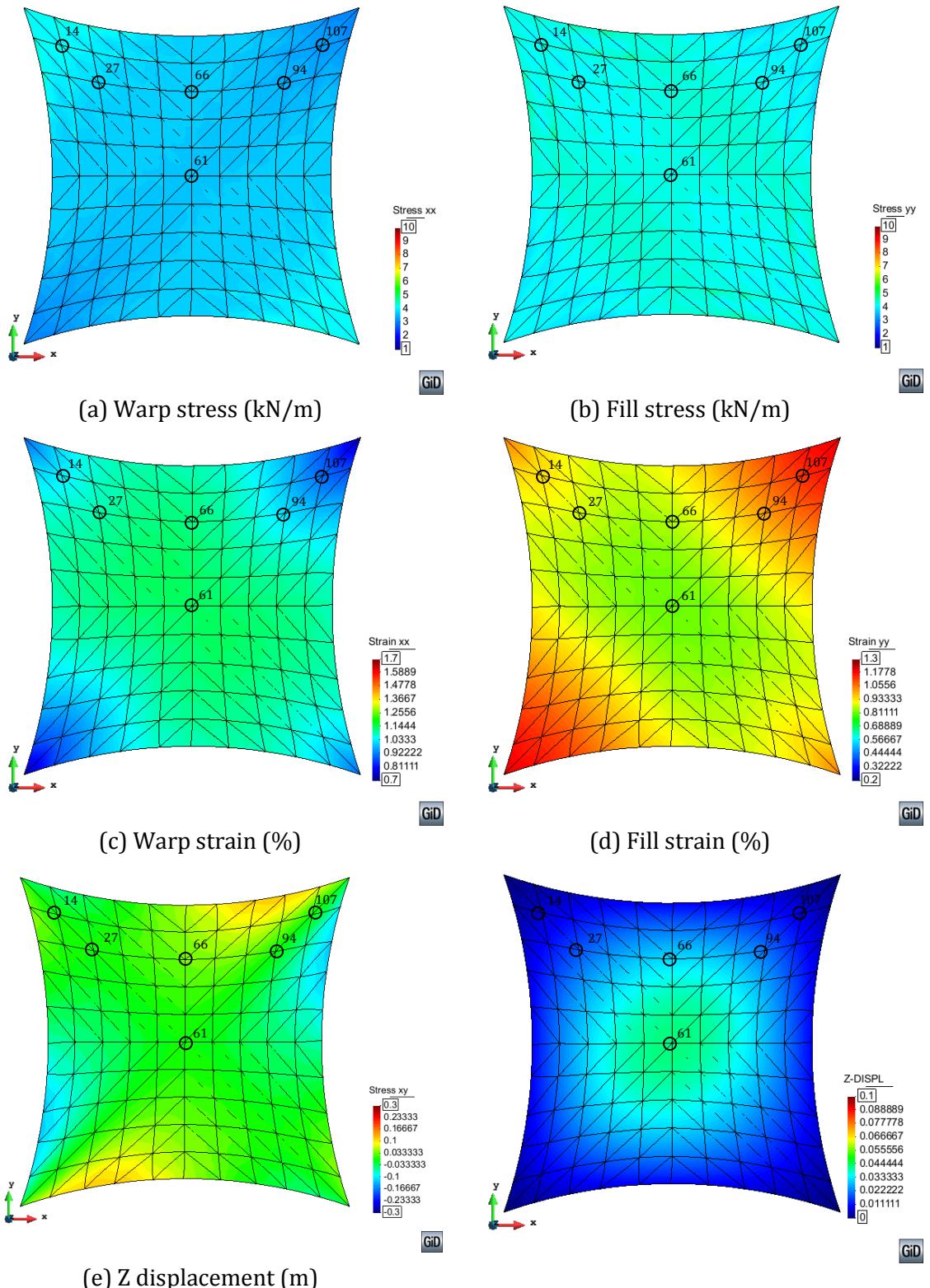
**Figure 5-36: Hypar snow loading PVC history network nodal results.**



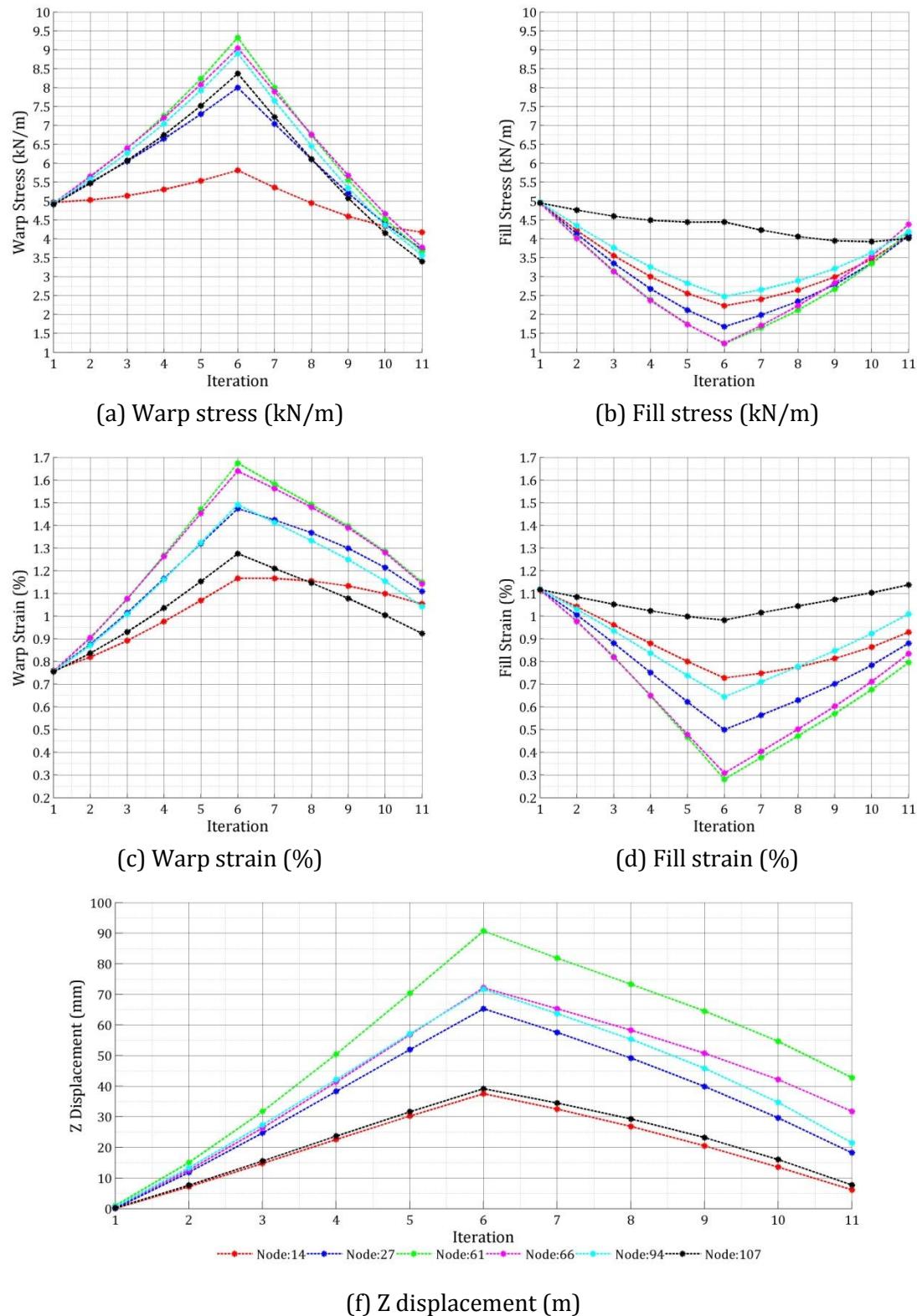
**Figure 5-37: Hypar wind loading PVC history network maximum load (Iteration 1: 0.0kN/m<sup>2</sup>)**



**Figure 5-38: Hypar wind loading PVC history network maximum load (Iteration 6: 1.0kN/m<sup>2</sup>)**



**Figure 5-39: Hypar snow loading PVC history network maximum load (Iteration 11: 0.0kN/m<sup>2</sup>)**



**Figure 5-40: Hypar wind loading PVC history nodal results.**

Under both the snow and wind loading it can be clearly seen that the stress paths followed at each of the selected nodes differ between loading and unloading (Figure 5-36, Figure 5-40). The recovery and increase of residual strain is of particular note. It is also noted that stress levels which fall below the level of stress explored in the training data remain within a reasonable range. This indicates that the network is capable of generalising beyond the training data.

|  | Warp Stress<br>(kN/m) |      | Fill Stress (kN/m) |      | Displacement (mm) |       |          |
|--|-----------------------|------|--------------------|------|-------------------|-------|----------|
|  | Max                   | Min  | Max                | Max  | z                 |       | Absolute |
|  |                       |      |                    |      | Max               | Min   | Max      |
| Prestress<br>(Response Surface Network)        | 5.01                  | 4.97 | 5.01               | 4.98 | 0.1               | -0.1  | 0.2      |
| Prestress:<br>Iteration 1<br>(History Network) | 4.98                  | 4.89 | 4.98               | 4.94 | 0.9               | 0.0   | 0.9      |
| Snow Load<br>(Response Surface Network)        | 5.50                  | 2.83 | 8.04               | 5.23 | 0.0               | -52.3 | 52.3     |
| Snow Load:<br>Iteration 6<br>(History Network) | 5.34                  | 2.52 | 7.88               | 5.06 | 0.0               | -48.7 | 48.7     |
| Wind Load<br>(Response Surface Network)        | 10.20                 | 5.16 | 5.80               | 1.17 | 70.7              | 0.0   | 70.7     |
| Wind Load:<br>Iteration 6<br>(History Network) | 9.89                  | 4.82 | 5.52               | 1.04 | 90.7              | 0.0   | 90.7     |

Table 5-11: Hypar analysis results summary

Simulations of the two load cases along with a prestress load case are repeated on the same form found mesh using the PVC response surface style network from Section 5.2. An initial strain of 0.121% in the warp direction and 0.113% is applied in the fill direction in order to induce the required prestress. Both network models generate results within approximately 0.1kN/m of the desired pre-stress and maximum displacements less than

1mm. The maximum and minimum stress results for both snow and wind load cases are all within 0.5kN/m of each other. In the snow loading case the maximum absolute displacement from both network simulations is within 5mm of each other. However, in the higher wind loading case the maximum absolute displacement simulated by the history network is 20mm greater than that simulated by the response surface network.

## 5.4 Summary and Conclusions

Derivation of the implied stiffness matrix has been presented and validated for both response surface style and history neural network material models. The use of the implied stiffness matrix along with the network material model within the finite element formulation developed in Chapter 3 is described.

A patch test with the response surface style network produces results that fit within the response surface used for training. When training the response surface style networks it has been demonstrated that addition training data outside of the range of available experimental data may be used to stabilise simulation of structures which undergo loading that causes stress to decrease below prestress. The additional data was produced using stress strain pairs generated using a plane stress material model with wrinkling criteria applied.

A set of iteratively loaded patch tests with varying load steps demonstrate that the history network is most effective when the load step is the same or similar to the load step used in generating the training data set. It also demonstrates that the history network simulation is likely to fail to reach an acceptable solution where the load step is greater than that used in training. This indicates that the simulations expected stress range and required load step should be considered when generating training data.

Simulation of the hypar using the history network and iterative loading generates results in a similar range to those generated using the response surface network at peak load. However, it is demonstrated that the history network captures stress history effects including the build up and recovery of residual strain. Also at selected nodes in the mesh it can be seen that the different stress paths are followed when loading and unloading. This form of model has potential for the detailed simulation of fabric installation. It could also provide information on expected residual displacement caused by repeated cyclic loading. This could be used in the planning of maintenance such as re-tensioning required to prevent wrinkled or slack areas forming within the fabric membrane.



**Chapter 6.**  
**Reliability Analysis**



## Contents

|  |     |
|--|-----|
| Chapter 6. Reliability Analysis .....  | 275 |
| 6.1 Limit State Functions and Statistical Variables .....                                      | 280 |
| 6.2 Solution Procedure.....  | 281 |
| 6.3 Sensitivity Analysis .....   | 282 |
| 6.3.1 Analytical derivatives .....   | 283 |
| 6.3.2 Validation of partial derivatives using finite differences.....                          | 298 |
| 6.3.3 Validation using Monte Carlo Simulation.....   | 302 |
| 6.4 Neural Network Variability.....  | 307 |
| 6.5 Response Surface Style Neural Network Reliability Analysis .....                           | 310 |
| 6.5.2 Validation of implied stiffness partial derivative with finite difference<br>method..... | 311 |
| 6.5.3 Reliability analysis of a Hypar.....   | 313 |
| 6.6 Summary and Conclusions.....   | 323 |

## Figures

|   |     |
|---|-----|
| Figure 6-1: Finite element patch for finite difference reliability simulation.....                      | 299 |
| Figure 6-2: Finite element patch for Monte Carlo simulation.....  | 302 |
| Figure 6-3: Cumulative probability plots and intercepts from Monte Carlo simulation ...                 | 305 |
| Figure 6-4: Hypar structure.....  | 307 |
| Figure 6-5: Hypar Stress Plots From 5 Randomly Initiated Networks Trained using<br>Identical Data. .... | 309 |
| Figure 6-6: Hypar structure form-found mesh with selected nodes .....                                   | 315 |
| Figure 6-7: Results Variability Across 30 Networks.....   | 316 |
| Figure 6-8: Hypar snow results using mean implied stiffness values .....                                | 317 |
| Figure 6-9: Implied stiffness values for hypar mesh. ....   | 318 |
| Figure 6-10: Limit State 1 and 2 Reliability Outputs .....  | 320 |
| Figure 6-11: Limit State 3 Reliability Outputs.....   | 321 |
| Figure 6-12: Limit State 4, Without Direction Factor.....   | 322 |
| Figure 6-13: Limit State 4 With Direction Factor.....   | 322 |

## Tables

|  |     |
|--|-----|
| Table 6-1: Sensitivities, finite difference method using perturbation of 0.1% .....  | 300 |
| Table 6-2: Sensitivities, finite difference method using perturbation of 0.001% .....  | 300 |
| Table 6-3: Sensitivities, finite difference method using perturbation of 0.00001%.....   | 301 |
| Table 6-4: Analytically Derived HL_RF Probability of Failure.....  | 303 |
| Table 6-5: Comparison between analytically derived HL_RF probability of failure,<br>$\Phi(-\beta)$ , and Monte Carlo probability of failure, $p_f$ ..... | 304 |
| Table 6-6: Sensitivities, finite difference method using perturbation of 0.001%<br>(Implied Stiffness Variables) .....                                   | 312 |
| Table 6-7: Sensitivities, finite difference method using perturbation of 0.00001%<br>(Implied Stiffness Variables) .....                                 | 313 |
| Table 6-8: Non network related statistical variables.....  | 314 |
| Table 6-9: Network related statistical variables (global mean). ....   | 319 |

## Chapter 6. Reliability Analysis

Reliability analysis is typically used to assess the structural safety. In Eurocode 0 (EC0) the acceptable value of the safety index determined through reliability analysis is defined for a variety of different classes of structure. Partial factors, applied to both actions (loads) and resistance (material strength and stiffness), are determined to provide the same level of safety through a process of calibration that uses the specified safety index in EC0. The partial factors take into account uncertainty in the same way as a reliability analysis, but don't require a reliability analysis to be performed for every design. Due to the geometrically non-linear response of fabric structures to load it is suggested that application of partial factors to actions may be inappropriate. Therefore, existing design guides adopt either a permissible stress or combined approach with a limit state check for overload conditions [130]. The permissible stress approach employs stress reduction factors applied to the strength of the architectural fabric.

In the European Design Guide it is noted that across various available standards a minimum stress factor of 5 is frequently used. However, a relatively wide range of factors, from 3 to 9.5, is reported accounting for various load cases, load effects, environmental impacts and material states. These combined factors do not provide an unambiguous definition of structural reliability. Reliability analysis utilises probabilistic information that quantifies the uncertainty attached to a specific model variable including loads, material properties and geometry.

The reliability analysis presented here builds on the work of Zhang [31, 8]. Derivatives of the equations of the new element formulation are required in order to calculate the gradients of the limit state functions. The analytical partial derivative, along with the reliability algorithm is, presented here and validated using the finite difference method and Monte Carlo Simulation. The algorithm and partial derivatives are then adapted to use the implied stiffness matrix in place of the plane stress stiffness terms in order to facilitate an approximation to the first order reliability analysis for the neural network material models detailed in Chapters 4 and 5. This chapter serves as a first pass reliability analysis of fabric structures using neural network material models.

## 6.1 Limit State Functions and Statistical Variables

Four separate limit state functions,  $G_{1-4}(X_{si})$ , are considered. The structure is deemed to have failed when  $G_{1-4}(X_{si}) < 0$ . The first two functions, (6.1) and (6.2), represent the ultimate failure of the architectural fabric in the principle warp and fill material directions. This is where the maximum stress in the direction in question,  $\sigma_{max}^w$  or  $\sigma_{max}^f$ , exceeds the permissible stress,  $\sigma_{per}^w$  or  $\sigma_{per}^f$ . Permissible stress is usually taken as a factored value of the ultimate strength. The third function represents failure due to wrinkling or complete loss of tension. In this case, the minimum principle stress,  $\sigma_{min}^p$ , must be greater than the permissible stress which is set at zero,  $\sigma_{per}^p = 0$ . This is based on the principle stress wrinkling criteria, equations (3.135) to (3.137), Section 3.1.13. The other criteria, principle strain and mixed criteria, equations (3.138) to (3.143), would be implemented in the same way. The final limit state represents excessive deformation. In this case the limit state is violated when the absolute maximum deformation,  $\delta_{max}$ , exceeds the allowable limit,  $\delta_{per}$ .

$$G_1(X_{si}) = \sigma_{per}^w - \sigma_{max}^w \quad (6.1)$$

$$G_2(X_{si}) = \sigma_{per}^f - \sigma_{max}^f \quad (6.2)$$

$$G_3(X_{si}) = \sigma_{min}^p - \sigma_{per}^p \quad (6.3)$$

$$G_4(X_{si}) = \delta_{per} - |\delta|_{max} \quad (6.4)$$

Where a plane stress material model with unconstrained Poisson's ratios is employed, each limit state is a function of eight independent statistical variables,  $X_{si}$ . Seven statistical basic variables represent uncertainty in the material properties, namely;

$$X_1 = E_w, X_2 = E_f, X_3 = \nu_{fw}, X_4 = \nu_{wf}, X_5 = G_{wf}, X_6 = \sigma_{ult}^f, X_7 = \sigma_{ult}^w, \quad (6.5)$$

where  $E_w$  and  $E_f$  are Young's modulus in the warp and fill directions respectively,  $\nu_{fw}$  and  $\nu_{wf}$  are unconstrained Poisson's ratios,  $G_{wf}$  is the shear modulus and  $\sigma_{ult}^w$  and  $\sigma_{ult}^f$  are the fabrics ultimate strengths in the warp and fill directions, respectively.

The remaining statistical basic variable accounts for uncertainty associated with the imposed load coefficient,  $t_{load}$ , with,

$$X_8 = \frac{F}{F_c} = t_{load} \quad (6.6)$$

where  $F$  is the applied load and  $F_c$  is the deterministic design load. In cases where Poisson's ratio is constrained to follow the reciprocal relationship, the total number of independent statistical variables is reduced to seven.

## 6.2 Solution Procedure

The solution procedure is based on the Hasofer-Lind and Rackwitz-Fiessler (HL\_RF) algorithm[16, 31] :

1. Define the appropriate limit state function and failure criterion.
2. Assume initial values of the design point  $x_{si}^*, i = 1, 2, \dots, n$  (normally mean value), and calculate the corresponding value of the limit state functions  $G_i()$ .
3. For those non-normally distributed variables, compute the mean and standard deviation at the design point of the equivalent normal distribution using (6.7) and (6.8). Then transform the random variables to the reduced coordinate system using (6.9).

$$\mu_{X_{si}}^N = F_{X_{si}}^{-1}(0.5) = \text{median of } X_{si} \quad (6.7)$$

$$\sigma_{X_{si}}^N = \frac{x_{si}^* - \mu_{X_{si}}^N}{\Phi^{-1}[F_{X_{si}}(x_{si}^*)]} \quad (6.8)$$

$$x'_{si}^* = \frac{x_{si}^* - \mu_{X_{si}}^N}{\sigma_{X_{si}}^N} \quad (6.9)$$

4. Compute the partial derivative  $\frac{\partial G}{\partial X_{si}}$  evaluated at the design point  $x_{si}^*$ .
5. Compute the partial derivative  $\frac{\partial G}{\partial X'_{si}}$  using the chain rule of differentiation as,

$$\frac{\partial G}{\partial X'_{si}} = \frac{\partial G}{\partial X_{si}} \frac{\partial X_{si}}{\partial X'_{si}} = \frac{\partial G}{\partial X_{si}} \sigma_{X_{si}}^N \quad (6.10)$$

6. Compute  $L$

$$L = \sqrt{\sum_{i=1}^n \left( \frac{\partial G}{\partial X'_{si}} \right)^2} \quad (6.11)$$

7. Calculate the direction cosines, alpha,

$$\alpha_i = \frac{\frac{\partial G}{\partial X'_{si}}}{L} \quad (6.12)$$

8. Compute the safety index  $\beta$  based on the values of the design points calculated in step 6 as,

$$\beta = \sqrt{\sum_{i=1}^n (\alpha_i)^2} \quad (6.13)$$

9. Compute the new values of design points ( $x'_{si}^*$ ) in the reduced space using (6.14).

$$x'_{si(k+1)}^* = -\alpha_i \left( \beta + \frac{G(X_{si})}{L} \right) \quad (6.14)$$

10. Check the convergence of  $\beta$ .

11. Compute the new values of the design point ( $x_{si(k+1)}^*$ ) in the original space as,

$$x_{si(k+1)}^* = \mu_{X_{si}}^N + \sigma_{X_{si}}^N x'_{si(k+1)}^* \quad (6.15)$$

Update the value of the limit state function  $G(X_{si})$  for the new design point and check if the design points are on the limit state function (i.e.  $|G(X_{si})|$  is very close to zero, e.g. within 0.001 for the examples presented in this chapter). If the convergence criteria in step 7 are satisfied and  $|G(X_{si})|$  is approximately zero, then stop. Otherwise, repeat steps 3 to 8 until convergence.

### 6.3 Sensitivity Analysis

The gradients of the limit state functions are required for steps 4 and 5 of the solution procedure presented above. They are calculated from the partial derivatives of the limit state functions with respect to the statistical variable, as in,

$$\frac{\partial G_1}{\partial X_{si}} = \frac{\partial(\sigma_{per}^w - \sigma_{max}^w)}{\partial X_{si}} = \frac{\partial \sigma_{per}^w}{\partial X_{si}} - \frac{\partial \sigma_{max}^w}{\partial X_{si}} \quad (6.16)$$

$$\frac{\partial G_2}{\partial X_{si}} = \frac{\partial(\sigma_{per}^f - \sigma_{max}^f)}{\partial X_{si}} = \frac{\partial \sigma_{per}^f}{\partial X_{si}} - \frac{\partial \sigma_{max}^f}{\partial X_{si}} \quad (6.17)$$

$$\frac{\partial G_3}{\partial X_{si}} = \frac{\partial(\sigma_{min}^p - \sigma_{per}^p)}{\partial X_{si}} = \frac{\partial \sigma_{min}^p}{\partial X_{si}} - \frac{\partial \sigma_{per}^p}{\partial X_{si}} \quad (6.18)$$

$$\frac{\partial G_4}{\partial X_{si}} = \frac{\partial(\delta_{per} - \delta_{max})}{\partial X_{si}} = \frac{\partial \delta_{per}}{\partial X_{si}} - \frac{\partial \delta_{max}}{\partial X_{si}}. \quad (6.19)$$

The gradients calculated by (6.16) to (6.19) effectively constitute a sensitivity analysis of structural response with respect to the statistical variables. Therefore, the gradients provide useful information about how the statistical variables influence the reliability of a structure. In some cases, where a variable is identified as having a very limited impact (e.g. the sensitivity is very low), the computational efficiency of the reliability analysis may be greatly improved by eliminating these statistical variables that may then be classified as deterministic.

### 6.3.1 Analytical derivatives

The derivative of displacement with respect to the statistical variables is first required to be found. The simplest form of a general deterministic linear finite element analysis is represented by,

$$K\delta = F \text{ or } \delta = K^{-1}F \quad (6.20)$$

where  $K$  is the structural stiffness matrix,  $\delta$  is the array of nodal displacements and  $F$  is the array of nodal forces.

Differentiating (6.20) with respect to the statistical variables using the chain rule gives,

$$\frac{dK}{dX_{si}}\delta + K\frac{d\delta}{dX_{si}} = \frac{dF}{dX_{si}} \quad (6.21)$$

Given that,

$$K = \int_V B^T EB dV, \text{ then,} \quad (6.22)$$

$$\begin{aligned} \frac{dK}{dX_{si}} &= \int_V \frac{dB}{dX_{si}}^T EB + B^T \frac{dE}{dX_{si}} B + B^T E \frac{dB}{dX_{si}} dV = \int_V \left[ \frac{\partial B^T}{\partial \delta} \frac{\partial \delta}{\partial X_{si}} \right]^T EB + B^T \frac{dE}{dX_{si}} B + \\ &B^T E \frac{\partial B}{\partial \delta} \frac{\partial \delta}{\partial X_{si}} dV \end{aligned} \quad (6.23)$$

Numerical integration with 12 Gauss Points is used to compute the integral of (6.22).

Therefore,

$$K = \int_V B^T EB dV \approx \sum_{i=1}^{n=12} [B_i^T EB_i] \times GW_i \times Area \quad (6.24)$$

The following sub-derivatives are defined:

$$dK_1 \frac{\partial \delta}{\partial X_{si}} = \int_V \left[ \frac{\partial B}{\partial \delta} \frac{\partial \delta}{\partial X_{si}} \right]^T EB dV \approx \sum_{i=1}^{n=12} \left[ \left[ \frac{\partial B_i}{\partial \delta} \frac{\partial \delta}{\partial X_{si}} \right]^T EB_i \right] \times GW_i \times Area \quad (6.25)$$

$$dK_2 = \int_V B^T \frac{dE}{dX_{si}} B dV \approx \sum_{i=1}^{n=12} \left[ B_i^T \frac{dE}{dX_{si}} B_i \right] \times GW_i \times Area \quad (6.26)$$

$$dK_3 \frac{\partial \delta}{\partial X_{si}} = \int_V B^T E \left[ \frac{\partial B}{\partial \delta} \frac{\partial \delta}{\partial X_{si}} \right] dV \approx \sum_{i=1}^{n=12} \left[ B_i^T E \left[ \frac{\partial B_i}{\partial \delta} \frac{\partial \delta}{\partial X_{si}} \right] \right] \times GW_i \times Area. \quad (6.27)$$

As the area of the element is also a function of the displacement, an additional derivative term is required, as in,

$$dK_4 \frac{\partial \delta}{\partial X_{si}} = \sum_{i=1}^{n=12} [B_i^T EB_i] \times GW_i \times \frac{\partial Area}{\partial \delta} \frac{\partial \delta}{\partial X_{si}} \quad (6.28)$$

Substitution of (6.25) to (6.28) into (6.23) leads to,

$$\left[ dK_1 \frac{\partial \delta}{\partial X_{si}} + dK_2 + dK_3 \frac{\partial \delta}{\partial X_{si}} + dK_4 \frac{\partial \delta}{\partial X_{si}} \right] \delta + K \frac{\partial \delta}{\partial X_{si}} = \frac{dF}{dX_{si}} \quad (6.29)$$

Collecting together terms containing  $\frac{\partial\delta}{\partial X_{si}}$  yields,

$$[dK_1\delta + dK_3\delta + dK_4\delta + K] \frac{\partial\delta}{\partial X_{si}} = \frac{dF}{dX_{si}} - dK_2\delta. \quad (6.30)$$

in which the terms within the bracket may be described by the single variable,  $dK$ , as,

$$dK = dK_1\delta + dK_3\delta + dK_4\delta + K. \quad (6.31)$$

Substitution of (6.31) into (6.30) gives,

$$dK \frac{\partial\delta}{\partial X_{si}} = \frac{dF}{dX_{si}} - dK_2\delta. \quad (6.32)$$

Rearranging for  $\frac{\partial\delta}{\partial X_{si}}$  yields,

$$\frac{\partial\delta}{\partial X} = dK^{-1} \left[ \frac{dF}{dX} - dK_2\delta \right], \quad (6.33)$$

and concludes the derivation of the derivative of displacement with respect to the statistical variables.

The derivative of stress with respect to the statistical variable makes use of the displacement sensitivity, as in,

$$\sigma = E\varepsilon = EB\delta \quad (6.34)$$

The derivative of the stress vector is thus given by,

$$\frac{\partial\sigma}{\partial X_{si}} = \frac{\partial E}{\partial X_{si}} B\delta + E \frac{\partial B}{\partial X_{si}} \delta + EB \frac{\partial\delta}{\partial X_{si}} \quad (6.35)$$

It is not possible to find the derivative of the B-matrix with respect to the statistical variable directly. Using the chain rule the derivative of the B-matrix may be expressed as a function of the displacement derivative as,

$$\frac{\partial B}{\partial X_{si}} = \frac{\partial B}{\partial \delta} \frac{\partial \delta}{\partial X_{si}} = dB \frac{\partial \delta}{\partial X_{si}}. \quad (6.36)$$

Substituting (6.36) into (6.35) leads to,

$$\frac{\partial\sigma}{\partial X_{si}} = \frac{\partial E}{\partial X_{si}} B\delta + EdB \frac{\partial\delta}{\partial X_{si}} \delta + EB \frac{\partial\delta}{\partial X_{si}}. \quad (6.37)$$

The minimum principle stress is given by,

$$\sigma_{min}^p = 0.5(\sigma_f + \sigma_w) - \sqrt{\left(\frac{\sigma_w - \sigma_f}{2}\right)^2 + (\gamma_{wf})^2}. \quad (6.38)$$

Therefore, based on (6.38) the derivative is given by,

$$\frac{\partial \sigma_{min}^p}{\partial X_{si}} = 0.5 \left( \frac{\partial \sigma_f}{\partial X_{si}} + \frac{\partial \sigma_w}{\partial X_{si}} \right) - \frac{0.5(\sigma_w - \sigma_f) \left( \frac{\partial \sigma_f}{\partial X_{si}} \frac{\partial \sigma_w}{\partial X_{si}} \right) + 2\sigma_{wf} \frac{\partial \sigma_{wf}}{\partial X_{si}}}{2\sqrt{\frac{1}{4}(\sigma_w - \sigma_f)^2 + (\sigma_{wf})^2}}. \quad (6.39)$$

As in the case of the  $B$ -matrix, (6.36), the derivatives of the membrane transformation matrix,  $T^m$ , cable transformation matrix,  $T^c$ , and cable force,  $P$ , with respect to the statistical variables cannot be found directly. Again the chain rule is evoked, and use is made of the displacement derivative to give,

$$\frac{\partial T^m}{\partial X_{si}} = \frac{\partial T^m}{\partial \delta} \frac{\partial \delta}{\partial X_{si}} = dT^m \frac{\partial \delta}{\partial X_{si}} \quad (6.40)$$

$$\frac{\partial T^c}{\partial X_{si}} = \frac{\partial T^c}{\partial \delta} \frac{\partial \delta}{\partial X_{si}} = dT^c \frac{\partial \delta}{\partial X_{si}} \quad (6.41)$$

$$\frac{\partial P}{\partial X_{si}} = \frac{\partial P}{\partial \delta} \frac{\partial \delta}{\partial X_{si}} = dP \frac{\partial \delta}{\partial X_{si}}. \quad (6.42)$$

### 6.3.1.1 Elastic Stiffness Matrix Derivative

Assuming that the elastic stiffness matrix,  $E$ , is given by,

$$[E] = \begin{bmatrix} E_w & \frac{E_w v_{fw}}{(1-v_{wf}v_{fw})} & 0 \\ \frac{E_f v_{wf}}{(1-v_{wf}v_{fw})} & \frac{E_f}{(1-v_{wf}v_{fw})} & 0 \\ 0 & 0 & G_{wf} \end{bmatrix}, \quad (6.43)$$

it is relatively simple to find the partial derivatives of  $E$  with respect to the statistical variables,  $\frac{\partial E}{\partial X_{si}}$ . The partial derivative of  $E$  with respect to the young's modulus in the warp direction,  $E_w$ , is given by,

$$\left[ \frac{\partial E}{\partial X_1} \right] = \begin{bmatrix} \frac{1}{(1-v_{wf}v_{fw})} & \frac{v_{fw}}{(1-v_{wf}v_{fw})} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (6.44)$$

and similarly with respect to young's modulus in the fill direction,  $E_f$ , is given by,

$$\left[ \frac{\partial E}{\partial X_2} \right] = \begin{bmatrix} 0 & 0 & 0 \\ \frac{v_{wf}}{(1-v_{wf}v_{fw})} & \frac{1}{(1-v_{wf}v_{fw})} & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (6.45)$$

The partial derivatives of  $E$  with respect to the Poisson's ratios  $v_{fw}$  and  $v_{wf}$  are given by,

$$\left[ \frac{\partial E}{\partial X_3} \right] = \begin{bmatrix} \frac{E_w}{v_{wf}(1-v_{wf}v_{fw})^2} & \frac{E_w v_{fw}}{v_{wf}(1-v_{wf}v_{fw})^2} + \frac{E_w}{(1-v_{wf}v_{fw})} & 0 \\ \frac{E_f v_{wf}}{v_{wf}(1-v_{wf}v_{fw})^2} & \frac{E_f}{v_{wf}(1-v_{wf}v_{fw})^2} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (6.46)$$

and

$$\left[ \frac{\partial E}{\partial X_4} \right] = \begin{bmatrix} \frac{E_w}{v_{fw}(1-v_{wf}v_{fw})^2} & \frac{E_w v_{fw}}{v_{fw}(1-v_{wf}v_{fw})^2} & 0 \\ \frac{E_f v_{wf}}{v_{fw}(1-v_{wf}v_{fw})^2} + \frac{E_f}{(1-v_{wf}v_{fw})} & \frac{E_f}{v_{fw}(1-v_{wf}v_{fw})^2} & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (6.47)$$

Finally the partial derivatives of  $E$  with respect to the shear modulus,  $G_{wf}$ , is given by,

$$\left[ \frac{\partial E}{\partial X_5} \right] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (6.48)$$

As  $E$  is not a function of the fabric ultimate strength or the applied load,

$$\left[ \frac{\partial E}{\partial X_{6-8}} \right] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (6.49)$$

where  $X_{6-8}$  is  $\sigma_{ult}^w$ ,  $\sigma_{ult}^f$  and  $t_{load}$  respectively.

### 6.3.1.2 Transformation matrix derivative

A demonstrated in Section 3.1.5 the local Z-direction basis vector is derived from the cross product of the natural co-ordinate basis vectors. When written explicitly equation (3.56) becomes

$$\hat{Z} = \frac{\hat{x} \times \hat{y}}{\|\hat{x} \times \hat{y}\|} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \hat{\xi}_1 & \hat{\xi}_2 & \hat{\xi}_3 \\ \hat{\eta}_1 & \hat{\eta}_2 & \hat{\eta}_3 \end{vmatrix} = \frac{(\hat{\xi}_2 \hat{\eta}_3 - \hat{\xi}_3 \hat{\eta}_2)}{\|\vec{Z}\|} \hat{i} + \frac{(\hat{\xi}_3 \hat{\eta}_1 - \hat{\xi}_1 \hat{\eta}_3)}{\|\vec{Z}\|} \hat{j} + \frac{(\hat{\xi}_1 \hat{\eta}_2 - \hat{\xi}_2 \hat{\eta}_1)}{\|\vec{Z}\|} \hat{k} \quad (6.50)$$

where,

$$\|\vec{Z}\| = \sqrt{(\hat{\xi}_2 \hat{\eta}_3 - \hat{\xi}_3 \hat{\eta}_2)^2 + (\hat{\xi}_3 \hat{\eta}_1 - \hat{\xi}_1 \hat{\eta}_3)^2 + (\hat{\xi}_1 \hat{\eta}_2 - \hat{\xi}_2 \hat{\eta}_1)^2} \quad (6.51)$$

Therefore, the directional cosines describing the local Z-direction are given by,

$$n_1 = \frac{(\hat{\xi}_2 \hat{\eta}_3 - \hat{\xi}_3 \hat{\eta}_2)}{\|\vec{Z}\|} \quad (6.52)$$

$$n_2 = \frac{(\hat{\xi}_3 \hat{\eta}_1 - \hat{\xi}_1 \hat{\eta}_3)}{\|\vec{Z}\|} \quad (6.53)$$

$$n_3 = \frac{(\hat{\xi}_1 \hat{\eta}_2 - \hat{\xi}_2 \hat{\eta}_1)}{\|\vec{Z}\|} \quad (6.54)$$

The local  $X$ -direction basis vector is obtained from the cross product of the global fill material direction basis vector,  $\hat{\theta}_f$ , with the local  $Z$ -direction basis vector. When written explicitly, using the form of (3.56) and (3.57), (3.58) becomes

$$\hat{X} = \frac{\hat{\theta}_f \times \hat{Z}}{\|\hat{\theta}_f \times \hat{Z}\|} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \hat{\theta}_{f_1} & \hat{\theta}_{f_2} & \hat{\theta}_{f_3} \\ \hat{Z}_1 & \hat{Z}_2 & \hat{Z}_3 \end{vmatrix} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ -\sin \theta & \cos \theta & 0 \\ n_1 & n_2 & n_3 \end{vmatrix} = \frac{(\cos(\theta) n_3)}{\|\vec{X}\|} \hat{i} + \frac{(\sin(\theta) n_3)}{\|\vec{X}\|} \hat{j} + \frac{(-\sin(\theta) n_2 - \cos(\theta) n_1)}{\|\vec{X}\|} \hat{k}, \quad (6.55)$$

where,

$$\|\vec{X}\| = \sqrt{(\cos(\theta) n_3)^2 + (\sin(\theta) n_3)^2 + (-\sin(\theta) n_2 - \cos(\theta) n_1)^2}. \quad (6.56)$$

Therefore,

$$l_1 = \frac{(\cos(\theta) n_3)}{\|\vec{X}\|} \quad (6.57)$$

$$l_2 = \frac{(\sin(\theta) n_3)}{\|\vec{X}\|} \quad (6.58)$$

$$l_3 = \frac{(-\sin(\theta) n_2 - \cos(\theta) n_1)}{\|\vec{X}\|}. \quad (6.59)$$

Finally the local  $Y$ -direction basis vector derived from the cross product of the local  $X$ -direction basis vector and the local  $Z$ -direction material direction basis vector. When written explicitly, using the form of (3.57) and (3.58), (3.59) becomes

$$\hat{Y} = \frac{\hat{Z} \times \hat{X}}{\|\hat{Z} \times \hat{X}\|} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \hat{Z}_1 & \hat{Z}_2 & \hat{Z}_3 \\ \hat{X}_1 & \hat{X}_2 & \hat{X}_3 \end{vmatrix} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ n_1 & n_2 & n_3 \\ l_1 & l_2 & l_3 \end{vmatrix} = \frac{(n_2 l_3 - n_3 l_2)}{\|\vec{Y}\|} \hat{i} + \frac{(n_3 l_1 - n_1 l_3)}{\|\vec{Y}\|} \hat{j} + \frac{(n_1 l_2 - n_2 l_1)}{\|\vec{Y}\|} \hat{k},$$

where,

$$\|\vec{Y}\| = \sqrt{(n_2 l_3 - n_3 l_2)^2 + (n_3 l_1 - n_1 l_3)^2 + (n_1 l_2 - n_2 l_1)^2}. \quad (6.60)$$

Therefore,

$$m_1 = \frac{(n_2 l_3 - n_3 l_2)}{\|\vec{Y}\|} \quad (6.61)$$

$$m_2 = \frac{(n_3 l_1 - n_1 l_3)}{\|\vec{Y}\|} \quad (6.62)$$

$$m_3 = \frac{(n_1 l_2 - n_2 l_1)}{\|\vec{Y}\|}. \quad (6.63)$$

The partial derivative of the transformation matrix with respect to the global displacements,  $\{\delta\}_i$ , is given by

$$dT^m = \frac{\partial T}{\partial \delta} = \begin{bmatrix} \frac{\partial l_1}{\partial \delta} & \frac{\partial l_2}{\partial \delta} & \frac{\partial l_3}{\partial \delta} \\ \frac{\partial m_1}{\partial \delta} & \frac{\partial m_2}{\partial \delta} & \frac{\partial m_3}{\partial \delta} \\ \frac{\partial n_1}{\partial \delta} & \frac{\partial n_2}{\partial \delta} & \frac{\partial n_3}{\partial \delta} \end{bmatrix}. \quad (6.64)$$

The updated global nodal co-ordinates of an element may be defined by the dot product of the initial nodal co-ordinates and the element global nodal displacements

$$C = C_0 \cdot \delta = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \\ x_5 & y_5 & z_5 \\ x_6 & y_6 & z_6 \end{bmatrix} \cdot \begin{bmatrix} u_1 & v_1 & w_1 \\ u_2 & v_2 & w_2 \\ u_3 & v_3 & w_3 \\ u_4 & v_4 & w_4 \\ u_5 & v_5 & w_5 \\ u_6 & v_6 & w_6 \end{bmatrix} = \begin{bmatrix} x_1 + u_1 & y_1 + v_1 & z_1 + w_1 \\ x_2 + u_2 & y_2 + v_2 & z_2 + w_2 \\ x_3 + u_3 & y_3 + v_3 & z_3 + w_3 \\ x_4 + u_4 & y_4 + v_4 & z_4 + w_4 \\ x_5 + u_5 & y_5 + v_5 & z_5 + w_5 \\ x_6 + u_6 & y_6 + v_6 & z_6 + w_6 \end{bmatrix} \quad (6.65)$$

Substitution of the above definition of the global nodal co-ordinates into the equations for the natural co-ordinate basis vectors  $\hat{\xi}$  (3.53) and  $\hat{\eta}$  (3.54) yields,

$$\begin{aligned} \hat{\xi}_1 &= (4\xi_1 - 1)(x_1 + u_1) - (4\xi_3 + 1)(x_3 + u_3) + (4\xi_2)(x_4 + u_4) - \\ &\quad (4\xi_2)(x_5 + u_5) + 4(\xi_3 - \xi_1)(x_6 + u_6), \end{aligned} \quad (6.66)$$

$$\begin{aligned} \hat{\xi}_2 &= (4\xi_1 - 1)(y_1 + v_1) - (4\xi_3 + 1)(y_3 + v_3) + (4\xi_2)(y_4 + v_4) - \\ &\quad (4\xi_2)(y_5 + v_5) + 4(\xi_3 - \xi_1)(y_6 + v_6), \end{aligned} \quad (6.67)$$

$$\begin{aligned} \hat{\xi}_3 &= (4\xi_1 - 1)(z_1 + w_1) - (4\xi_3 + 1)(z_3 + w_3) + (4\xi_2)(z_4 + w_4) - \\ &\quad (4\xi_2)(z_5 + w_5) + 4(\xi_3 - \xi_1)(z_6 + w_6), \end{aligned} \quad (6.68)$$

and

$$\begin{aligned} \hat{\eta}_1 &= (4\xi_2 - 1)(x_2 + u_2) - (4\xi_3 + 1)(x_3 + u_3) + (4\xi_1)(x_4 + u_4) + \\ &\quad 4(\xi_3 - \xi_2)(x_5 + u_5) - 4(\xi_1)(x_6 + u_6), \end{aligned} \quad (6.69)$$

$$\begin{aligned} \hat{\eta}_2 &= (4\xi_2 - 1)(y_2 + v_2) - (4\xi_3 + 1)(y_3 + v_3) + (4\xi_1)(y_4 + v_4) + \\ &\quad 4(\xi_3 - \xi_2)(y_5 + v_5) - 4(\xi_1)(y_6 + v_6), \end{aligned} \quad (6.70)$$

$$\begin{aligned} \hat{\eta}_3 &= (4\xi_2 - 1)(z_2 + w_2) - (4\xi_3 + 1)(z_3 + w_3) + (4\xi_1)(z_4 + w_4) + \\ &\quad 4(\xi_3 - \xi_2)(z_5 + w_5) - 4(\xi_1)(z_6 + w_6). \end{aligned} \quad (6.71)$$

Defining

$$\begin{aligned} a_{23} &= \hat{\xi}_2 \hat{\eta}_3 & a_{31} &= \hat{\xi}_3 \hat{\eta}_1 & a_{12} &= \hat{\xi}_1 \hat{\eta}_2 \\ b_{32} &= \hat{\xi}_3 \hat{\eta}_2 & b_{13} &= \hat{\xi}_1 \hat{\eta}_3 & b_{21} &= \hat{\xi}_2 \hat{\eta}_1, \end{aligned} \quad (6.72)$$

$$\|\vec{Z}\| = c_1 = (a_{23} - b_{32})^2 + (a_{31} - b_{13})^2 + (a_{12} - b_{21})^2, \quad (6.73)$$

and noting that the derivatives of the basis vectors are arranged within  $n$  by  $m$  matrices in the same form as (6.65) where  $n$  indicates the element node ( $n = 1 \rightarrow 6$ ) and  $m$  indicates the global displacement direction  $u, v$  and  $w$  ( $m = 1 \rightarrow 3$ ),

$$\partial \hat{\xi}_1 = \begin{bmatrix} 4\xi_1 - 1 & 0 & 0 \\ 0 & 0 & 0 \\ -4\xi_3 + 1 & 0 & 0 \\ 4\xi_2 & 0 & 0 \\ -4\xi_2 & 0 & 0 \\ 4(\xi_3 - \xi_1) & 0 & 0 \end{bmatrix} \quad \partial \hat{\xi}_2 = \begin{bmatrix} 0 & 4\xi_1 - 1 & 0 \\ 0 & 0 & 0 \\ 0 & -4\xi_3 + 1 & 0 \\ 0 & 4\xi_2 & 0 \\ 0 & -4\xi_2 & 0 \\ 0 & 4(\xi_3 - \xi_1) & 0 \end{bmatrix} \quad \partial \hat{\xi}_3 = \begin{bmatrix} 0 & 0 & 4\xi_1 - 1 \\ 0 & 0 & 0 \\ 0 & 0 & -4\xi_3 + 1 \\ 0 & 0 & 4\xi_2 \\ 0 & 0 & -4\xi_2 \\ 0 & 0 & 4(\xi_3 - \xi_1) \end{bmatrix} \quad (6.74)$$

$$\partial \hat{\eta}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 4\xi_2 - 1 & 0 & 0 \\ -4\xi_3 + 1 & 0 & 0 \\ 4\xi_1 & 0 & 0 \\ 4(\xi_3 - \xi_2) & 0 & 0 \\ -4\xi_1 & 0 & 0 \end{bmatrix} \quad \partial \hat{\eta}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4\xi_2 - 1 & 0 \\ 0 & -4\xi_3 + 1 & 0 \\ 0 & 4\xi_1 & 0 \\ 0 & 4(\xi_3 - \xi_2) & 0 \\ 0 & -4\xi_1 & 0 \end{bmatrix} \quad \partial \hat{\eta}_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 4\xi_2 - 1 \\ 0 & 0 & -4\xi_3 + 1 \\ 0 & 0 & 4\xi_1 \\ 0 & 0 & 4(\xi_3 - \xi_2) \\ 0 & 0 & -4\xi_1 \end{bmatrix}, \quad (6.75)$$

then the derivatives of (6.72) and (6.73) are given by

$$\frac{\partial a_{23}}{\partial \delta_{(n,m)}} = \partial \hat{\xi}_{2(n,m)} \hat{\eta}_3 + \partial \hat{\eta}_{3(n,m)} \hat{\xi}_2 \quad (6.76)$$

$$\frac{\partial a_{31}}{\partial \delta_{(n,m)}} = \partial \hat{\xi}_{3(n,m)} \hat{\eta}_1 + \partial \hat{\eta}_{1(n,m)} \hat{\xi}_3 \quad (6.77)$$

$$\frac{\partial a_{12}}{\partial \delta_{(n,m)}} = \partial \hat{\xi}_{1(n,m)} \hat{\eta}_2 + \partial \hat{\eta}_{2(n,m)} \hat{\xi}_1 \quad (6.78)$$

$$\frac{\partial b_{32}}{\partial \delta_{(n,m)}} = \partial \hat{\xi}_{3(n,m)} \hat{\eta}_2 + \partial \hat{\eta}_{3(n,m)} \hat{\xi}_2 \quad (6.79)$$

$$\frac{\partial b_{13}}{\partial \delta_{(n,m)}} = \partial \hat{\xi}_{1(n,m)} \hat{\eta}_3 + \partial \hat{\eta}_{3(n,m)} \hat{\xi}_1 \quad (6.80)$$

$$\frac{\partial b_{21}}{\partial \delta_{(n,m)}} = \partial \hat{\xi}_{2(n,m)} \hat{\eta}_1 + \partial \hat{\eta}_{1(n,m)} \hat{\xi}_2 \quad (6.81)$$

$$\begin{aligned} \frac{\partial c_1}{\partial \delta_{(n,m)}} &= 2(a_{23} - b_{32}) \left( \frac{\partial a_{23}}{\partial \delta_{(n,m)}} - \frac{\partial b_{32}}{\partial \delta_{(n,m)}} \right) + 2(a_{31} - b_{13}) \left( \frac{\partial a_{31}}{\partial \delta_{(n,m)}} - \frac{\partial b_{13}}{\partial \delta_{(n,m)}} \right) + \\ &\quad 2(a_{12} - b_{21}) \left( \frac{\partial a_{12}}{\partial \delta_{(n,m)}} - \frac{\partial b_{21}}{\partial \delta_{(n,m)}} \right) \end{aligned} \quad (6.82)$$

Substituting the identities in (6.72) and (6.73) into the  $Z$ -direction cosines, (6.52) to (6.54), and using the derivatives (6.76) to (6.82) the derivatives of the  $Z$ -direction cosines are given by,

$$\frac{\partial n_1}{\partial \delta_{(n,m)}} = \frac{\partial}{\partial \delta_{(n,m)}} (a_{23} - b_{32})(c_1)^{-1/2} - \frac{1}{2}(c_1)^{-3/2} \frac{\partial c_1}{\partial \delta_{(n,m)}} (a_{23} - b_{32}) = \quad (6.83)$$

$$\left( \frac{\partial a_{23}}{\partial \delta_{(n,m)}} - \frac{\partial b_{32}}{\partial \delta_{(n,m)}} \right) (c_1)^{-1/2} - \frac{1}{2} (c_1)^{-3/2} \frac{\partial c_1}{\partial \delta_{(n,m)}} (a_{23} - b_{32}) .$$

Similarly,

$$\frac{\partial n_2}{\partial u_{(n,m)}} = \left( \frac{\partial a_{31}}{\partial u_{(n,m)}} - \frac{\partial b_{13}}{\partial u_{(n,m)}} \right) (c_1)^{-1/2} - \frac{1}{2} (c_1)^{-3/2} \frac{\partial c_1}{\partial u_{(n,m)}} (a_{31} - b_{13}) \quad (6.84)$$

$$\frac{\partial n_3}{\partial u_{(n,m)}} = \left( \frac{\partial a_{12}}{\partial u_{(n,m)}} - \frac{\partial b_{21}}{\partial u_{(n,m)}} \right) (c_1)^{-1/2} - \frac{1}{2} (c_1)^{-3/2} \frac{\partial c_1}{\partial u_{(n,m)}} (a_{12} - b_{21}) . \quad (6.85)$$

Defining

$$\|\vec{X}\| = c_2 = (\cos(\theta) n_3)^2 + (\sin(\theta) n_3)^2 + (-\sin(\theta) n_2 - \cos(\theta) n_1)^2 , \quad (6.86)$$

differentiation with respect to the displacements yields,

$$\begin{aligned} \frac{\partial c_2}{\partial \delta_{(n,m)}} &= 2(\cos(\theta) n_3) \left( \cos(\theta) \frac{\partial n_3}{\partial \delta_{(n,m)}} \right) + 2(\sin(\theta) n_3) \left( \sin(\theta) \frac{\partial n_3}{\partial \delta_{(n,m)}} \right) + \\ &\quad 2(-\sin(\theta) n_2 - \cos(\theta) n_1) \left( -\sin(\theta) \frac{\partial n_2}{\partial \delta_{(n,m)}} - \cos(\theta) \frac{\partial n_1}{\partial \delta_{(n,m)}} \right) . \end{aligned} \quad (6.87)$$

Substituting (6.86) into the  $X$ -direction cosines, (6.57) to (6.59), and using the derivatives (6.83) to (6.85) and (6.87), the derivatives of the  $X$ -direction cosines are given by,

$$\frac{\partial l_1}{\partial \delta_{(n,m)}} = \left( \cos(\theta) \frac{\partial n_3}{\partial \delta_{(n,m)}} \right) (c_2)^{-1/2} - \frac{1}{2} (c_2)^{-3/2} \frac{\partial c_2}{\partial \delta_{(n,m)}} (\cos(\theta) n_3) \quad (6.88)$$

$$\frac{\partial l_2}{\partial \delta_{(n,m)}} = \left( \sin(\theta) \frac{\partial n_3}{\partial \delta_{(n,m)}} \right) (c_2)^{-1/2} - \frac{1}{2} (c_2)^{-3/2} \frac{\partial c_2}{\partial \delta_{(n,m)}} (\sin(\theta) n_3) \quad (6.89)$$

$$\begin{aligned} \frac{\partial l_3}{\partial \delta_{(n,m)}} &= \\ &\quad \left( -\sin(\theta) \frac{\partial n_2}{\partial \delta_{(n,m)}} - \cos(\theta) \frac{\partial n_1}{\partial \delta_{(n,m)}} \right) (c_2)^{-1/2} - \\ &\quad \frac{1}{2} (c_2)^{-3/2} \frac{\partial c_2}{\partial \delta_{(n,m)}} (-\sin(\theta) n_2 - \cos(\theta) n_1) . \end{aligned} \quad (6.90)$$

With,

$$\begin{aligned} e_{23} &= n_2 l_3 & e_{31} &= n_3 l_1 & e_{12} &= n_1 l_2 \\ f_{32} &= n_3 l_2 & f_{13} &= n_1 l_3 & f_{21} &= n_2 l_1 \end{aligned} \quad (6.91)$$

$$\|\vec{Y}\| = c_3 = (e_{23} - f_{32})^2 + (e_{31} - f_{13})^2 + (e_{12} - f_{21})^2 , \quad (6.92)$$

the derivatives of (6.91) and (6.92) are given by,

$$\frac{\partial e_{23}}{\partial u_{(n,m)}} = \frac{\partial n_2}{\partial u_{(n,m)}} l_3 + \frac{\partial l_3}{\partial u_{(n,m)}} n_2 \quad (6.93)$$

$$\frac{\partial e_{31}}{\partial u_{(n,m)}} = \frac{\partial n_3}{\partial u_{(n,m)}} l_1 + \frac{\partial l_1}{\partial u_{(n,m)}} n_3 \quad (6.94)$$

$$\frac{\partial e_{12}}{\partial u_{(n,m)}} = \frac{\partial n_1}{\partial u_{(n,m)}} l_2 + \frac{\partial l_2}{\partial u_{(n,m)}} n_1 \quad (6.95)$$

$$\frac{\partial f_{32}}{\partial \delta_{(n,m)}} = \frac{\partial n_3}{\partial \delta_{(n,m)}} l_2 + \frac{\partial l_2}{\partial \delta_{(n,m)}} n_3 \quad (6.96)$$

$$\frac{\partial f_{13}}{\partial \delta_{(n,m)}} = \frac{\partial n_1}{\partial \delta_{(n,m)}} l_3 + \frac{\partial l_3}{\partial \delta_{(n,m)}} n_1 \quad (6.97)$$

$$\frac{\partial f_{21}}{\partial \delta_{(n,m)}} = \frac{\partial n_2}{\partial \delta_{(n,m)}} l_1 + \frac{\partial l_1}{\partial \delta_{(n,m)}} n_2 \quad (6.98)$$

$$\begin{aligned} \frac{\partial c_3}{\partial \delta_{(n,m)}} &= 2(e_{23} - f_{32}) \left( \frac{\partial e_{23}}{\partial \delta_{(n,m)}} - \frac{\partial f_{32}}{\partial \delta_{(n,m)}} \right) + 2(e_{31} - f_{13}) \left( \frac{\partial e_{31}}{\partial \delta_{(n,m)}} - \frac{\partial f_{13}}{\partial \delta_{(n,m)}} \right) + \\ &\quad 2(e_{12} - f_{21}) \left( \frac{\partial e_{12}}{\partial \delta_{(n,m)}} - \frac{\partial f_{21}}{\partial \delta_{(n,m)}} \right). \end{aligned} \quad (6.99)$$

Substituting (6.91) and (6.92) into the  $Y$ -direction cosines, (6.61) to (6.63), and using the derivatives (6.93) to (6.99) the derivatives of the  $Y$ -direction cosines are given by

$$\frac{\partial m_1}{\partial \delta_{(n,m)}} = \left( \frac{\partial e_{23}}{\partial \delta_{(n,m)}} - \frac{\partial f_{32}}{\partial \delta_{(n,m)}} \right) (c_3)^{-1/2} - \frac{1}{2} (c_3)^{-3/2} \frac{\partial c_3}{\partial \delta_{(n,m)}} (e_{23} - f_{32}) \quad (6.100)$$

$$\frac{\partial m_2}{\partial \delta_{(n,m)}} = \left( \frac{\partial e_{31}}{\partial \delta_{(n,m)}} - \frac{\partial f_{13}}{\partial \delta_{(n,m)}} \right) (c_1)^{-1/2} - \frac{1}{2} (c_1)^{-3/2} \frac{\partial c_3}{\partial \delta_{(n,m)}} (e_{31} - f_{13}) \quad (6.101)$$

$$\frac{\partial m_3}{\partial \delta_{(n,m)}} = \left( \frac{\partial e_{12}}{\partial \delta_{(n,m)}} - \frac{\partial f_{21}}{\partial \delta_{(n,m)}} \right) (c_1)^{-1/2} - \frac{1}{2} (c_1)^{-3/2} \frac{\partial c_3}{\partial \delta_{(n,m)}} (e_{12} - f_{21}). \quad (6.102)$$

Revisiting (6.40),  $\frac{\partial T^m}{\partial X_{si}}$  may now be written as,

$$\begin{aligned} \frac{\partial T^m(i,j)}{\partial X_{sj}} &= [dT^m(i,j)] \left[ \frac{\partial \delta}{\partial X_{si}} \right] = \\ &\quad \left[ \frac{\partial T^m(i,j)}{\partial \delta_{1,1}} \quad \frac{\partial T^m(i,j)}{\partial \delta_{1,2}} \quad \frac{\partial T^m(i,j)}{\partial \delta_{1,3}} \quad \dots \quad \frac{\partial T^m(i,j)}{\partial \delta_{6,1}} \quad \frac{\partial T^m(i,j)}{\partial \delta_{6,2}} \quad \frac{\partial T^m(i,j)}{\partial \delta_{6,3}} \right] \begin{bmatrix} \frac{\partial \delta_{1,1}}{\partial X_{si}} \\ \frac{\partial \delta_{1,2}}{\partial X_{si}} \\ \frac{\partial \delta_{1,3}}{\partial X_{si}} \\ \vdots \\ \frac{\partial \delta_{6,1}}{\partial X_{si}} \\ \frac{\partial \delta_{6,2}}{\partial X_{si}} \\ \frac{\partial \delta_{6,3}}{\partial X_{si}} \end{bmatrix} \end{aligned} \quad (6.103)$$

$$\frac{\partial T^m}{\partial X_{si}} = \begin{bmatrix} \frac{\partial T^m(1,1)}{\partial X_{si}} & \frac{\partial T^m(1,2)}{\partial X_{si}} & \frac{\partial T^m(1,3)}{\partial X_{si}} \\ \frac{\partial T^m(2,1)}{\partial X_{si}} & \frac{\partial T^m(2,2)}{\partial X_{si}} & \frac{\partial T^m(2,3)}{\partial X_{si}} \\ \frac{\partial T^m(3,1)}{\partial X_{si}} & \frac{\partial T^m(3,2)}{\partial X_{si}} & \frac{\partial T^m(3,3)}{\partial X_{si}} \end{bmatrix}. \quad (6.104)$$

### 6.3.1.3 B-matrix derivatives

Using a similar form as above, the derivatives of the linear part of the B-matrix with respect to the displacements may be expressed as,

$$\frac{\partial B_0}{\partial \delta_{(n,m)}} = \begin{bmatrix} \frac{\partial B_{0(1,1)}}{\partial \delta_{(n,m)}} & \frac{\partial B_{0(1,2)}}{\partial \delta_{(n,m)}} & \frac{\partial B_{0(1,3)}}{\partial \delta_{(n,m)}} & \dots & \frac{\partial B_{0(1,18)}}{\partial \delta_{(n,m)}} \\ \frac{\partial B_{0(2,1)}}{\partial \delta_{(n,m)}} & \frac{\partial B_{0(2,2)}}{\partial \delta_{(n,m)}} & \frac{\partial B_{0(2,3)}}{\partial \delta_{(n,m)}} & \dots & \frac{\partial B_{0(2,18)}}{\partial \delta_{(n,m)}} \\ \frac{\partial B_{0(3,1)}}{\partial \delta_{(n,m)}} & \frac{\partial B_{0(3,2)}}{\partial \delta_{(n,m)}} & \frac{\partial B_{0(3,3)}}{\partial \delta_{(n,m)}} & \dots & \frac{\partial B_{0(3,18)}}{\partial \delta_{(n,m)}} \end{bmatrix} \quad (6.105)$$

Revisiting equation (3.80) the derivatives of the first term of each row of the linear B-matrix with respect to the global displacements may be expressed explicitly as,

$$\frac{\partial B_{0(1,1)}}{\partial \delta_{(n,m)}} = l_1 \frac{\partial \frac{\partial N_1}{\partial X}}{\partial \delta_{(n,m)}} + \frac{\partial l_1}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial X} \quad (6.106)$$

$$\frac{\partial B_{0(2,1)}}{\partial \delta_{(n,m)}} = m_1 \frac{\partial \frac{\partial N_1}{\partial Y}}{\partial \delta_{(n,m)}} + \frac{\partial m_1}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial Y} \quad (6.107)$$

$$\frac{\partial B_{0(3,1)}}{\partial \delta_{(n,m)}} = l_1 \frac{\partial \frac{\partial N_1}{\partial Y}}{\partial \delta_{(n,m)}} + \frac{\partial l_1}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial Y} + m_1 \frac{\partial \frac{\partial N_1}{\partial X}}{\partial \delta_{(n,m)}} + \frac{\partial m_1}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial X} \quad (6.108)$$

The remaining terms are easily obtained from the preceding procedure. The direction cosines and their partial derivatives have been found in Section 6.3.1.1. Therefore, all that remains is to find the derivatives of  $\frac{\partial N_i}{\partial X}$  and  $\frac{\partial N_i}{\partial Y}$  with respect to the displacements.

With (3.83) and (3.84)

$$\frac{\partial N_i}{\partial X} = \Gamma_{11} \frac{\partial N_i}{\partial \xi} + \Gamma_{12} \frac{\partial N_i}{\partial \eta}. \quad (6.109)$$

The shape function derivatives contained in  $D_{N,\xi,\eta}$  (3.50), are independent of the displacements. Therefore, the derivative of (6.109) with respect to displacement is given by,

$$\frac{\partial \frac{\partial N_i}{\partial X}}{\partial \delta_{(n,m)}} = \frac{\partial \Gamma_{11}}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial \xi} + \frac{\partial \Gamma_{12}}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial \eta} = \frac{\partial \Gamma_{11}}{\partial \delta_{(n,m)}} D_{N,\xi,\eta(1,i)} + \frac{\partial \Gamma_{12}}{\partial \delta_{(n,m)}} D_{N,\xi,\eta(2,i)}. \quad (6.110)$$

Similarly  $\frac{\partial N_i}{\partial Y}$  and the corresponding derivative is given by,

$$\frac{\partial N_i}{\partial Y} = \Gamma_{21} \frac{\partial N_i}{\partial \xi} + \Gamma_{22} \frac{\partial N_i}{\partial \eta} \quad (6.111)$$

$$\frac{\partial \frac{\partial N_i}{\partial Y}}{\partial \delta_{(n,m)}} = \frac{\partial \Gamma_{21}}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial \xi} + \frac{\partial \Gamma_{22}}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial \eta} = \frac{\partial \Gamma_{21}}{\partial \delta_{(n,m)}} D_{N,\xi,\eta(1,i)} + \frac{\partial \Gamma_{22}}{\partial \delta_{(n,m)}} D_{N,\xi,\eta(2,i)}. \quad (6.112)$$

Combining (6.110) to (6.112) and using the form of (3.84)

$$\begin{aligned} \frac{\partial D_{N,X,Y}}{\partial \delta_{(n,m)}} &= \frac{\partial \Gamma}{\partial \delta_{(n,m)}} D_{N,\xi,\eta} = \\ &\left[ \begin{array}{cccccc|cccccc} \frac{\partial \Gamma_{11}}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial \xi} + \frac{\partial \Gamma_{12}}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial \eta} & \dots & \dots & \dots & \dots & \frac{\partial \Gamma_{11}}{\partial \delta_{(n,m)}} \frac{\partial N_6}{\partial \xi} + \frac{\partial \Gamma_{12}}{\partial \delta_{(n,m)}} \frac{\partial N_6}{\partial \eta} \\ \frac{\partial \Gamma_{21}}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial \xi} + \frac{\partial \Gamma_{22}}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial \eta} & \dots & \dots & \dots & \dots & \frac{\partial \Gamma_{21}}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial \xi} + \frac{\partial \Gamma_{22}}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial \eta} \end{array} \right]. \end{aligned} \quad (6.113)$$

From (3.83) the terms of the inverse Jacobian,  $\Gamma$ , expressed in terms of the Jacobian, are

$$\Gamma_{11} = (J_{22}) \times (J_{11}J_{22} - J_{12}J_{21})^{-1} \quad (6.114)$$

$$\Gamma_{12} = (-J_{12}) \times (J_{11}J_{22} - J_{12}J_{21})^{-1} \quad (6.115)$$

$$\Gamma_{21} = (-J_{21}) \times (J_{11}J_{22} - J_{12}J_{21})^{-1} \quad (6.116)$$

$$\Gamma_{22} = (J_{11}) \times (J_{11}J_{22} - J_{12}J_{21})^{-1}. \quad (6.117)$$

The derivatives of (6.114) to (6.117) are,

$$\begin{aligned} \frac{\partial \Gamma_{11}}{\partial \delta_{(n,m)}} &= \frac{\partial J_{22}}{\partial \delta_{(n,m)}} (J_{11}J_{22} - J_{12}J_{21})^{-1} - (J_{11}J_{22} - J_{12}J_{21})^{-2} \frac{\partial}{\partial \delta_{(n,m)}} (J_{11}J_{22} - \\ &J_{12}J_{21}) J_{22} \end{aligned} \quad (6.118)$$

$$\begin{aligned} \frac{\partial \Gamma_{12}}{\partial \delta_{(n,m)}} &= -\frac{\partial J a_{12}}{\partial \delta_{(n,m)}} (J_{11}J_{22} - J_{12}J_{21})^{-1} - (J_{11}J_{22} - J_{12}J_{21})^{-2} \frac{\partial}{\partial \delta_{(n,m)}} (J_{11}J_{22} - \\ &J_{12}J_{21}) - J a_{12} \end{aligned} \quad (6.119)$$

$$\begin{aligned} \frac{\partial \Gamma_{21}}{\partial \delta_{(n,m)}} &= -\frac{\partial J_{21}}{\partial \delta_{(n,m)}} \times (J_{11}J_{22} - J_{12}J_{21})^{-1} - (J_{11}J_{22} - J_{12}J_{21})^{-2} \frac{\partial}{\partial \delta_{(n,m)}} (J_{11}J_{22} - \\ &J_{12}J_{21}) - J_{21} \end{aligned} \quad (6.120)$$

$$\begin{aligned} \frac{\partial \Gamma_{22}}{\partial u_{(n,m)}} &= \frac{\partial J_{11}}{\partial u_{(n,m)}} \times (J_{11}J_{22} - J_{12}J_{21})^{-1} - (J_{11}J_{22} - J_{12}J_{21})^{-2} \frac{\partial}{\partial u_{(n,m)}} (J_{11}J_{22} - \\ &J_{12}J_{21}) J_{11} \end{aligned} \quad (6.121)$$

With (3.67),

$$J_{22} = \hat{\eta} \cdot \hat{Y} = \sum_{i=1}^{i=3} \hat{\eta}_i m_i, \quad (6.122)$$

then,

$$\frac{\partial J_{22}}{\partial \delta_{(n,m)}} = \sum_{i=1}^{i=3} \hat{\eta}_i \frac{\partial m_i}{\partial \delta_{(n,m)}} + \frac{\partial \hat{\eta}_i}{\partial \delta_{(n,m)}} m_i. \quad (6.123)$$

Similarly,

$$J_{11} = \hat{\xi} \cdot \hat{X} = \sum_{i=1}^{i=3} \hat{\xi}_i l_i \quad (6.124)$$

$$\frac{\partial J_{11}}{\partial \delta_{(n,m)}} = \sum_{i=1}^{i=3} \hat{\xi}_i \frac{\partial l_i}{\partial \delta_{(n,m)}} + \frac{\partial \hat{\xi}_i}{\partial \delta_{(n,m)}} l_i \quad (6.125)$$

$$J_{12} = \hat{\xi} \cdot \hat{Y} = \sum_{i=1}^{i=3} \hat{\xi}_i m_i \quad (6.126)$$

$$\frac{\partial J_{12}}{\partial \delta_{(n,m)}} = \sum_{i=1}^{i=3} \hat{\xi}_i \frac{\partial m_i}{\partial \delta_{(n,m)}} + \frac{\partial \hat{\xi}_i}{\partial \delta_{(n,m)}} m_i \quad (6.127)$$

$$J_{21} = \hat{\eta} \cdot \hat{X} = \sum_{i=1}^{i=3} \hat{\eta}_i l_i \quad (6.128)$$

$$\frac{\partial J_{21}}{\partial \delta_{(n,m)}} = \sum_{i=1}^{i=3} \hat{\eta}_i \frac{\partial l_i}{\partial \delta_{(n,m)}} + \frac{\partial \hat{\eta}_i}{\partial \delta_{(n,m)}} l_i \quad (6.129)$$

Following the product rule, the derivative of the determinant is given by,

$$\frac{\partial}{\partial \delta_{(n,m)}} (J_{11} J_{22} - J_{12} J_{21}) = \left( J_{11} \frac{\partial J_{22}}{\partial \delta_{(n,m)}} + \frac{\partial J_{11}}{\partial \delta_{(n,m)}} J_{22} \right) - \left( J_{12} \frac{\partial J_{21}}{\partial \delta_{(n,m)}} + \frac{\partial J_{12}}{\partial \delta_{(n,m)}} J_{21} \right) \quad (6.130)$$

A similar procedure is followed to find the derivatives of the non-linear part of the B-matrix with respect to the displacements. Writing the definition of the non-linear B-matrix, (3.94), explicitly yields,

$$B_L = [A][G] = \quad (6.131)$$

$$\begin{bmatrix} \sum_{i=1}^{i=6} A_{1,i} \times G_{i1} & \sum_{i=1}^{i=6} A_{1,i} \times G_{i2} & \sum_{i=1}^{i=6} A_{1,i} \times G_{i3} & \dots & \sum_{i=1}^{i=6} A_{1,i} \times G_{i18} \\ \sum_{i=1}^{i=6} A_{2,i} \times G_{i1} & \sum_{i=1}^{i=6} A_{2,i} \times G_{i2} & \sum_{i=1}^{i=6} A_{2,i} \times G_{i3} & \dots & \sum_{i=1}^{i=6} A_{2,i} \times G_{i18} \\ \sum_{i=1}^{i=6} A_{3,i} \times G_{i1} & \sum_{i=1}^{i=6} A_{3,i} \times G_{i2} & \sum_{i=1}^{i=6} A_{3,i} \times G_{i3} & \dots & \sum_{i=1}^{i=6} A_{3,i} \times G_{i18} \end{bmatrix}.$$

Therefore, the derivatives of the second order B-matrix terms are given by,

$$\frac{\partial B_L}{\partial \delta_{(n,m)}} = \begin{bmatrix} \sum_{i=1}^{i=6} A_{1,i} \frac{\partial G_{i,1}}{\partial \delta_{(n,m)}} + G_{i,1} \frac{\partial A_{1,i}}{\partial \delta_{(n,m)}} & \sum_{i=1}^{i=6} A_{1,i} \frac{\partial G_{i,2}}{\partial \delta_{(n,m)}} + G_{i,2} \frac{\partial A_{1,i}}{\partial \delta_{(n,m)}} & \dots & \sum_{i=1}^{i=6} A_{1,i} \frac{\partial G_{i,18}}{\partial \delta_{(n,m)}} + G_{i,18} \frac{\partial A_{1,i}}{\partial \delta_{(n,m)}} \\ \sum_{i=1}^{i=6} A_{2,i} \frac{\partial G_{i,1}}{\partial \delta_{(n,m)}} + G_{i,1} \frac{\partial A_{2,i}}{\partial \delta_{(n,m)}} & \sum_{i=1}^{i=6} A_{2,i} \frac{\partial G_{i,2}}{\partial \delta_{(n,m)}} + G_{i,2} \frac{\partial A_{2,i}}{\partial \delta_{(n,m)}} & \dots & \sum_{i=1}^{i=6} A_{2,i} \frac{\partial G_{i,18}}{\partial \delta_{(n,m)}} + G_{i,18} \frac{\partial A_{2,i}}{\partial \delta_{(n,m)}} \\ \sum_{i=1}^{i=6} A_{3,i} \frac{\partial G_{i,1}}{\partial \delta_{(n,m)}} + G_{i,1} \frac{\partial A_{3,i}}{\partial \delta_{(n,m)}} & \sum_{i=1}^{i=6} A_{3,i} \frac{\partial G_{i,2}}{\partial \delta_{(n,m)}} + G_{i,2} \frac{\partial A_{3,i}}{\partial \delta_{(n,m)}} & \dots & \sum_{i=1}^{i=6} A_{3,i} \frac{\partial G_{i,18}}{\partial \delta_{(n,m)}} + G_{i,18} \frac{\partial A_{3,i}}{\partial \delta_{(n,m)}} \end{bmatrix}. \quad (6.132)$$

Using the form of (6.105), the derivatives of the G-matrix terms with respect to the displacements may be expressed as,

$$\frac{\partial G}{\partial \delta_{(n,m)}} = \begin{bmatrix} \frac{\partial G_{(1,1)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(1,2)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(1,3)}}{\partial \delta_{(n,m)}} & \dots & \frac{\partial G_{(1,18)}}{\partial \delta_{(n,m)}} \\ \frac{\partial G_{(2,1)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(2,2)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(2,3)}}{\partial \delta_{(n,m)}} & \dots & \frac{\partial G_{(2,18)}}{\partial \delta_{(n,m)}} \\ \frac{\partial G_{(3,1)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(3,2)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(3,3)}}{\partial \delta_{(n,m)}} & \dots & \frac{\partial G_{(3,18)}}{\partial \delta_{(n,m)}} \\ \frac{\partial G_{(4,1)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(4,2)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(4,3)}}{\partial \delta_{(n,m)}} & \dots & \frac{\partial G_{(4,18)}}{\partial \delta_{(n,m)}} \\ \frac{\partial G_{(5,1)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(5,2)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(5,3)}}{\partial \delta_{(n,m)}} & \dots & \frac{\partial G_{(5,18)}}{\partial \delta_{(n,m)}} \\ \frac{\partial G_{(6,1)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(6,2)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(6,3)}}{\partial \delta_{(n,m)}} & \dots & \frac{\partial G_{(6,18)}}{\partial \delta_{(n,m)}} \\ \frac{\partial G_{(7,1)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(7,2)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(7,3)}}{\partial \delta_{(n,m)}} & \dots & \frac{\partial G_{(7,18)}}{\partial \delta_{(n,m)}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial G_{(18,1)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(18,2)}}{\partial \delta_{(n,m)}} & \frac{\partial G_{(18,3)}}{\partial \delta_{(n,m)}} & \dots & \frac{\partial G_{(18,18)}}{\partial \delta_{(n,m)}} \end{bmatrix}. \quad (6.133)$$

The derivatives of the first term of each row of the G-matrix with respect to the global displacements may be expressed explicitly as,

$$\frac{\partial G_{(1,1)}}{\partial \delta_{(n,m)}} = l_1 \frac{\partial \frac{\partial N_1}{\partial X}}{\partial \delta_{(n,m)}} + \frac{\partial l_1}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial X} \quad (6.134)$$

$$\frac{\partial G_{(2,1)}}{\partial \delta_{(n,m)}} = m_1 \frac{\partial \frac{\partial N_1}{\partial X}}{\partial \delta_{(n,m)}} + \frac{\partial m_1}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial X} \quad (6.135)$$

$$\frac{\partial G_{(3,1)}}{\partial \delta_{(n,m)}} = n_1 \frac{\partial \frac{\partial N_1}{\partial X}}{\partial \delta_{(n,m)}} + \frac{\partial n_1}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial X} \quad (6.136)$$

$$\frac{\partial G_{(1,1)}}{\partial \delta_{(n,m)}} = l_1 \frac{\partial \frac{\partial N_1}{\partial Y}}{\partial \delta_{(n,m)}} + \frac{\partial l_1}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial Y} \quad (6.137)$$

$$\frac{\partial G_{(2,1)}}{\partial \delta_{(n,m)}} = m_1 \frac{\partial \frac{\partial N_1}{\partial Y}}{\partial \delta_{(n,m)}} + \frac{\partial m_1}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial Y} \quad (6.138)$$

$$\frac{\partial G_{(3,1)}}{\partial \delta_{(n,m)}} = n_1 \frac{\partial \frac{\partial N_1}{\partial Y}}{\partial \delta_{(n,m)}} + \frac{\partial n_1}{\partial \delta_{(n,m)}} \frac{\partial N_1}{\partial Y} \quad (6.139)$$

As before, the remaining terms are easily obtained in the same way, including the partial derivatives of the direction cosines (e.g. see Section 6.3.1.1) and the derivatives of  $\frac{\partial N_i}{\partial X}$  and  $\frac{\partial N_i}{\partial Y}$  with respect to the displacements.

The final derivatives to be found are those of matrix A. inspection of (3.85) yields,

$$\frac{\partial A}{\partial \delta_{(n,m)}} = \begin{bmatrix} \frac{\partial \frac{\partial U}{\partial X}}{\partial \delta_{(n,m)}} & \frac{\partial \frac{\partial V}{\partial X}}{\partial \delta_{(n,m)}} & \frac{\partial \frac{\partial W}{\partial X}}{\partial \delta_{(n,m)}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial \frac{\partial U}{\partial Y}}{\partial \delta_{(n,m)}} & \frac{\partial \frac{\partial V}{\partial Y}}{\partial \delta_{(n,m)}} & \frac{\partial \frac{\partial W}{\partial Y}}{\partial \delta_{(n,m)}} \\ \frac{\partial \frac{\partial U}{\partial Y}}{\partial \delta_{(n,m)}} & \frac{\partial \frac{\partial V}{\partial Y}}{\partial \delta_{(n,m)}} & \frac{\partial \frac{\partial W}{\partial Y}}{\partial \delta_{(n,m)}} & \frac{\partial \frac{\partial U}{\partial X}}{\partial \delta_{(n,m)}} & \frac{\partial \frac{\partial V}{\partial X}}{\partial \delta_{(n,m)}} & \frac{\partial \frac{\partial W}{\partial X}}{\partial \delta_{(n,m)}} \end{bmatrix} \quad (6.140)$$

To find the terms of  $\frac{\partial A}{\partial \delta_{(n,m)}}$  the derivatives of  $\{\Delta\}$  are found with respect to the displacements, as in,

$$\frac{\partial \Delta}{\partial \delta_{(n,m)}} = \begin{bmatrix} \frac{\partial \frac{\partial U}{\partial X}}{\partial \delta_{(n,m)}} \\ \frac{\partial \frac{\partial V}{\partial X}}{\partial \delta_{(n,m)}} \\ \frac{\partial \frac{\partial W}{\partial X}}{\partial \delta_{(n,m)}} \\ \frac{\partial \frac{\partial U}{\partial Y}}{\partial \delta_{(n,m)}} \\ \frac{\partial \frac{\partial V}{\partial Y}}{\partial \delta_{(n,m)}} \\ \frac{\partial \frac{\partial W}{\partial Y}}{\partial \delta_{(n,m)}} \end{bmatrix} = \frac{\partial G}{\partial \delta_{(n,m)}} \{\delta_i\} + G \frac{\partial \{\delta_i\}}{\partial \delta_{(n,m)}} \quad (6.141)$$

Therefore, the derivatives of  $\{\Delta\}$  may be expressed as,

$$\frac{\partial \frac{\partial U}{\partial X}}{\partial \delta_{(n,m)}} = \sum_{i=1}^{i=6} \left[ \left( \frac{\partial l_1}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial X} + l_1 \frac{\partial \frac{\partial N_i}{\partial X}}{\partial \delta_{(n,m)}} \right) \delta_{(i,1)} + l_1 \frac{\partial N_i}{\partial X} \frac{\partial \delta_{(i,1)}}{\partial \delta_{(n,m)}} \right] + \sum_{i=1}^{i=6} \left[ \left( \frac{\partial l_2}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial X} + l_2 \frac{\partial \frac{\partial N_i}{\partial X}}{\partial \delta_{(n,m)}} \right) \delta_{(i,2)} + l_2 \frac{\partial N_i}{\partial X} \frac{\partial \delta_{(i,2)}}{\partial \delta_{(n,m)}} \right] + \sum_{i=1}^{i=6} \left[ \left( \frac{\partial l_3}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial X} + l_3 \frac{\partial \frac{\partial N_i}{\partial X}}{\partial \delta_{(n,m)}} \right) \delta_{(i,3)} + l_3 \frac{\partial N_i}{\partial X} \frac{\partial \delta_{(i,3)}}{\partial \delta_{(n,m)}} \right] \quad (6.142)$$

$$\frac{\partial \frac{\partial V}{\partial X}}{\partial \delta_{(n,m)}} = \sum_{i=1}^{i=6} \left[ \left( \frac{\partial m_1}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial X} + m_1 \frac{\partial \frac{\partial N_i}{\partial X}}{\partial \delta_{(n,m)}} \right) \delta_{(i,1)} + m_1 \frac{\partial N_i}{\partial X} \frac{\partial \delta_{(i,1)}}{\partial \delta_{(n,m)}} \right] + \sum_{i=1}^{i=6} \left[ \left( \frac{\partial m_2}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial X} + m_2 \frac{\partial \frac{\partial N_i}{\partial X}}{\partial \delta_{(n,m)}} \right) \delta_{(i,2)} + m_2 \frac{\partial N_i}{\partial X} \frac{\partial \delta_{(i,2)}}{\partial \delta_{(n,m)}} \right] + \sum_{i=1}^{i=6} \left[ \left( \frac{\partial m_3}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial X} + m_3 \frac{\partial \frac{\partial N_i}{\partial X}}{\partial \delta_{(n,m)}} \right) \delta_{(i,3)} + m_3 \frac{\partial N_i}{\partial X} \frac{\partial \delta_{(i,3)}}{\partial \delta_{(n,m)}} \right] \quad (6.143)$$

$$\frac{\partial \frac{\partial U}{\partial X}}{\partial \delta_{(n,m)}} = \sum_{i=1}^{i=6} \left[ \left( \frac{\partial n_1}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial X} + n_1 \frac{\partial \frac{\partial N_i}{\partial X}}{\partial \delta_{(n,m)}} \right) \delta_{(i,1)} + n_1 \frac{\partial N_i}{\partial X} \frac{\partial \delta_{(i,1)}}{\partial \delta_{(n,m)}} \right] + \sum_{i=1}^{i=6} \left[ \left( \frac{\partial n_2}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial X} + n_2 \frac{\partial \frac{\partial N_i}{\partial X}}{\partial \delta_{(n,m)}} \right) \delta_{(i,2)} + n_2 \frac{\partial N_i}{\partial X} \frac{\partial \delta_{(i,2)}}{\partial \delta_{(n,m)}} \right] + \sum_{i=1}^{i=6} \left[ \left( \frac{\partial n_3}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial X} + n_3 \frac{\partial \frac{\partial N_i}{\partial X}}{\partial \delta_{(n,m)}} \right) \delta_{(i,3)} + n_3 \frac{\partial N_i}{\partial X} \frac{\partial \delta_{(i,3)}}{\partial \delta_{(n,m)}} \right] \quad (6.144)$$

$$\frac{\partial \frac{\partial U}{\partial Y}}{\partial \delta_{(n,m)}} = \sum_{i=1}^{i=6} \left[ \left( \frac{\partial l_1}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial Y} + l_1 \frac{\partial \frac{\partial N_i}{\partial Y}}{\partial \delta_{(n,m)}} \right) \delta_{(i,1)} + l_1 \frac{\partial N_i}{\partial Y} \frac{\partial \delta_{(i,1)}}{\partial \delta_{(n,m)}} \right] + \sum_{i=1}^{i=6} \left[ \left( \frac{\partial l_2}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial Y} + l_2 \frac{\partial \frac{\partial N_i}{\partial Y}}{\partial \delta_{(n,m)}} \right) \delta_{(i,2)} + l_2 \frac{\partial N_i}{\partial Y} \frac{\partial \delta_{(i,2)}}{\partial \delta_{(n,m)}} \right] + \sum_{i=1}^{i=6} \left[ \left( \frac{\partial l_3}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial Y} + l_3 \frac{\partial \frac{\partial N_i}{\partial Y}}{\partial \delta_{(n,m)}} \right) \delta_{(i,3)} + l_3 \frac{\partial N_i}{\partial Y} \frac{\partial \delta_{(i,3)}}{\partial \delta_{(n,m)}} \right] \quad (6.145)$$

$$\frac{\partial \frac{\partial V}{\partial Y}}{\partial \delta_{(n,m)}} = \sum_{i=1}^{i=6} \left[ \left( \frac{\partial m_1}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial Y} + m_1 \frac{\partial \frac{\partial N_i}{\partial Y}}{\partial \delta_{(n,m)}} \right) \delta_{(i,1)} + m_1 \frac{\partial N_i}{\partial Y} \frac{\partial \delta_{(i,1)}}{\partial \delta_{(n,m)}} \right] + \sum_{i=1}^{i=6} \left[ \left( \frac{\partial m_2}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial Y} + m_2 \frac{\partial \frac{\partial N_i}{\partial Y}}{\partial \delta_{(n,m)}} \right) \delta_{(i,2)} + m_2 \frac{\partial N_i}{\partial Y} \frac{\partial \delta_{(i,2)}}{\partial \delta_{(n,m)}} \right] + \sum_{i=1}^{i=6} \left[ \left( \frac{\partial m_3}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial Y} + m_3 \frac{\partial \frac{\partial N_i}{\partial Y}}{\partial \delta_{(n,m)}} \right) \delta_{(i,3)} + m_3 \frac{\partial N_i}{\partial Y} \frac{\partial \delta_{(i,3)}}{\partial \delta_{(n,m)}} \right] \quad (6.146)$$

$$\frac{\partial \frac{\partial U}{\partial Y}}{\partial \delta_{(n,m)}} = \sum_{i=1}^{i=6} \left[ \left( \frac{\partial n_1}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial Y} + n_1 \frac{\partial \frac{\partial N_i}{\partial Y}}{\partial \delta_{(n,m)}} \right) \delta_{(i,1)} + n_1 \frac{\partial N_i}{\partial Y} \frac{\partial \delta_{(i,1)}}{\partial \delta_{(n,m)}} \right] + \sum_{i=1}^{i=6} \left[ \left( \frac{\partial n_2}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial Y} + n_2 \frac{\partial \frac{\partial N_i}{\partial Y}}{\partial \delta_{(n,m)}} \right) \delta_{(i,2)} + n_2 \frac{\partial N_i}{\partial Y} \frac{\partial \delta_{(i,2)}}{\partial \delta_{(n,m)}} \right] + \sum_{i=1}^{i=6} \left[ \left( \frac{\partial n_3}{\partial \delta_{(n,m)}} \frac{\partial N_i}{\partial Y} + n_3 \frac{\partial \frac{\partial N_i}{\partial Y}}{\partial \delta_{(n,m)}} \right) \delta_{(i,3)} + n_3 \frac{\partial N_i}{\partial Y} \frac{\partial \delta_{(i,3)}}{\partial \delta_{(n,m)}} \right] \quad (6.147)$$

From equation (6.36),  $\frac{\partial B_0}{\partial X_{si}}$  may now be written as,

$$\frac{\partial B_0(i,j)}{\partial X_{si}} = [dB_0(i,j)] \left[ \frac{\partial \delta}{\partial X_{si}} \right] = \begin{bmatrix} \frac{\partial B_0(i,j)}{\partial \delta_{1,1}} & \frac{\partial B_0(i,j)}{\partial \delta_{1,2}} & \frac{\partial B_0(i,j)}{\partial \delta_{1,3}} & \dots & \frac{\partial B_0(i,j)}{\partial \delta_{6,1}} & \frac{\partial B_0(i,j)}{\partial \delta_{6,2}} & \frac{\partial B_0(i,j)}{\partial \delta_{6,3}} \end{bmatrix} \begin{bmatrix} \frac{\partial u_{1,1}}{\partial X_{si}} \\ \frac{\partial u_{1,2}}{\partial X_{si}} \\ \frac{\partial u_{1,3}}{\partial X_{si}} \\ \vdots \\ \frac{\partial u_{6,1}}{\partial X_{si}} \\ \frac{\partial u_{6,2}}{\partial X_{si}} \\ \frac{\partial u_{6,3}}{\partial X_{si}} \end{bmatrix}, \quad (6.148)$$

and,

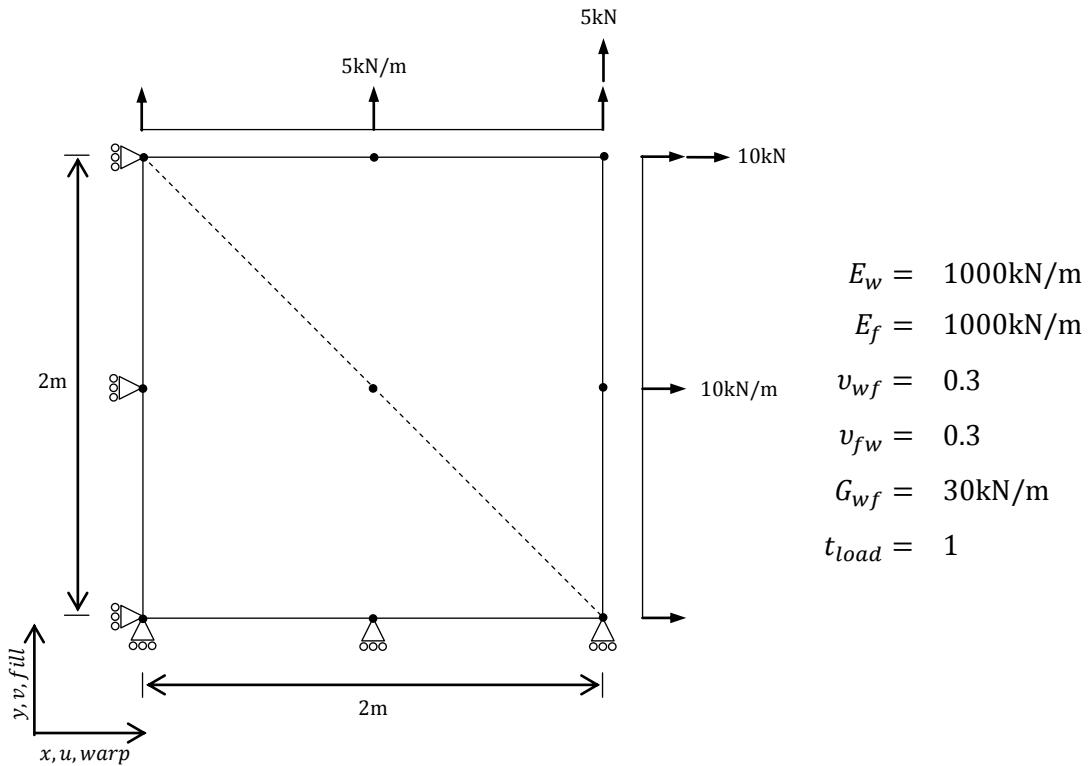
$$\frac{\partial B_L(i,j)}{\partial X_{si}} = [dB_L(i,j)] \left[ \frac{\partial u}{\partial X_{si}} \right] = \quad (6.149)$$

$$\frac{\partial B}{\partial X_{si}} = \begin{bmatrix} \frac{\partial B_L(i,j)}{\partial u_{1,1}} & \frac{\partial B_L(i,j)}{\partial u_{1,2}} & \frac{\partial B_L(i,j)}{\partial u_{1,3}} & \dots & \frac{\partial B_L(i,j)}{\partial u_{6,1}} & \frac{\partial B_L(i,j)}{\partial u_{6,2}} & \frac{\partial B_L(i,j)}{\partial u_{6,3}} \end{bmatrix} \begin{bmatrix} \frac{\partial u_{1,1}}{\partial X_{si}} \\ \frac{\partial u_{1,2}}{\partial X_{si}} \\ \frac{\partial u_{1,3}}{\partial X_{si}} \\ \vdots \\ \frac{\partial u_{6,1}}{\partial X_{si}} \\ \frac{\partial u_{6,2}}{\partial X_{si}} \\ \frac{\partial u_{6,3}}{\partial X_{si}} \end{bmatrix}$$

$$\frac{\partial B}{\partial X_{si}} = \begin{bmatrix} \frac{\partial B_0(1,1)}{\partial X_{si}} + \frac{\partial B_L(1,1)}{\partial X_{si}} & \dots & \frac{\partial B_0(1,18)}{\partial X_{si}} + \frac{\partial B_L(1,18)}{\partial X_{si}} \\ \frac{\partial B_0(2,1)}{\partial X_{si}} + \frac{\partial B_L(2,1)}{\partial X_{si}} & \dots & \frac{\partial B_0(2,18)}{\partial X_{si}} + \frac{\partial B_L(2,18)}{\partial X_{si}} \\ \frac{\partial B_0(3,1)}{\partial X_{si}} + \frac{\partial B_L(3,1)}{\partial X_{si}} & \dots & \frac{\partial B_0(3,18)}{\partial X_{si}} + \frac{\partial B_L(3,18)}{\partial X_{si}} \end{bmatrix}. \quad (6.150)$$

### 6.3.2 Validation of partial derivatives using finite differences

The analytical derivation for the limit state function sensitivities (partial derivatives) are validated using the finite difference method (FDM). A 2 metre square patch of fabric is discretised into 2 elements and restrained along two edges by rolling restraints illustrated in Figure 6-1. A 10kN/m uniformly distributed load (UDL) is applied in the x-direction and with 5kN/m applied in the y-direction along the free edge. Point loads of 5kN in the x-direction and 10kN in the y-direction are also applied in order to fix the position of the extreme stresses and displacements and to induce shear stress. Without the application of a point load a uniform stress is induced and the position of the extreme stresses and displacement is affected by small systematic errors. This leads to errors in the finite difference method. The warp material direction is aligned with the global  $x$ -direction and the fill material direction with the global  $y$ -direction. No prestress is applied to the material. Therefore, no form finding is required. An initial analysis to equilibrium is performed with all variable set as in Figure 6-1.



**Figure 6-1: Finite element patch for finite difference reliability simulation**

For each of the 8 statistical variables, a perturbation equal to a percentage of the variable's initial value is added while holding all other variables constant. The analysis is re-run to reach equilibrium. The finite difference derivatives for the required limit state, maximum warp stress, maximum fill stress, minimum principle stress or displacement may then be calculated. In general the finite difference method may be expressed as follows,

$$\frac{\partial f(a)}{\partial a} \approx \frac{f(a+h) - f(a)}{h} \quad (6.151)$$

where  $f$  is the function of interest and  $a$  is the statistical variable of interest,  $h$  is the value of perturbation equal to a percentage of the initial value of the statistical variable.

The magnitude of the perturbation ( $h$ ) has an impact on the accuracy of the numerically estimated partial derivative obtained using FDM. Each FDM limit state sensitivity has therefore been calculated using three perturbations – 0.1%, 0.001% and 0.00001%. The results are summarised in Table 6-1, 6-2 and 6-3, respectively. It should also be noted that the accuracy of the analysis has an impact on the derivative values derived using FDM.

As shown in Table 6-1 a perturbation of 0.1% produces generally poor agreement between the analytical and FDM limit state function sensitivities, indicating that it is potentially too great or that the analytical sensitivities are incorrect.

|            | $\frac{\partial(\sigma_{per}^w - \sigma_{max}^w)}{\partial X_{si}}$ : max stress warp      |                     |        | $\frac{\partial(\sigma_{per}^f - \sigma_{max}^f)}{\partial X_{si}}$ : max stress fill |                     |        |
|------------|--|---------------------|--------|---|---------------------|--------|
| $X_{si}$   | Finite difference method   | Analytical approach | %error | Finite difference method  | Analytical approach | %error |
| $E_w$      | -1.10E-03  | -1.17E-03           | 6.60%  | 2.60E-04  | 3.21E-04            | 23.40% |
| $E_f$      | -1.84E-04  | -1.96E-04           | 6.38%  | -1.28E-03   | -1.32E-03           | 3.03%  |
| $v_{fw}$   | -1.39E-01  | -1.35E-01           | -2.65% | -8.95E-01   | -8.87E-01           | -0.90% |
| $v_{wf}$   | -1.58E+00  | -1.57E+00           | -0.92% | 6.23E-02  | 6.55E-02            | 5.24%  |
| $G_{wf}$   | 6.80E-02   | 7.00E-02            | 3.02%  | 6.51E-02  | 6.73E-02            | 3.34%  |
| $t_{load}$ | -3.54E+01  | -3.53E+01           | -0.24% | -1.81E+01   | -1.80E+01           | -0.60% |
|            | $\frac{\partial(\sigma_{min}^p - \sigma_{per}^p)}{\partial X_{si}}$ : min principal stress |                     |        | $\frac{\partial(\delta_{per} - \delta_{max})}{\partial X_{si}}$ : max displacement    |                     |        |
| $X_{si}$   | Finite difference method   | Analytical approach | %error | Finite difference method  | Analytical approach | %error |
| $E_w$      | -3.76E-03  | -4.08E-03           | 8.31%  | 5.95E-05  | 6.50E-05            | 9.19%  |
| $E_f$      | -4.58E-04  | -5.02E-04           | 9.56%  | -4.18E-06   | -4.60E-06           | 10.18% |
| $v_{fw}$   | 8.03E-01   | 8.02E-01            | -0.13% | 1.08E-02  | 1.08E-02            | -0.01% |
| $v_{wf}$   | -2.33E-01  | -2.36E-01           | 1.49%  | -1.67E-03   | -1.66E-03           | -0.16% |
| $G_{wf}$   | 1.68E-01   | 1.73E-01            | 2.95%  | 2.52E-04  | 2.60E-04            | 2.82%  |
| $t_{load}$ | -7.21E+00  | -7.15E+00           | -0.84% | -6.83E-02   | -6.82E-02           | -0.09% |

Table 6-1: Sensitivities, finite difference method using perturbation of 0.1%

|            | $\frac{\partial(\sigma_{per}^w - \sigma_{max}^w)}{\partial X_{si}}$ : max stress warp      |                     |        | $\frac{\partial(\sigma_{per}^f - \sigma_{max}^f)}{\partial X_{si}}$ : max stress fill |                     |        |
|------------|--|---------------------|--------|---|---------------------|--------|
| $X_{si}$   | Finite difference method   | Analytical approach | %error | Finite difference method  | Analytical approach | %error |
| $E_w$      | -1.17E-03  | -1.17E-03           | 0.08%  | 3.20E-04  | 3.21E-04            | 0.27%  |
| $E_f$      | -1.96E-04  | -1.96E-04           | 0.01%  | -1.32E-03   | -1.32E-03           | 0.04%  |
| $v_{fw}$   | -1.33E-01  | -1.35E-01           | 1.61%  | -8.89E-01   | -8.87E-01           | -0.23% |
| $v_{wf}$   | -1.57E+00  | -1.57E+00           | 0.05%  | 6.46E-02  | 6.55E-02            | 1.39%  |
| $G_{wf}$   | 7.00E-02   | 7.00E-02            | 0.04%  | 6.73E-02  | 6.73E-02            | 0.02%  |
| $t_{load}$ | -3.53E+01  | -3.53E+01           | 0.00%  | -1.80E+01   | -1.80E+01           | -0.01% |
|            | $\frac{\partial(\sigma_{min}^p - \sigma_{per}^p)}{\partial X_{si}}$ : min principal stress |                     |        | $\frac{\partial(\delta_{per} - \delta_{max})}{\partial X_{si}}$ : max displacement    |                     |        |
| $X_{si}$   | Finite difference method   | Analytical approach | %error | Finite difference method  | Analytical approach | %error |
| $E_w$      | -4.07E-03  | -4.08E-03           | 0.13%  | 6.49E-05  | 6.50E-05            | 0.09%  |
| $E_f$      | -5.01E-04  | -5.02E-04           | 0.12%  | -4.60E-06   | -4.60E-06           | 0.10%  |
| $v_{fw}$   | 8.05E-01   | 8.02E-01            | -0.28% | 1.08E-02  | 1.08E-02            | -0.02% |
| $v_{wf}$   | -2.35E-01  | -2.36E-01           | 0.31%  | -1.66E-03   | -1.66E-03           | 0.06%  |
| $G_{wf}$   | 1.73E-01   | 1.73E-01            | 0.07%  | 2.60E-04  | 2.60E-04            | 0.03%  |
| $t_{load}$ | -7.14E+00  | -7.15E+00           | 0.07%  | -6.82E-02   | -6.82E-02           | 0.00%  |

Table 6-2: Sensitivities, finite difference method using perturbation of 0.001%

|            | $\frac{\partial(\sigma_{per}^w - \sigma_{max}^w)}{\partial X_{si}}$ : max stress warp      |                     |        | $\frac{\partial(\sigma_{per}^f - \sigma_{max}^f)}{\partial X_{si}}$ : max stress fill |                     |        |
|------------|--|---------------------|--------|---|---------------------|--------|
| $X_{si}$   | Finite difference method   | Analytical approach | %error | Finite difference method  | Analytical approach | %error |
| $E_w$      | -1.17E-03  | -1.17E-03           | 0.02%  | 3.20E-04  | 3.21E-04            | 0.06%  |
| $E_f$      | -1.96E-04  | -1.96E-04           | -0.05% | -1.32E-03   | -1.32E-03           | 0.01%  |
| $v_{fw}$   | -1.33E-01  | -1.35E-01           | 1.55%  | -8.89E-01   | -8.87E-01           | -0.21% |
| $v_{wf}$   | -1.57E+00  | -1.57E+00           | 0.06%  | 6.47E-02  | 6.55E-02            | 1.33%  |
| $G_{wf}$   | 7.00E-02   | 7.00E-02            | 0.01%  | 6.73E-02  | 6.73E-02            | -0.01% |
| $t_{load}$ | -3.53E+01  | -3.53E+01           | 0.00%  | -1.80E+01   | -1.80E+01           | 0.00%  |
|            | $\frac{\partial(\sigma_{min}^p - \sigma_{per}^p)}{\partial X_{si}}$ : min principal stress |                     |        | $\frac{\partial(\delta_{per} - \delta_{max})}{\partial X_{si}}$ : max displacement    |                     |        |
| $X_{si}$   | Finite difference method   | Analytical approach | %error | Finite difference method  | Analytical approach | %error |
| $E_w$      | -4.08E-03  | -4.08E-03           | 0.04%  | 6.50E-05  | 6.50E-05            | 0.00%  |
| $E_f$      | -5.01E-04  | -5.02E-04           | 0.03%  | -4.60E-06   | -4.60E-06           | 0.00%  |
| $v_{fw}$   | 8.04E-01   | 8.02E-01            | -0.25% | 1.08E-02  | 1.08E-02            | -0.02% |
| $v_{wf}$   | -2.36E-01  | -2.36E-01           | 0.29%  | -1.66E-03   | -1.66E-03           | 0.06%  |
| $G_{wf}$   | 1.73E-01   | 1.73E-01            | 0.04%  | 2.60E-04  | 2.60E-04            | 0.00%  |
| $t_{load}$ | -7.14E+00  | -7.15E+00           | 0.08%  | -6.82E-02   | -6.82E-02           | 0.00%  |

Table 6-3: Sensitivities, finite difference method using perturbation of 0.00001%

The former statement is supported by the results given in Table 6-2 where a perturbation of 0.001% produces good agreement (less than 1% error) between the analytical approach and FDM. This is shown across all limit states and all statistical variables except  $v_{fw}$  for limit state 1 and  $v_{wf}$  for limit state 2. The greater persistent error exhibited by the Poisson's ratio sensitivities at the lower perturbation is in part due to the greater impact of these variables in comparison to Young's modulus and the shear modulus. As highlighted in equations (6.46) and (6.47), unlike in the case of Young's moduli, a change in either  $v_{fw}$  or  $v_{wf}$  will result in a change in the stress in both material directions. As a result, the relationship between stress and strain with respect to the Poisson's ratios is more complex and interdependent than the other statistical variables. This is highlighted by the  $\frac{\partial E_{1,2}}{\partial v_{fw}}$  term for limit state 1 and the  $\frac{\partial E_{2,1}}{\partial v_{wf}}$  term for limit state 2.

It is also the case that the analytical derivatives only account for the sensitivity of the stress at the Gauss point where the limit state stress or displacement occurs. The stress and strain values at all other Gauss points are assumed to be fixed and therefore have no impact on the limit state function. When using the finite difference method the statistical variables are updated at all Gauss points and the analysis is re-run. Therefore, the

behaviour of the entire structure is taken into account in equation(6.151). This leads to a greater difference between analytical and FDM for the variables which induce a more complex and interdependent response.

In Table 6-3 it is clear that the agreement between analytical and FDM sensitivities remains good with respect to  $E_w$ ,  $E_f$ ,  $G_{wf}$  and  $t_{load}$  and in most cases the error decreases with a further reduction in perturbation to 0.00001%. However, the sensitivities with respect to  $v_{xy}$  and  $v_{yx}$  still show poorer agreement but still less than an acceptable value of 2%. This indicates that the difference is not due to errors caused by the size of perturbation.

### 6.3.3 Validation using Monte Carlo Simulation

Further validation of the reliability analysis is undertaken using Monte Carlo simulation. The simulation is the same as that used above in Section 6.15 with the exception of the loading. UDLs of 20kN/m in the x-direction and 30kN/m in the y-direction along with additional point load of 10kN in the top corner node are applied. The loading has been increased in order to increase the probability of failure and reduce the number of simulation required for the Monte Carlo simulation..

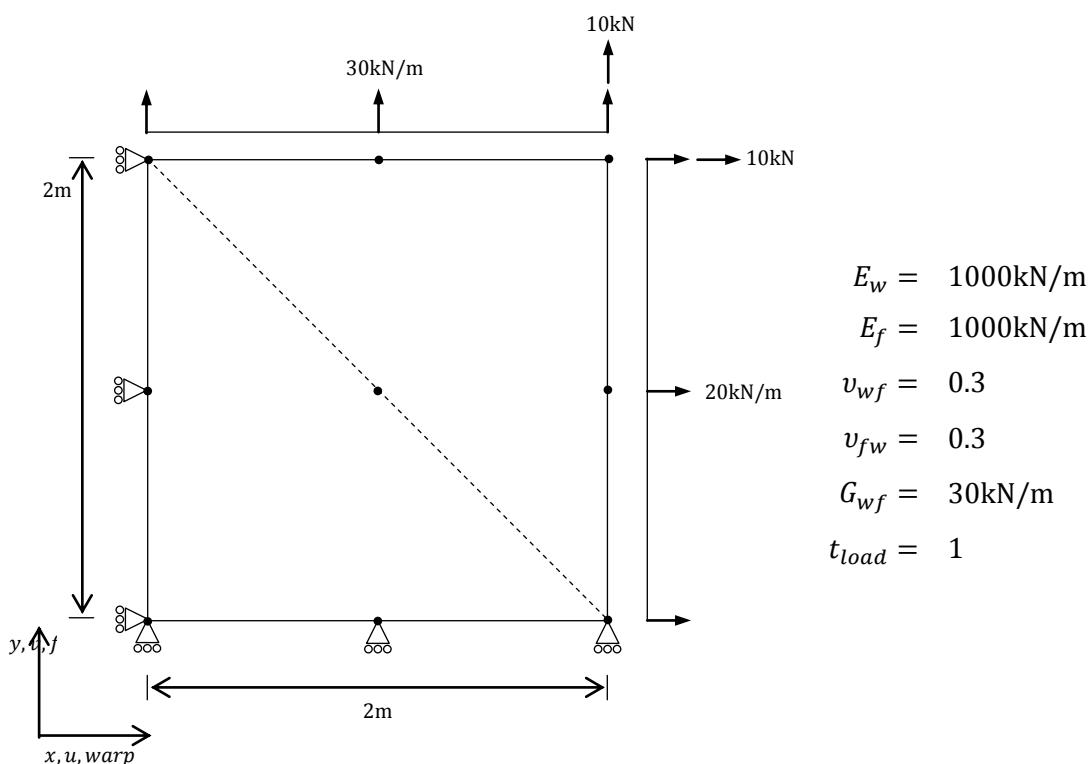


Figure 6-2: Finite element patch for Monte Carlo simulation

Initially an analytical reliability analysis is run for each limit state and the probability of failure is found from a standard normal table and the safety index  $\beta$  as shown in Table 6-4.

|                | $G_1(X_{si}) = \sigma_{per}^w - \sigma_{max}^w$ | $G_2(X_{si}) = \sigma_{per}^f - \sigma_{max}^f$ | $G_3(X_{si}) = \sigma_{min}^p - \sigma_{per}^p$ | $G_4(X_{si}) = D_{al} - D_{max}$ |
|----------------|---|---|---|----------------------------------|
| $\beta$        | 2.724   | 2.160   | 1.437   | 1.323                            |
| $\Phi(-\beta)$ | 0.3264e-2                                       | 0.1539e-1                                       | 0.7494e-1                                       | 0.9342e-1                        |

**Table 6-4: Analytically Derived HL\_RF Probability of Failure**

These values of  $\beta$  are outside the accepted range for a structure according to Eurocode 0 (EC0), where the minimum value is 3.8. The probability of failure for the structure is, therefore, greater than would be permitted by EC0. However, for an anticipated very small probability of failure a very large number of Monte Carlo simulations would be required to predict the failure probability with sufficient accuracy. It is, therefore, for the purposes of validation, to adopt a larger failure probability at this stage.

A Monte Carlo simulation is run using normally distributed random variables generated using the Box Muller method. Variables  $u_1$  and  $u_2$  are generated in pairs using random numbers,  $r_1$  and  $r_2$ , in the range 1 to 0, and the means and standard deviations of the uncertain variables,  $\mu_{X_{si}}$  and  $\sigma_{X_{si}}$ , as shown in (6.152) and (6.153). The variables are independent. Therefore, new pairs of random numbers are used for the generation of each variable set. The random numbers are generated using the Fortran 'pseudo' random number generator and therefore are actually pseudo random numbers. The current local time is used to derive the seed value.

$$u_1 = \mu_{X_{si}} + \sigma_{X_{si}}((-2 \ln r_1)^{1/2} \sin 2\pi r_2) \quad (6.152)$$

$$u_2 = \mu_{X_{si}} + \sigma_{X_{si}}((-2 \ln r_1)^{1/2} \cos 2\pi r_2) \quad (6.153)$$

20,000 separate analyses were undertaken and each limit state value recorded. The probability of failure may be found from (6.154) where  $N$  is the number of runs and  $I[G(X_{sj}) \leq 0]$  is an indicator function which equals 1 if  $[G(X_{sj}) \leq 0]$  is true and 0 if  $[G(X_{sj}) \leq 0]$  is false.

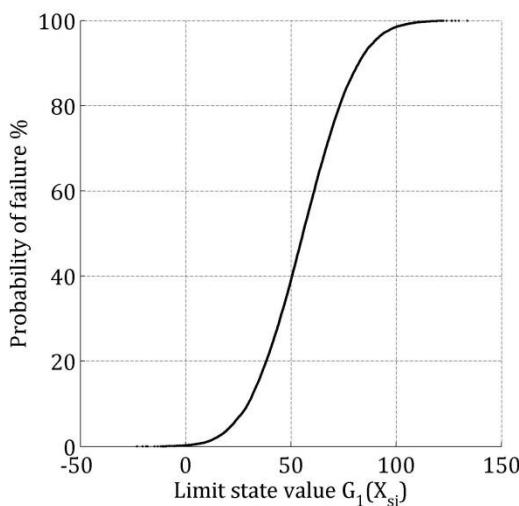
$$p_f \approx \frac{1}{N} \sum_{j=1}^N I[G(X_{sj}) \leq 0] \quad (6.154)$$

|   | $G_1(X_{si}) = \sigma_{per}^w - \sigma_{max}^w$ | $G_2(X_{si}) = \sigma_{per}^f - \sigma_{max}^f$ | $G_3(X_{si}) = \sigma_{min}^p - \sigma_{per}^p$ | $G_4(X_{si}) = \delta_{al} - \delta_{max}$ |
|---|---|---|---|--|
| <b>Reliability Analysis <math>\Phi(-\beta)</math> (Table 6-4)</b> | 0.0033  | 0.0154  | 0.0749  | 0.0934                                     |
| <b>Monte Carlo Simulation with equation (6.154)</b>               | 0.0027  | 0.0152  | 0.0735  | 0.0939                                     |
| <b>% error</b>  | -18.18%   | -1.30%  | -1.87%  | 0.54%                                      |
| <b>Monte Carlo Simulation with CDF estimate (Figure 6-3)</b>      | 0.0029  | 0.0154  | 0.0736  | 0.0940                                     |
| <b>% error</b>  | -12.12%   | 0.00%   | -1.74%  | 0.64%                                      |

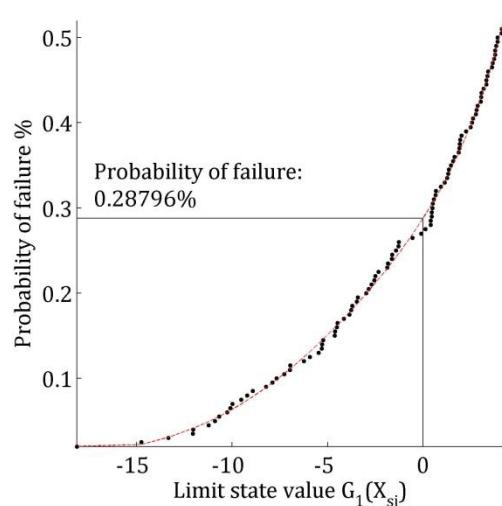
**Table 6-5: Comparison between analytically derived HL\_RF probability of failure,  $\Phi(-\beta)$ , and Monte Carlo probability of failure,  $p_f$**

The values can also be sorted from low to high and used to plot a cumulative probability function from which the approximate probability of failure may be read as demonstrated in Figure 6-3(a)-(h).

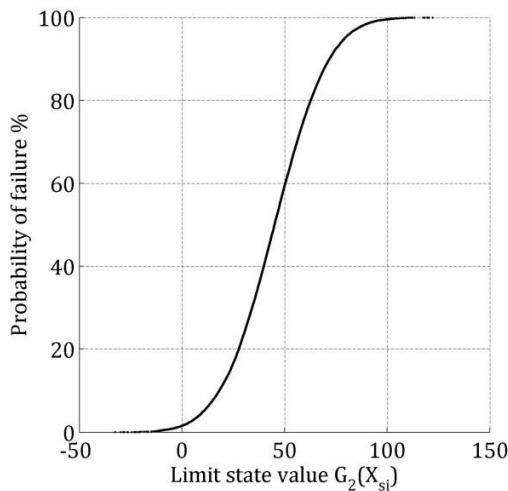
It can be seen that the probability of failure derived from the analytical and Monte Carlo simulation are in generally good agreement. The high percentage error given by  $G_1$  is due to the relatively low probability of failure for this limit state. The accuracy of the Monte Carlo simulation may be improved by increasing the quantity of simulations undertaken.



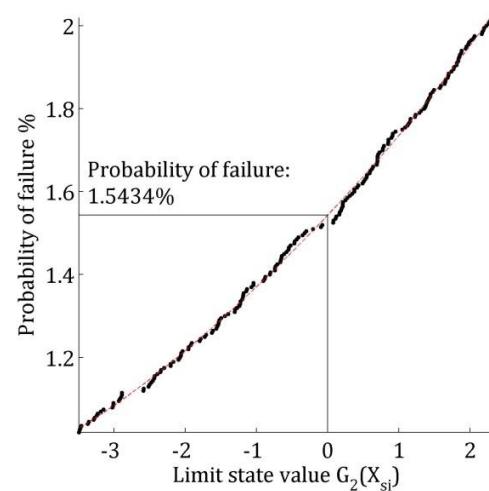
**(a) Cumulative Probability**



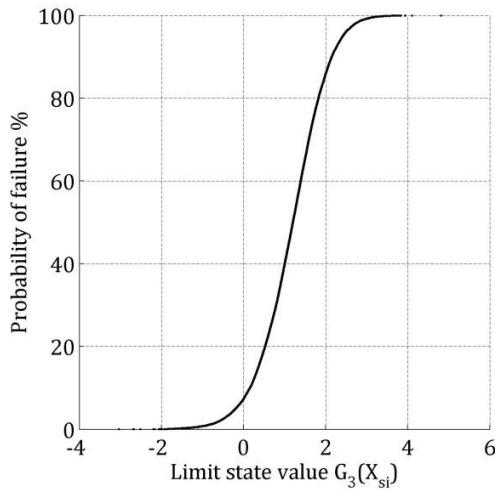
**(b) 50 points either side of intercept**



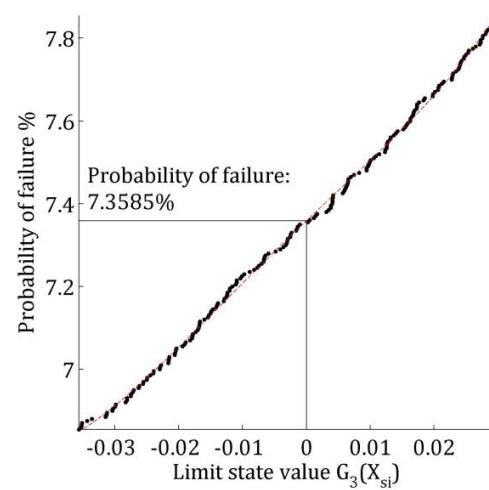
(c) Cumulative Probability



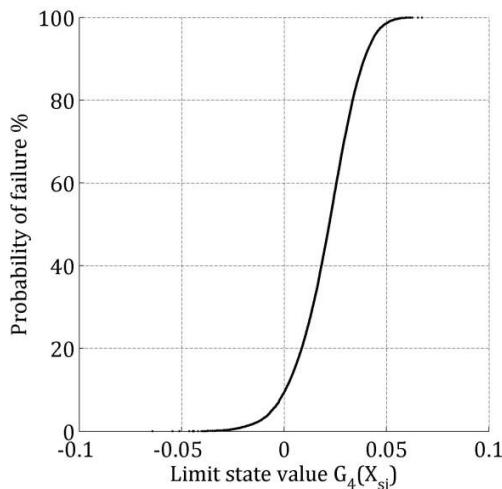
(d) 100 points either side of intercept



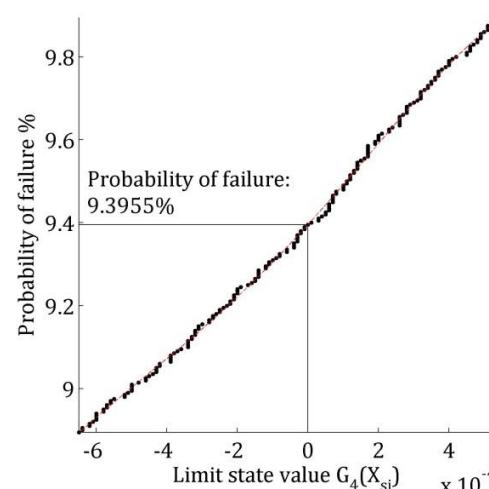
(e) Cumulative Probability



(f) 100 points either side of intercept



(g) Cumulative Probability



(h) 100 points either side of intercept

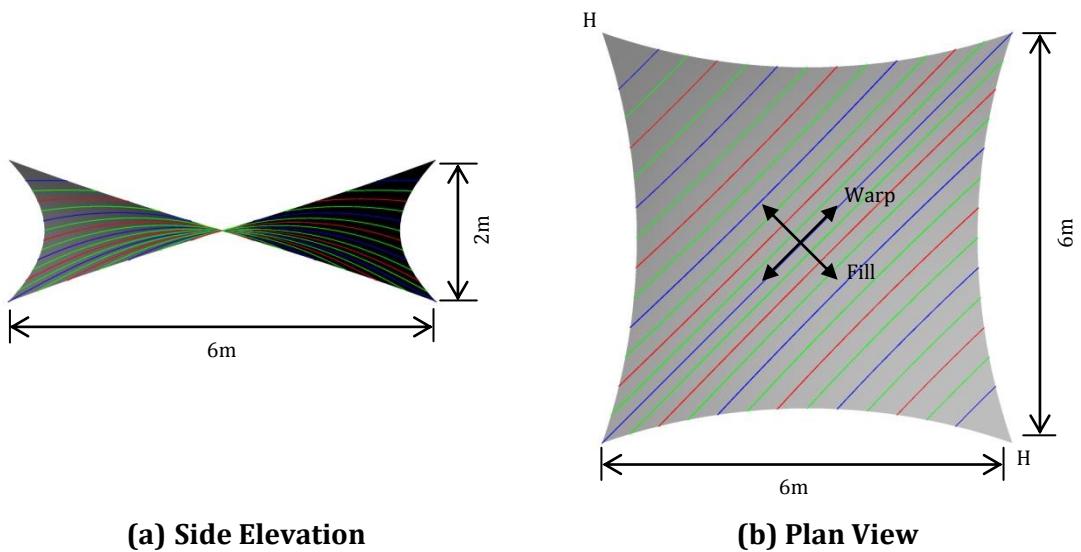
**Figure 6-3: Cumulative probability plots and intercepts from Monte Carlo simulation**



## 6.4 Neural Network Variability

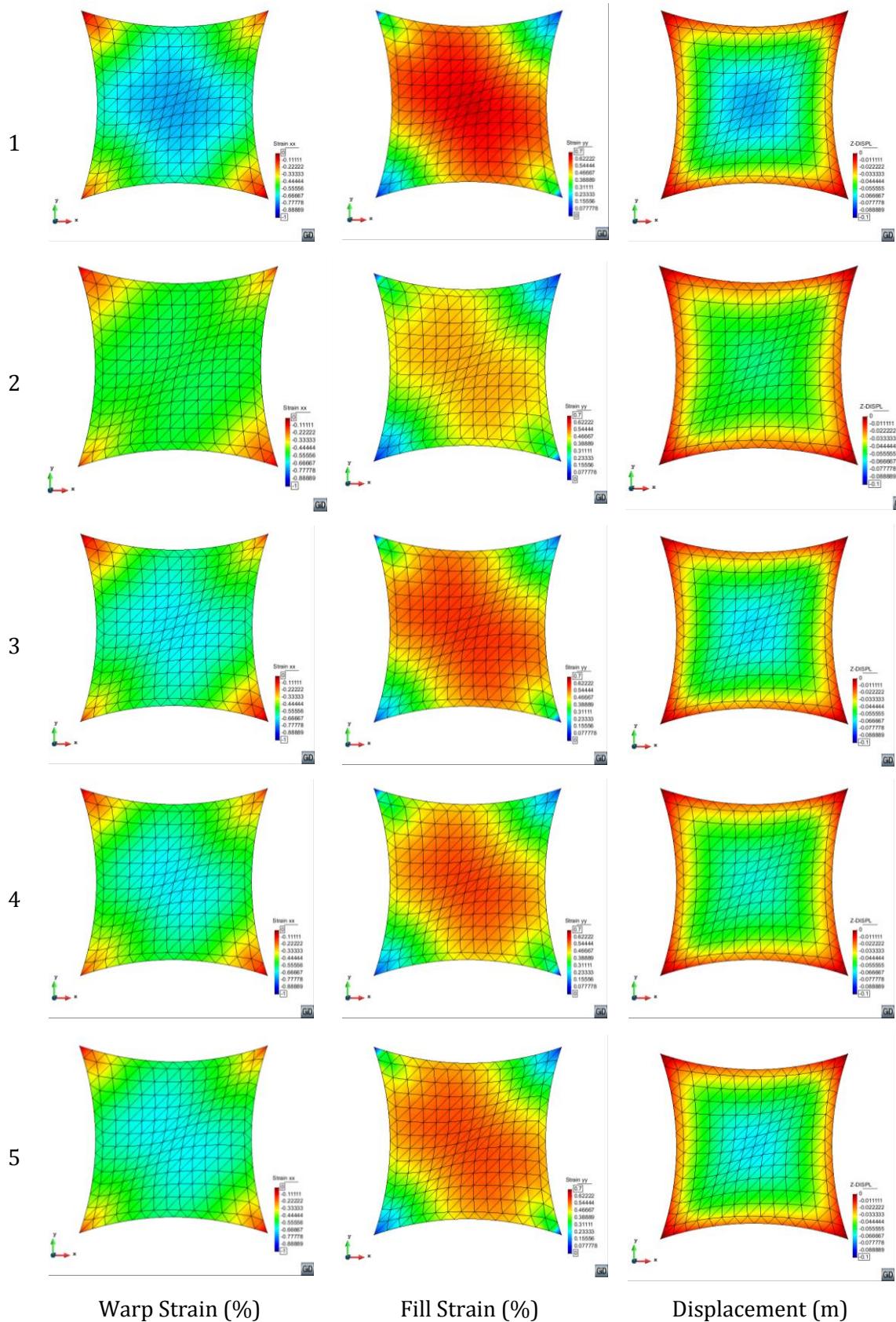
The training of neural networks involves a number of processes that contribute to variation between networks trained to represent the same material (see Section 4.4.5, and Section 5.3.1). The key processes include division of training data sets using randomised data selection and the random initiation of the weights and biases at the beginning of network training. The process of training, results in neural network material models that offer a non-unique solution to the stress strain relationship of a material. Therefore, neural network material models contain epistemic uncertainty. This is highlighted in Figure 6-5. A set of 5 networks has been trained using data taken from Section 5.2 for a response surface style network. The network weights and biases are randomly initialised at the beginning of training. However each network is presented with identical training data sets.

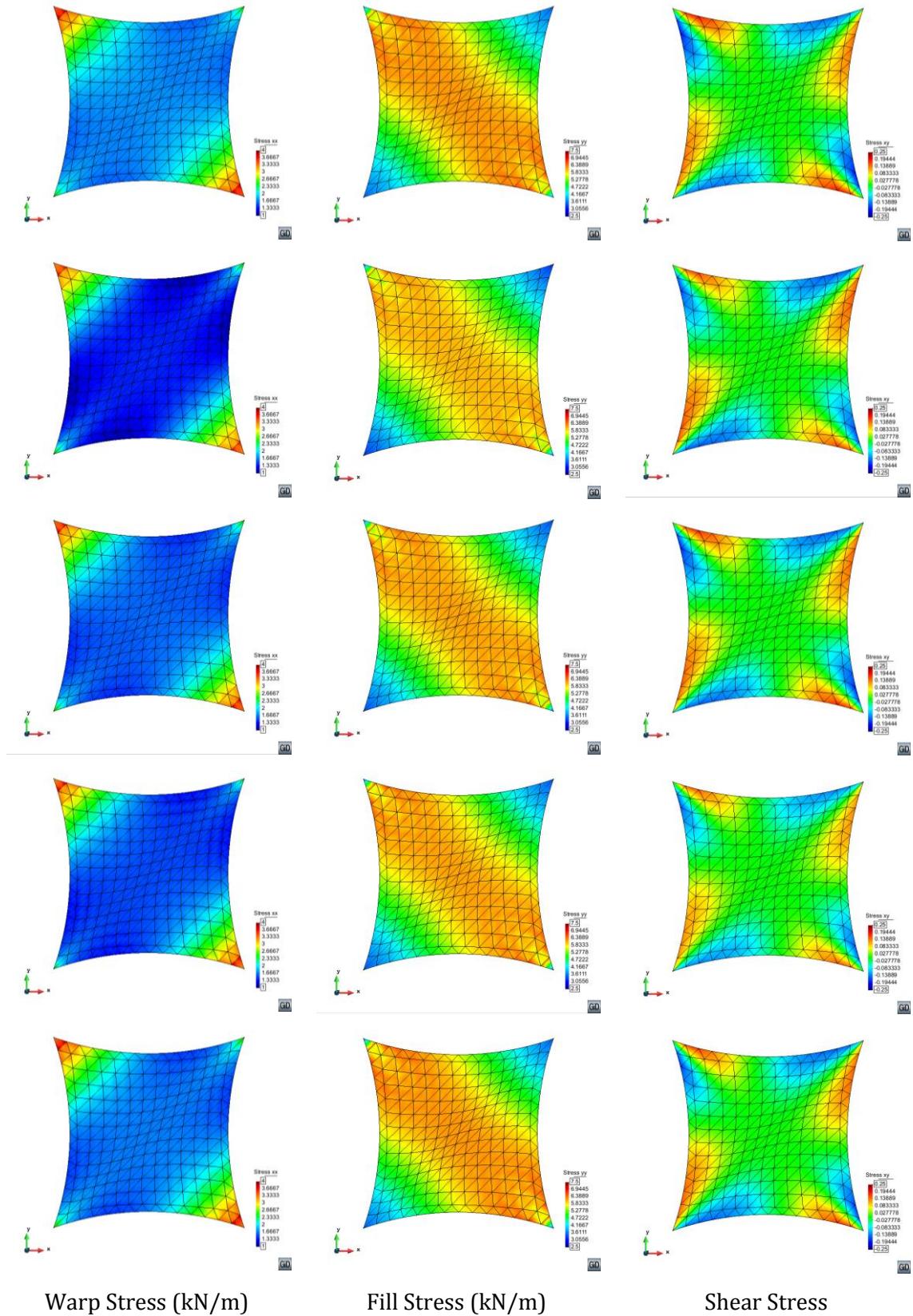
The 5 networks are each used in the simulation of the hypar structure used throughout this thesis. A typical global snow load of  $-0.6\text{kN/m}^2$  is applied to the structure.



**Figure 6-4: Hypar structure.**

It can clearly be seen in Figure 6-5 that each network produces different stress, strain and displacement distributions with varying maximum and minimum values, despite being trained using identical data. Each legend has been set to the same range in order to make the variations clear. To account for this variation in design some form of reliability analysis incorporating network simulation is required.





**Figure 6-5: Hypar Stress Plots From 5 Randomly Initiated Networks Trained using Identical Data.**

## 6.5 Response Surface Style Neural Network Reliability Analysis

In order to include the neural network epistemic uncertainty within reliability analysis it is necessary to replace the statistical variables describing the plane stress material model with variables describing the network material model within the reliability analysis. This is a complex problem due the high number of variables to consider. It is also not possible to derive meaningful statistical information describing variability of the network weights and biases as each variable cannot be considered in isolation. However, the implied stiffness matrix, Chapter 5, offers an approximation to the network response and is derived at each Gauss point for each dynamic relaxation energy peak. The implied stiffness matrix approximates the network stress strain relationship by effectively fitting a surface tangential to network stress strain response surface at the Gauss point's current position on the response surface. The 4 variables of the implied stiffness matrix may be used in place of the 4 variables of the plane stress material model for the purposes of reliability analysis. Therefore, the 8 statistical variables for reliability analysis become,

$$X_1^n = E_{1,1}^{GP,n}, X_2^n = E_{1,2}^{GP,n}, X_3^n = E_{2,1}^{GP,n}, X_4^n = E_{2,2}^{GP,n}, X_5^n = G_{wf}, X_6^n = \sigma_{ult}^f, X_7^n = \sigma_{ult}^w, X_8 = t_{load}, \quad (6.155)$$

where GP refers to the Gauss point position of the current limit state and n refers to the current load increment. The reliability analysis is run using the mean implied stiffness matrices and associated standard deviations at each Gauss point calculated at the final energy peak for a specific structural simulation in place of the neural network material model.

An added level of complexity is introduced when considering an incrementally loaded simulation as required when using a history network model. The variables at each load increment must be considered as separate statistical variable. Therefore, a simulation with 10 load increments has 62 statistical variables. This will require a very large computational effort and the development of incremental partial derivatives. Therefore, for this initial study, only simulation of a single load increment with a response surface style neural network is explored in this chapter. Development of an incremental reliability analysis will be briefly explored in the further work section of Chapter 7, but is beyond the scope of this thesis.

The analytical partial derivatives of the limit state functions, Sections 6.3 and 6.3.1, remain largely the same with the obvious exception of the elastic stiffness derivatives.

### 6.5.1.1 Implied Elastic Stiffness Matrix Derivatives

Partial differentiation of the implied stiffness matrix,  $E^{implied}$ , with respect to its 4 network related terms is straightforward and is given by,

$$\begin{aligned} \left[ \frac{\partial E^{implied}}{\partial X_1} \right] &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \left[ \frac{\partial E^{implied}}{\partial X_2} \right] = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \left[ \frac{\partial E^{implied}}{\partial X_3} \right] = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ \left[ \frac{\partial E^{implied}}{\partial X_4} \right] &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \end{aligned} \quad (6.156)$$

The partial derivatives of  $E^{implied}$  with respect to the shear modulus,  $G_{wf}$ , remains the same and,

$$\left[ \frac{\partial E}{\partial X_5} \right] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (6.157)$$

The implied stiffness matrix is fixed at the mean value for a given structural simulation.

Therefore the partial derivatives of the stiffness matrix with respect to the fabric ultimate strength and the applied load remains,

$$\left[ \frac{\partial E}{\partial X_{6-8}} \right] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (6.158)$$

where  $X_{6-8}$  is  $\sigma_{ult}^w$ ,  $\sigma_{ult}^f$  and  $t_{load}$ , respectively.

It is noteworthy that in a true neural network material model simulation, the implied elastic stiffness matrix varies depending on the current stress strain state of the Gauss point at which the implied stiffness matrix is defined. Therefore, the implied stiffness matrix is a function of applied load. Further work may be required to investigate the variation of the implied stiffness matrix with respect to load. This level of complexity is neglected for the purposes of this simplified case where the initial values of the implied stiffness matrix are considered fixed for the purposes of reliability analysis.

### 6.5.2 Validation of implied stiffness partial derivative with finite difference method.

As with the plane stress reliability analysis, Section 6.3.2, the partial derivatives of the limit state function with respect to the statistical variables may be validated using the finite difference method. A simple 2 element patch, Figure 6-1, is again employed for this purpose. This patch is not suitable for detailed analysis but is selected for speed of analysis due to its small stiffness matrix.

As in the validation of the plane stress derivatives , Section 6.3.2, the finite difference method is performed using 2 different levels of perturbation, 0.001% and 0.00001%, noting that the 0.1% perturbation has been previously discounted as unsuitable.

The finite difference derivatives generated by a perturbation of 0.001% are in generally good agreement with the analytically derived values, with <1% difference across all variables with the exception of  $E_{2,2}$  for limit state 1 and  $E_{2,1}$  for limit state 2. Decreasing the perturbation to 0.00001% further reduces the error. Only the error between the analytical and finite difference derivatives of limit state 2 with respect to  $E_{2,1}$  remains greater than 1%. This derivative has a value 2 orders of magnitude smaller than the other the derivatives in this set. Therefore, it is likely that the error is due to the relatively small magnitude of the derivative in question, resulting in the impact of error introduced by the finite difference method being more significant.

| $X_{si}$   | $\frac{\partial(\sigma_{per}^w - \sigma_{max}^w)}{\partial X_{si}}$ : max stress warp      |                     |        | $\frac{\partial(\sigma_{per}^f - \sigma_{max}^f)}{\partial X_{si}}$ : max stress fill |                     |         |
|------------|--|---------------------|--------|---|---------------------|---------|
|            | Finite difference method   | Analytical approach | %error | Finite difference method  | Analytical approach | %error  |
| $E_{1,1}$  | -1.45E-03  | -1.44E-03           | -0.74% | 1.41E-03  | 1.40E-03            | -0.67%  |
| $E_{1,2}$  | 3.35E-04   | 3.35E-04            | -0.22% | -1.02E-03   | -1.01E-03           | -0.16%  |
| $E_{2,1}$  | -1.05E-03  | -1.03E-03           | -1.01% | 5.78E-05  | 4.55E-05            | -21.23% |
| $E_{2,2}$  | 1.31E-04   | 1.40E-04            | 6.80%  | -1.48E-03   | -1.49E-03           | 0.66%   |
| $G_{wf}$   | 8.26E-02   | 8.27E-02            | 0.03%  | 8.76E-02  | 8.77E-02            | 0.04%   |
| $t_{load}$ | -3.54E+01  | -3.54E+01           | 0.00%  | -1.91E+01   | -1.91E+01           | -0.01%  |
| $X_{si}$   | $\frac{\partial(\sigma_{min}^p - \sigma_{per}^p)}{\partial X_{si}}$ : min principal stress |                     |        | $\frac{\partial(\delta_{per} - \delta_{max})}{\partial X_{si}}$ : max displacement    |                     |         |
|            | Finite difference method   | Analytical approach | %error | Finite difference method  | Analytical approach | %error  |
| $E_{1,1}$  | -6.74E-03  | -6.73E-03           | -0.01% | 9.63E-05  | 9.65E-05            | 0.11%   |
| $E_{1,2}$  | 3.27E-03   | 3.27E-03            | -0.14% | -2.09E-05   | -2.09E-05           | -0.02%  |
| $E_{2,1}$  | 2.80E-03   | 2.81E-03            | 0.22%  | -4.24E-05   | -4.23E-05           | -0.02%  |
| $E_{2,2}$  | -1.45E-03  | -1.44E-03           | -0.43% | 8.21E-06  | 8.22E-06            | 0.12%   |
| $G_{wf}$   | 1.91E-01   | 1.91E-01            | 0.09%  | 3.82E-04  | 3.82E-04            | 0.03%   |
| $t_{load}$ | -5.64E+00  | -5.65E+00           | 0.14%  | -8.46E-02   | -8.46E-02           | 0.00%   |

**Table 6-6: Sensitivities, finite difference method using perturbation of 0.001% (Implied Stiffness Variables)**

|            | $\frac{\partial(\sigma_{per}^w - \sigma_{max}^w)}{\partial X_{si}}$ : max stress warp      |                     |        | $\frac{\partial(\sigma_{per}^f - \sigma_{max}^f)}{\partial X_{si}}$ : max stress fill |                     |        |
|------------|--|---------------------|--------|---|---------------------|--------|
| $X_{si}$   | Finite difference method   | Analytical approach | %error | Finite difference method  | Analytical approach | %error |
| $E_{1,1}$  | -1.44E-03  | -1.44E-03           | -0.12% | 1.40E-03  | 1.40E-03            | -0.18% |
| $E_{1,2}$  | 3.36E-04   | 3.35E-04            | -0.53% | -1.02E-03   | -1.01E-03           | -0.26% |
| $E_{2,1}$  | -1.03E-03  | -1.03E-03           | 0.41%  | 3.91E-05  | 4.55E-05            | 16.26% |
| $E_{2,2}$  | 1.39E-04   | 1.40E-04            | 0.78%  | -1.48E-03   | -1.49E-03           | 0.11%  |
| $G_{wf}$   | 8.27E-02   | 8.27E-02            | 0.00%  | 8.77E-02  | 8.77E-02            | 0.00%  |
| $t_{load}$ | -3.54E+01  | -3.54E+01           | 0.00%  | -1.91E+01   | -1.91E+01           | 0.00%  |
|            | $\frac{\partial(\sigma_{min}^p - \sigma_{per}^p)}{\partial X_{si}}$ : min principal stress |                     |        | $\frac{\partial(\delta_{per} - \delta_{max})}{\partial X_{si}}$ : max displacement    |                     |        |
| $X_{si}$   | Finite difference method   | Analytical approach | %error | Finite difference method  | Analytical approach | %error |
| $E_{1,1}$  | -6.74E-03  | -6.73E-03           | -0.06% | 9.65E-05  | 9.65E-05            | 0.00%  |
| $E_{1,2}$  | 3.27E-03   | 3.27E-03            | -0.20% | -2.09E-05   | -2.09E-05           | -0.03% |
| $E_{2,1}$  | 2.83E-03   | 2.81E-03            | -0.64% | -4.24E-05   | -4.23E-05           | -0.03% |
| $E_{2,2}$  | -1.45E-03  | -1.44E-03           | -0.27% | 8.22E-06  | 8.22E-06            | -0.04% |
| $G_{wf}$   | 1.91E-01   | 1.91E-01            | 0.06%  | 3.82E-04  | 3.82E-04            | 0.00%  |
| $t_{load}$ | -5.64E+00  | -5.65E+00           | 0.15%  | -8.46E-02   | -8.46E-02           | 0.00%  |

**Table 6-7: Sensitivities, finite difference method using perturbation of 0.00001% (Implied Stiffness Variables)**

### 6.5.3 Reliability analysis of a Hypar

The partial derivatives described in Section 6.5 will be used in the reliability analysis of the hypar structure demonstrated in Section 6.4. The hypar is patterned as shown in Figure 6-4. It is subjected to a typical snow load of 0.6kN/m<sup>2</sup>. The non-neural-network related statistical variables used in the reliability analysis are summarised in

| Coefficient of variation | 0.1 | 0.1 | 0.2 | 0.2 |
|--------------------------|-----|-----|-----|-----|
|--------------------------|-----|-----|-----|-----|

Table 6-8. The maximum permissible stress in both directions is set at approximately 25% of the ultimate limit strength of a typical PVC coated polyester fabric, and is related to tear propagation. This is the maximum level of stress allowed in the biaxial testing used to generate the network training data. Beyond this level of stress the network is unlikely to be able to reliably generalise the fabric response as it is outside of the extents of the training response surface, Section 2.3.

| Variable                 | $G_{xy}$ | $t_{load}$ | $\sigma_{per}^w$ | $\sigma_{per}^f$ |
|--------------------------|----------|------------|------------------|------------------|
| Mean                     | 30       | 1          | 30               | 30               |
| Standard Deviation       | 3        | 0.1        | 6                | 6                |
| Coefficient of variation | 0.1      | 0.1        | 0.2              | 0.2              |

**Table 6-8: Non network related statistical variables.**

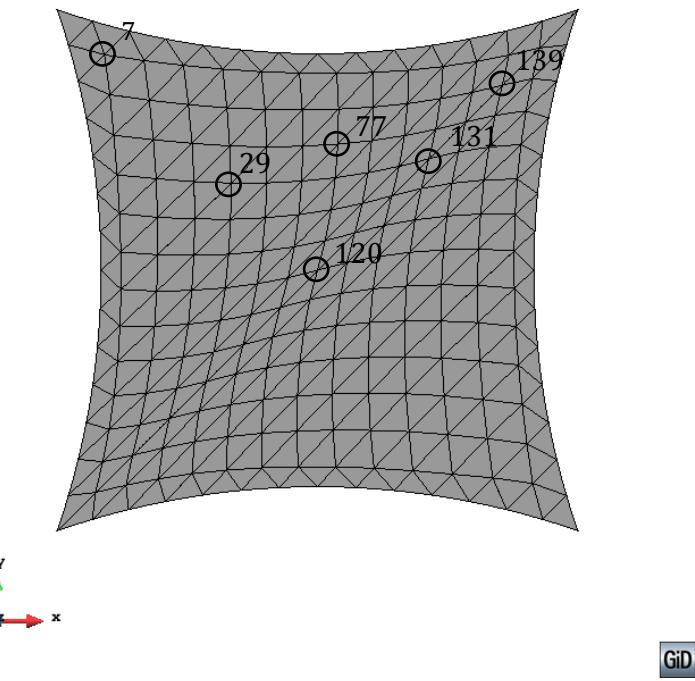
#### 6.5.3.1 *Derivation of Implied Stiffness matrix statistical variables.*

The process of generating statistical information, i.e. means and standard deviations, for the elastic stiffness constants is completed in two steps.

In the first step, a set of 30 response surface style networks are trained using experimental data. The response surface style networks each take warp and fill strains as the (2), have 10 hidden nodes, and provide warp and fill stresses as the (2) outputs,. Construction of training data sets and network initiation may done using random values (see section 6.4). Here, a single training data set is derived from the available experimental data for PVC coated polyester architectural fabric and used as training data for each network. In other words, each network is trained using identical data. The weights and bias values are initiated randomly. This means that the training algorithm begins at a different position in the solution space for each network. This results in a set of 30 unique network material models for the architectural fabric.

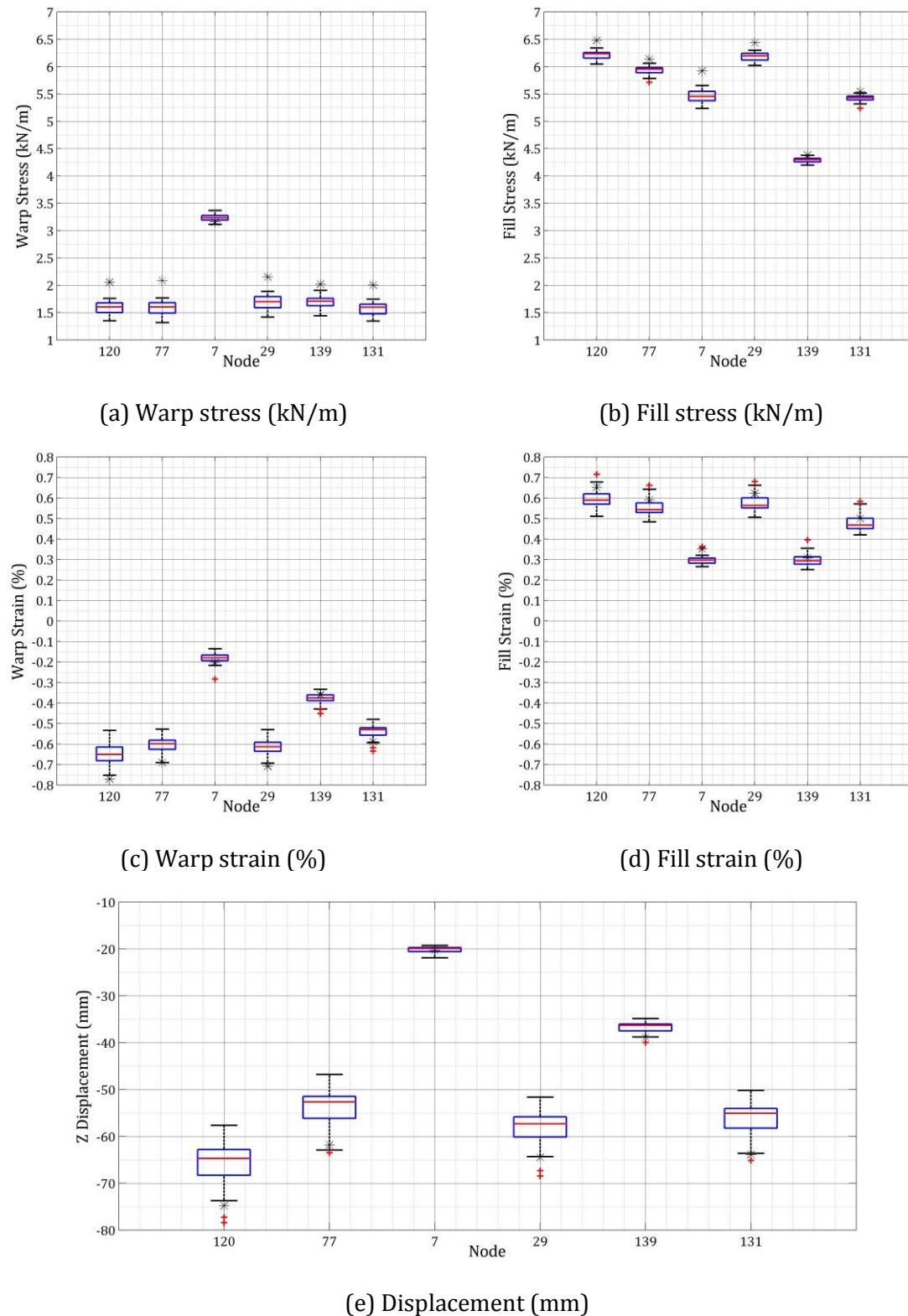
In the second step, each network is used in a simulation of the structure on which the reliability analysis is to be performed. The structure is loaded as it will be in the reliability analysis. The implied stiffness matrix for each Gauss point of the mesh at the equilibrium state, after the final dynamic relaxation energy peak, is taken as output. These sets of implied stiffness values are used to derive a mean implied stiffness matrix and associated standard deviation for each Gauss point. The reliability analysis is then undertaken using these values. In each reliability iteration the statistical information and limit state function at the current design point is used to calculate the direction cosines,  $\alpha$ , and the safety index,  $\beta$  (equations (6.9) to (6.13)). These values are then used along with each Gauss point's mean and standard deviation in order to update all the implied elastic stiffness matrices (equations (6.14) and (6.15)). The process is repeated until the reliability analysis convergence criteria are met.

In order to illustrate the variation between each network simulation, stress, strain and displacement data has been output for 6 key nodes in the hypar mesh, Figure 6-6.

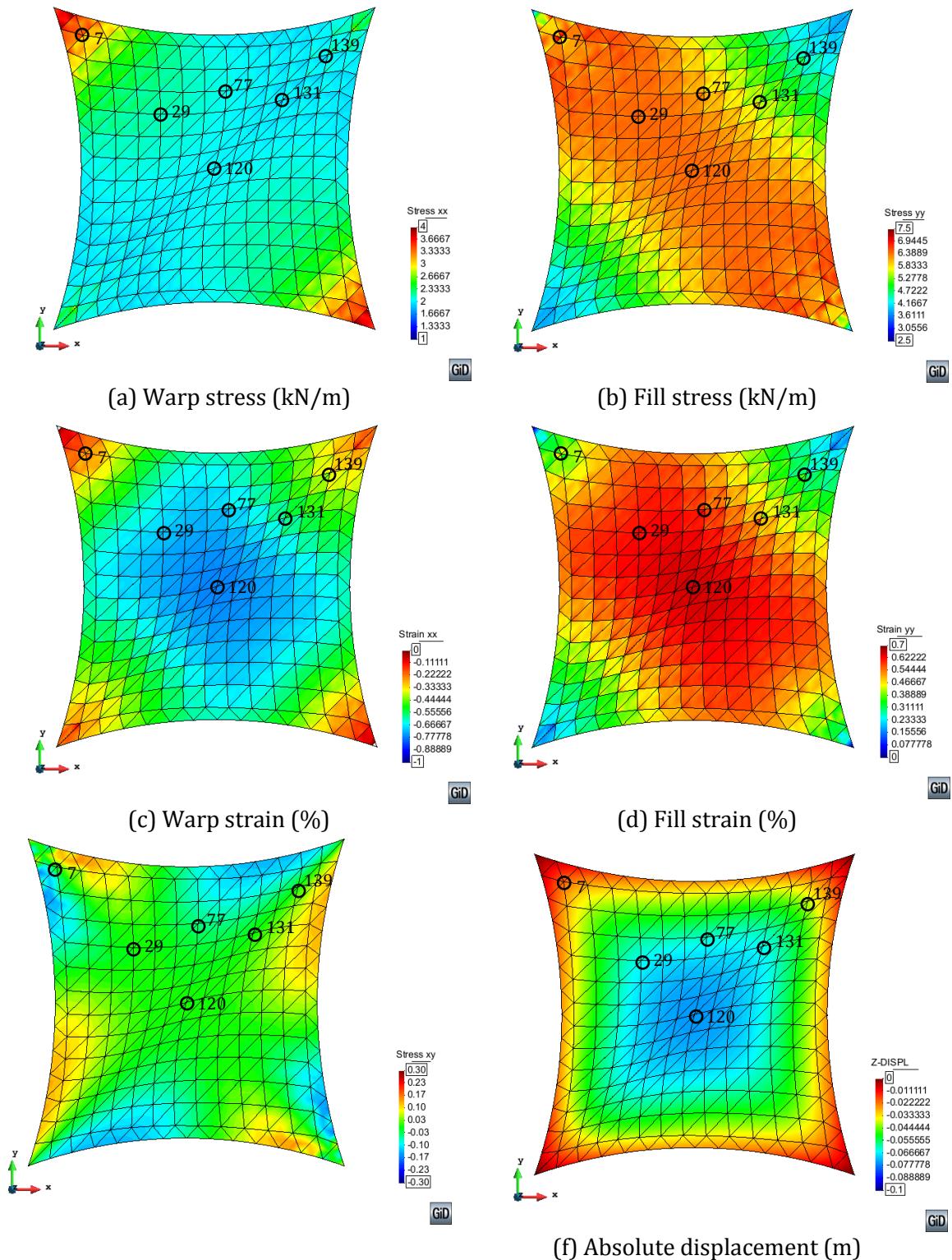


**Figure 6-6: Hypar structure form-found mesh with selected nodes**

The results extracted for each node are plotted as box plots, Figure 6-7, where the median is denoted by the red line and the box contains all results between the 25th and 75th percentile. The stress results produce the least outliers, denoted by a red +, with a typical range of approximately 0.5kN/m or less. The strain and displacement results produce a greater number of outliers and relatively larger ranges. This is expected as an applied load drives the analysis and the level of stress is more or less prescribed. The nodal displacement, which induced the strain required as input to the network model in order reach equilibrium with the applied load, is found. Therefore, the higher level of variation between displacement and strain results is expected.



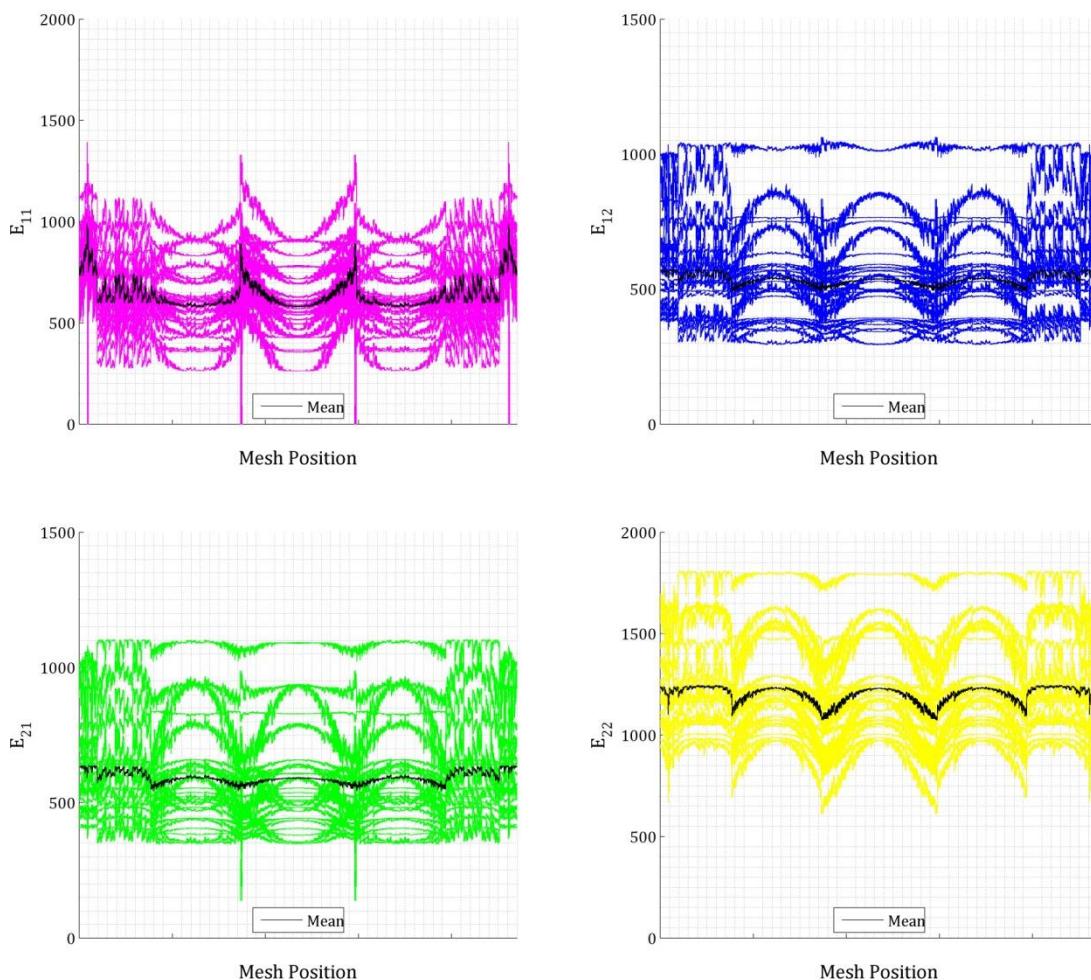
**Figure 6-7: Results Variability Across 30 Networks**



**Figure 6-8: Hypar snow results using mean implied stiffness values**

The results of the hypar simulation based on mean implied stiffness values are shown in Figure 6-8. The stress, strain and displacement values for the selected nodes are plotted as \* in Figure 6-7. For these latter values in both material directions, the stress values are

greater than the range denoted by the box plots. The fill strains are generally greater than the 75th percentile but still within the total range. The warp strains are generally at the lower end of the range. These result in an overall deflection that is similar to the minimum deflection observed in the network simulations. The error is likely to be caused by the large strain step used in conjunction with the implied stiffness matrix. Accuracy of the implied stiffness approximation is inversely proportional to the strain step. Iterative loading would improve the accuracy of the implied stiffness solution, as the implied stiffness would vary across the response surface. However, as previously discussed, this would require the derivation and implementation of an iterative reliability analysis and greatly increase the number of partial derivatives required. These results may also indicate that a more suitable candidate for the initial implied stiffness values would be the values derived from the network that generates the median stress result. However, the mean values are deemed to produce results sufficiently similar to the typical network response to serve as a substitute for the network in this initial study.



**Figure 6-9: Implied stiffness values for hypar mesh.**

The hypar mesh contains a total of 4704 Gauss points, for which the implied stiffness values for each network along with the mean implied stiffness constants are plotted in Figure 6-9. The symmetry of the plots is a result of the symmetrical mesh that is generated systematically. It is evident from Figure 6-9 that the implied stiffness values show a high degree of variation between networks. They also vary across the mesh. This demonstrates the power of the network to capture the non-linear behaviour of the fabric material. Overall means of the mean and standard deviation values for each implied stiffness variable is shown in Table 6-9 to give an indication of the typical statistical properties.

| Variable                        | $E_{11}$ | $E_{12}$ | $E_{21}$ | $E_{22}$ |
|---------------------------------|----------|----------|----------|----------|
| <b>Mean of means</b>            | 640.3    | 536.4    | 593.1    | 1203.1   |
| <b>Mean standard Deviation</b>  | 152.5    | 167.7    | 192.7    | 217.0    |
| <b>Coefficient of variation</b> | 0.24     | 0.31     | 0.32     | 0.18     |

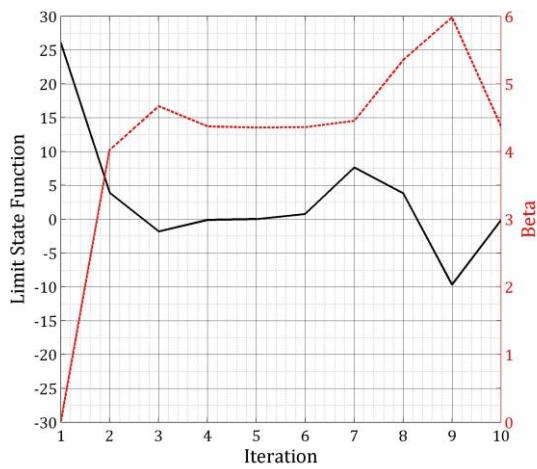
**Table 6-9: Network related statistical variables (Global mean).**

The reliability analysis is limited to 10 iterations as computation of the analytical limit state partial derivatives for each variable requires a significant computer resource. This is largely due to the inversion of the partial derivative stiffness matrix,  $dK^{-1}$ , required in the determination of the displacement derivatives (equation (6.33)). For the hypar structure this matrix is a 2511 by 2511 matrix and has to be inverted for each statistical variable for each iteration of the reliability algorithm. However, this remains a faster process than using the finite difference method. Nevertheless, for this size structure, 10 iterations of the reliability algorithm take approximately 10 hours to complete with the programme in its current form.

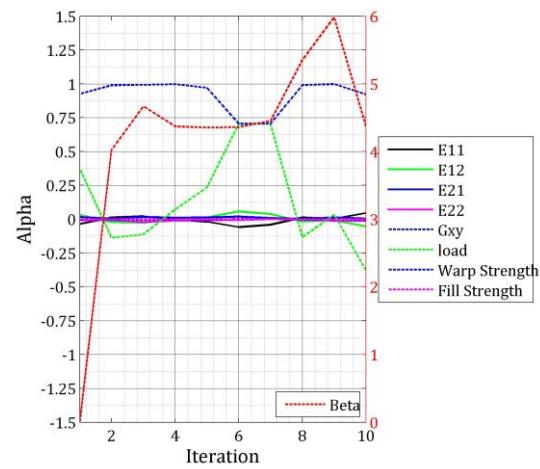
The statistical values for each gauss point are used in the reliability analysis of the strength limit states; limit states 1 and 2, equation (6.1) and (6.2). Limit state 1 does not fully converge to a stable result after 10 reliability iterations, Figure 6-10 (a) . However, the limit state function approaches zero between iterations 4 and 7 with a concurrent value of  $\beta$  that stabilises at approximately 4.4. Interestingly, and arguably providing confidence in the solution, a very similar value of  $\beta$  is obtained at iteration 10 with the value of the limit state close to zero. Limit State 2 converges to a stable solution after 5 iterations, with a  $\beta$  value of 3.81 and a limit state function of approximately 0.001, Figure 6-10 (b). Both limit states generate a  $\beta$  value greater than the minimum of 3.8 suggested in Eurocode 0 and represent a less than 0.007% probability of failure. As expected, the probability of failure due to material rupture is higher in the fill direction, corresponding

as it does with the high points of the hypar and greater stress levels under gravity loading. The warp direction would be the critical direction in an uplift case.

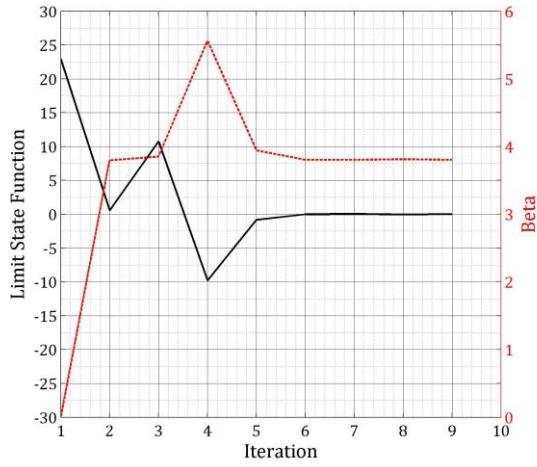
The values of  $\alpha$  for each statistical variable for both limit state functions Figure 6-10 (c) and (d) clearly show that the most critical variable is the material strength in the relevant direction. This is followed by the load factor. It is in fact questionable whether the remaining 6 statistical variables have any significant impact despite their large standard deviations.



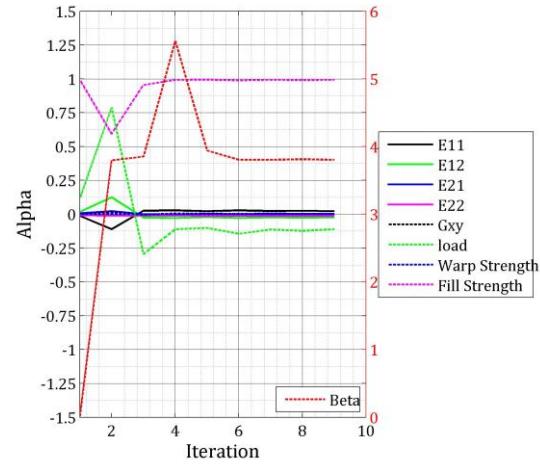
(a) Limit State 1,  $\beta \approx 4.4$



(b) Limit State 1,  $\alpha$ -values



(c) Limit State 2,  $\beta \approx 3.81$

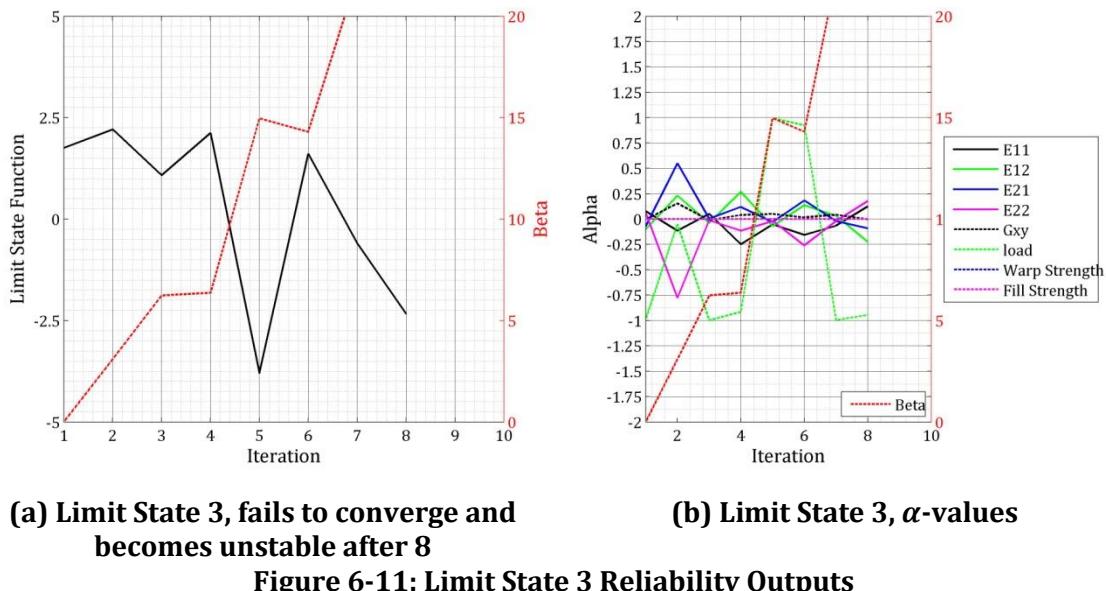


(d) Limit State 2,  $\alpha$ -values

Figure 6-10: Limit State 1 and 2 Reliability Outputs

The statistical values for each Gauss point were next used in the reliability analysis of the wrinkling and deformation serviceability limit states, limit state 3 and 4 respectively equation (6.3) and (6.4). The reliability analysis for limit state 3, the minimum stress limit state, becomes unstable after only a small number of iterations. The value of  $\beta$  becomes greater than 5 after only 3 reliability iterations and the mesh collapsed after 8 iterations,

Figure 6-12 (a). This may indicate that the failure due to wrinkling has an extremely low probability. However, it is more likely that the large standard deviation of the statistical variables combined with their complex interactions caused the analysis to become unstable. Attempts to stabilise the analysis using the lower standard deviations also produced similar results.



**(a) Limit State 3, fails to converge and becomes unstable after 8**

**(b) Limit State 3,  $\alpha$ -values**

**Figure 6-11: Limit State 3 Reliability Outputs**

Limit State 4 shows greater promise as the value of  $\beta$  did not diverge as quickly as for Limit state 3, Figure 6-12 (a). However, the limit state function oscillates approximately about its initial value and the  $\beta$  value shows no sign of converging. The load factor is clearly the most significant variable and is the oscillation of the  $\alpha$ -values of this variable is the main cause of the oscillation of the limit-state function. It appears that the load factor and  $E_{11}$  are working against each other, when the  $\alpha$ -value for  $E_{11}$  becomes a positive value the load factor alpha-value quickly switches from positive to negative, the same is true for the  $\alpha$ -value for  $E_{12}$ .

The oscillation was caused by ill conditioning between the limit state function, its partial derivatives and the direction of loading and therefore displacement. This was due to the use of the absolute value of displacement in the limit state function. Therefore, the displacement limit state function was reformulated to include a unit factor accounting for the principle direction of loading. This is deemed an acceptable solution as loading is typically applied approximately perpendicular to the fabric surface and therefore the principle direction of loading may be defined.

$$G_4(X_{si}) = \delta_{per} - (l^{dir})|\delta|_{max}$$

Principle load direction+'ve  $\rightarrow l^{dir} = 1$  (6.159)

Principle load direction-'ve  $\rightarrow l^{dir} = -1$

When using this form of the limit state function the reliability analysis becomes stable and converges in 10 iterations with a stable  $\beta$  value of 0.96 and a minimum limit state function value of -0.0002. A  $\beta$  value of 0.96 represents an unacceptably high probability of failure indicating that the design should be altered to increase allowable displacement or reduce displacement.

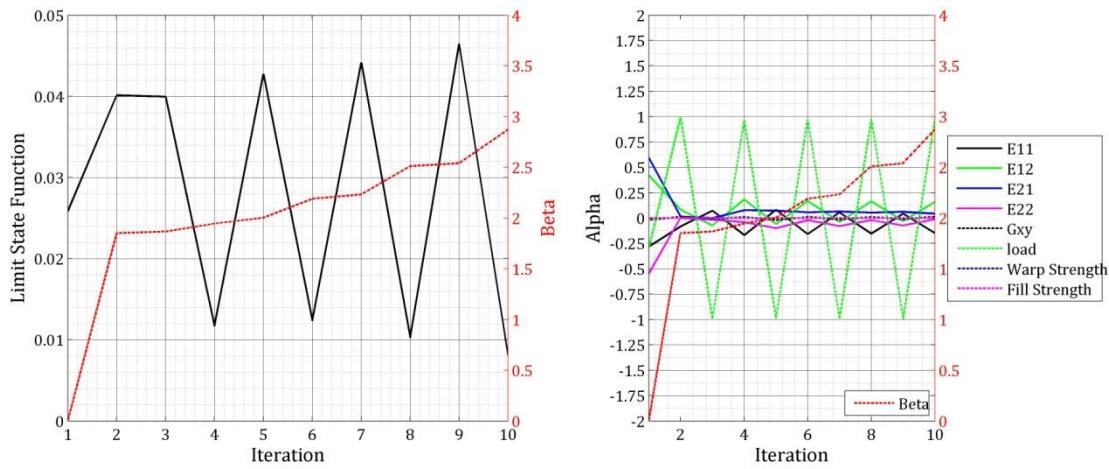


Figure 6-12: Limit State 4, Without Direction Factor

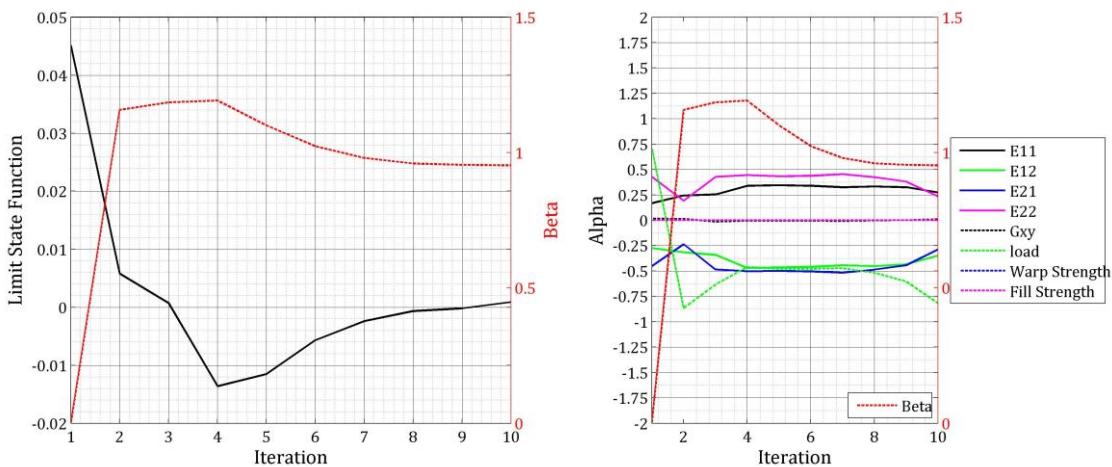


Figure 6-13: Limit State 4 With Direction Factor

## 6.6 Summary and Conclusions

In this chapter the analytical partial derivatives required for probabilistic reliability analyses using both plane stress and implied stiffness material models are derived and validated. The reliability solution procedure is based on the Hasofer-Lind and Rackwitz-Fiessler (HL\_RF) algorithm. Four limit state functions (failure in the warp direction, failure in the fill direction, wrinkling and maximum displacement) have been explored. The limit state functions are dependent on 8 statistical variables. The partial derivatives of each limit state function with respect to each statistical variable describes the sensitivity of that limit state to the variable.

Validation of the analytical sensitivities has been demonstrated using comparisons with partial derivatives obtained from the finite difference method and based on the analysis of a simple 2-element biaxially loaded patch. Varying levels of perturbation were investigated. Excellent agreement is demonstrated between the analytical and finite difference derivatives apart from where the derivative was very low or was incompatible with the level of perturbation, where good agreement has been achieved.

Monte Carlo simulation has been used to validate the probabilities of failure for the four limit state functions obtained from the reliability analysis. 20,000 Monte Carlo simulations were run with normally distributed statistical variables. The probability of failure was estimated using an indicator function and cumulative frequency plots. Excellent agreement between the reliability analysis and Monte Carlo simulations has been achieved for 3 of the 4 limit states. The limit state that showed the least agreement had a probability of failure that was lower by a factor of 10 suggesting that insufficient number of analysis runs had been completed to provide a reliable measure of the probability of failure from the Monte Carlo simulation.

A procedure for reliability analysis using an implied stiffness material model to approximate the network response has been explored. A set of 30 networks was trained and used to generate statistical information for the reliability analysis of a realistic hypar structure. Reliability analyses of the ultimate limit state functions for warp and fill stresses converged acceptably, with  $\beta$  values of 4.4 for the warp direction and 3.8 in the fill. The sensitivity of the analysis to the material properties is shown to be relatively low. The reliability analysis of the principle stress serviceability limit functions proved to be unstable and failed to reach acceptable levels of convergence. The analysis was not stabilised by the reduction of the statistical uncertainty of the material model. The displacement limit state was initially unstable. However, the modification of the limit state

function with a direction term lead to a stable convergence. The resulting  $\beta$  value represent an unacceptable high probability of failure. Significant further work is required in order to develop a robust reliability analysis procedure for neural network material models. However, initial work presented here indicates that with further development of the procedure network reliability analysis will be possible.

**Chapter 7.**  
**Conclusions and Further Work**



## Contents

|  |     |
|--|-----|
| Chapter 7. Conclusions and Further Work..... | 325 |
| 7.1 Conclusions.....                         | 329 |
| 7.2 Recommendations for Further Work .....   | 335 |



## Chapter 7. Conclusions and Further Work

### 7.1 Conclusions

A large displacement finite element formulation based on a 6 node isoperimetric triangular membrane element has been developed and demonstrated, Section 3.1 to 3.2. The element formulation allows the designer to prescribe the element local coordinate system to align with the desired fabric warp and fill direction. Geodesic control strings have been implemented to control the panel edges and centre lines without effecting the final form-found minimal surface. These features allow the patterning of the structure to be incorporated into structural mesh.

The finite element formulation implemented as a Fortran console application has been used to perform form finding, using a zero elastic stiffness matrix, and static load analysis, using a plane stress orthotropic elastic stiffness matrix, on realistic fabric structural meshes. A wrinkling procedure based on combined stress and stiffness criteria is implemented within the static analysis. Where the wrinkling criteria is triggered discontinuities in resulting stress field indicate that the analysis has potentially become unstable.

A patterning procedure using the 6 node isoperimetric element is presented, Section 3.3. The stress plot output from the patterning procedure allows the investigation of 'locked in' forces induced during the flattening of the form found panels. Discontinuities between neighbouring panel edge length are investigated and found to be relatively small. The formulation includes all of the features identified as requirements for architectural fabric simulation in the literature review, Section 2.1, and is suitable for the implementation of a neural network material model.

Biaxial load profiles have been developed for PVC coated polyester and PTFE coated glass architectural fabrics with additional load ratios between the typical 0:1, 1:2, 1:1, 2:1 and 1:0 load ratio arms, Section 4.3.2. The resulting experimental data with residual strain removed has been used to represent fabric behaviour in the form of a strain-stress response surface.

Response surface style neural network material models that have been trained and tested using the data sets, Section 4.3.3. Networks trained with PVC data are capable of

accurately generalising the PVC material response surface when presented with unseen data. Problems caused by the steepness of the PTFE response surface, leading to one-to-many mappings, have been overcome by the addition of a third stress ratio input. Stress to strain neural networks have been used to generate additional stress strain data points to further explore network generalisation. Some cases of over fitting have been identified. This highlights the importance of comprehensive network validation with unseen data prior to implementation.

Initially the load history dependant behaviour of architectural fabric has been investigated using uniaxial experimental data generated using 3 unique load profiles, Section 4.4.2. A network architecture with 4 inputs namely current strain, previous strain, previous stress and internal variables has been developed to be trained with the uniaxial stress strain data.

A study into the effect of training data profile, load step and hidden node number was undertaken, Section 4.4.3. Network generalisation improved when the network was trained with data from the load profile with the widest variety of features. Network performance decreased as the difference between the testing and training profile load step increased. Networks with 4 hidden nodes produced unacceptably low  $R^2$  value more frequently than those with 7 or 10 nodes but no definitive conclusions regarding the sufficient number of nodes could be made from this study.

A further study into the required number of hidden layer nodes for the uniaxial load history network indicated that the networks performed best with between 9 and 17 nodes the hidden layer. With more than 17 nodes, error increased indicating that over-fitting had begun to occur. A network with 9 hidden nodes was trained with a data set taken from the most comprehensive load profile. The trained network was capable of reproducing the material response from the two unseen profiles with a high degree of accuracy.

The findings of the uniaxial studies were used to develop a set of five unique biaxial load profiles. Each of the five load profiles used a different load ratio, 0:1, 1:2, 1:1, 2:1 or 1:0, as a repeated conditioning cycle, Section 4.4.4. As in the most successful uniaxial profile, differing upper and lower loads were used throughout the profiles to capture the effect of previous stress magnitude as well as stress ratio. These profiles aimed to capture a wide variety of loading conditions to enable the production of comprehensive data sets for both training and testing. The profiles were used to generate data sets for PVC coated polyester and PTFE coated glass fabric.

The proposed biaxial load history neural network takes inputs of current and historical strain, historical stress and internal variables for the warp and fill material directions,

Section 4.4.5. A study into the number of nodes required by the biaxial load history network demonstrated that network performance increased with an increasing number of nodes, up to approximately 15 hidden nodes. Performance neither consistently increased nor decreased with increasing hidden node numbers, up to 40 nodes. Due to the high complexity of the training data over fitting was not identified.

PVC and PTFE neural network material models with 15 hidden nodes have been trained with training data sets combining cycles from each of the experimental load profiles. The trained networks were tested in recurrent mode, where historical stress and strain inputs are taken from previous network output, with the full experimental profiles containing both seen and unseen data. For both materials, the network output demonstrates excellent agreement with the target experimental data. The network material models accurately generalise the fabric response when presented with unseen data. This indicates that the experimental load profiles have successfully captured the architectural fabric material response.

The response surface and biaxial history neural network material models, Chapter 4, have been implemented in the finite element formulation, Chapter 3. Shear response has been assumed to be decoupled from direct stress and is calculated using a typical shear modulus. An implied stiffness matrix derived from partial derivatives of the network equations is used to generate the diagonal stiffness matrix terms required for the dynamic relaxation solution algorithm.

The response surface implied stiffness matrix formulation has been validated using a network trained using a data set derived from PTFE plane stress elastic constants, Section 5.1.1. The implied stiffness matrix was calculated at 3 points on the surface. Errors between the network implied stiffness and target plane stress stiffness matrix terms were consistently less than 0.02%. The history network implied stiffness matrix formulation was validated using the finite difference method, Section 5.1.2.

Wrinkling behaviour is incorporated into the response surface network by generating additional training data in the wrinkled zone, Section 5.2.1. This was done using a plane stress approximation of the material response and applying a modification using combined stress strain wrinkling criteria. A patch mesh simulated with loading to induce stress ratios within the PVC and PTFE response surface has been used to demonstrate the implemented neural network material models, Section 5.2.2. When plotted alongside the experimental data it can be visually confirmed that the stress strain results for both materials sit on the target surface.

The PVC neural network material model has been demonstrated in a full simulation of a hypar structure. In comparison to an equivalent plane stress simulations the Network produces similar stress and displacement patterns with some variations in magnitude. More significantly, the neural network produces a strikingly smoother stress distribution than the plane stress material model when wrinkling areas developed under wind loading. The network simulation is more stable as the modelled material response is continuous and the stiffness reduces as the simulation approaches compressive stress. The plane stress material model is modified only after the equilibrium condition has been found with areas of negative stress. This leads to potentially significant discontinuities in the modelled material response in subsequent solution iterations.

In order to implement the history neural network material model, the finite element formulation has been modified to include incremental loading, Section 5.3. The effect of the difference between training load step and simulation load step has been investigated, Section 5.3.1. In a set of simulations using varying load step and multiple PVC network models, the non-linear biaxial stress strain response of the architectural fabric was reproduced with varying degrees of accuracy. Error between target and network output was typically greater when the simulation load step differed from the training data load step. This is in agreement with findings when the networks were tested outside of the finite element code. Where the training data load step was smaller than the simulation load step, the error between network output and target output consistently became unacceptably large. Nevertheless numerous network models produced simulation results in excellent agreement with the experimental results and a network model has shown to be more accurate than a typical plane stress approximation.

A history neural network simulation of a hypar undergoing loading and unloading demonstrated that the material model effectively captures load history effects including residual strain build up, strain recovery and differing response to loading and unloading, Section 5.3.2. Under both snow and wind loading residual deformation in the global z-direction was observed, positive in the wind loading case and negative in the snow loading case. This is driven by residual strain in the warp and fill directions. Under snow loading, strain is recovered in the warp and builds up in the fill. Under wind loading, strain is recovered in the fill and builds up in the warp. Stress is reduced in both directions below prestress for both load cases.

Numerous studies throughout this thesis demonstrate the variation exhibited by neural networks due to the random initiation of network weights and biases and training data division. Varying performance has been consistently observed between randomly initiated

networks trained using identical data. This has been further illustrated via variation in hypar simulation results generated using a set of PVC network models, Section 6.2. It is proposed that during model development, multiple networks should be trained using a single comprehensive data set. The networks should then be rigorously tested using unseen data. The network demonstrating the best performance should be selected for the final simulation task.

In order to assess probability of failure given the uncertainty associated with the proposed neural network material model, a FORM reliability analysis procedure has been developed, Chapter 6. Initially the analytical derivatives of the proposed limit state functions have been derived and validated for the original plane stress formulation using the finite difference method, Section 6.1.5. The full plane stress FORM reliability analysis has been validated using a simple Monte Carlo simulation of a biaxially loaded patch.

Meaningful statistical descriptions of the neural network variability cannot be derived directly for the weights and biases of the network. It has been proposed that a set of implied stiffness matrix, derived for each Gauss Point of a mesh at equilibrium, may be used in place of a neural network material model. This enables an equivalent reliability analysis to be performed. The required partial derivatives have been developed and validated using the finite difference method. Statistical properties describing the variation of implied stiffness matrix terms at each gauss point of a hypar mesh have been derived from simulations using 30 randomly initiated trained network models, Section 6.3.3.1. The coefficients of variation are relatively high, ranging between 0.18 and 0.32.

The statistical properties have been used to perform reliability analysis of the hypar mesh from which they were derived. The strength limit states converge close to 0 in less than 10 FORM iterations and produce  $\beta$  values of 4.4 in the warp direction and 3.8 in the fill. These safety factors lie in an acceptable range as defined by EC0. The strength and load statistical variables are demonstrated to be the most influential in these analysis. Despite their high coefficients of variation, only minor contributions from the material model variables are observed. The displacement limit state has a much greater dependence on the material model variables. The analysis converges with a  $\beta$  value of 0.95, which represents an unacceptably large probability of failure. Attempts to increase the allowable displacement lead the finite element code to become unstable. The principle stress limit state analysis diverged away from zero producing  $\beta$  values which are too high to be meaningful.

This thesis serves as promising proof of concept for the development and implementation of architectural fabric neural network material models. It has been demonstrated that it is possible to train neural network material models, using biaxial experimental data, to

accurately generalise the non-linear and history dependent strain stress response of architectural fabric. The trained networks have been successfully implemented in a specialist finite element code and have been used in the simulation of a realistic fabric structures. High levels of epistemic uncertainty, introduced by the training process, has been observed and quantified for the trained neural networks. Initial investigations using reliability analysis have demonstrated that while network material model uncertainty has a significant impact on the probability of failure associated with the displacement serviceability limit state, the ultimate strength limit state probability of failure is largely unaffected.

## 7.2 Recommendations for Further Work

The form-finding procedure presented in this thesis is based on the classical soap-film analogy and as such does not allow form finding with orthotropic prestress. A wide variety of fabric structural forms may be investigated using pure minimal surfaces. However, the implementation of non isometric form finding would greatly increase the number of available forms.

Currently the patterning procedure does not include seam length control allowing the development of minor discontinuities between edge lengths of neighbouring panels. Some form of geometric control is required to ensure compatibility of the final panels. The application of compensation to produce the final panel geometries should be investigated.

Currently shear behaviour is not included in the proposed neural network fabric model. The addition of inputs to a neural network model is reasonably arbitrary. For instance it is straight forward to add an additional shear strain input to the response surface network. However, the development of a network that includes both direct and shear stresses is limited by the collection of sufficiently comprehensive data. Various shear testing methodologies have been proposed which facilitate the application of both direct and shear stresses. These methodologies may be adapted to allow the generation of sufficiently comprehensive data required for network training.

It has been demonstrated that a finite element simulation of a test specimen may be incorporated into the network training process to directly extract constitutive behaviour from boundary measurements of load and displacement. This form of self-learning simulation was initially demonstrated using idealised panels and bars [117]. More recently it has been applied to non-liner connections [116] and soils [111]. This training methodology offers a possible opportunity to capture fabric response through testing that simulates the loading conditions undergone by insitu fabric structures. For example the development of a doubly curved surface form an initially flat panel. The development of specialist experimental equipment would be required in order to implement this methodology. Data may also potentially be gathered from insitu fabric structures.

Benchmarking of the proposed neural network fabric material model simulations against physical tests would also be an ideal way of validating the material models . Digital image correlation (DIC) has been applied to architectural fabric biaxial [6] and shear [64] tests to provide detailed strain and displacement distribution data. Initial attempts were made to simulate a full biaxial test specimen but the unrestrained edges of the slits in the arms

caused the simulation to become unstable. Development of a model suitable for the representation of the unrestrained arms would facilitate a direct comparison between a biaxial test simulation and DIC results. Rich displacement and strain data for simulation validation could also be gathered from full scale structural tests or insitu structures using DIC.

The reliability analysis procedure requires further development in order to improve stability to allow analysis of the serviceability limit state. As described in Chapter 6, in order to perform reliability analysis on an iteratively loaded fabric structure the limit state function derivatives must be derived for each load increment. Taking the displacement derivative with respect to the first implied stiffness term,  $E_{1,1}^n$ , as an example. The displacement derivative for the first load increment with respect to  $E_{1,1}^{n=1}$  is the same as those used in Section 6.3 for a single load increment. At the second load increment the total displacement is dependent on the variables of both the first and second load increments,  $E_{1,1}^{n=1}$  and  $E_{1,1}^{n=2}$ . If the maximum load occurs at the 5th increment the final maximum displacement is a function of  $E_{1,1}^{n=1 \rightarrow 5}$ . The following 5 derivatives are required,

$$\frac{d\delta^{max}}{dE_{1,1}^{n=1}}, \frac{d\delta^{max}}{dE_{1,1}^{n=2}}, \frac{d\delta^{max}}{dE_{1,1}^{n=3}}, \frac{d\delta^{max}}{dE_{1,1}^{n=4}}, \frac{d\delta^{max}}{dE_{1,1}^{n=5}}$$

These derivatives maybe accessed using the finite difference method but this would be very computationally expensive. Therefore, analytical definition of these terms is a requirement in order to perform incremental load reliability analysis.

Computational expense is a limiting factor not only when considering incremental load reliability analysis. It also limits the complexity and density of meshes that can be analysed using the code developed in this Thesis. It would be beneficial to future research using this code to redevelop the Fortran code to make it more user friendly and more efficient. Initial efforts have been made using Matlab scripts and Rhino Grasshopper to provide graphical user interfaces. It would be of further benefit to rewrite the code in an object orientated language such as C# to facilitate better variable management, better integration with powerful graphical tools such as Grasshopper and clear user interfaces.

## Bibliography

1. Bradatsch, J., et al., *Form*, in *European Design Guide for Tensile Surface Structures*. 2004, TensiNet. p. 43 - 96.
2. FabriTec Structures. *Skysong ASU*. [cited 2014 04/02/2014]; Available from: <http://www.fabritecstructures.com/portfolio-skysong.html#ad-image-0>.
3. Kostick, M. *Hubert H. Humphrey Metrodome Roof Snow Collapse of 2010*. 2012 [cited 2013 20/12/2013]; Available from: <http://failures.wikispaces.com/Hubert+H.+Humphrey+Metrodome+Roof+Snow+Collapse+of+2010>.
4. Stremski, M. *Dallas Cowboys Indoor Practice Facility Collapse (May 2, 2009)*. 2012 [cited 2013 20/11/2013]; Available from: <http://failures.wikispaces.com/Dallas+Cowboy+Indoor+Practice+Facility+Collapse#Dallas>
5. Blum, R., H. Bogner, and G. Nemoz, *Material Properties and Testing*, in *European Design Guide for Tensile Surface Structures*. 2004, TensiNet. p. 219-242.
6. Dinh, T.D., et al., *A new elasto-plastic material model for coated fabric*. Engineering Structures, 2014. **71**(0): p. 222-233.
7. Forster, B. and M. Mollaert, *Engineering Fabric Architecture*, in *European Design Guide for Tensile Surface Structures*. 2004, TensiNet. p. 25 - 42.
8. Gosling, P.D., B.N. Bridgens, and L. Zhang, *Adoption of a reliability approach for membrane structure analysis*. Structural Safety, 2013. **40**(0): p. 39-50.
9. Nouri-Baranger, T., *Computational methods for tension-loaded structures*. Archives of Computational Methods in Engineering, 2004. **11**(2): p. 143-186.
10. Otto, F., *Tensile structures*. Vol. 1. 1967: Cambridge, Mass. : M. I. T. Press
11. Gellin, S. and R.M.O. Pauletti, *Necking limits of conoid membrane structures with variable stress ratio*. Engineering Structures, 2013. **50**(0): p. 90-95.
12. Liddell, I., *The Engineering of Surface Structures*, in *Patterns 5*, R. McElhinney and M. Dickson, Editors. 1998, Buro Happold Consulting Engineers
13. Argyris, J.H., T. Angelopoulos, and B. Bichat, *GENERAL METHOD FOR THE SHAPE FINDING OF LIGHTWEIGHT TENSION STRUCTURES*. Computer Methods in Applied Mechanics and Engineering, 1974. **3**(1): p. 135-149.
14. Schek, H.J., *The force density method for form finding and computation of general networks*. Computer Methods in Applied Mechanics and Engineering, 1974. **3**(1): p. 115-134.
15. Linkwitz, K., *About formfinding of double-curved structures*. Engineering Structures, 1999. **21**(8): p. 709-718.
16. Pauletti, R.M.O. and P.M. Pimenta, *The natural force density method for the shape finding of taut structures*. Computer Methods in Applied Mechanics and Engineering, 2008. **197**(49-50): p. 4419-4428.
17. Ströbel, D., Singer, P., *Computational Modelling of Lightweight Structures; Formfinding, Load Analysis and Cutting Pattern Generation*, in *Textile Roofs 2005, The Tenth international workshop on the design and practical realisation of architectural membrane structures*. 2005: Berlin, Germany.
18. GmbH, t. *Easy - Software for the integrated planning and calculation of lightweight surface structures*. [cited 2015 14/02/2015]; Available from: <http://technet-gmbh.com/>.

19. Bletzinger, K.-U. and E. Ramm, *Structural optimization and form finding of light weight structures*. Computers & Structures, 2001. **79**(22–25): p. 2053-2062.
20. Bletzinger, K.-U., et al., *Computational methods for form finding and optimization of shells and membranes*. Computer Methods in Applied Mechanics and Engineering, 2005. **194**(30–33): p. 3438-3452.
21. Bletzinger, K.-U. and E. Ramm, *A General Finite Element Approach to the Form Finding of Tensile Structures by the Updated Reference Strategy*. International Journal of Space Structures, 1999. **14**(2): p. 131-145.
22. Bonet, J. and J. Mahaney, *Form finding of membrane structures by the updated reference method with minimum mesh distortion*. International Journal of Solids and Structures, 2001. **38**(32–33): p. 5469-5480.
23. Barnes, M.R., *Form-finding and analysis of tension space structures by dynamic relaxation*. 1977, City University London.
24. Wakefield, D.S., *Dynamic relaxation analysis of pretensioned networks supported by compression arches*. 1980, City University London: City University London.
25. Barnes, M., *Form Finding and Analysis of Tension Structures by Dynamic Relaxation*. International Journal of Space Structures, 1999. **14**(2): p. 89-104.
26. Kilian, A. and J. Ochsendorf, *Particle-spring systems for structural form finding*. Journal-International Association for Shell and Spatial Structures, 2005. **148**: p. 77.
27. Aish, F., et al., *The use of a particle method for the modelling of isotropic membrane stress for the form finding of shell structures*. Computer-Aided Design, 2015. **61**(0): p. 24-31.
28. Veenendaal, D. and P. Block, *An overview and comparison of structural form finding methods for general networks*. International Journal of Solids and Structures, 2012. **49**(26): p. 3741-3753.
29. Wakefield, D.S., *Engineering analysis of tension structures: theory and practice*. Engineering Structures, 1999. **21**(8): p. 680-690.
30. Gosling, P.D., *Numerical Modelling of Stable Minimal Surfaces*. 1992, University of Warwick.
31. Zhang, L., *Reliability Analysis of Fabric Structures*, in *School of Civil Engineering and Geosciences*. 2010, Newcastle University: Newcastle, UK.
32. Stanuszek, M., *FE analysis of large deformations of membranes with wrinkling*. Finite Elements in Analysis and Design, 2003. **39**(7): p. 599-618.
33. Gosling, P.D. and W.J. Lewis, *Optimal structural membranes—I. Formulation of a curved quadrilateral element for surface definition*. Computers & Structures, 1996. **61**(5): p. 871-883.
34. Cook, R.D., D.S. Malkus, and M.E. Plesch, *Concepts and Applications of Finite Element Analysis*. Third ed. 1989, Canada: John Wiley & Sons, Inc.
35. Pargana, J.B., D. Lloyd-Smith, and B.A. Izzuddin, *Fully integrated design and analysis of Tensioned Fabric Structures: Finite elements and case studies*. Engineering Structures, 2010. **32**(4): p. 1054-1068.
36. Gosling, P.D., et al., *Analysis and design of membrane structures: Results of a round robin exercise*. Engineering Structures, 2013. **48**(0): p. 313-328.
37. Valdebenito, M.A., H.J. Pradlwarter, and G.I. Schuëller, *The role of the design point for calculating failure probabilities in view of dimensionality and structural nonlinearities*. Structural Safety, 2010. **32**(2): p. 101-111.
38. Katafygiotis, L.S. and K.M. Zuev, *Geometric insight into the challenges of solving high-dimensional reliability problems*. Probabilistic Engineering Mechanics, 2008. **23**(2–3): p. 208-218.
39. Bletzinger, K.-U., et al., *Optimal shapes of mechanically motivated surfaces*. Computer Methods in Applied Mechanics and Engineering, 2010. **199**(5–8): p. 324-333.
40. Der Kiureghian, A. and T. Dakessian, *Multiple design points in first and second-order reliability*. Structural Safety, 1998. **20**(1): p. 37-49.

41. Koduru, S.D. and T. Haukaas, *Feasibility of FORM in finite element reliability analysis*. Structural Safety, 2010. **32**(2): p. 145-153.
42. Pargana, J.B., D. Lloyd-Smith, and B.A. Izzuddin, *Advanced material model for coated fabrics used in tensioned fabric structures*. Engineering Structures, 2007. **29**(7): p. 1323-1336.
43. MSAJ, *MSAJ/M-02-1995. Testing Method for Elastic Constants of Membrane Materials*. 1995, Membrane Structures Association of Japan.
44. Day, A.S. *Stress strain equations for non-linear behaviour of coated woven fabrics, IASS symposium proceedings: shells, membranes and space frames*. 1986. Osaka: Elsevier.
45. Tan, P., L. Tong, and G.P. Steven, *Modelling for predicting the mechanical properties of textile composites—A review*. Composites Part A: Applied Science and Manufacturing, 1997. **28**(11): p. 903-922.
46. Bassett, R.J., R. Postle, and N. Pan, *Experimental methods for measuring fabric mechanical properties: A review and analysis*. Textile Research Journal, 1999. **69**(11): p. 866-875.
47. Cao, J., et al., *Characterization of mechanical behavior of woven fabrics: Experimental methods and benchmark results*. Composites Part A: Applied Science and Manufacturing, 2008. **39**(6): p. 1037-1053.
48. ASTM, *ASTM D5035 - 11 Standard Test Method for Breaking Force and Elongation of Textile Fabrics (Strip Method)*. 2011, ASTM International
49. ISO, *ISO 1421:1998 Rubber- or plastics-coated fabrics -- Determination of tensile strength and elongation at break*. 1998.
50. ISO, *ISO 13934-1:2013 Textiles - Tensile properties of fabrics - Part 1: Determination of maximum force and elongation at maximum force using the strip method*. 2013.
51. ISO, *ISO 13934-2:2014 Textiles - Tensile properties of fabrics - Part 2: Determination of maximum force using the grab method*. 2014.
52. Chen, S., X. Ding, and H. Yi, *On the Anisotropic Tensile Behaviors of Flexible Polyvinyl Chloride-coated Fabrics*. Textile Research Journal, 2007. **77**(6): p. 369-374.
53. Quaglini, V., C. Corazza, and C. Poggi, *Experimental characterization of orthotropic technical textiles under uniaxial and biaxial loading*. Composites Part A: Applied Science and Manufacturing, 2008. **39**(8): p. 1331-1342.
54. Meng, L. and M.E. Wu, *Study on stress relaxation and creep properties of PTFE membrane*. Jianzhu Cailiao Xuebao/Journal of Building Materials, 2012. **15**(2): p. 206-210.
55. Zhang, Y., Q. Zhang, and H. Lv, *Mechanical properties of polyvinylchloride-coated fabrics processed with Preconstraint® technology*. Journal of Reinforced Plastics and Composites, 2012. **31**(23): p. 1670-1684.
56. Zhang, Y., et al., *Mechanical properties of PTFE coated fabrics*. Journal of Reinforced Plastics and Composites, 2010. **29**(24): p. 3624-3630.
57. Blum, R., H. Bogner, and G. Nemoz, *Testing methods and standards, in European design guide for tensile structures*, B. Forster and M. Mollaert, Editors. 2004, Tensinet. p. 293-322.
58. Ambroziak, A. and P. Kłosowski, *Mechanical testing of technical woven fabrics*. Journal of Reinforced Plastics and Composites, 2013. **32**(10): p. 726-739.
59. Bridgens, B.N., *Architectural fabric properties: determination, representation & prediction, in School of Civil Engineering & Geosciences*. 2005, Newcastle University.
60. Bridgens, B.N. and P.D. Gosling, *A new biaxial test protocol for architectural fabrics*. in *IASS 2004 symposium: shell and spatial structures from models to realisation*. 2004. Montpellier: IASS.
61. Galliot, C. and R.H. Luchsinger, *Uniaxial and biaxial mechanical properties of ETFE foils*. Polymer Testing, 2011. **30**(4): p. 356-365.

62. Kabche, J.P., M.L. Peterson, and W.G. Davids, *Effect of inflation pressure on the constitutive response of coated woven fabrics used in airbeams*. Composites Part B: Engineering, 2011. **42**(3): p. 526-537.
63. Vysochina, K., et al., *Identification of shear stiffness of soft orthotropic textile composites: Part I - Development of a mixed method for shear elastic constant identification*. Journal of Industrial Textiles, 2005. **35**(2): p. 137-155.
64. Colman, A.G., et al., *Shear behaviour of architectural fabrics subjected to biaxial tensile loads*. Composites Part A: Applied Science and Manufacturing, 2014. **66**(0): p. 163-174.
65. Galliot, C. and R.H. Luchsinger, *The shear ramp: A new test method for the investigation of coated fabric shear behaviour – Part I: Theory*. Composites Part A: Applied Science and Manufacturing, 2010. **41**(12): p. 1743-1749.
66. Galliot, C. and R.H. Luchsinger, *A simple model describing the non-linear biaxial tensile behaviour of PVC-coated polyester fabrics for use in finite element analysis*. Composite Structures, 2009. **90**(4): p. 438-447.
67. Minami, H., *A multi-step linear approximation method for nonlinear analysis of stress and deformation of coated plain-weave fabric*. Journal of Textile Engineering, 2006. **52**(5): p. 189-195.
68. Gosling, P.D. and B.N. Bridgens, *Material Testing Computational Mechanics A New Philosophy For Architectural Fabrics*. International Journal of Space Structures, 2008. **23**: p. 215-232.
69. Oasys. *GSA Fabric*. 14/11/2010; Available from: <http://www.oasys-software.com>.
70. Bridgens, B.N. and P.D. Gosling, *Direct stress-strain representation for coated woven fabrics*. Computers & Structures, 2004. **82**(23-26): p. 1913-1927.
71. MSAJ, *MSAJ/M-01-1993. Testing Method for in-plane shear stiffness of membrane materials*. 1993, Membrane Structures Association of Japan.
72. K., C.K., *Composite materials science and engineering*. 3rd ed. ed. 2012: New York ; London : Springer.
73. Stubbs, N. and S. Thomas, *A nonlinear elastic constitutive model for coated fabrics*. Mechanics of Materials, 1984. **3**(2): p. 157-168.
74. Kato, S., T. Yoshino, and H. Minami, *Formulation of constitutive equations for fabric membranes based on the concept of fabric lattice model*. Engineering Structures, 1999. **21**(8): p. 691-708.
75. Pan, N., *Analysis of woven fabric strengths: Prediction of fabric strength under uniaxial and biaxial extensions*. Composites Science and Technology, 1996. **56**(3): p. 311-327.
76. Sun, H., N. Pan, and R. Postle, *On the Poisson's ratios of a woven fabric*. Composite Structures, 2005. **68**(4): p. 505-510.
77. Xue, P., J. Cao, and J. Chen, *Integrated micro/macro-mechanical model of woven fabric composites under large deformation*. Composite Structures, 2005. **70**(1): p. 69-80.
78. Potluri, P. and V.S. Thammandra, *Influence of uniaxial and biaxial tension on meso-scale geometry and strain fields in a woven composite*. Composite Structures, 2007. **77**(3): p. 405-418.
79. Boisse, P., et al., *Finite element simulations of textile composite forming including the biaxial fabric behaviour*. Composites Part B: Engineering, 1997. **28**(4): p. 453-464.
80. Boisse, P., A. Gasser, and G. Hivet, *Analyses of fabric tensile behaviour: determination of the biaxial tension-strain surfaces and their use in forming simulations*. Composites Part A: Applied Science and Manufacturing, 2001. **32**(10): p. 1395-1414.
81. Ivanov, I. and A. Tabiei, *Three-dimensional computational micro-mechanical model for woven fabric composites*. Composite Structures, 2001. **54**(4): p. 489-496.
82. Cui, L. and D. Sheng, *Genetic algorithms in probabilistic finite element analysis of geotechnical problems*. Computers and Geotechnics, 2005. **32**(8): p. 555-563.

83. Feng, X.-T., et al., *Identification of visco-elastic models for rocks using genetic programming coupled with the modified particle swarm optimization algorithm*. International Journal of Rock Mechanics and Mining Sciences, 2006. **43**(5): p. 789-801.
84. Feng, X.-T. and C. Yang, *Genetic evolution of nonlinear material constitutive models*. Computer Methods in Applied Mechanics and Engineering, 2001. **190**(45): p. 5957-5973.
85. Feng, X.-T. and C. Yang, *Coupling recognition of the structure and parameters of nonlinear constitutive material models using hybrid evolutionary algorithms*. International Journal for Numerical Methods in Engineering, 2004. **59**(9): p. 1227-1250.
86. Badnava, H., et al., *Determination of combined hardening material parameters under strain controlled cyclic loading by using the genetic algorithm method*. Journal of Mechanical Science and Technology, 2012. **26**(10): p. 3067-3072.
87. Lostado, R., et al., *Using genetic algorithms to optimize the material behaviour model in finite element models of processes with cyclic loads*. Journal of Strain Analysis for Engineering Design, 2011. **46**(2): p. 143-159.
88. Ghaboussi, J., *Advances of Soft Computing in Engineering*, in *Advances in Neural Networks in Computational Mechanics and Engineering*, Z. Waszczyszyn, Editor. 2010, Springer Vienna. p. 191-236.
89. Waszczyszyn, Z. and L. Ziemiański, *Neural networks in mechanics of structures and materials – new results and prospects of applications*. Computers & Structures, 2001. **79**(22-25): p. 2261-2276.
90. Wu, Y. and Y. Ren. *Prediction of mechanical properties of hot rolled strips by BP artificial neural network*. in *2011 International Conference on Information Technology, Computer Engineering and Management Sciences, ICM 2011, September 24, 2011 - September 25, 2011*. 2011. Nanjing, Jiangsu, China: IEEE Computer Society.
91. Mo, B. and Z. Pan. *Weld quality prediction for resistance microwelding of fine cu wire based on the back-propagation neural network*. in *2nd International Conference on Advanced Design and Manufacturing Engineering, ADME 2012, August 16, 2012 - August 18, 2012*. 2012. Taiyuan, China: Trans Tech Publications.
92. Agarwal, M., *Systematic classification of neural-network-based control*. IEEE Control Systems Magazine, 1997. **17**(2): p. 75-93.
93. Kim, T.-H. *Pattern recognition using artificial neural network: A review*. in *4th International Conference on Information Security and Assurance, ISA 2010, June 23, 2010 - June 25, 2010*. 2010. Miyazaki, Japan: Springer Verlag.
94. Krishnaraj, V. *Optimisation of end milling parameters on aluminium/Sic composites using response surface and artificial neural network methodologies*. in *Soft Computing Techniques for Machining of Composites*. 2013. Kreuzstrasse 10, Zurich-Durnten, CH-8635, Switzerland: Trans Tech Publications Ltd.
95. Liujie, X., et al., *Optimisation of chemical composition of high speed steel with high vanadium content for abrasive wear using an artificial neural network*. Materials and Design, 2007. **28**(3): p. 1031-1037.
96. Ghaboussi, J., et al. *Neural network constitutive models determined from structural tests*. 1996.
97. Jain, A.K., M. Jianchang, and K.M. Mohiuddin, *Artificial neural networks: a tutorial*. Computer, 1996. **29**(3): p. 31-44.
98. Basheer, I.A. and M. Hajmeer, *Artificial neural networks: fundamentals, computing, design, and application*. Journal of Microbiological Methods, 2000. **43**(1): p. 3-31.
99. Ghaboussi, J., et al., *Autoprogressive training of neural network constitutive models*. International Journal for Numerical Methods in Engineering, 1998. **42**(1): p. 105-126.

100. Lefik, M. and B.A. Schrefler, *Artificial neural network as an incremental non-linear constitutive model for a finite element code*. Computer Methods in Applied Mechanics and Engineering, 2003. **192**(28-30): p. 3265-3283.
101. Gründig, L., Moncrieff, E., Singer, P., Ströbel, D., *A History of the principal developments and applications of the force density method in Germany 1970-1999*, in *IASS-IACM 2000, Fourth International Colloquium on Computation of Shell & Spatial Structures*. 2000: Chania-Crete, Greece.
102. Williams, C.J.K., *Use of structural analogy in generation of smooth surfaces for engineering purposes*. Computer-Aided Design, 1987. **19**(6): p. 310-322.
103. Borges, H.B. and J.C. Nievola. *Hierarchical classification using a competitive neural network*. in *2012 8th International Conference on Natural Computation, ICNC 2012, May 29, 2012 - May 31, 2012*. 2012. Chongqing, China: IEEE Computer Society.
104. Flanagan, J.A., *Self-organisation in Kohonen's SOM*. Neural Networks, 1996. **9**(7): p. 1185-1197.
105. Popescu, D., et al., *Competitive Hopfield neural network model for evaluating pedicle screw placement accuracy*. Strojniski Vestnik/Journal of Mechanical Engineering, 2012. **58**(9): p. 509-516.
106. Ming, G. *Fuzzy art neural network model and its application*. in *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS 2009, November 20, 2009 - November 22, 2009*. 2009. Shanghai, China: IEEE Computer Society.
107. Rafiq, M.Y., G. Bugmann, and D.J. Easterbrook, *Neural network design for engineering applications*. Computers & Structures, 2001. **79**(17): p. 1541-1552.
108. Kasperkiewicz, J., *The applications of ANNs in certain materials-analysis problems*. Journal of Materials Processing Technology, 2000. **106**(1-3): p. 74-79.
109. Flood, I., L. Muszynski, and S. Nandy, *Rapid analysis of externally reinforced concrete beams using neural networks*. Computers & Structures, 2001. **79**(17): p. 1553-1559.
110. Hadi, M.N.S., *Neural networks applications in concrete structures*. Computers & Structures, 2003. **81**(6): p. 373-381.
111. Fu, Q., et al., *Integration of laboratory testing and constitutive modeling of soils*. Computers and Geotechnics, 2007. **34**(5): p. 330-345.
112. Hashash, Y.M.A., et al. *Constitutive soil behavior representation via artificial neural networks: A shift from soil models to soil behavior data*. 2006.
113. Zhang, Z. and K. Friedrich, *Artificial neural networks applied to polymer composites: a review*. Composites Science and Technology, 2003. **63**(14): p. 2029-2044.
114. Yun, G.J., J. Ghaboussi, and A.S. Elnashai, *A new neural network-based model for hysteretic behavior of materials*. International Journal for Numerical Methods in Engineering, 2008. **73**(Compendex): p. 447-469.
115. Jung, S. and J. Ghaboussi, *Neural network constitutive model for rate-dependent materials*. Computers and Structures, 2006. **84**(15-16): p. 955-963.
116. Yun, G.J., J. Ghaboussi, and A.S. Elnashai, *Self-learning simulation method for inverse nonlinear modeling of cyclic behavior of connections*. Computer Methods in Applied Mechanics and Engineering, 2008. **197**(Compendex): p. 2836-2857.
117. Shin, H.S. and G.N. Pande, *On self-learning finite element codes based on monitored response of structures*. Computers and Geotechnics, 2000. **27**(3): p. 161-178.
118. Xing-xing, W. and L. Jin-guo. *A New Early Stopping Algorithm for Improving Neural Network Generalization*. in *Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on*. 2009.
119. Hashash, Y.M.A., S. Jung, and J. Ghaboussi, *Numerical implementation of a neural network based material model in finite element analysis*. International Journal for Numerical Methods in Engineering, 2004. **59**(7): p. 989-1005.

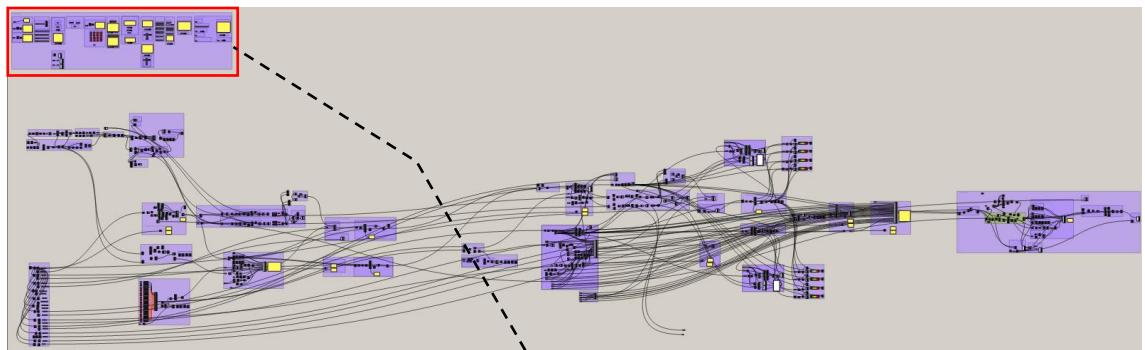
120. Shen, Y., et al., *Finite element analysis of V-ribbed belts using neural network based hyperelastic material model*. International Journal of Non-Linear Mechanics, 2005. **40**(6): p. 875-890.
121. Jung, S. and J. Ghaboussi, *Neural network constitutive model for rate-dependent materials*. Computers & Structures, 2006. **84**(15–16): p. 955-963.
122. Zienkiewicz, O.C., *The Finite Element Method*. 3rd ed. 1977, London ; New York: McGraw-Hill
123. Rossi, R., et al., *Simulation of light-weight membrane structures by wrinkling model*. International Journal for Numerical Methods in Engineering, 2005. **62**(15): p. 2127-2153.
124. Bridgens, B.N. and P.D. Gosling. *New developments in fabric structure analysis*. in *HKIE/IStructE Joint Structural Division Annual Seminar: Innovative Light-weight Structures and Sustainable Facades*. 2008. Hong Kong.
125. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning representations by back-propagating errors*. Nature, 1986. **323**: p. 533-536.
126. *Neural Network Toolbox*. [cited 2012 29th November]; Available from: <http://www.mathworks.co.uk/products/neural-network/>.
127. Hagan, M.T. and M.B. Manhaj, *Training Feedforward Networks with the Marquardt Algorithm*. IEEE Transactions on Neural Networks, 1994. **5**(6): p. 989-992.
128. [Anon], *BS EN ISO 1421. Rubber- or plastics-coated fabrics. Determination of tensile strength and elongation at break*. 1998, British Standards Institute.
129. Ghaboussi, J. and D.E. Sidarta, *New nested adaptive neural networks (NANN) for constitutive modeling*. Computers and Geotechnics, 1998. **22**(1): p. 29-52.
130. Barnes, M., B. Forster, and M. Dencher, *Structural Design Basis and Safety Criteria, in European Design Guide for Tensile Surface Structures*. 2004, TensiNet. p. 177 - 204.

## Appendix A. Grasshopper Mesh Generation

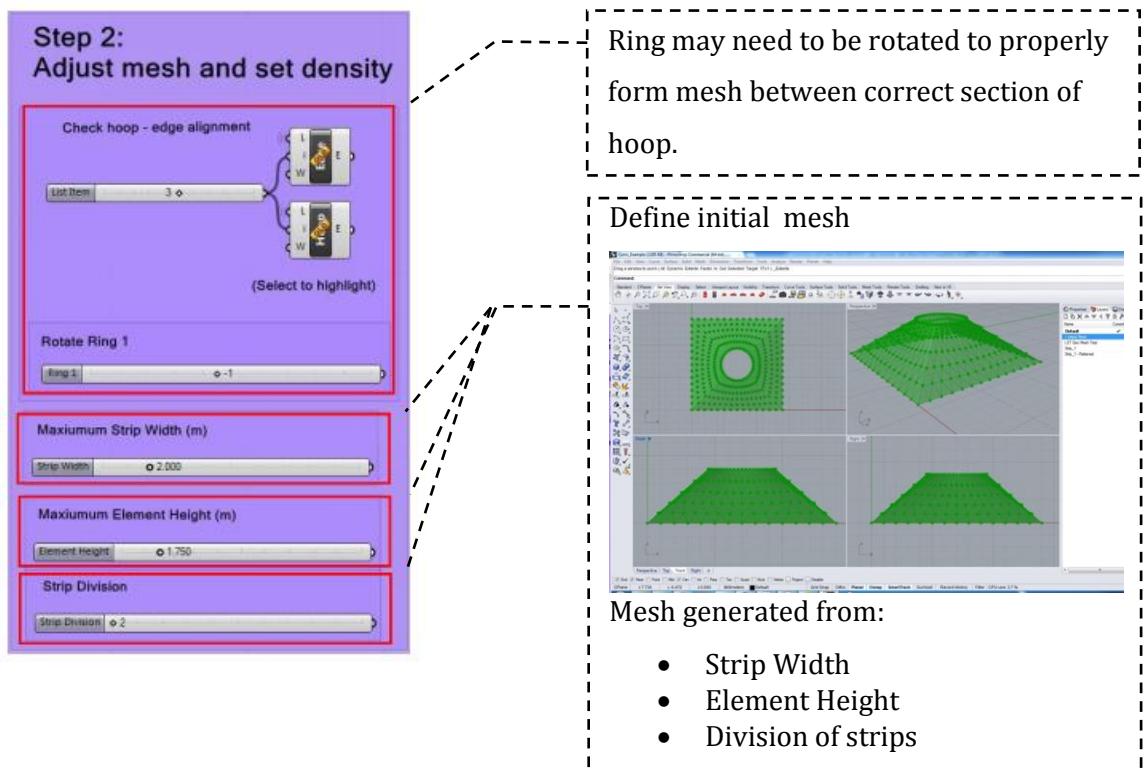
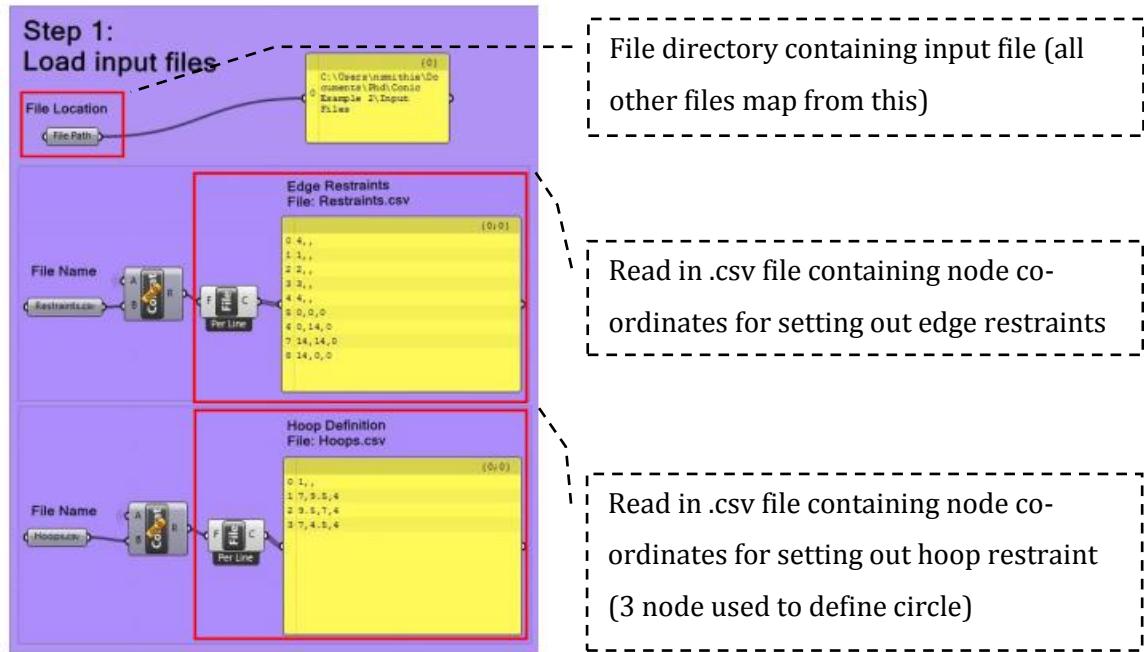
### Conic Mesh Generation

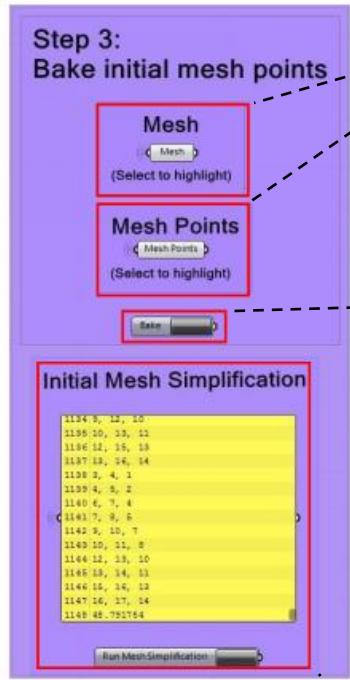
Grasshopper is a powerful open access algorithmic modelling plug-in for Rhino 3D. It has been used in this thesis for mesh generation as it allows complicated meshes to be generated with relative ease. A number of scripts have been generated for different structure types, only the conic script is shown here. However all work in a similar manner. As grasshopper is a Rhino plug-in it creates an excellent graphical interface and allows the mesh to be captured in the form of an exportable Rhino 3D file. The Grasshopper script interfaces with my Fortran code modules through the generation of .csv input files and by reading .txt Fortran module output files. Fortran modules may also be called from within the Grasshopper. For more information visit: <http://www.grasshopper3d.com/>

Overview of Conic Mesh Generation Grasshopper script:



This area of the script contains all of the objects for the user interface, these will be described in the following pages. The rest of the script performs the various functions of mesh generation, file generation



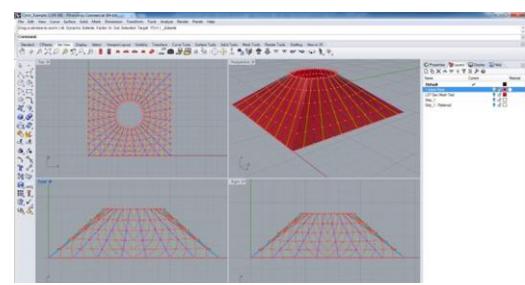


Preview mesh and node points in Rhino viewport

Bake initial mesh to Rhino view-port.

Panels are automatically generated with geodesic control strings along the edges (Green) and down the centre line (blue)

The panel warp direction angle is aligned with the centre line



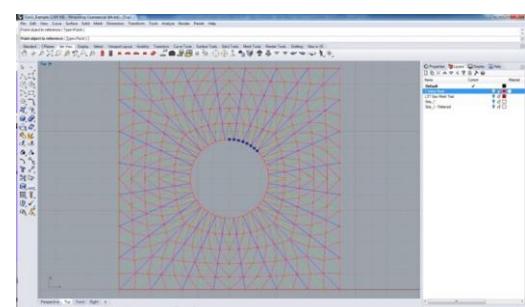
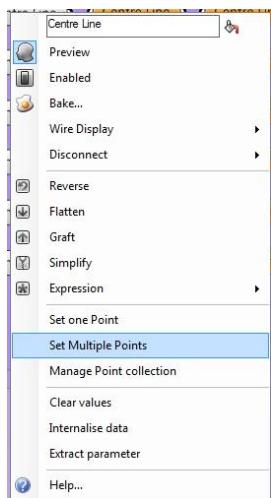
Mesh is simplified using Fortran script to remove any coincident nodes



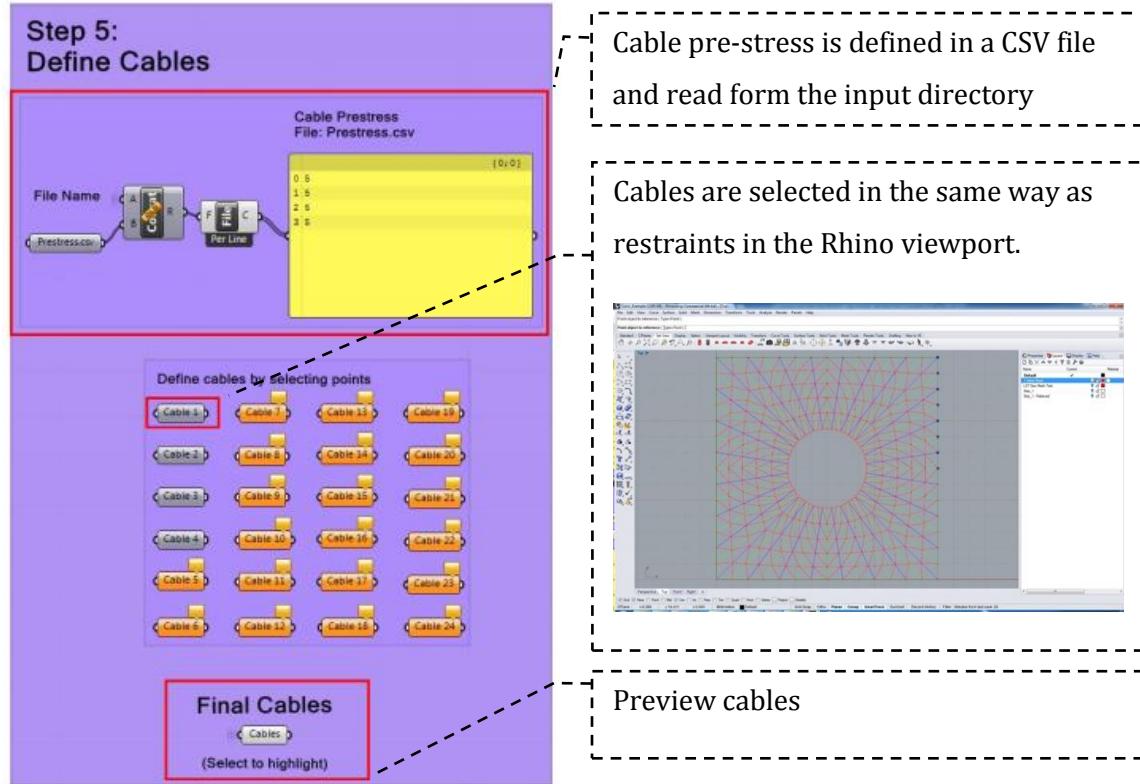
Hoop and edge restraints are selected from the baked mesh in the Rhino view port

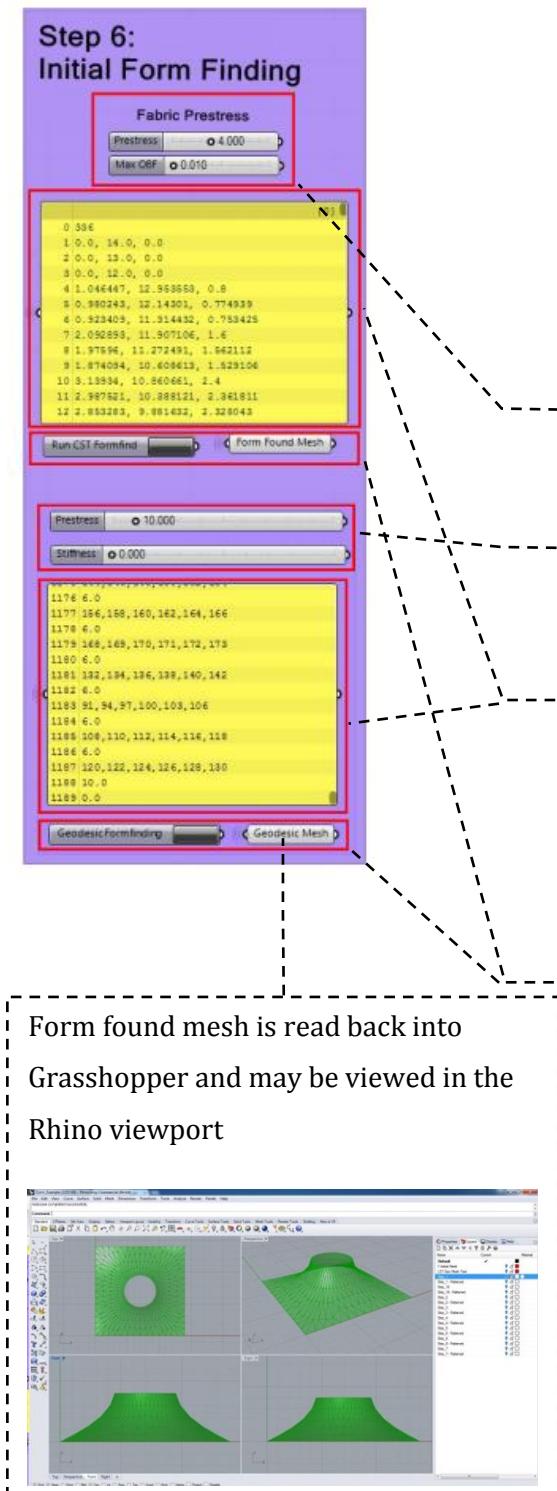
Right click on control and select "Set multiple points"

Selected points are highlighted in blue



Preview restraints (preview geometry is highlighted in green)





An initial form-finding is run using a CST element formulation (this is optional but helps to generate a regular mesh geometry)

Geodesic string elements may be ignored if not required

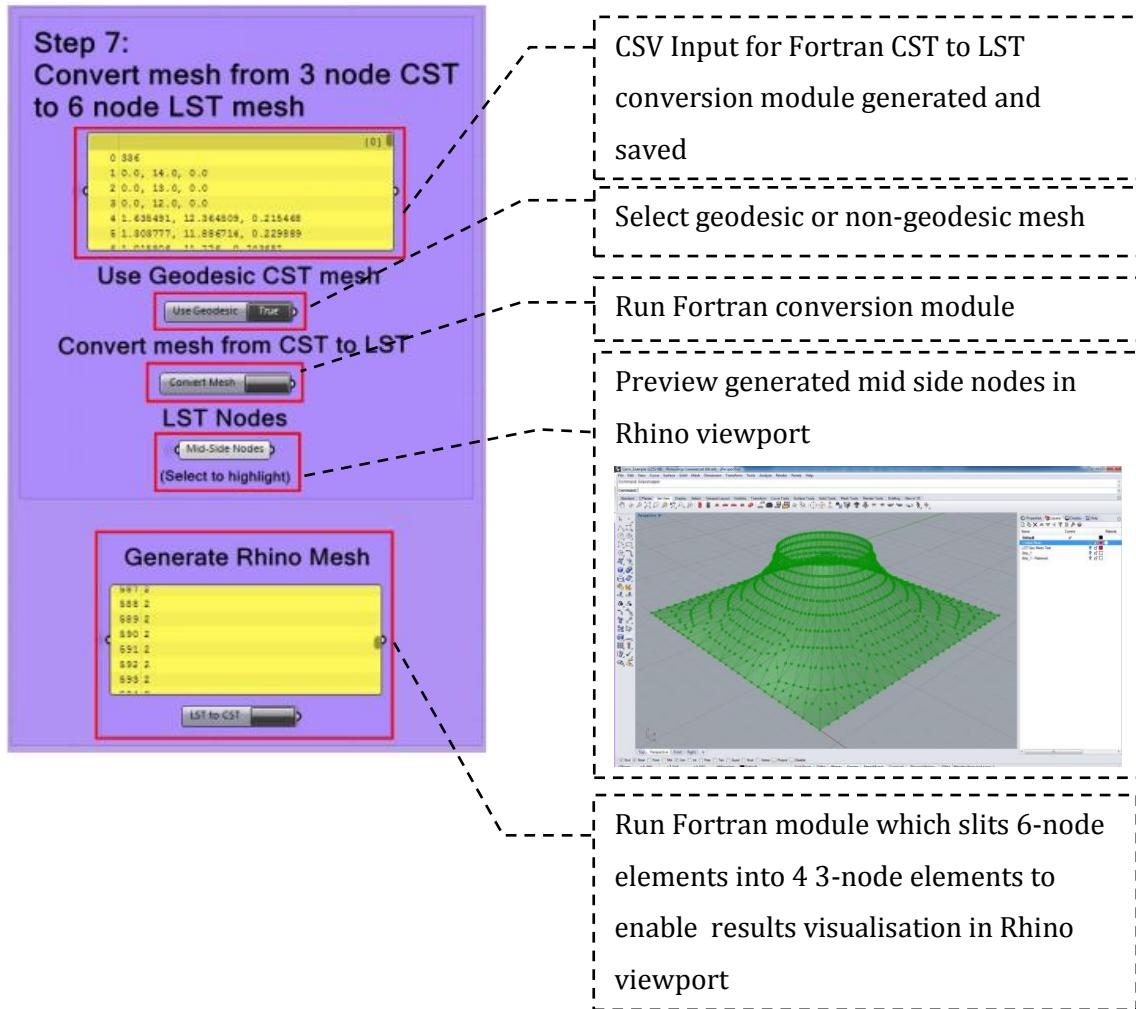
Fabric pre-stress and allowable out of balance force is set

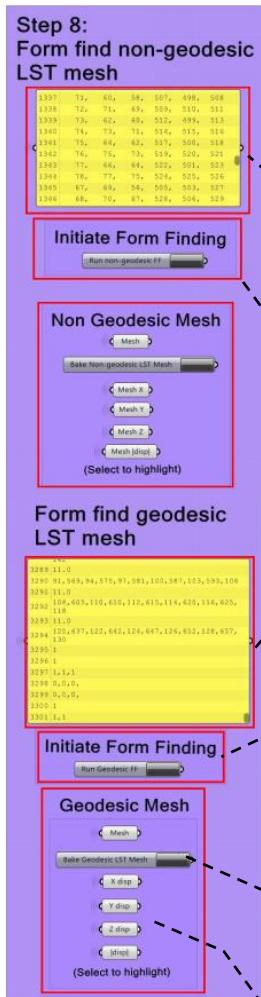
Geodesic string pre-stress and stiffness is set

Analysis input is automatically generated and is saved to a CSV file that in turn is read by the Fortran FE code

Fortran FE code if called from Grasshopper, analysis progress is displayed in a console:

```
C:\Users\lsmithie\Documents\PhD\Conic Example 2\CST geodesic FP\CST - Formfinding.exe
Max out of balance force: 4.8642893554982
Max out of balance force: 1.7938683344331
Max out of balance force: 1.51353979784238
Max out of balance force: 0.51353979784238
Max out of balance force: 0.34154431129517
Max out of balance force: 0.179234835580772
Max out of balance force: 0.179234835580772
Max out of balance force: 0.193336592882152
Max out of balance force: 0.193336592882152
Max out of balance force: 7.572233426486014E-002
Max out of balance force: 6.221958158608529E-002
Max out of balance force: 4.771841884848954E-002
Max out of balance force: 3.495214136809514E-002
Max out of balance force: 2.861661815912361E-002
Max out of balance force: 2.191625212121434E-002
Max out of balance force: 1.6640483862793469E-002
Max out of balance force: 1.49151367844182E-002
Max out of balance force: 1.151431367844182E-002
Max out of balance force: 7.977817972781668E-003
Convergence criteria met
Fortran Pause - Enter command CCR or <CCR> to continue.
```





Final LST mesh form-finding is run in the same way as the initial CST mesh form finding.

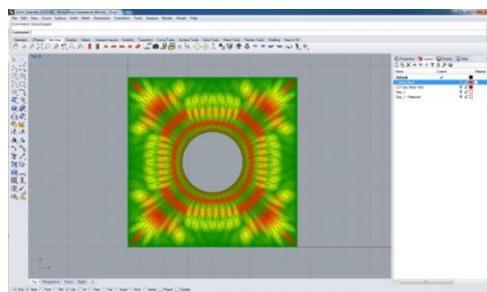
Again geodesic string elements may be ignored if not required

Analysis input is automatically generated and is saved to a CSV file that in turn is read by the Fortran FE code

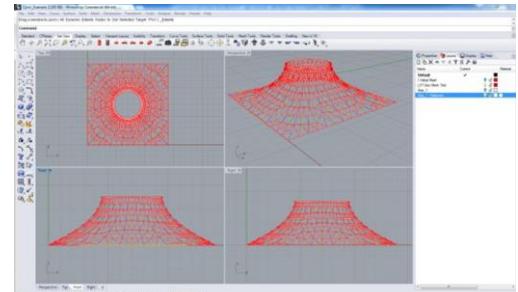
Fortran FE code if called from Grasshopper, analysis progress is displayed in a console:

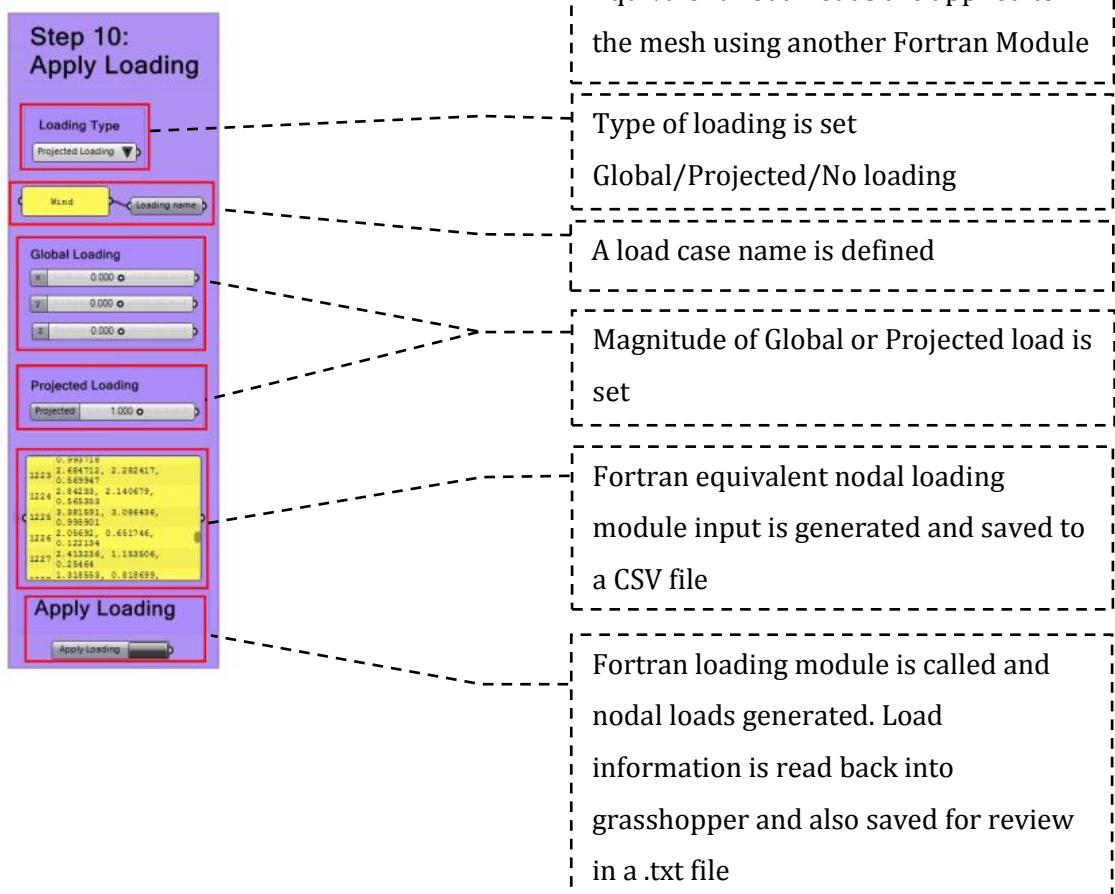
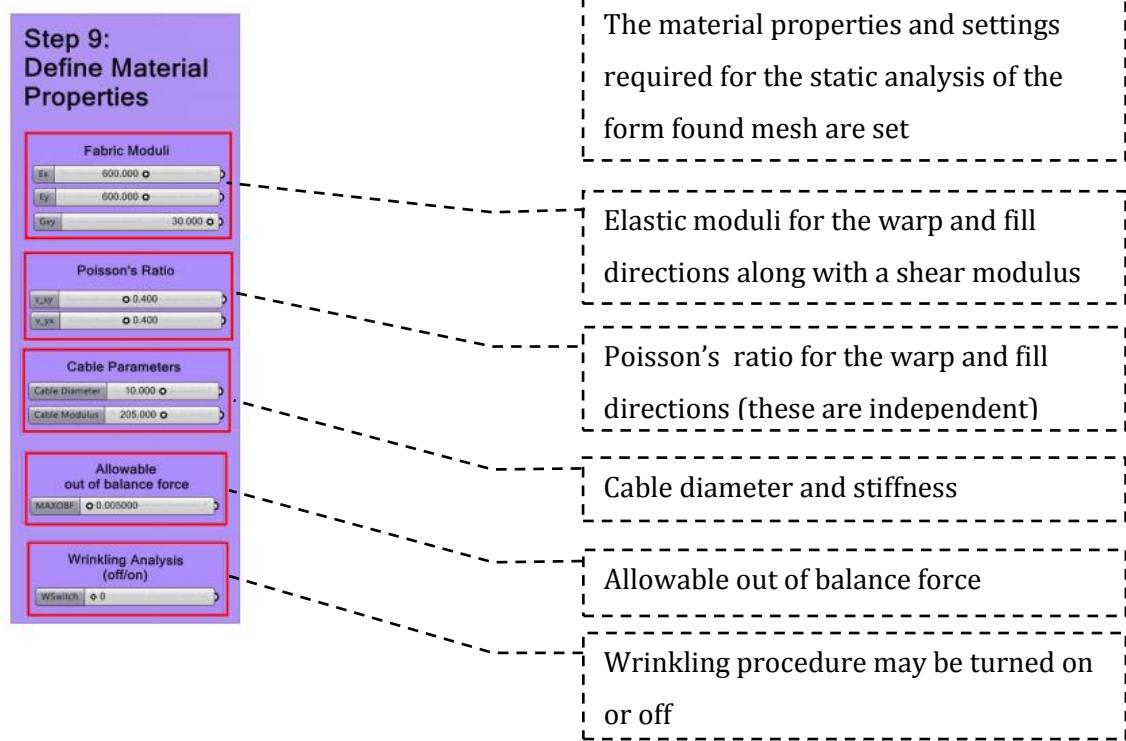
```
C:\Users\nsmithin\Documents\Phd\Conic Example 2\lst_geodesic_ff\History.net.exe
=====
=====LOADING=====
=====NOFLVIS=====
ITERATION: 1
Max out of balance force: 1.139317727658099
Max out of balance force: 0.231097859885734
Max out of balance force: 0.45244684139205BE-002
Max out of balance force: 6.45244684139205BE-002
Max out of balance force: 3.77013545271997E-002
Max out of balance force: 1.23464249847727BE-002
Max out of balance force: 1.48464249847727BE-002
Max out of balance force: 2.142331955657587E-002
Max out of balance force: 1.62331955657587E-002
Max out of balance force: 2.247515587850857E-003
Stress convergence reached. End of wrinkling analysis
=====
=====RESULTS=====
Warp Stress (GL)
Maximum Warp: 4.000000000000000
Element: .000000000000000
Gauss Point: 1.000000000000000
```

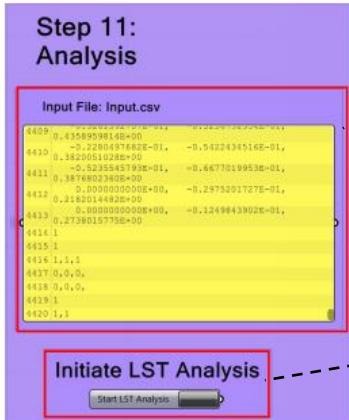
Displacement information may also be displayed on the form found in the Rhino viewport:



The form found mesh is read back into Grasshopper and may be baked to the Rhino viewport







Analysis input is automatically generated and is saved to a CSV file that in turn is read by the Fortran FE code

Fortran FE code if called from Grasshopper, analysis progress is displayed in a console:

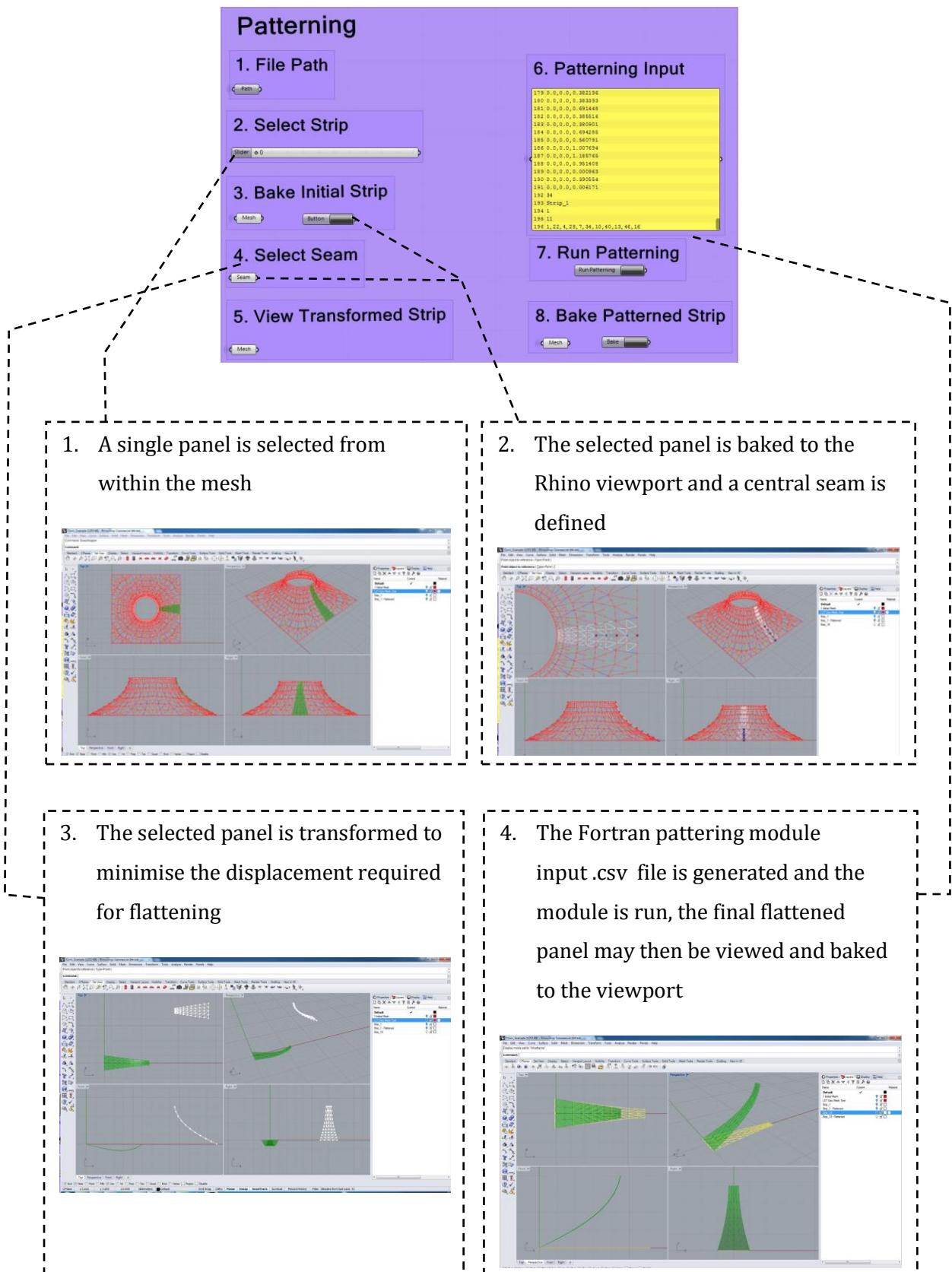
```

INCREMENTAL ANALYSIS =           1
NUMBER OF RATIO'S =           1
NODE   X RATIO  Y RATIO  Z RATIO
1.0000000000000000
1.0000000000000000
VOL STRESS   FILL STRESS   SHEAR
0.0000000000000000 0.0000000000000000E+000 0.0000000000000000E+000
WARP STRESS   FILL STRESS   SHEAR
0.0000000000000000 0.0000000000000000E+000 0.0000000000000000E+000
NO. SELECTED NODES =           1
SELECTED NODES  SELECTED ELEMENTS
1               1
*****LOADING*****
*****ANALYSIS*****
ITERATION
Max out of balance force:  0.545991725614823
Max out of balance force:  1.18803116344634
Max out of balance force:  0.816764169225399
Max out of balance force:  0.000000000000000
Max out of balance force:  0.377847224113953

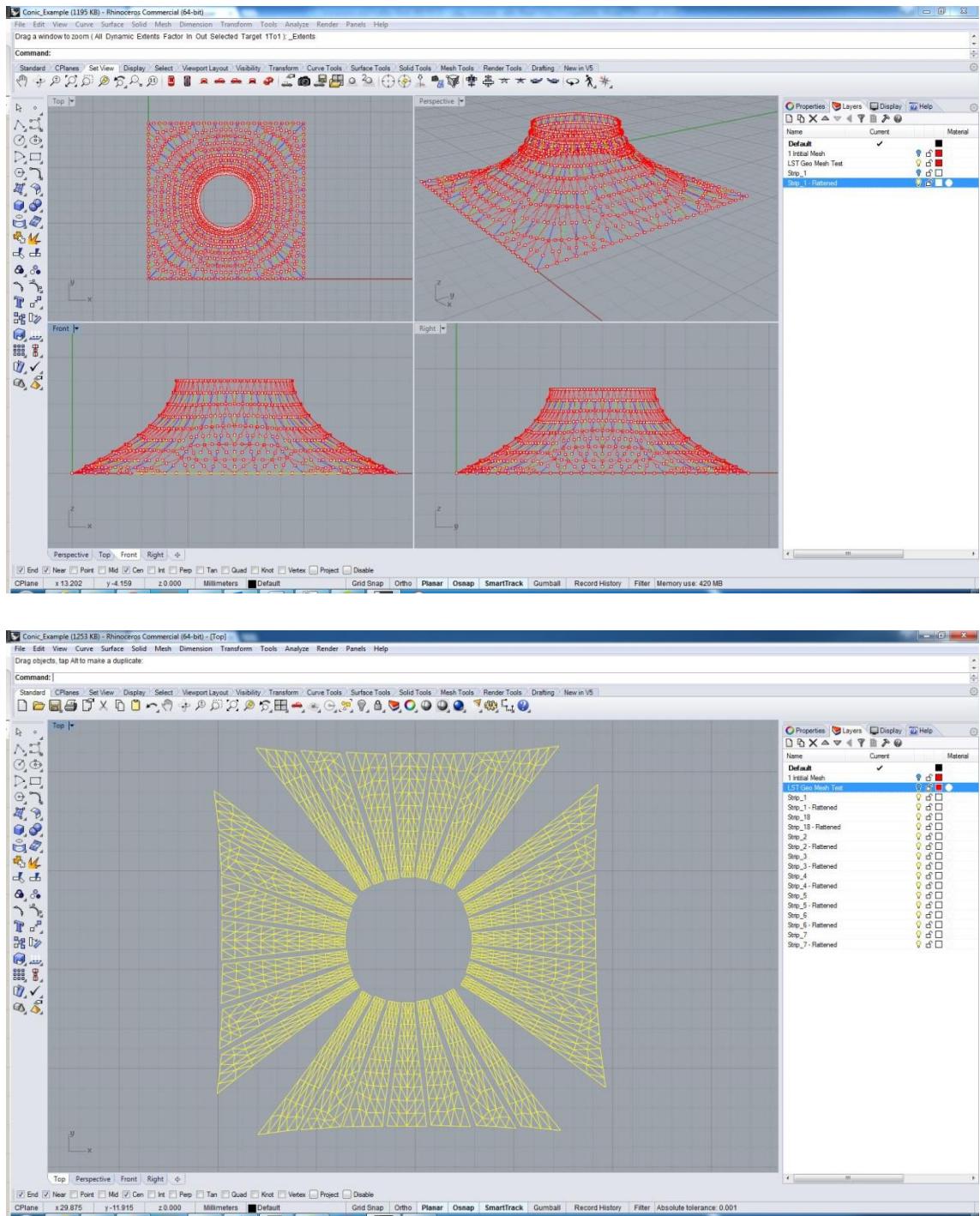
```

Analysis output including GiD results files are saved with the loading name as a prefix

Patterning of the form found mesh is also performed within the same Grasshopper script.



The final patterned panel meshes, baked to separate layers, and form found mesh may be manipulated in Rhino 3D or exported to other CAD packages in the form of a .dwg file

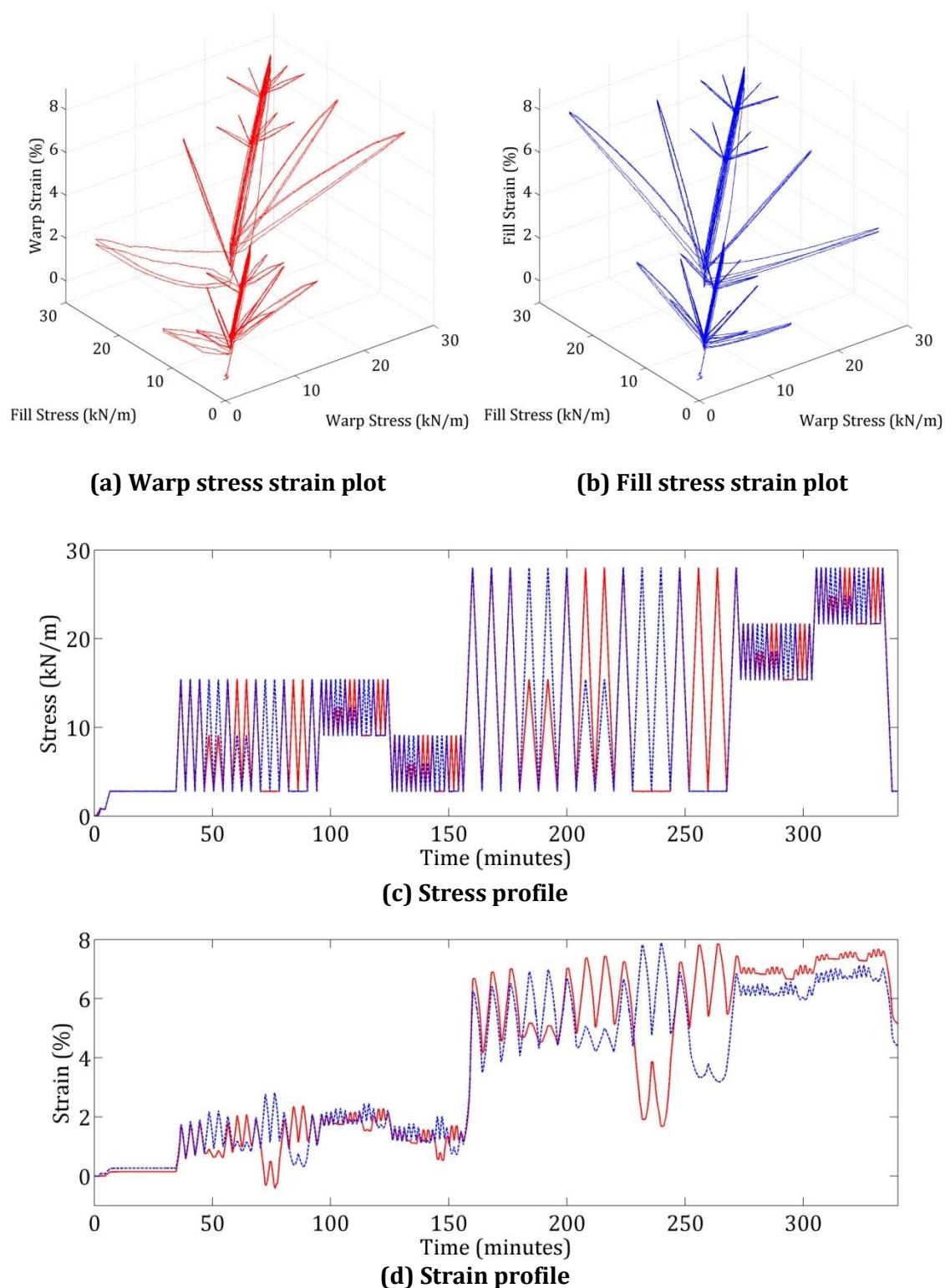


## Appendix B. Full Biaxial History Network Results

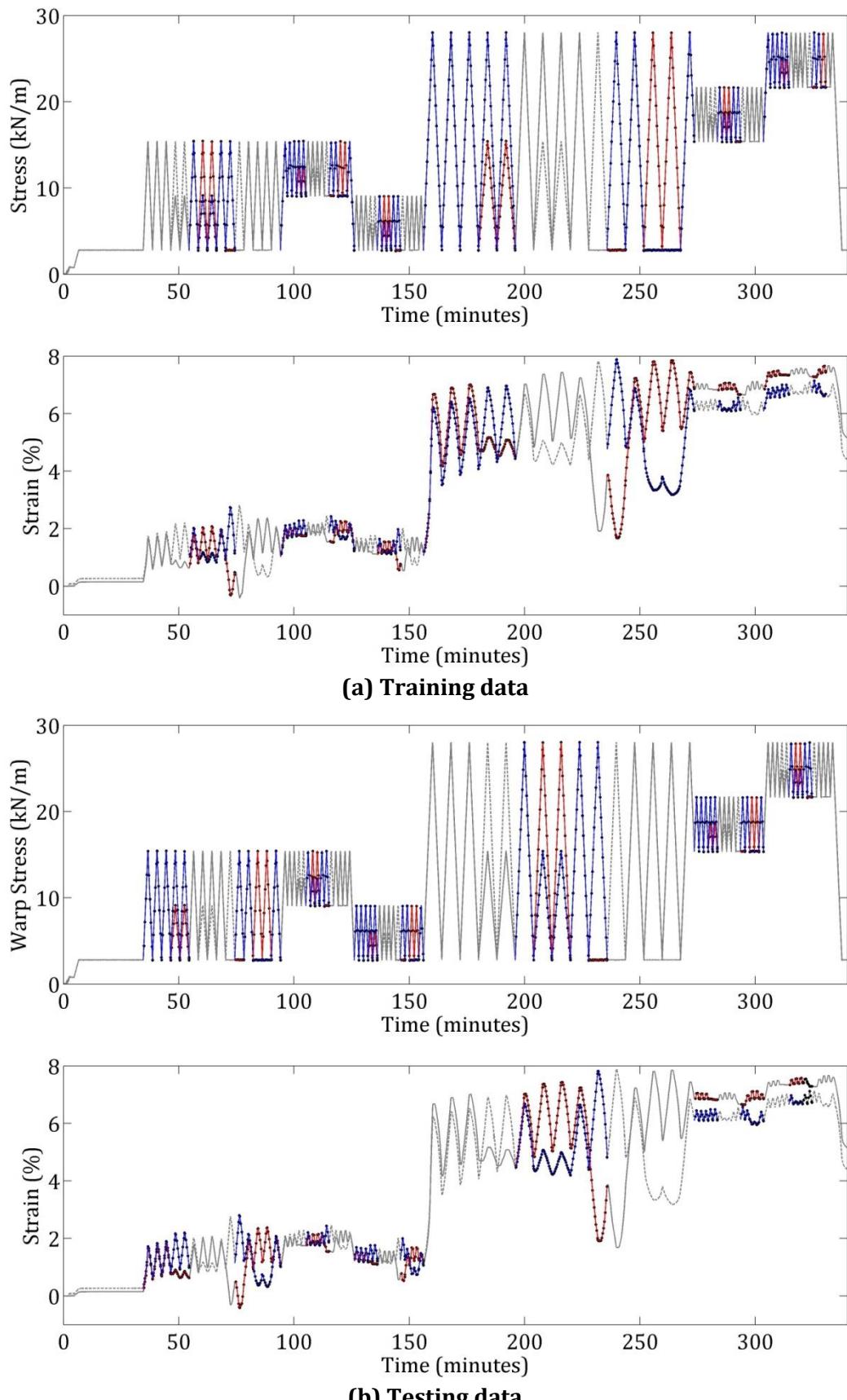
|   |     |
|---|-----|
| Figure B.1: PVC Biaxial 1:1 stress profile and results for history network training and validation.....                     | 359 |
| Figure B.2: PVC Biaxial 1:1 stress and resulting strain profiles divided into training and testing data respectively .....  | 360 |
| Figure B.3: PVC Biaxial 1:2 stress profile and results for history network training and validation.....                     | 361 |
| Figure B.4: PVC Biaxial 1:2 stress and resulting strain profiles divided into training and testing data respectively .....  | 362 |
| Figure B.5: PVC Biaxial 2:1 stress profile and results for history network training and validation.....                     | 363 |
| Figure B.6: PVC Biaxial 2:1 stress and resulting strain profiles divided into training and testing data respectively .....  | 364 |
| Figure B.7: PVC Biaxial 0:1 stress profile and results for history network training and validation.....                     | 365 |
| Figure B.8: PVC Biaxial 0:1 stress and resulting strain profiles divided into training and testing data respectively .....  | 366 |
| Figure B.9: PVC Biaxial 1:0 stress profile and results for history network training and validation.....                     | 367 |
| Figure B.10: PVC Biaxial 1:0 stress and resulting strain profiles divided into training and testing data respectively ..... | 368 |
| Figure B.11: PVC network tested with 1:1 profile in recurrent mode.....   | 369 |
| Figure B.12: PVC network tested with 1:1 profile in recurrent mode.....   | 370 |
| Figure B.13: PVC network tested with 0:1 profile in recurrent mode.....   | 371 |
| Figure B.14: PVC network tested with 0:1 profile in recurrent mode.....   | 372 |
| Figure B.15: PVC network tested with 1:0 profile in recurrent mode.....   | 373 |
| Figure B.16: PVC network tested with 1:0 profile in recurrent mode.....   | 374 |
| Figure B.17: PVC network tested with 1:2 profile in recurrent mode.....   | 375 |
| Figure B.18: PVC network tested with 1:2 profile in recurrent mode.....   | 376 |
| Figure B.19: PVC network tested with 2:1 profile in recurrent mode.....   | 377 |
| Figure B.20: PVC network tested with 2:1 profile in recurrent mode.....   | 378 |

|  |     |
|--|-----|
| Figure B.21: PTFE Biaxial 1:1 stress profile and results for history network training and validation.....                    | 379 |
| Figure B.22: PTFE Biaxial 1:1 stress and resulting strain profiles divided into training and testing data respectively ..... | 380 |
| Figure B.23: PTFE Biaxial 1:2 stress profile and results for history network training and validation.....                    | 381 |
| Figure B.24: PTFE Biaxial 1:2 stress and resulting strain profiles divided into training and testing data respectively ..... | 382 |
| Figure B.25: PTFE Biaxial 2:1 stress profile and results for history network training and validation.....                    | 383 |
| Figure B.26: PTFE Biaxial 2:1 stress and resulting strain profiles divided into training and testing data respectively ..... | 384 |
| Figure B.27: PTFE Biaxial 0:1 stress profile and results for history network training and validation.....                    | 385 |
| Figure B.28: PTFE Biaxial 0:1 stress and resulting strain profiles divided into training and testing data respectively ..... | 386 |
| Figure B.29: PTFE Biaxial 1:0 stress profile and results for history network training and validation.....                    | 387 |
| Figure B.30: PTFE Biaxial 1:0 stress and resulting strain profiles divided into training and testing data respectively ..... | 388 |
| Figure B.31: PTFE network tested with 1:1 profile in recurrent mode.....   | 389 |
| Figure B.32: PTFE network tested with 1:1 profile in recurrent mode.....   | 390 |
| Figure B.33: PTFE network tested with 0:1 profile in recurrent mode.....   | 391 |
| Figure B.34: PTFE network tested with 0:1 profile in recurrent mode.....   | 392 |
| Figure B.35: PTFE network tested with 1:0 profile in recurrent mode.....   | 393 |
| Figure B.36: PTFE network tested with 1:0 profile in recurrent mode.....   | 394 |
| Figure B.37: PTFE network tested with 1:2 profile in recurrent mode.....   | 395 |
| Figure B.38: PTFE network tested with 1:2 profile in recurrent mode.....   | 396 |
| Figure B.39: PTFE network tested with 2:1 profile in recurrent mode.....   | 397 |
| Figure B.40: PTFE network tested with 2:1 profile in recurrent mode.....   | 398 |





**Figure B.1: PVC Biaxial 1:1 stress profile and results for history network training and validation (— warp, .... fill)**



**Figure B.2: PVC Biaxial 1:1 stress and resulting strain profiles divided into training and testing data respectively (— warp, .... fill, • data point )**

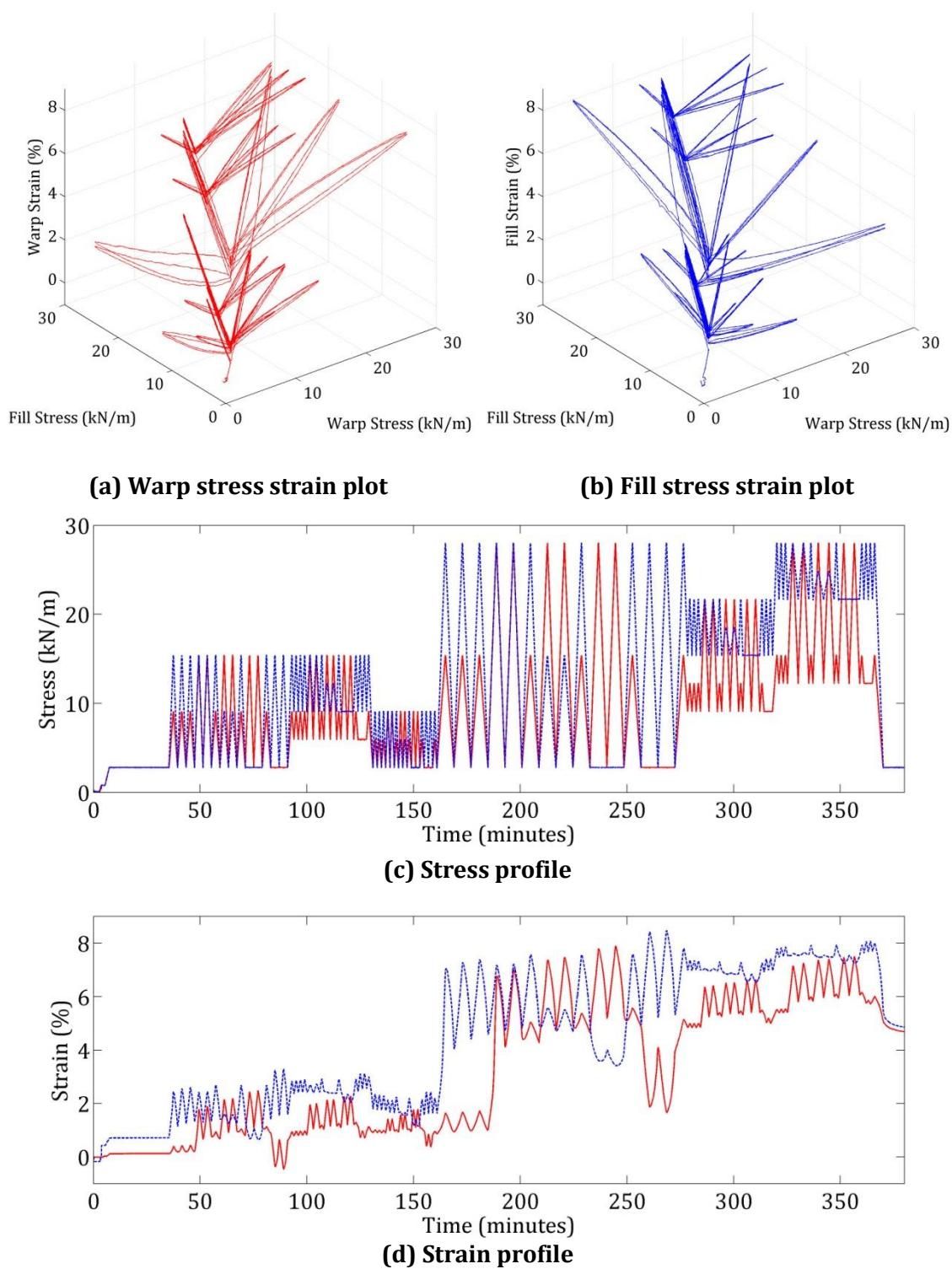
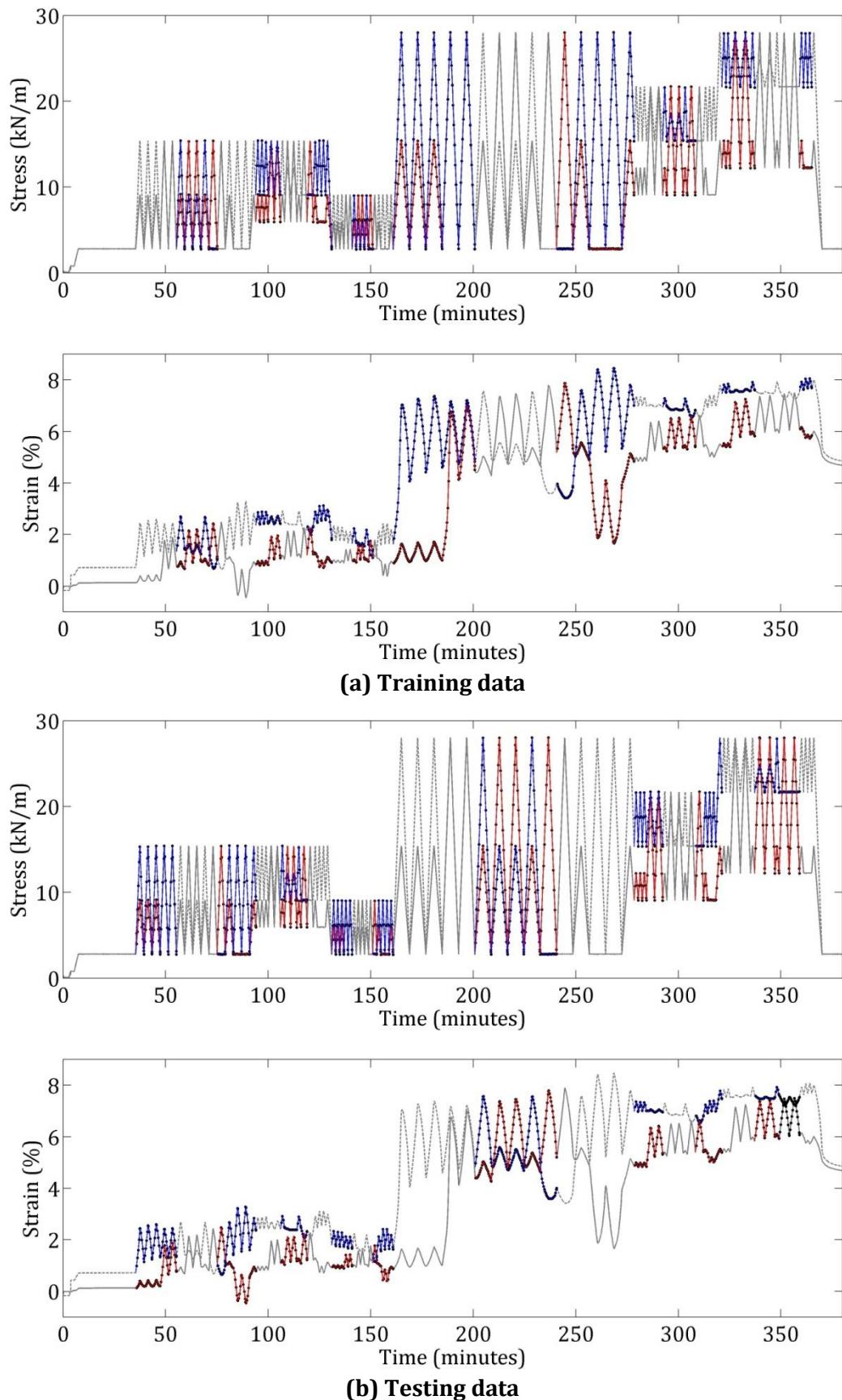
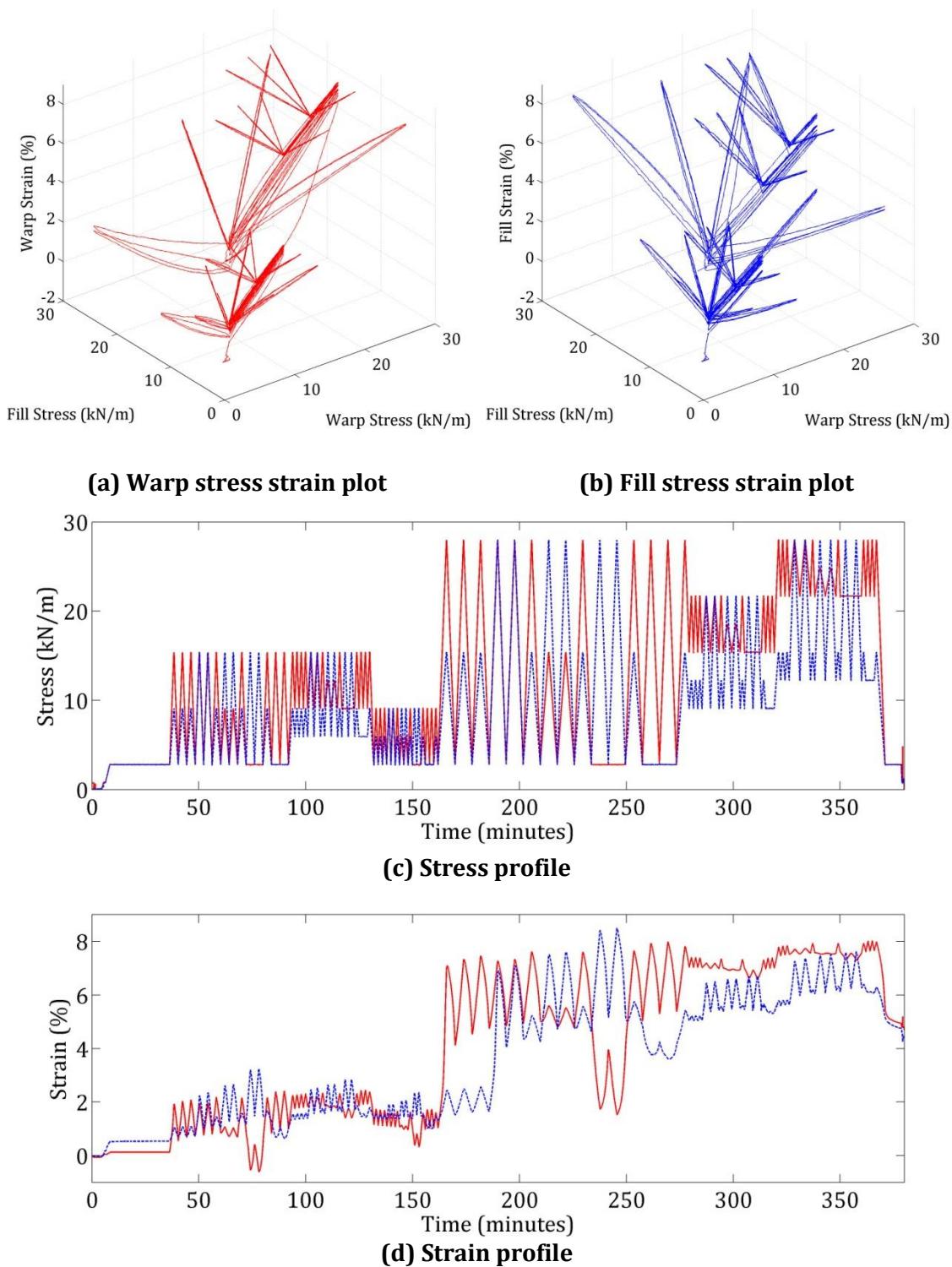


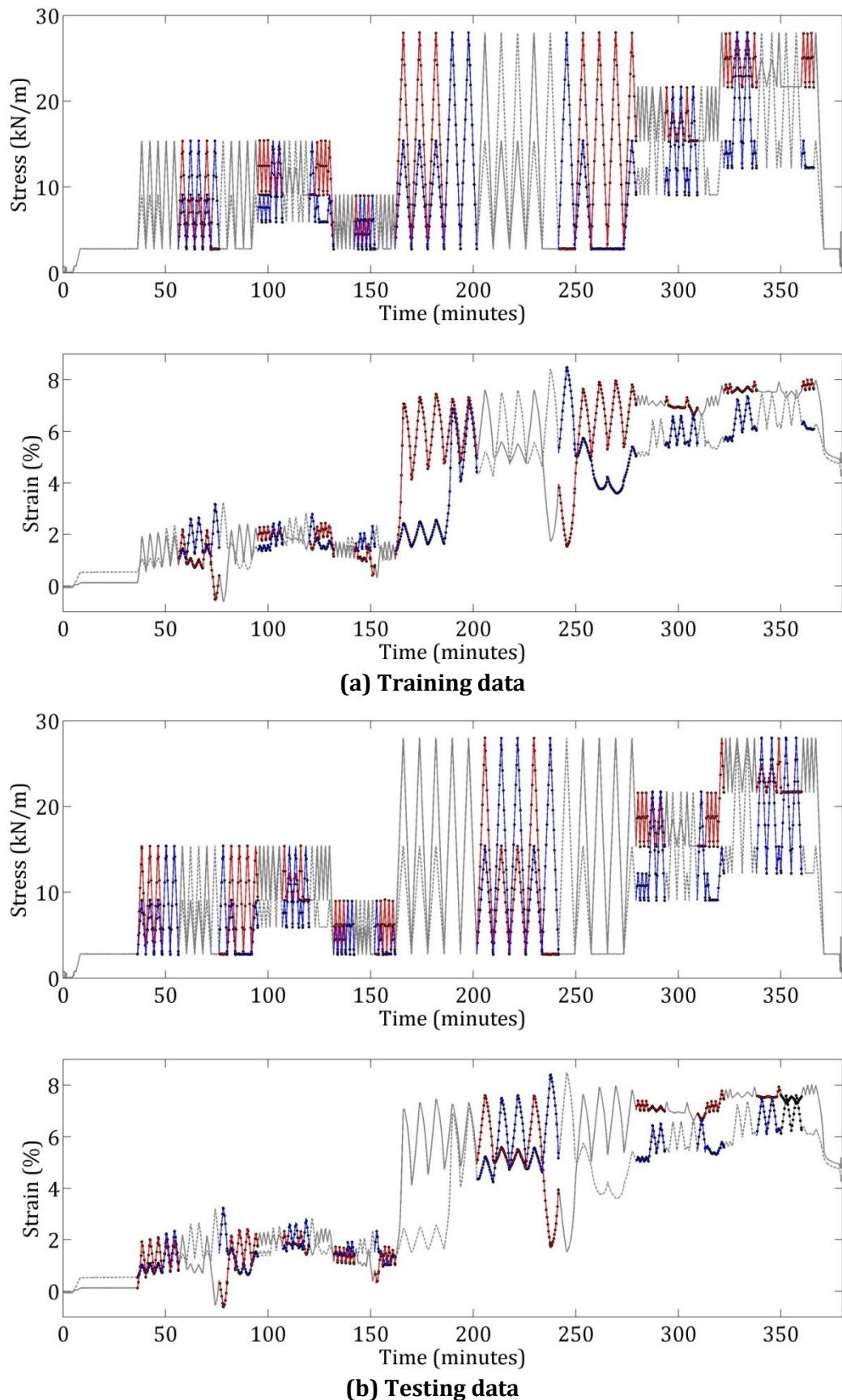
Figure B.3: PVC Biaxial 1:2 stress profile and results for history network training and validation (— warp, .... fill)



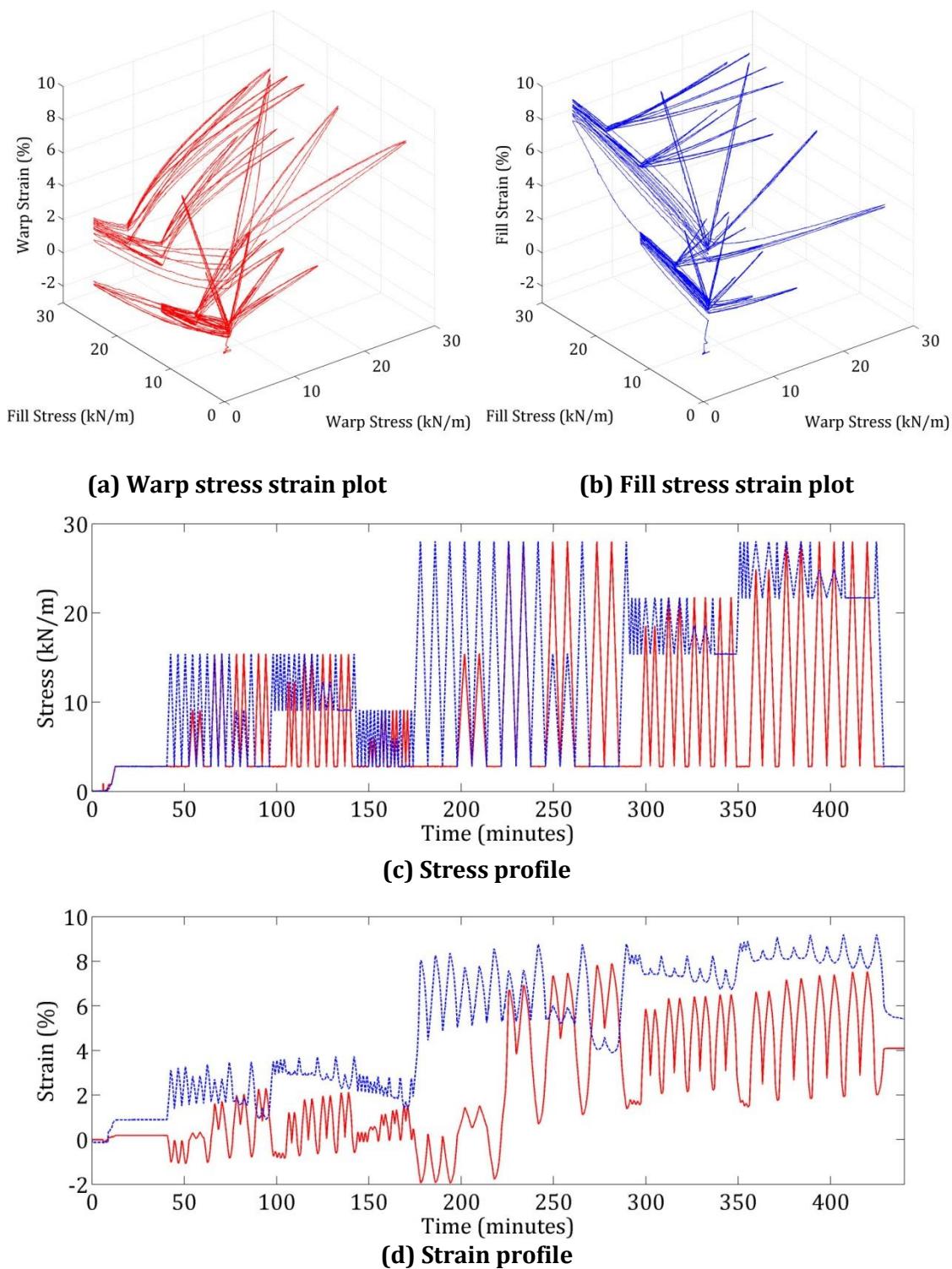
**Figure B.4: PVC Biaxial 1:2 stress and resulting strain profiles divided into training and testing data respectively** (— warp, .... fill, • data point)



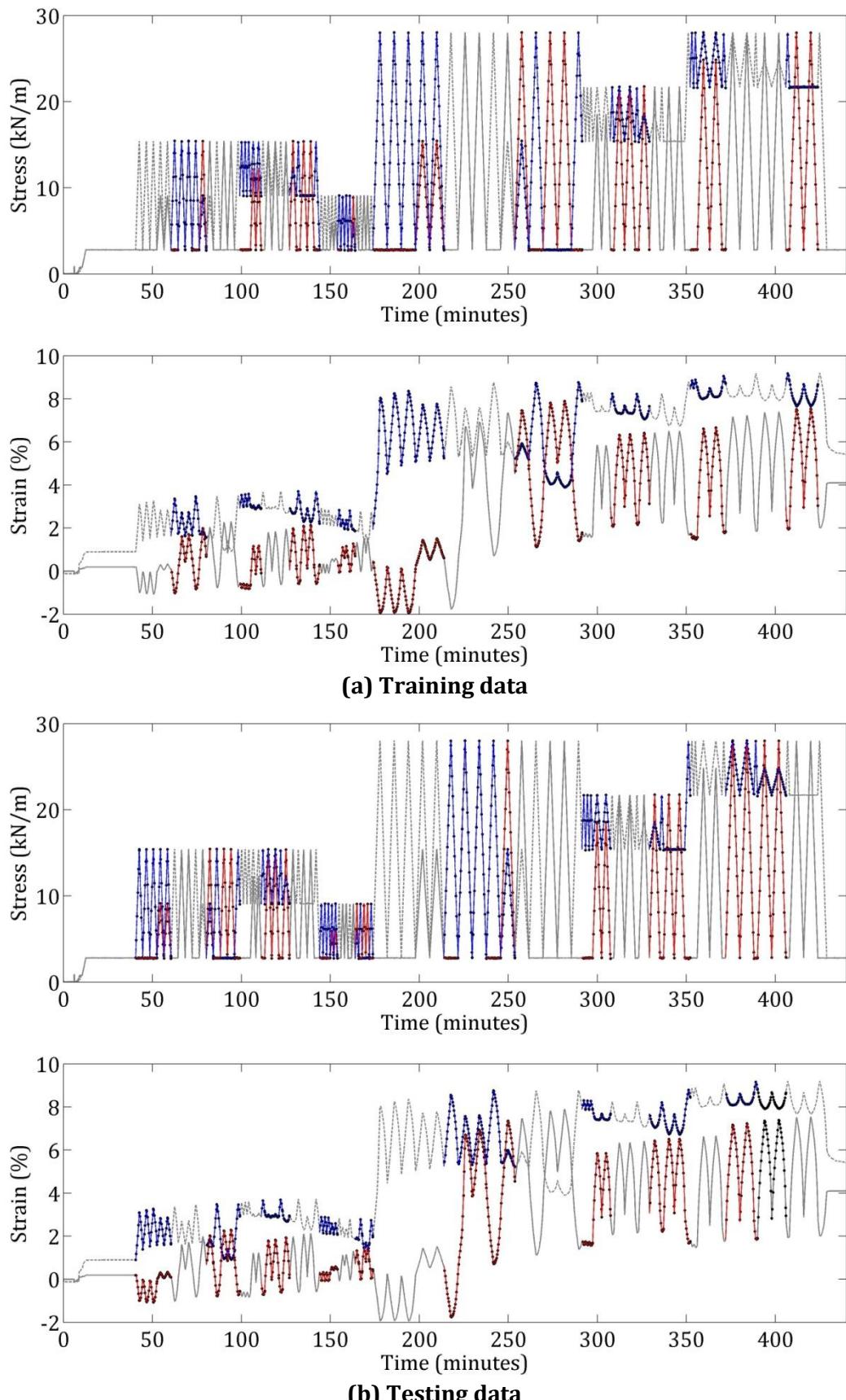
**Figure B.5: PVC Biaxial 2:1 stress profile and results for history network training and validation (— warp, .... fill)**



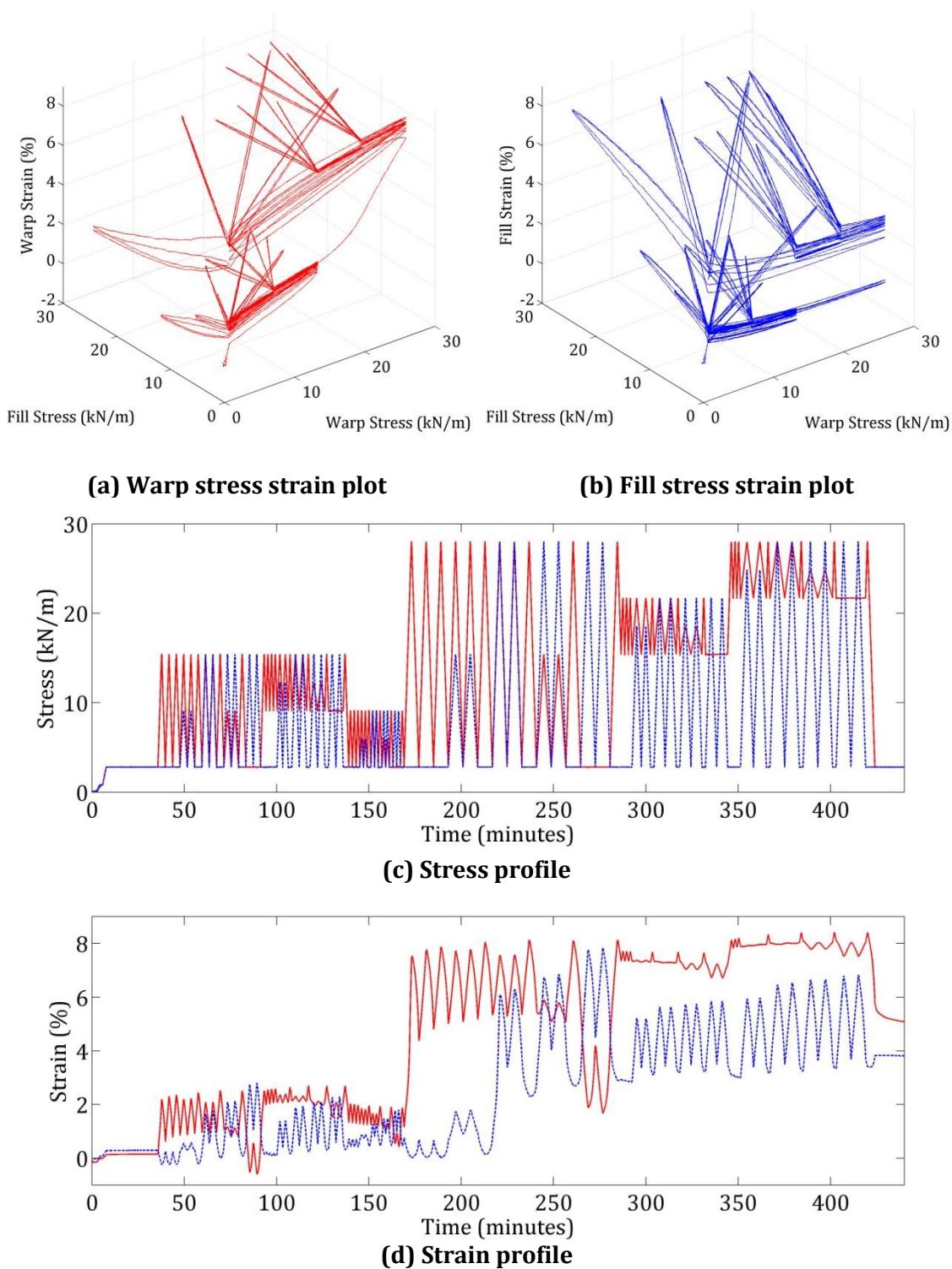
**Figure B.6: PVC Biaxial 2:1 stress and resulting strain profiles divided into training and testing data respectively** (— warp, .... fill, • data point)



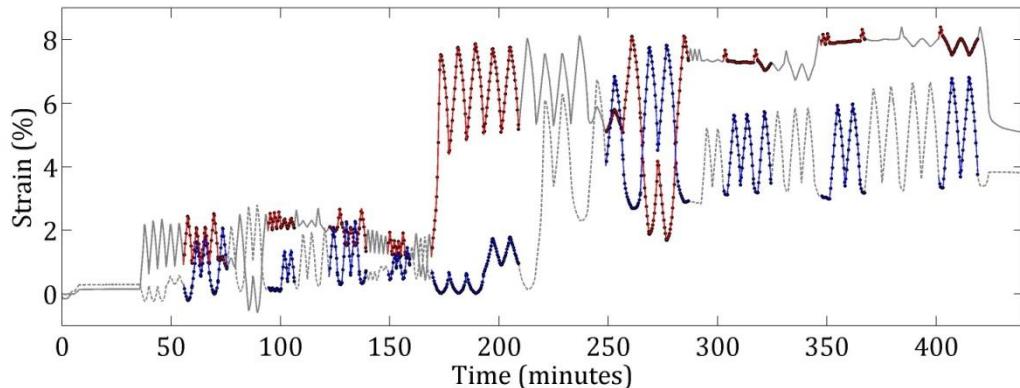
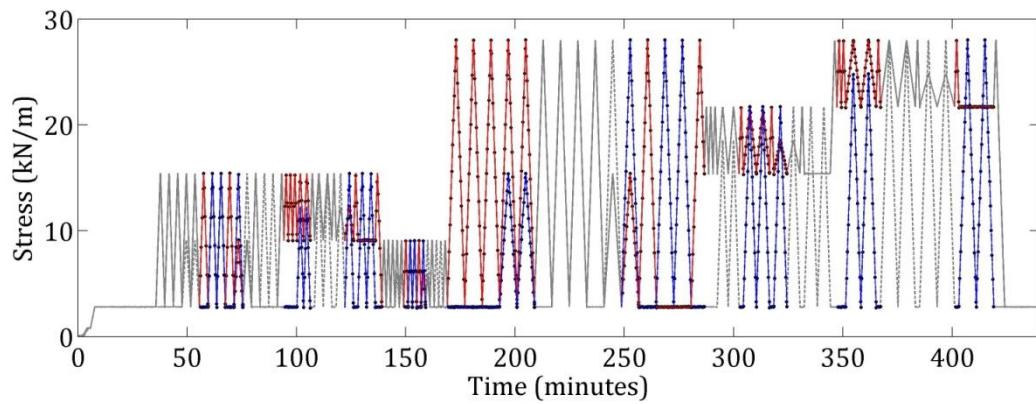
**Figure B.7: PVC Biaxial 0:1 stress profile and results for history network training and validation (— warp, .... fill)**



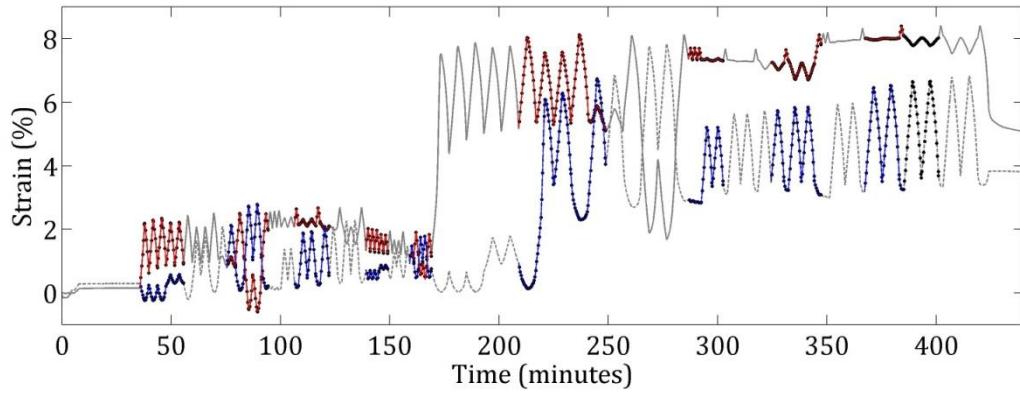
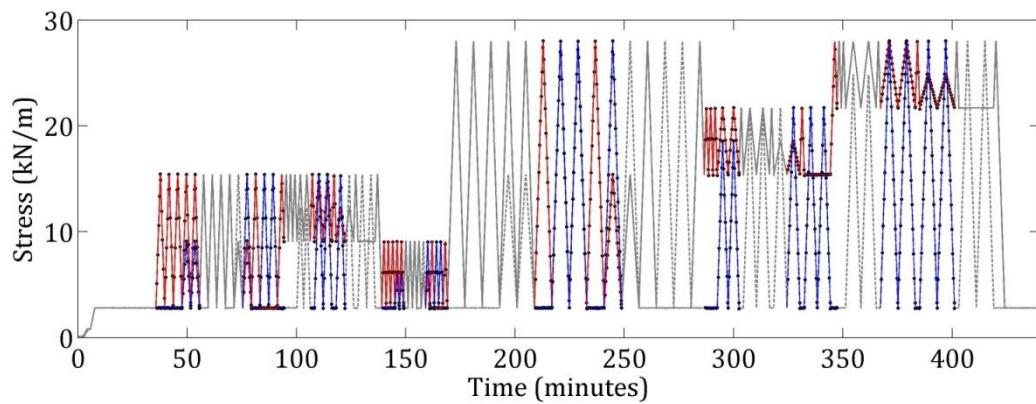
**Figure B.8: PVC Biaxial 0:1 stress and resulting strain profiles divided into training and testing data respectively (— warp, .... fill, • data point )**



**Figure B.9: PVC Biaxial 1:0 stress profile and results for history network training and validation (— warp, .... fill)**

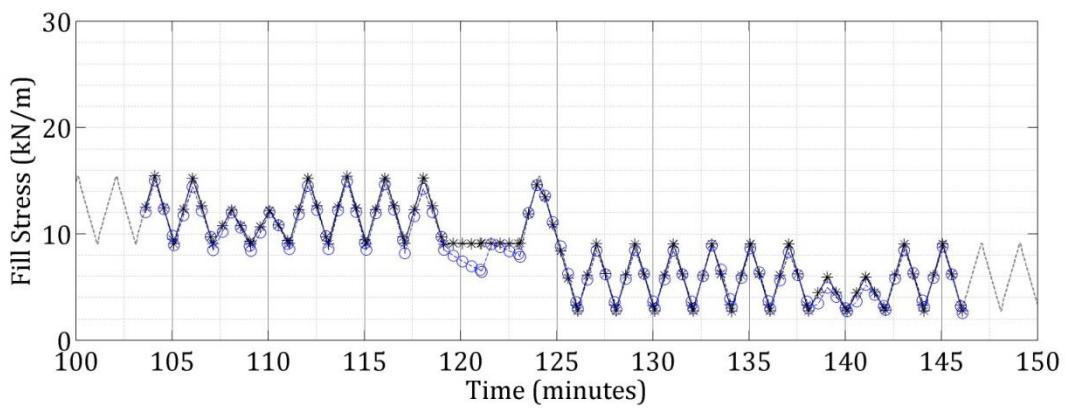
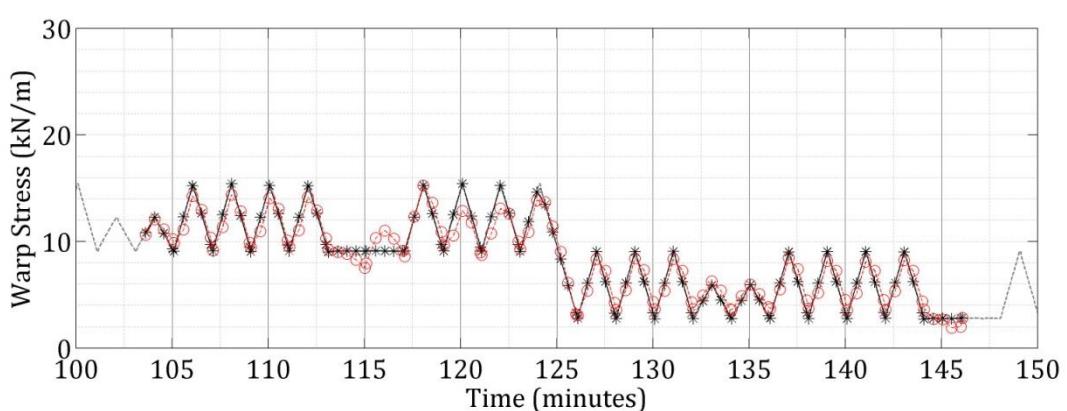
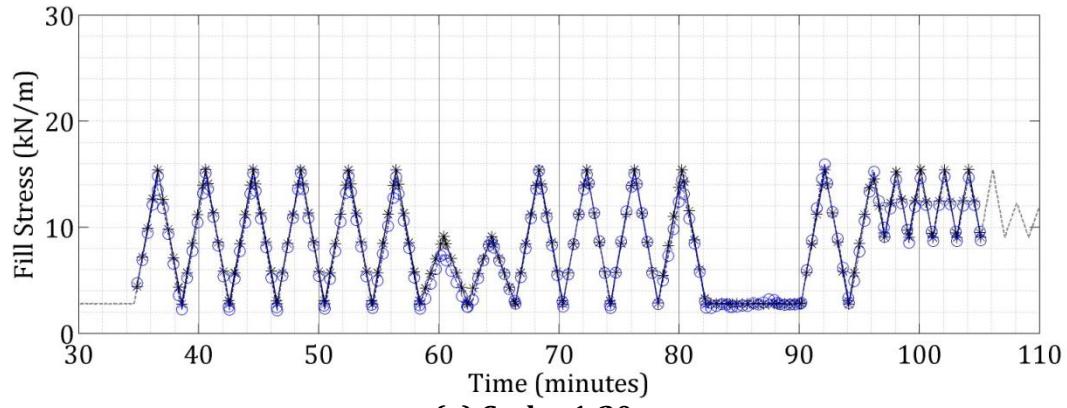
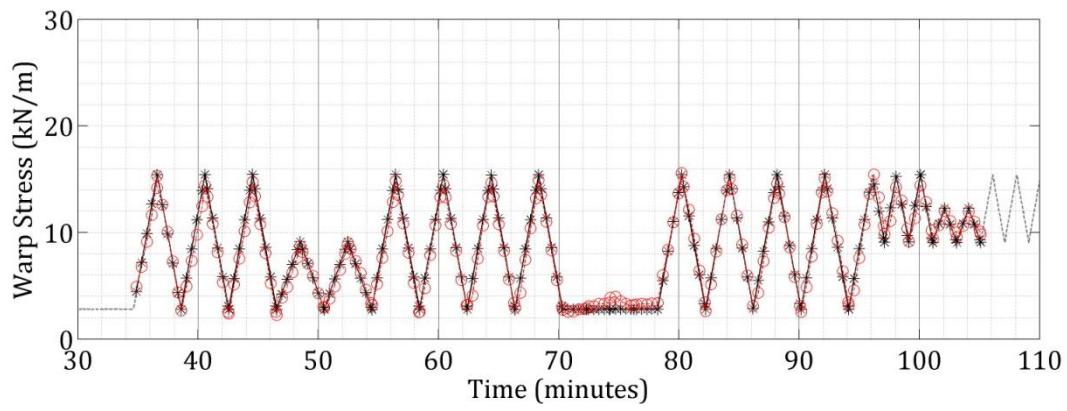


(a) Training data

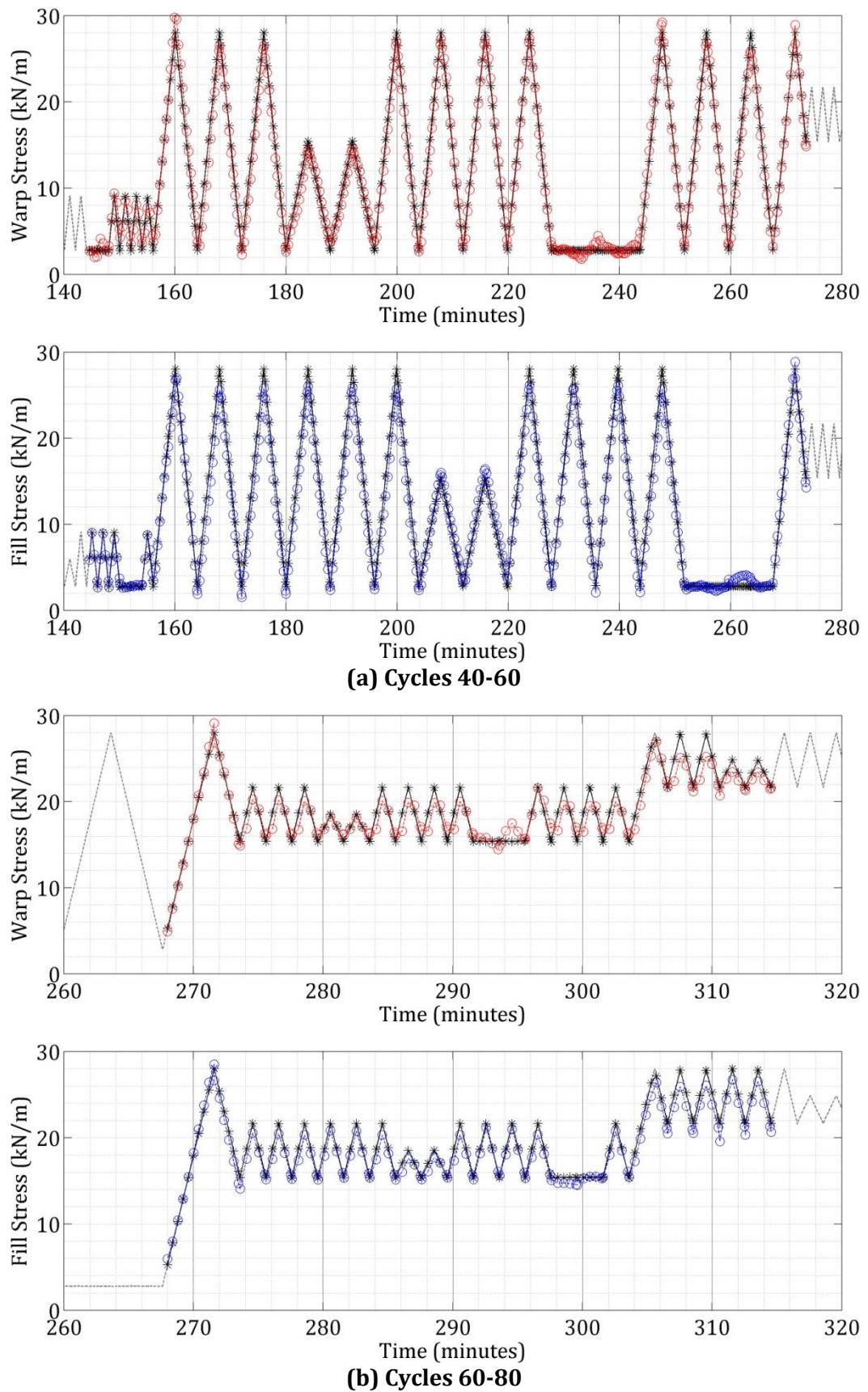


(b) Testing data

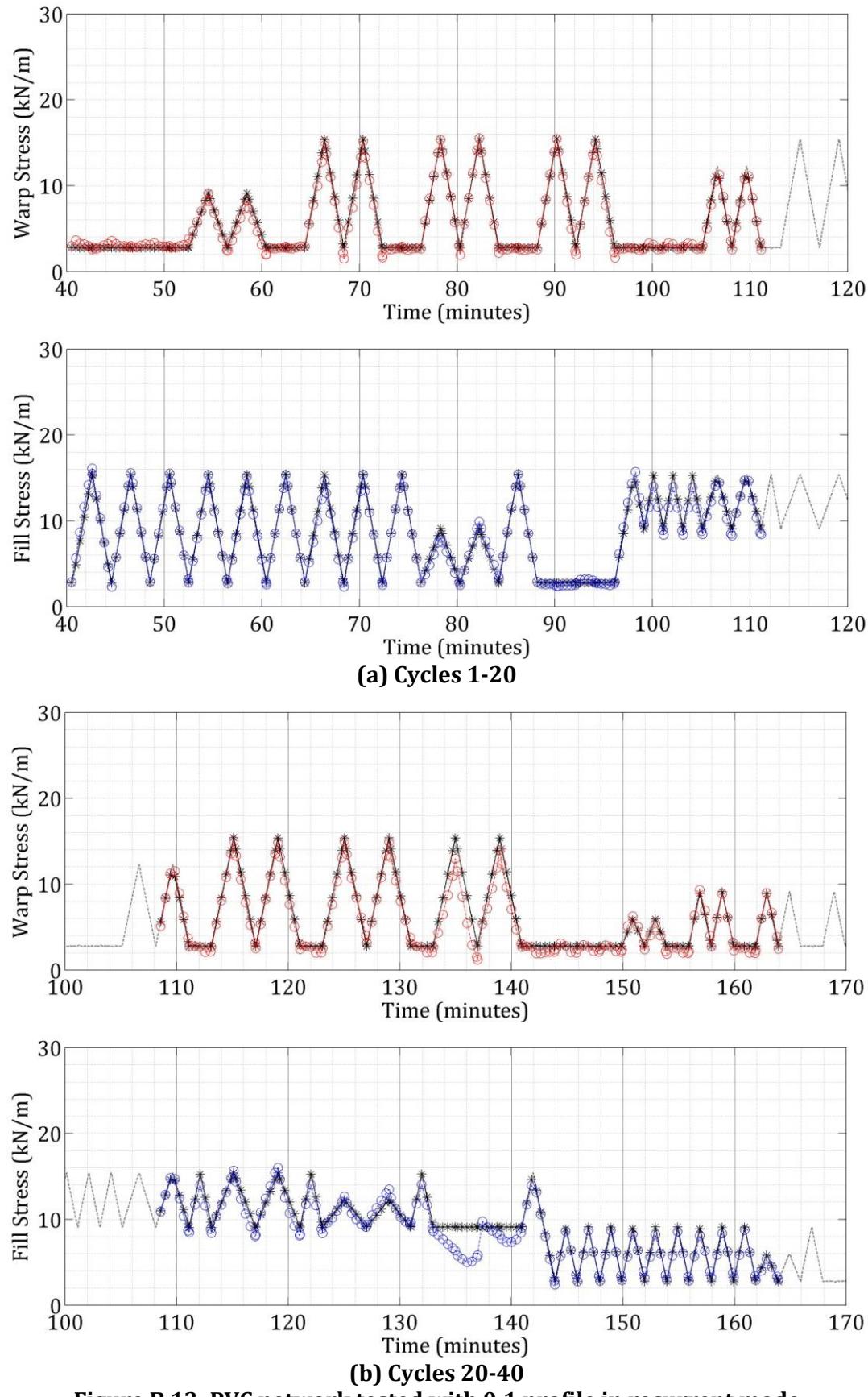
**Figure B.10:** PVC Biaxial 1:0 stress and resulting strain profiles divided into training and testing data respectively (— warp, .... fill, • data point )



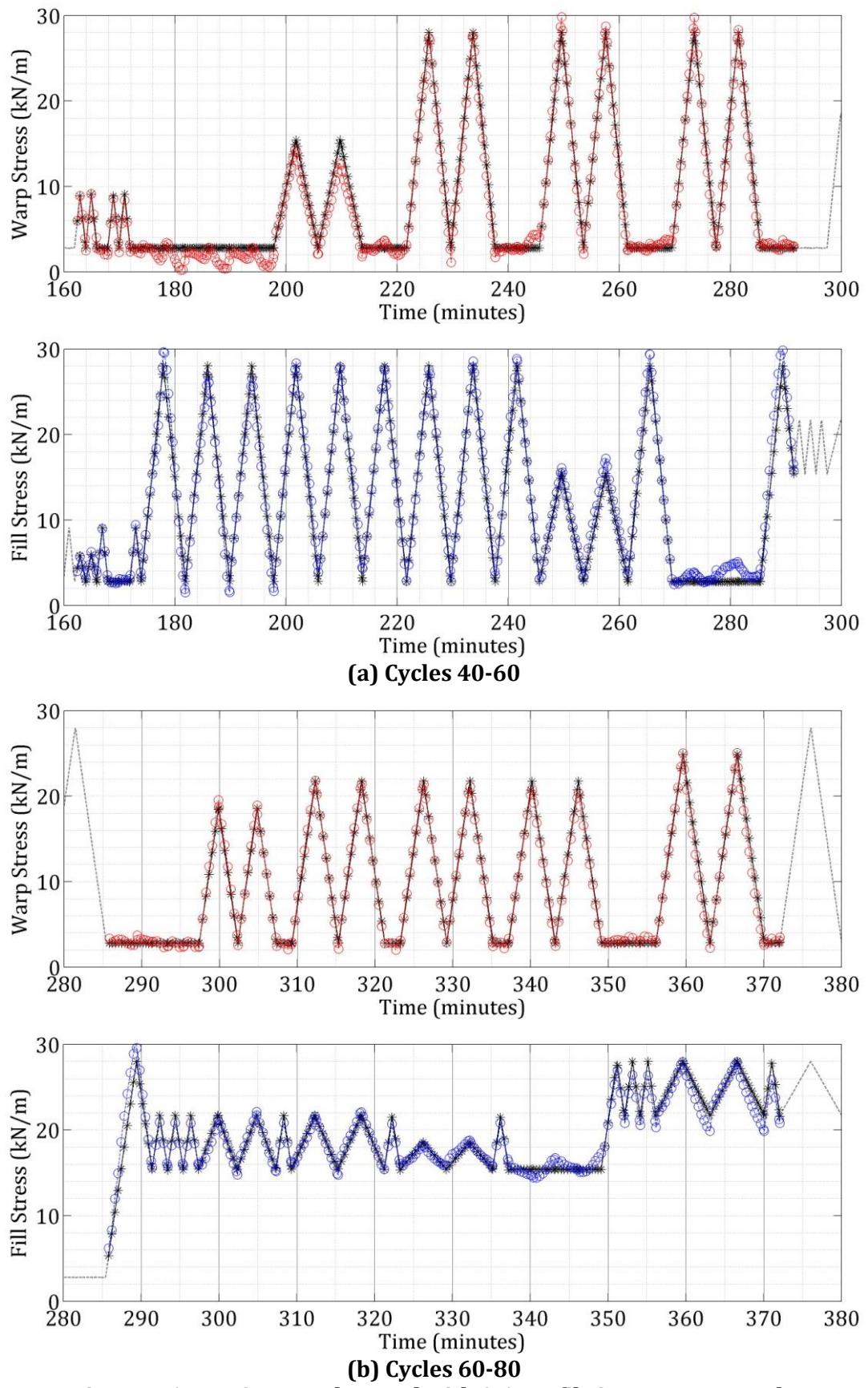
**Figure B.11: PVC network tested with 1:1 profile in recurrent mode**  
**(— warp, — fill, \* target, ° network output)**



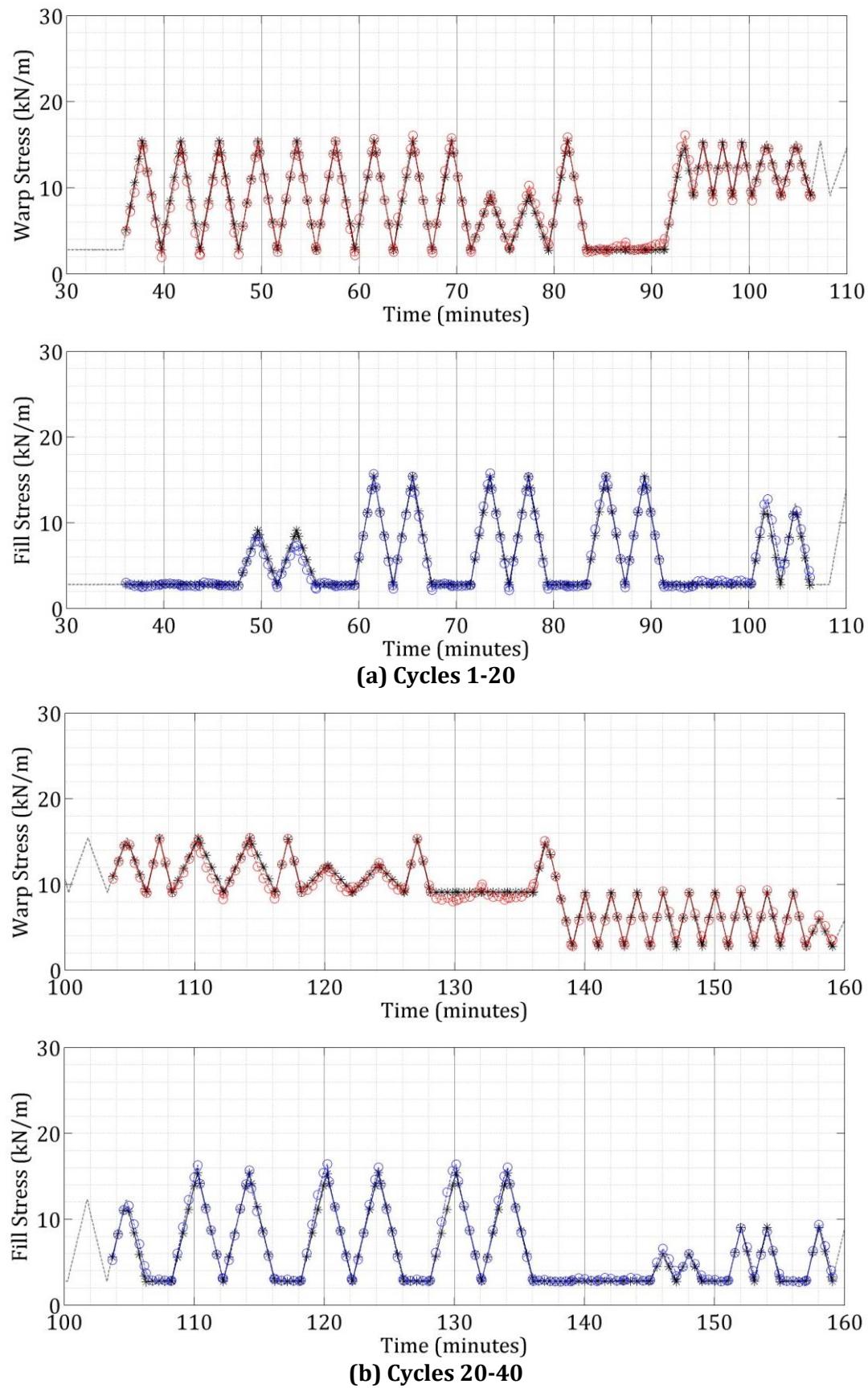
**Figure B.12: PVC network tested with 1:1 profile in recurrent mode**  
**(— warp, — fill, \* target, ○ network output)**



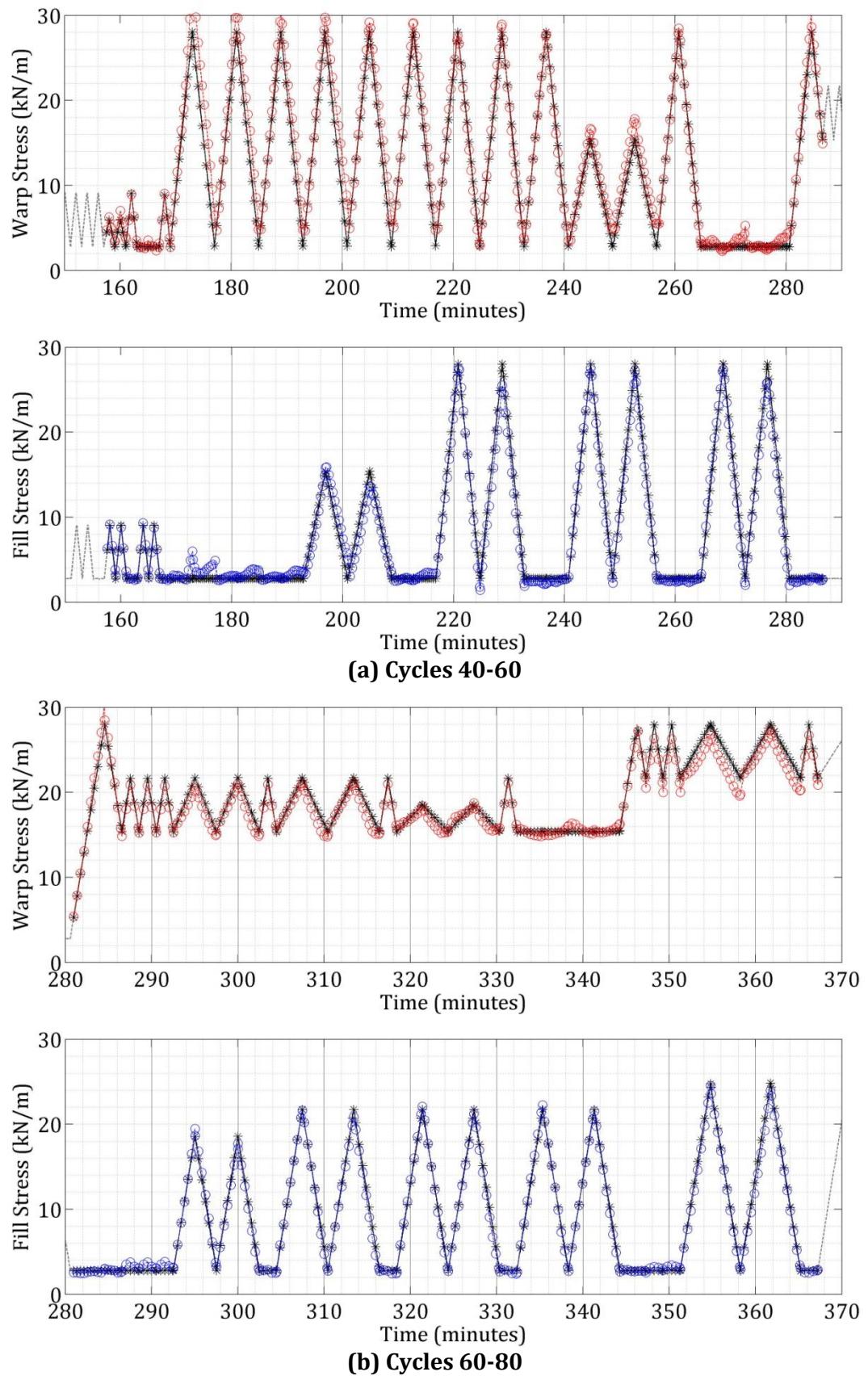
**Figure B.13: PVC network tested with 0:1 profile in recurrent mode**  
**(— warp, — fill, \* target, ° network output)**



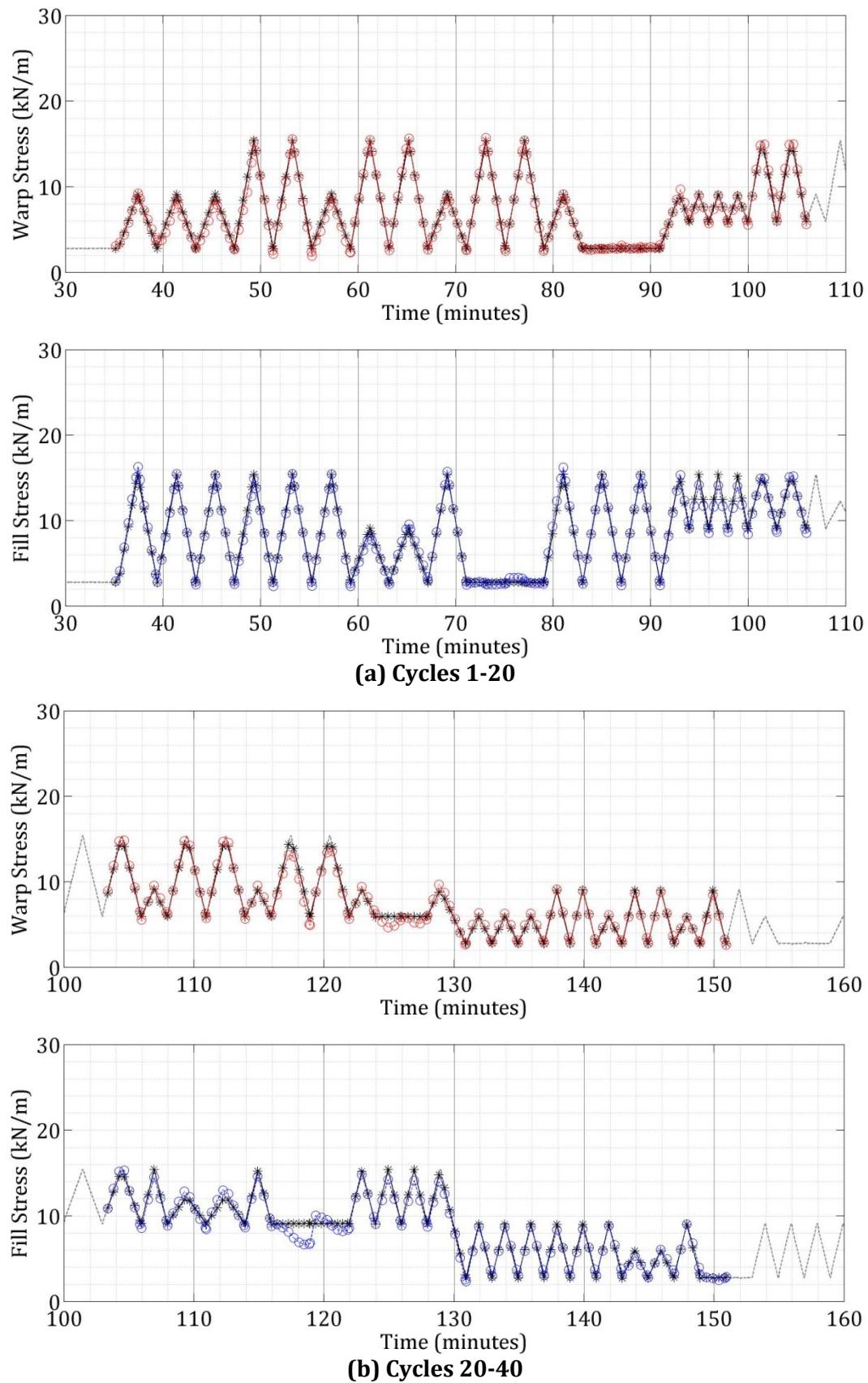
**Figure B.14: PVC network tested with 0:1 profile in recurrent mode**  
 (— warp, — fill, \* target, ○ network output)



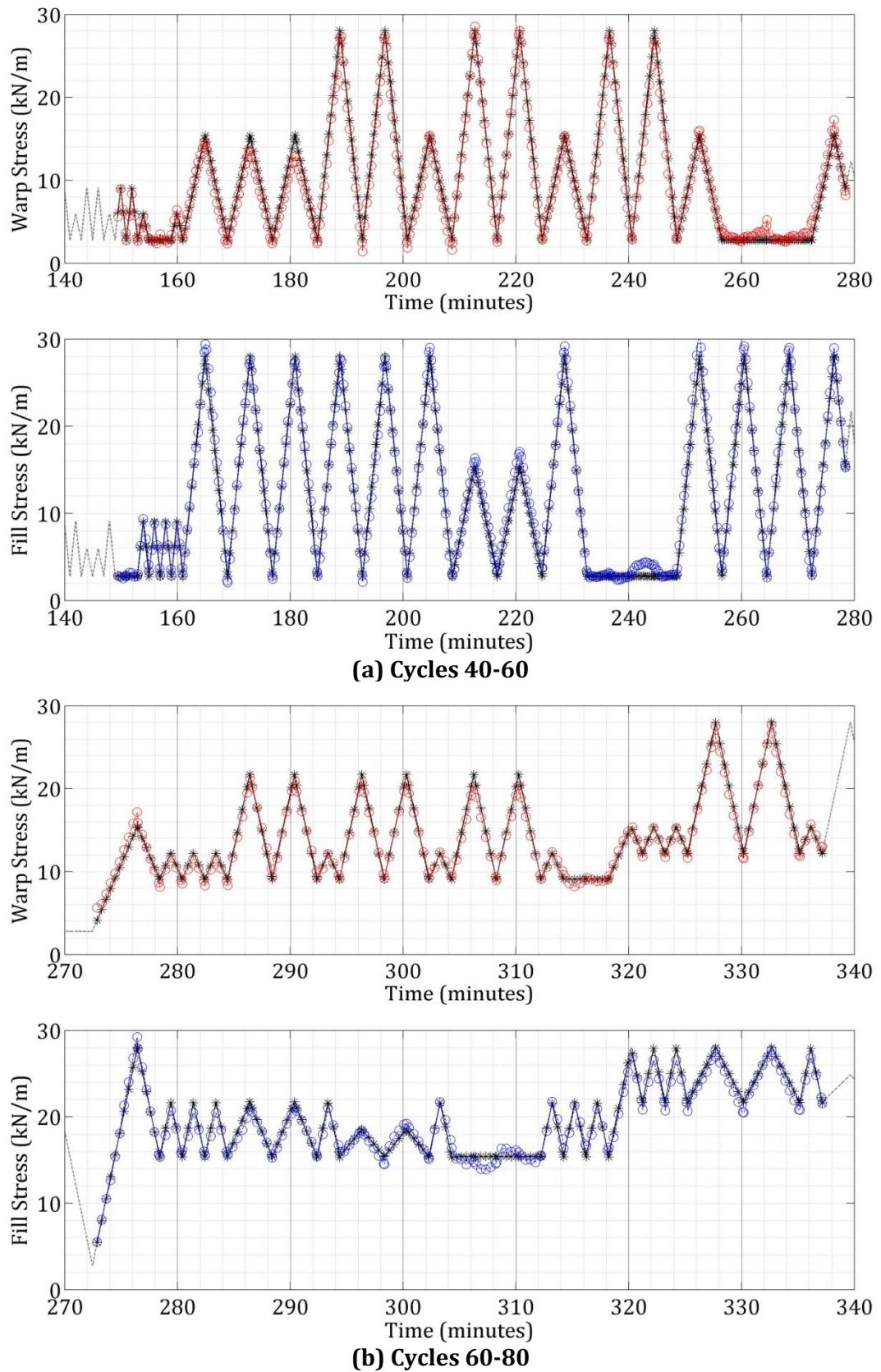
**Figure B.15: PVC network tested with 1:0 profile in recurrent mode**  
**(— warp, — fill, \* target, ◦ network output)**



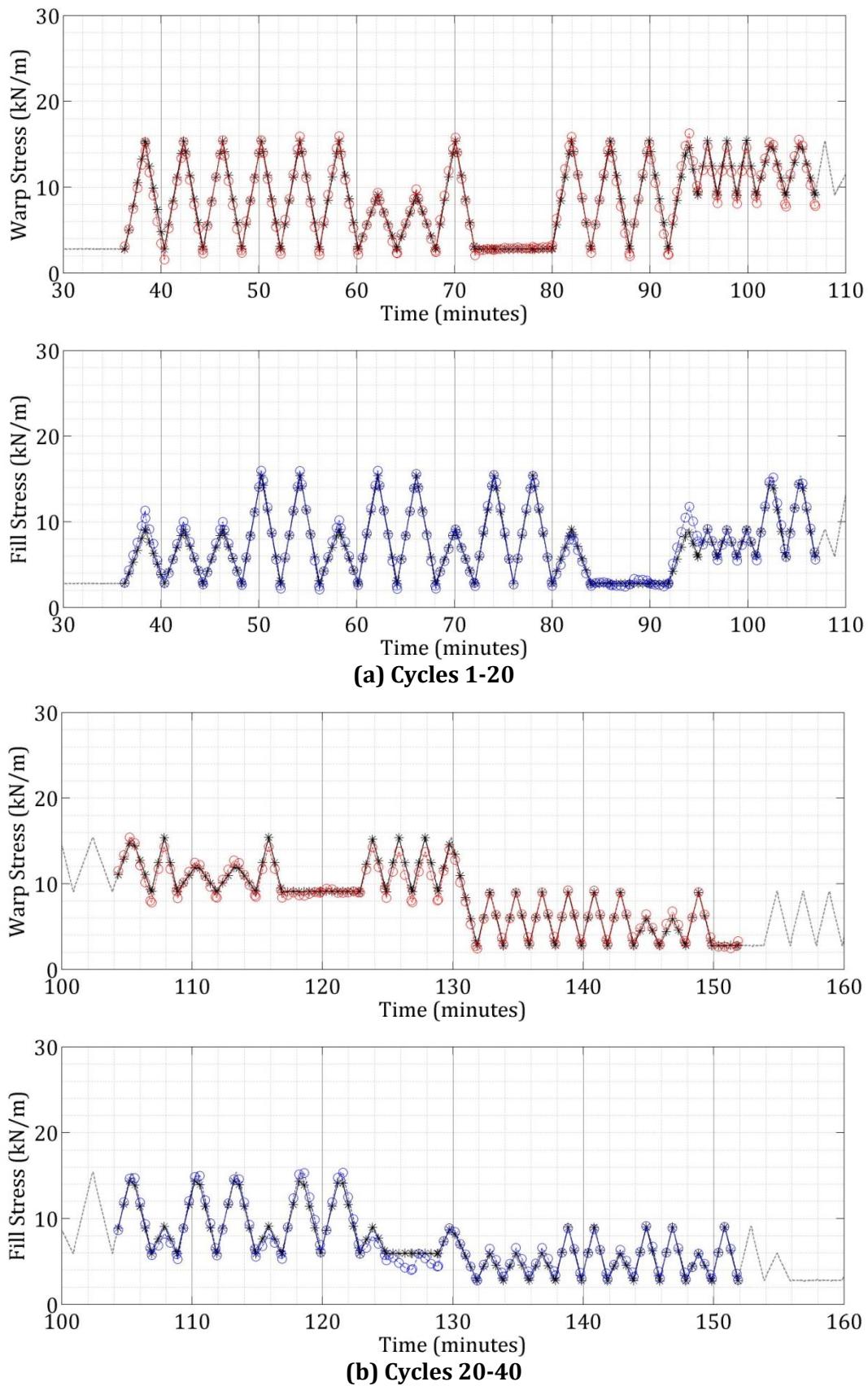
**Figure B.16: PVC network tested with 1:0 profile in recurrent mode**  
 (— warp, — fill, \* target, ° network output)



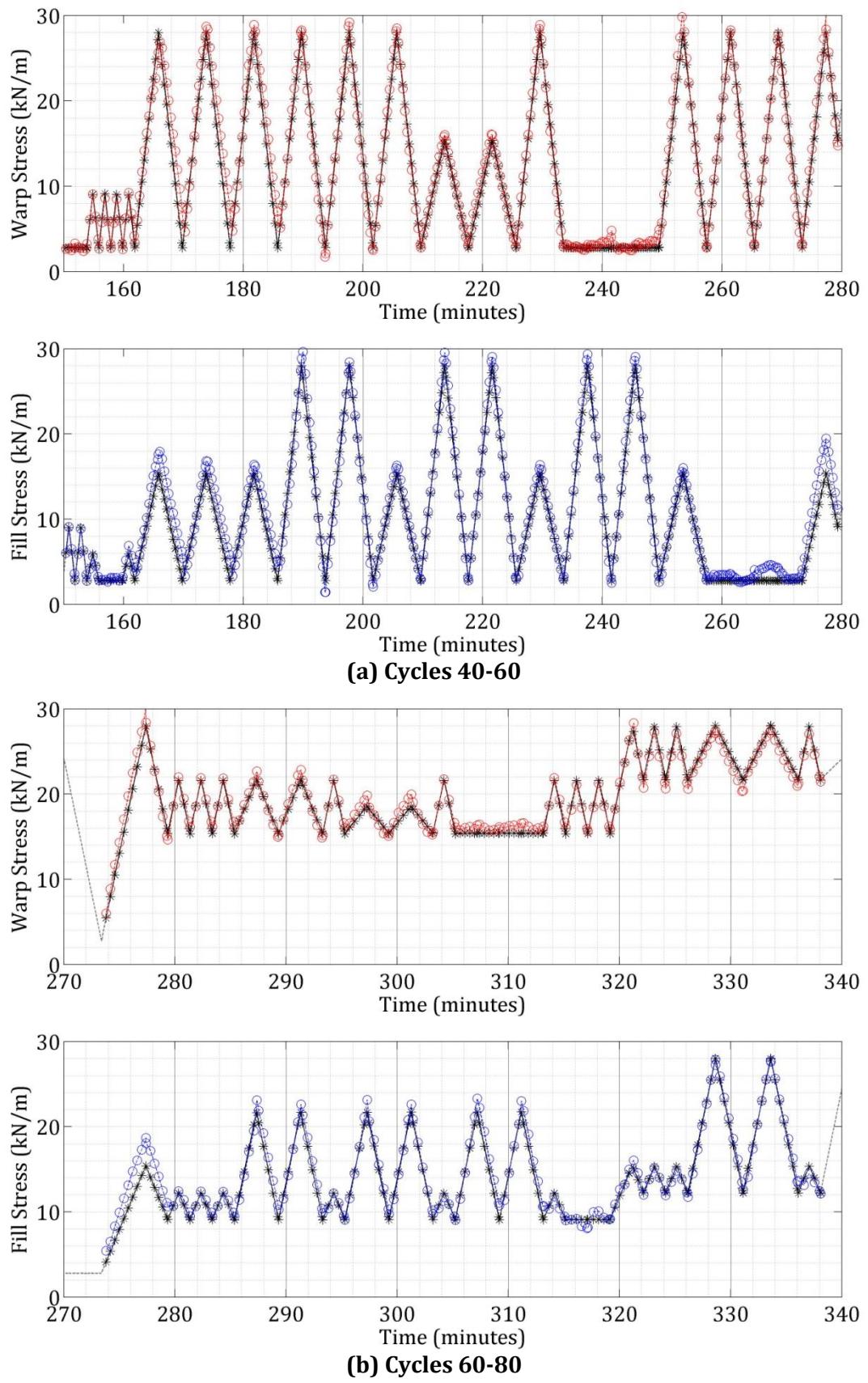
**Figure B.17: PVC network tested with 1:2 profile in recurrent mode**  
**(— warp, — fill, \* target, ° network output)**



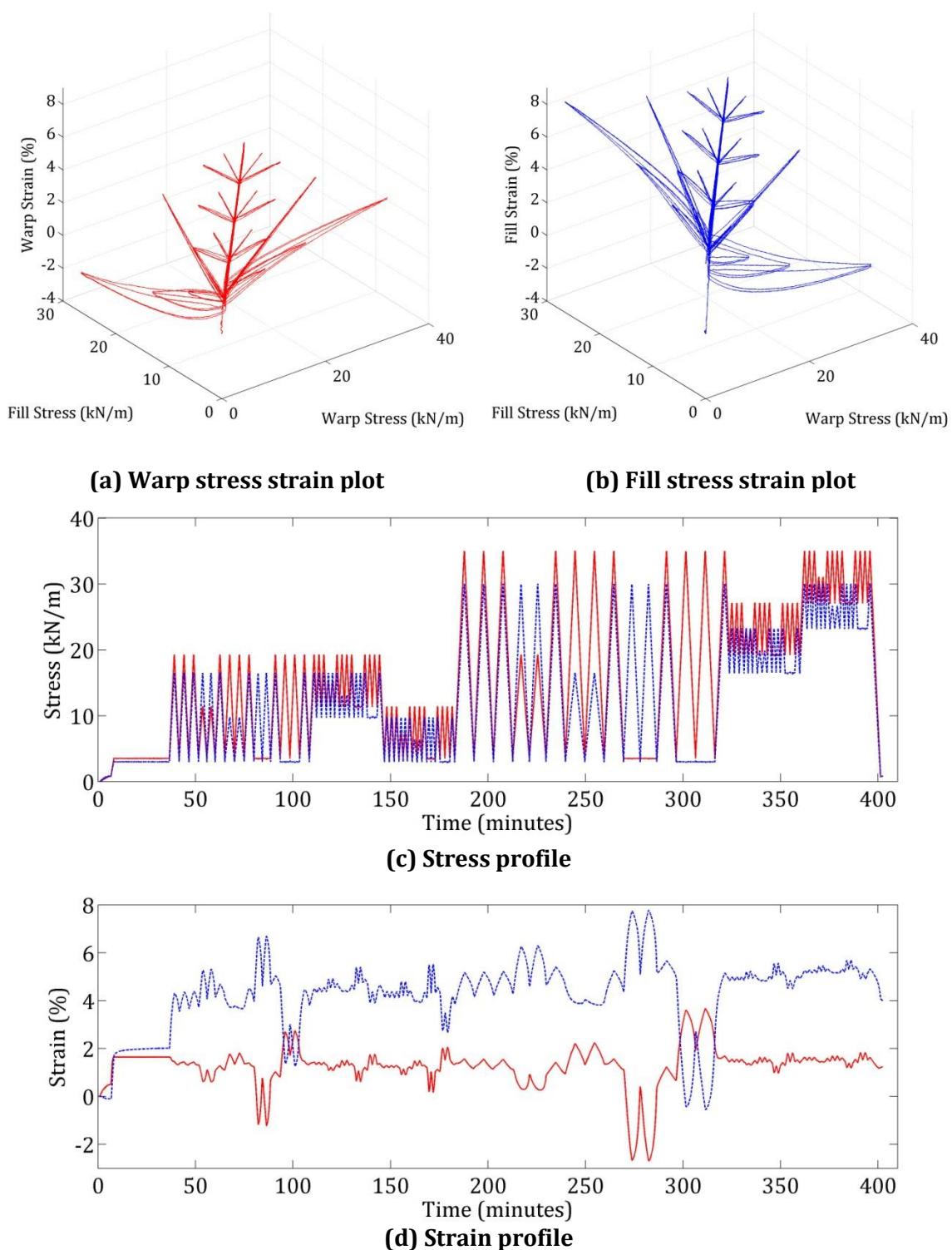
**Figure B.18: PVC network tested with 1:2 profile in recurrent mode**  
**(— warp, — fill, \* target, ° network output)**



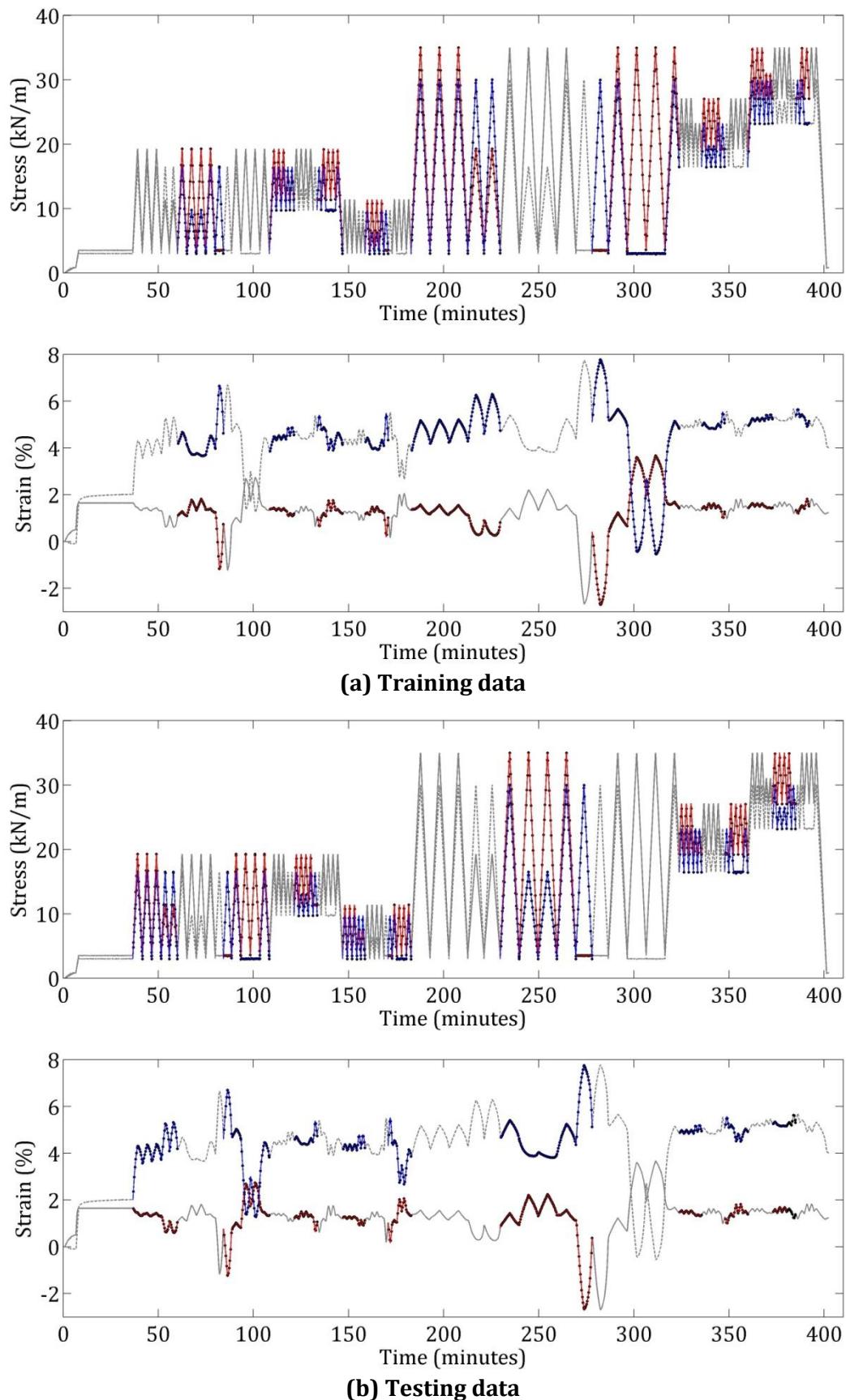
**Figure B.19: PVC network tested with 2:1 profile in recurrent mode**  
**(— warp, — fill, \* target, ° network output)**



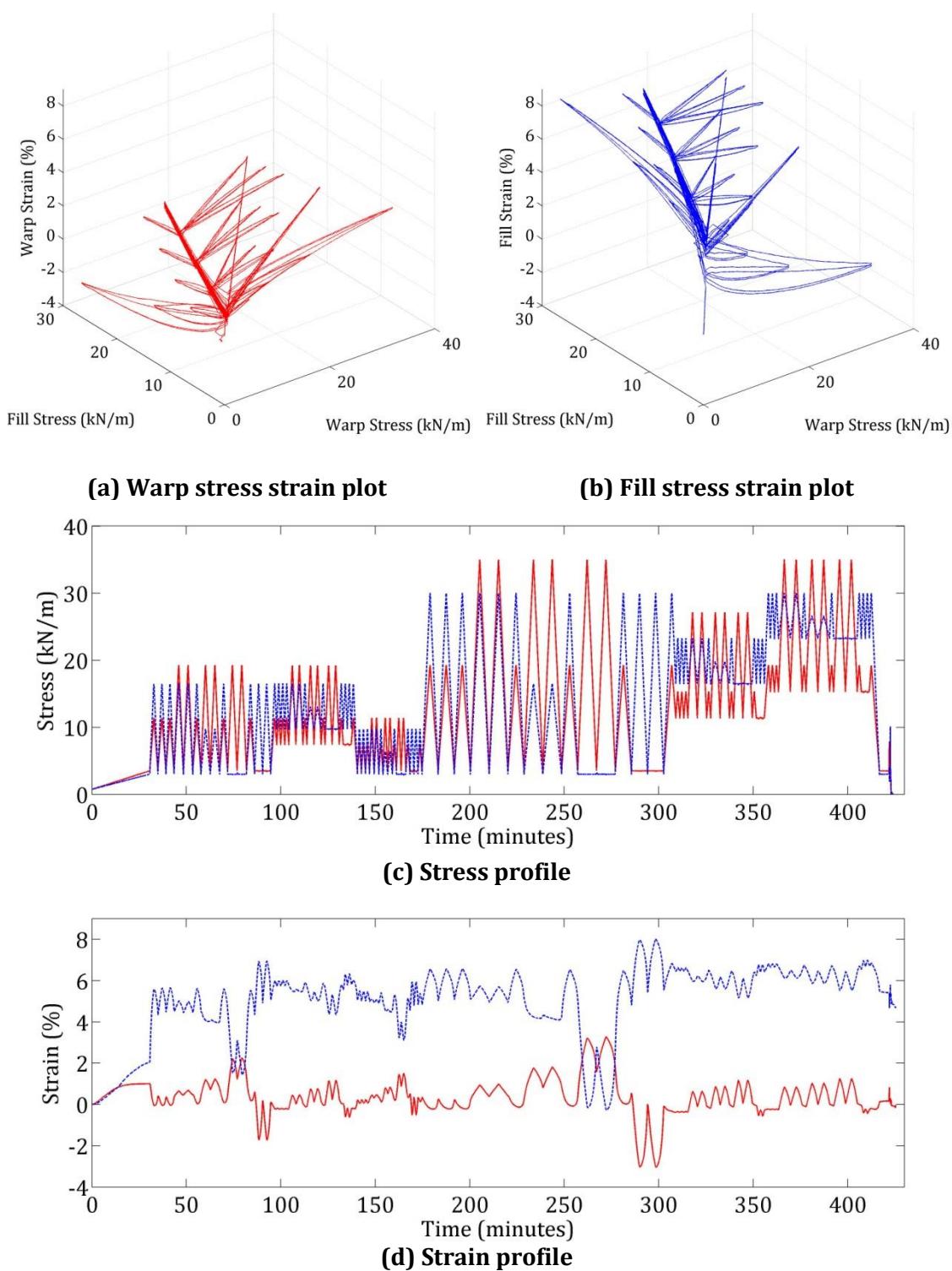
**Figure B.20: PVC network tested with 2:1 profile in recurrent mode**  
**(— warp, — fill, \* target, ° network output)**



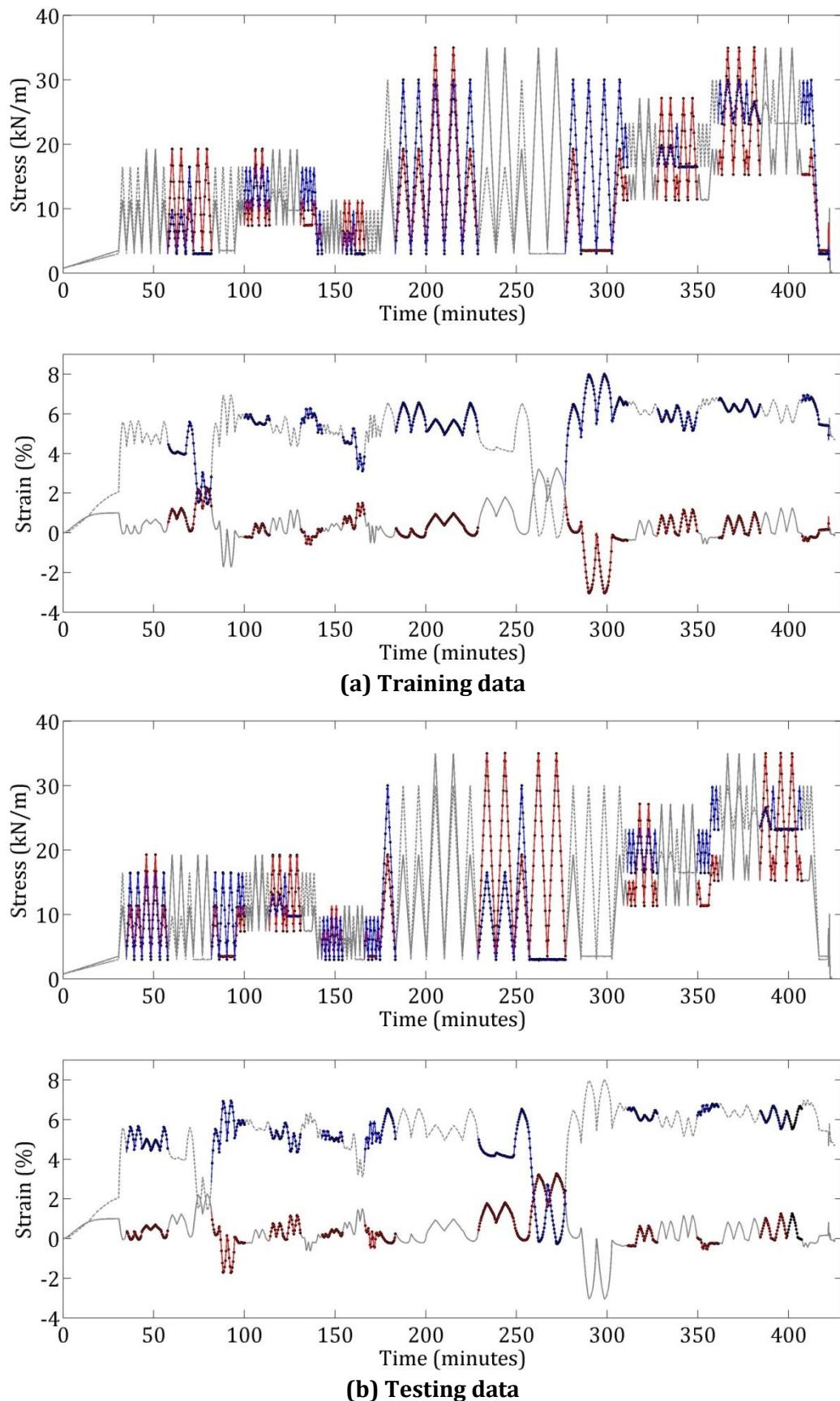
**Figure B.21: PTFE Biaxial 1:1 stress profile and results for history network training and validation(— warp, .... fill)**



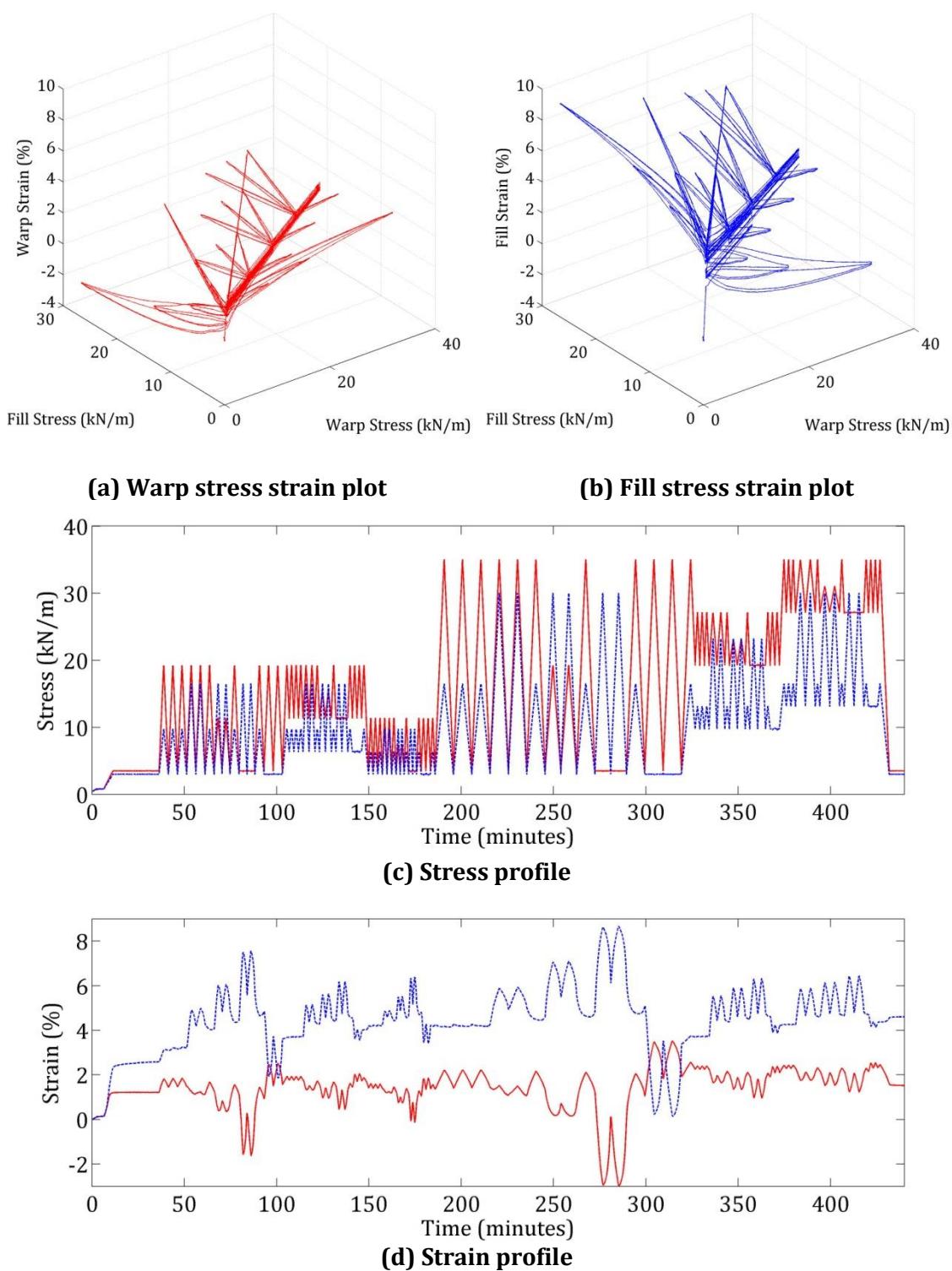
**Figure B.22: PTFE Biaxial 1:1 stress and resulting strain profiles divided into training and testing data respectively (— warp, .... fill, • data point )**



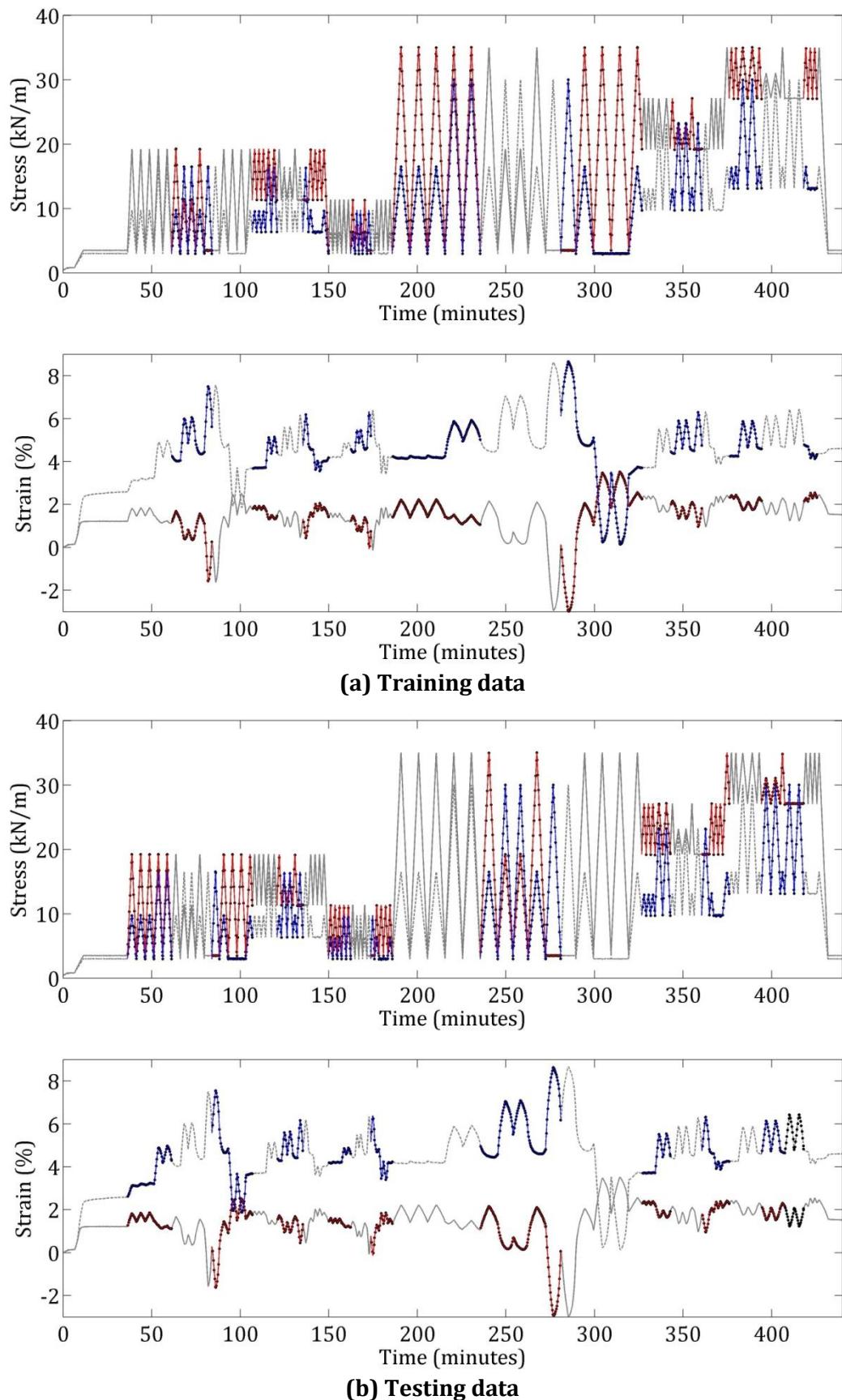
**Figure B.23: PTFE Biaxial 1:2 stress profile and results for history network training and validation (— warp, .... fill)**



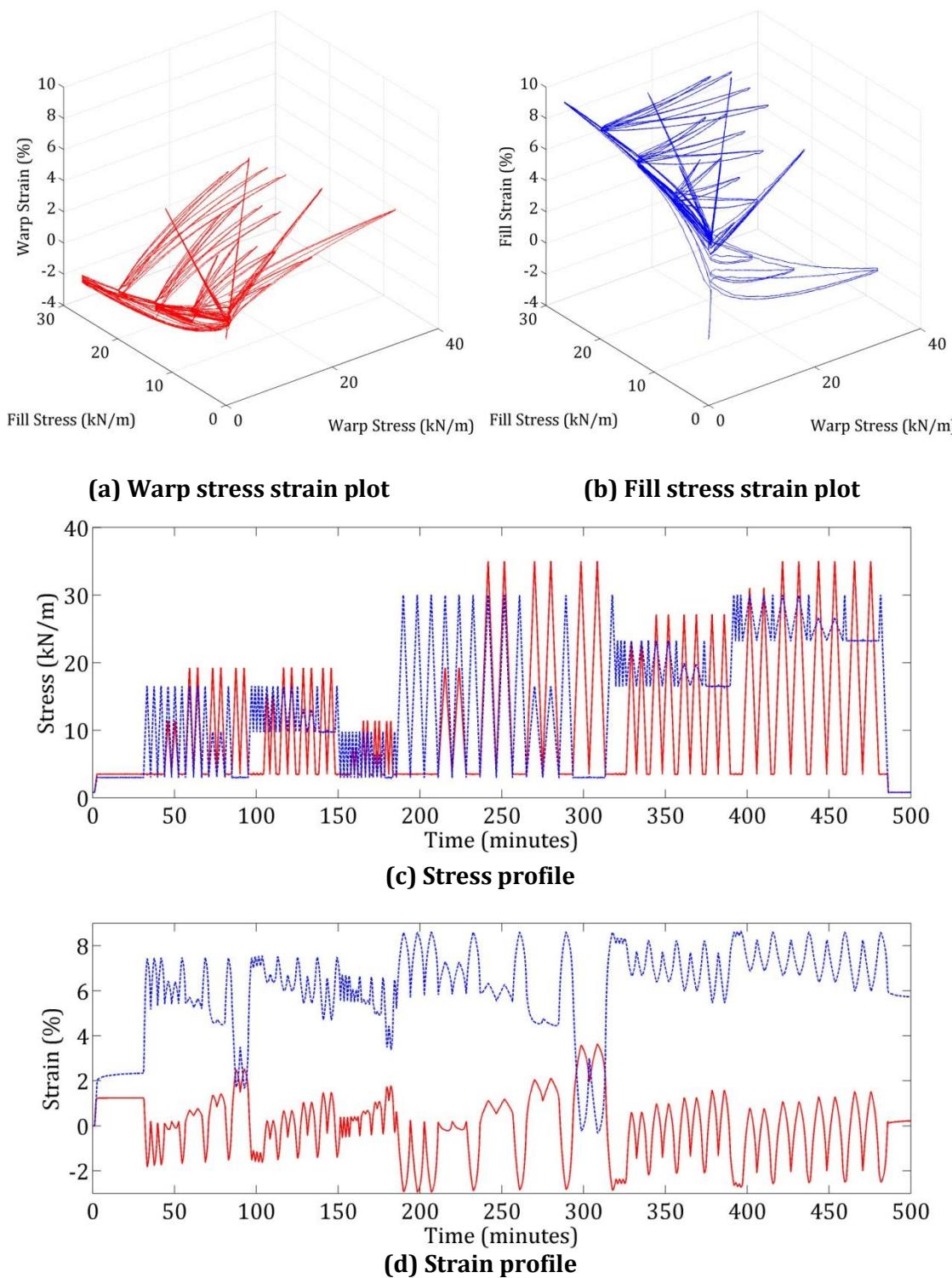
**Figure B.24: PTFE Biaxial 1:2 stress and resulting strain profiles divided into training and testing data respectively (— warp, .... fill, • data point )**



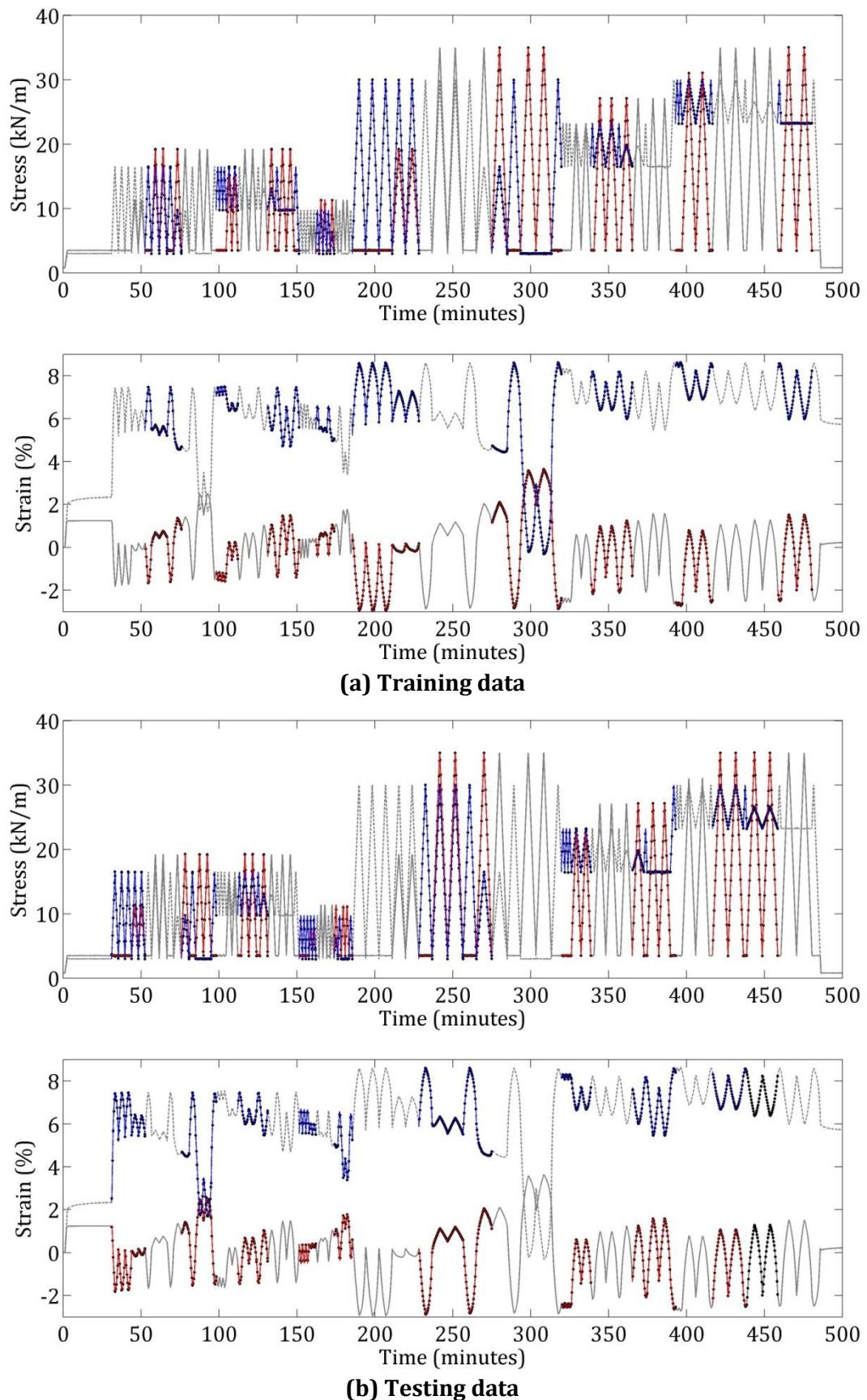
**Figure B.25: PTFE Biaxial 2:1 stress profile and results for history network training and validation (— warp, .... fill)**



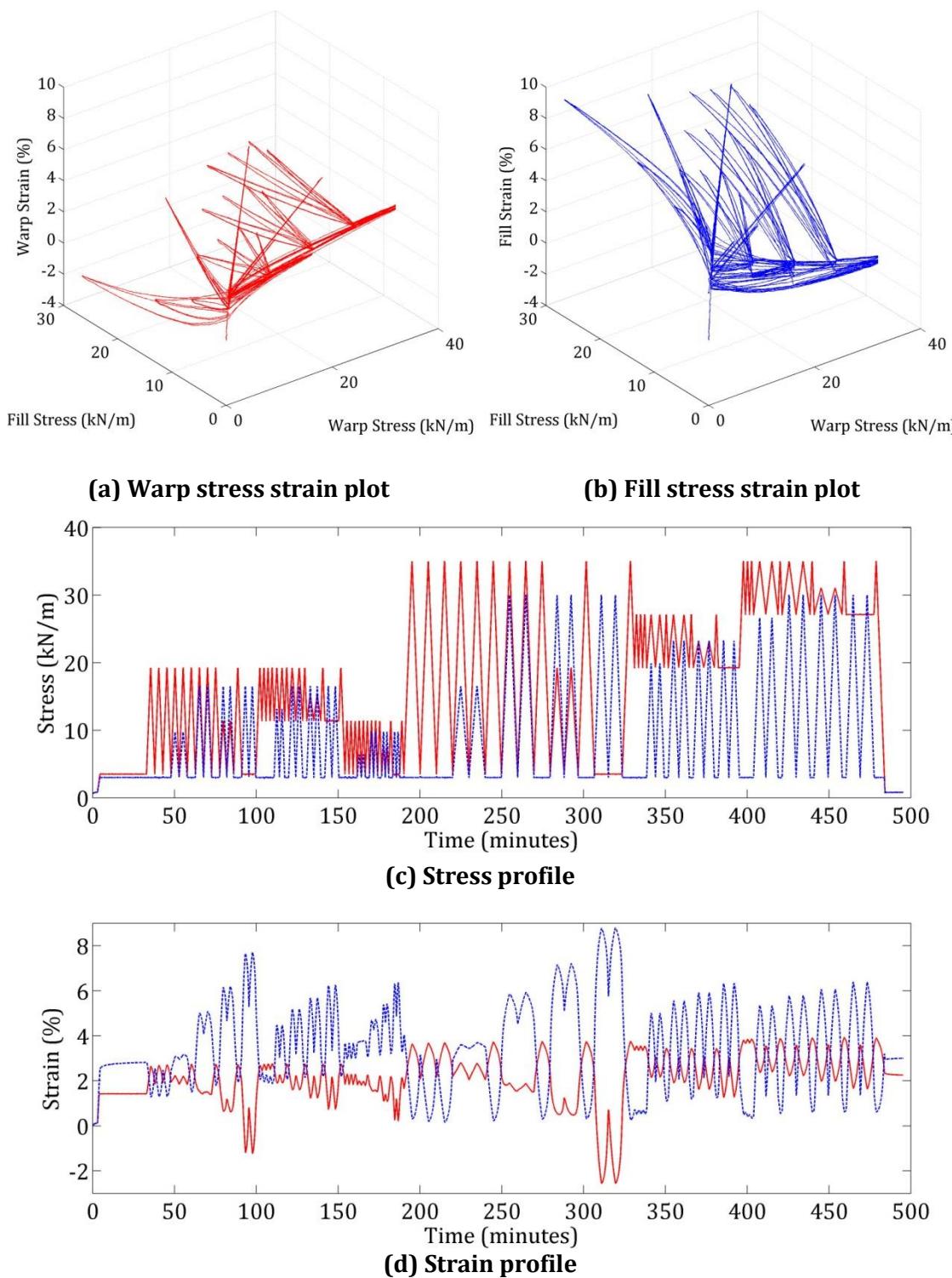
**Figure B.26: PTFE Biaxial 2:1 stress and resulting strain profiles divided into training and testing data respectively (— warp, .... fill, • data point )**



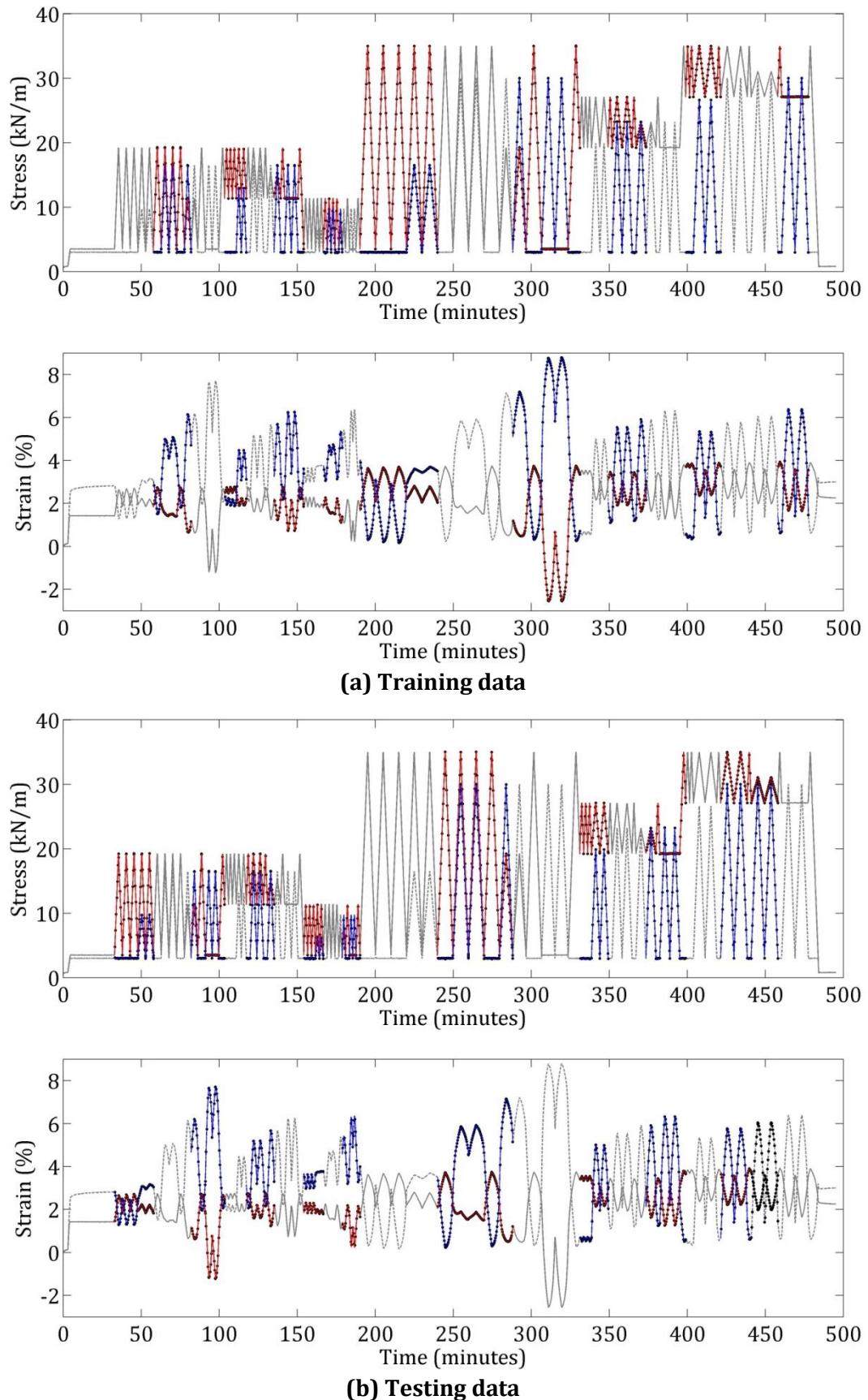
**Figure B.27: PTFE Biaxial 0:1 stress profile and results for history network training and validation (— warp, .... fill)**



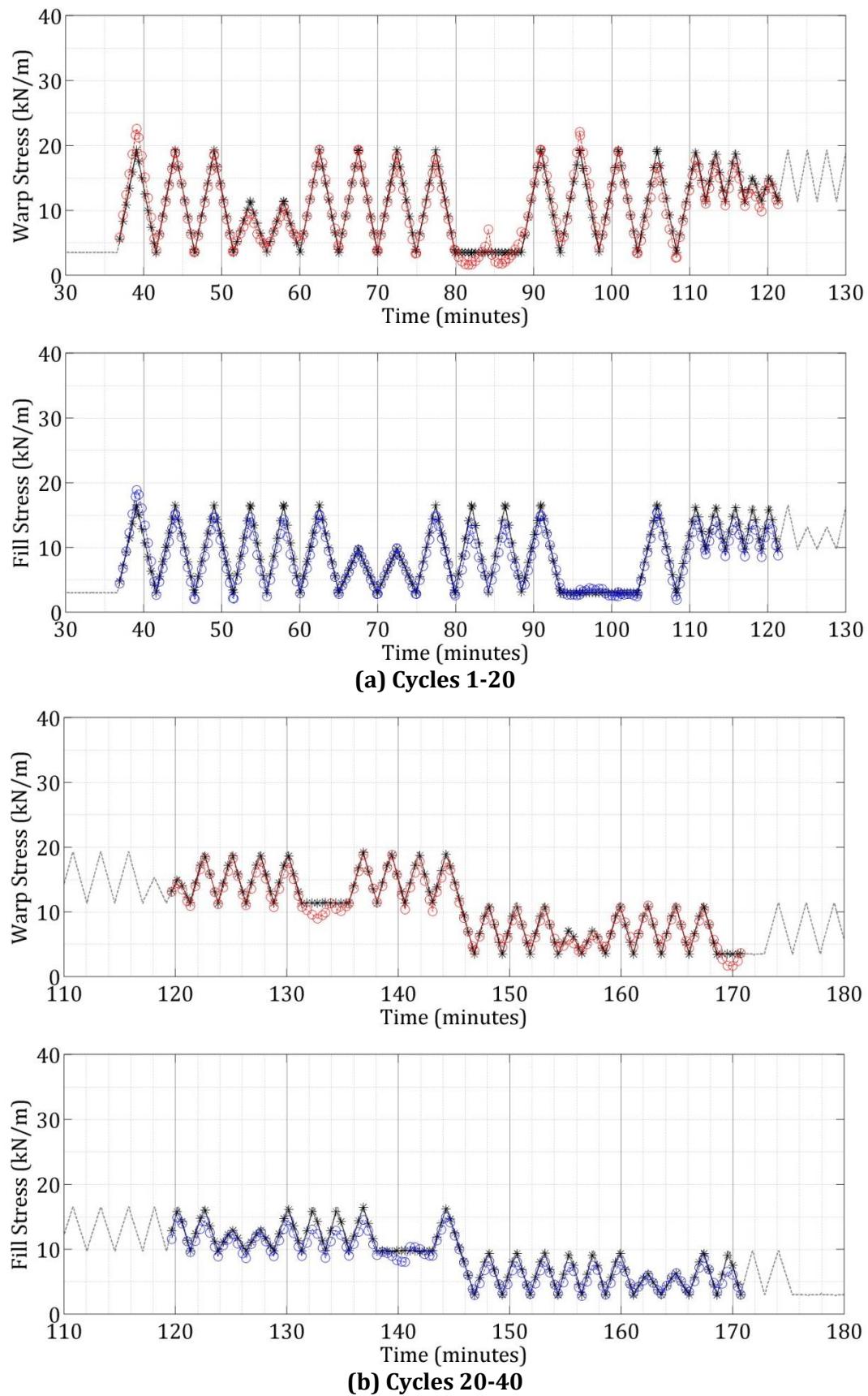
**Figure B.28: PTFE Biaxial 0:1 stress and resulting strain profiles divided into training and testing data respectively (— warp, .... fill, • data point )**



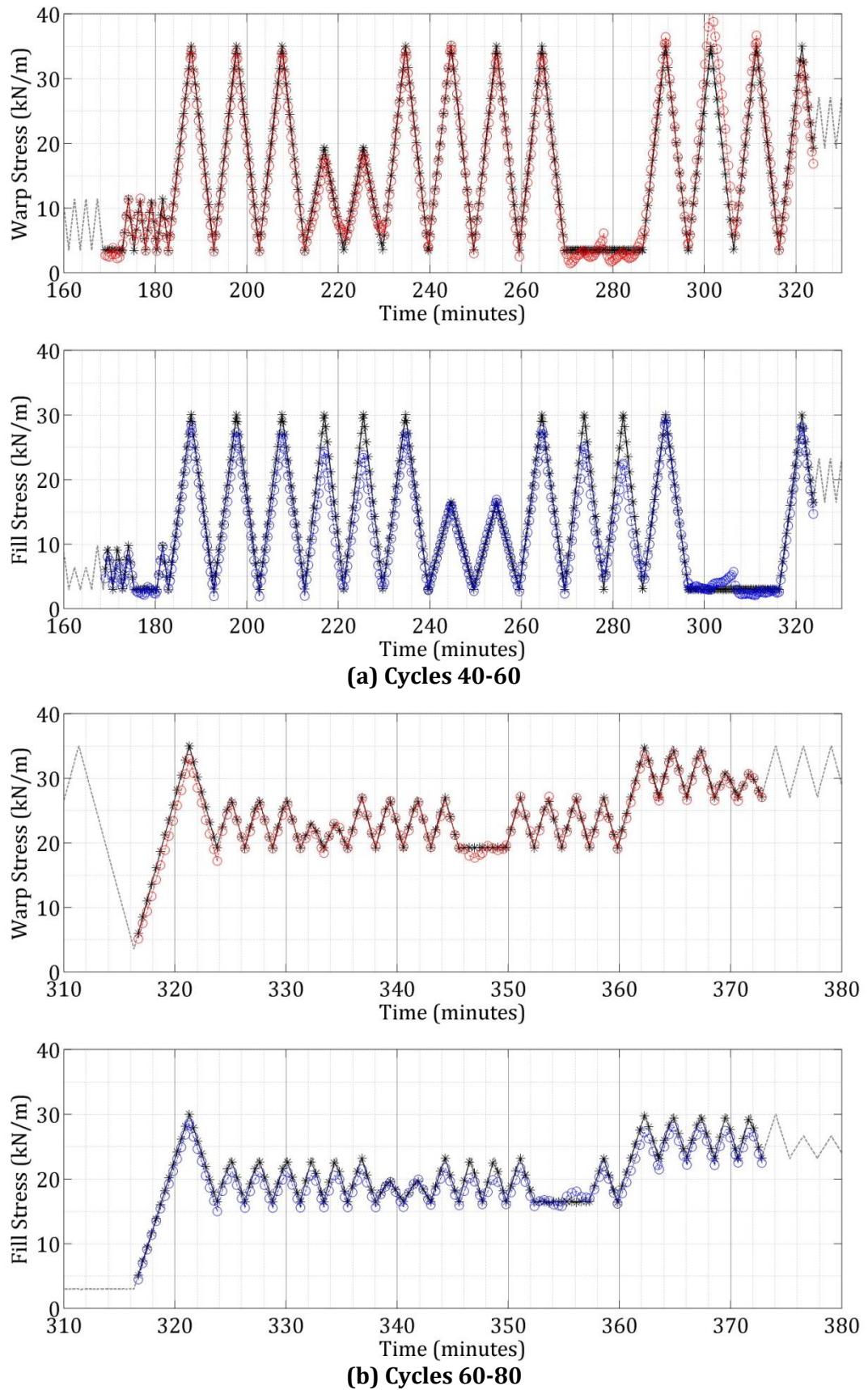
**Figure B.29: PTFE Biaxial 1:0 stress profile and results for history network training and validation (— warp, .... fill)**



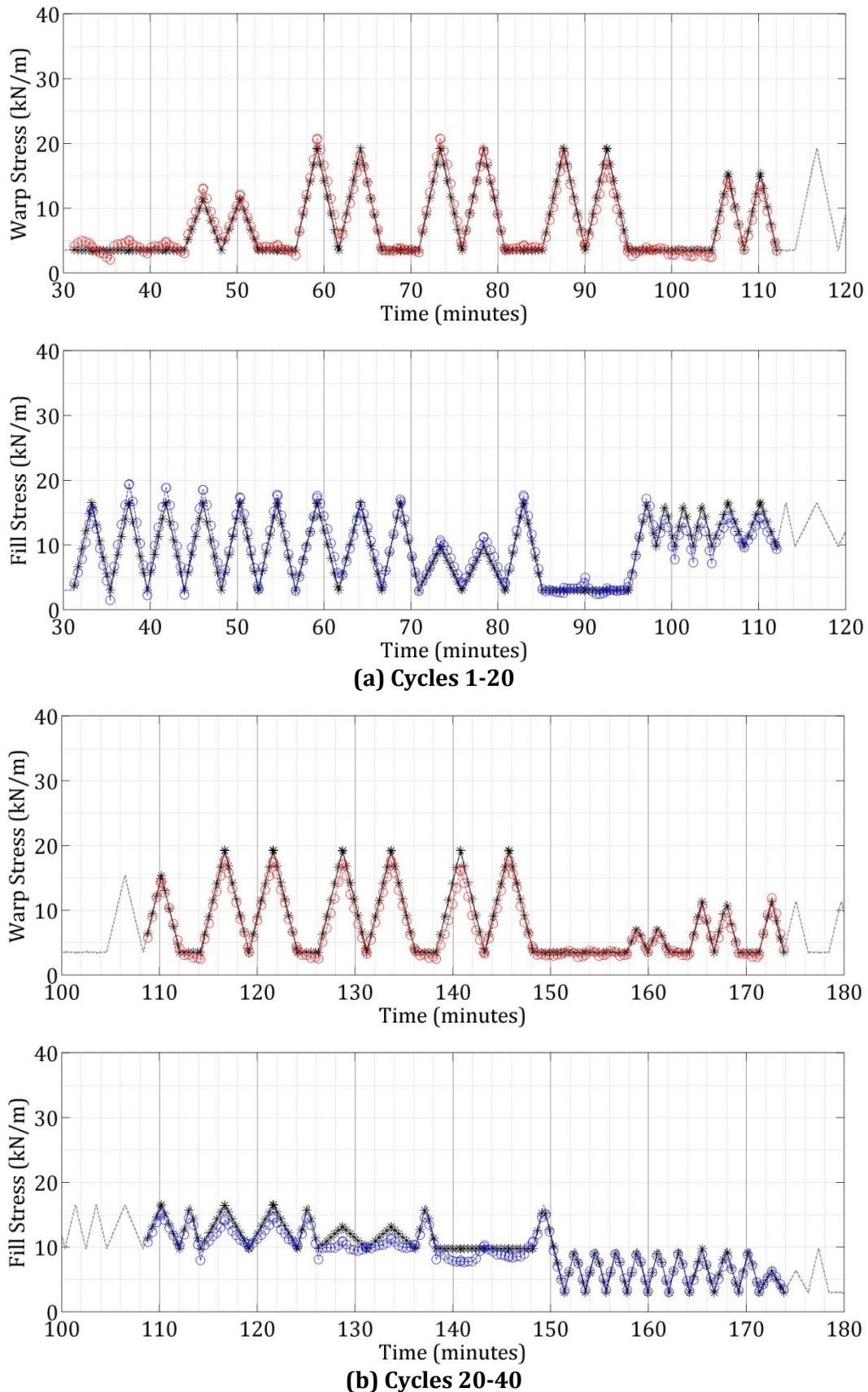
**Figure B.30:** PTFE Biaxial 1:0 stress and resulting strain profiles divided into training and testing data respectively (— warp, .... fill, • data point)



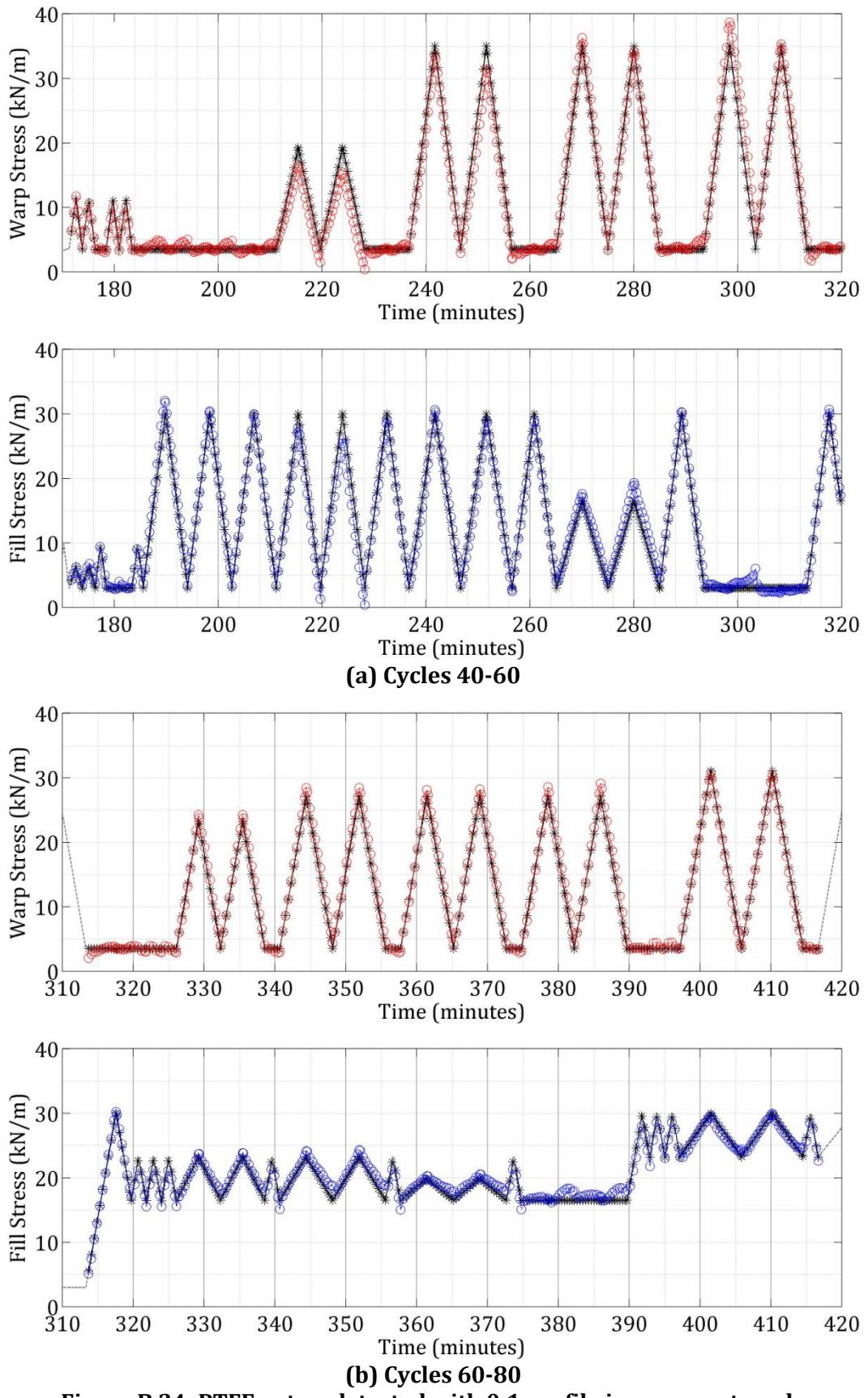
**Figure B.31:** PTFE network tested with 1:1 profile in recurrent mode  
 (— warp, — fill, \* target, ° network output)



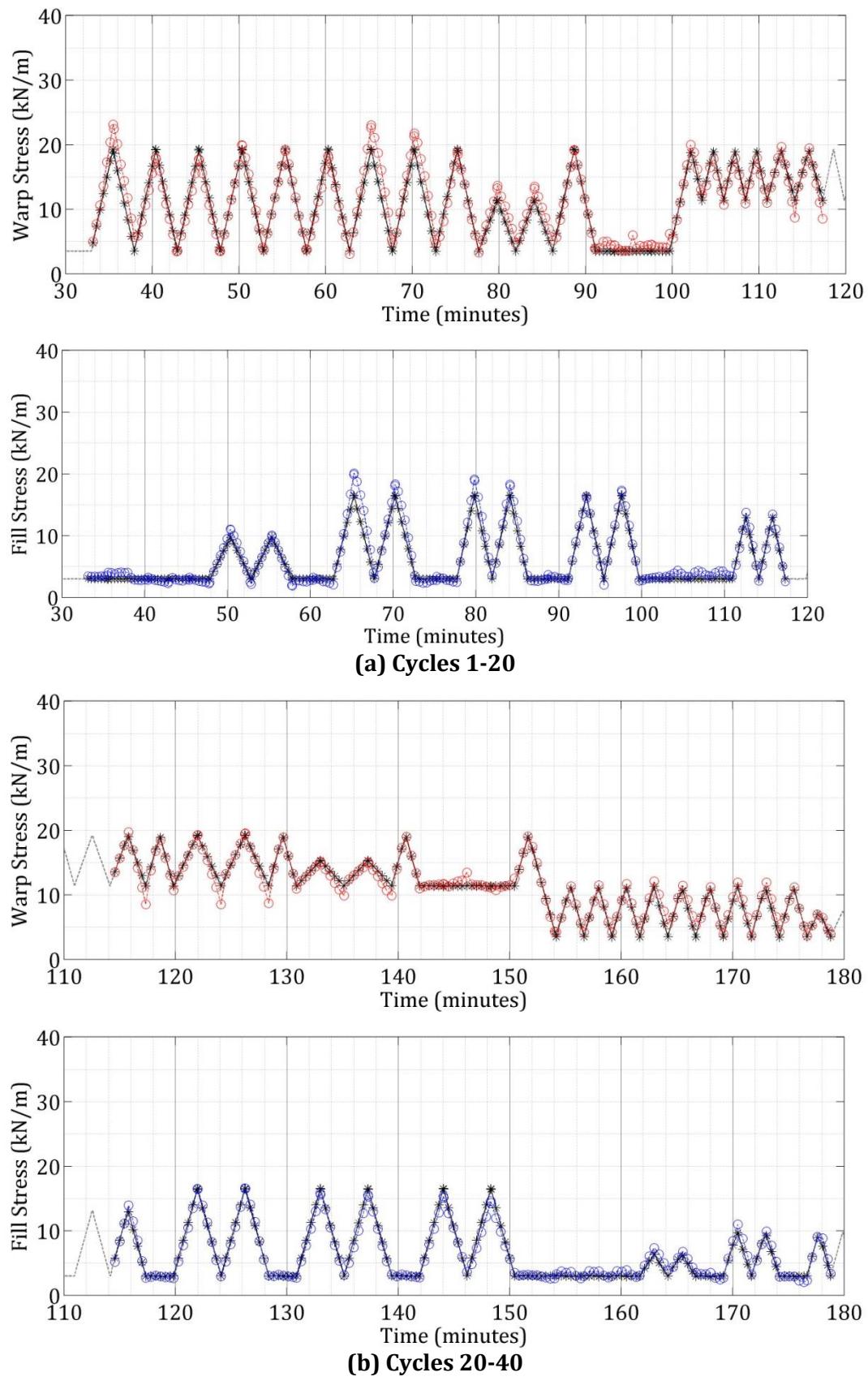
**Figure B.32: PTFE network tested with 1:1 profile in recurrent mode**  
**(— warp, — fill, \* target, ° network output)**



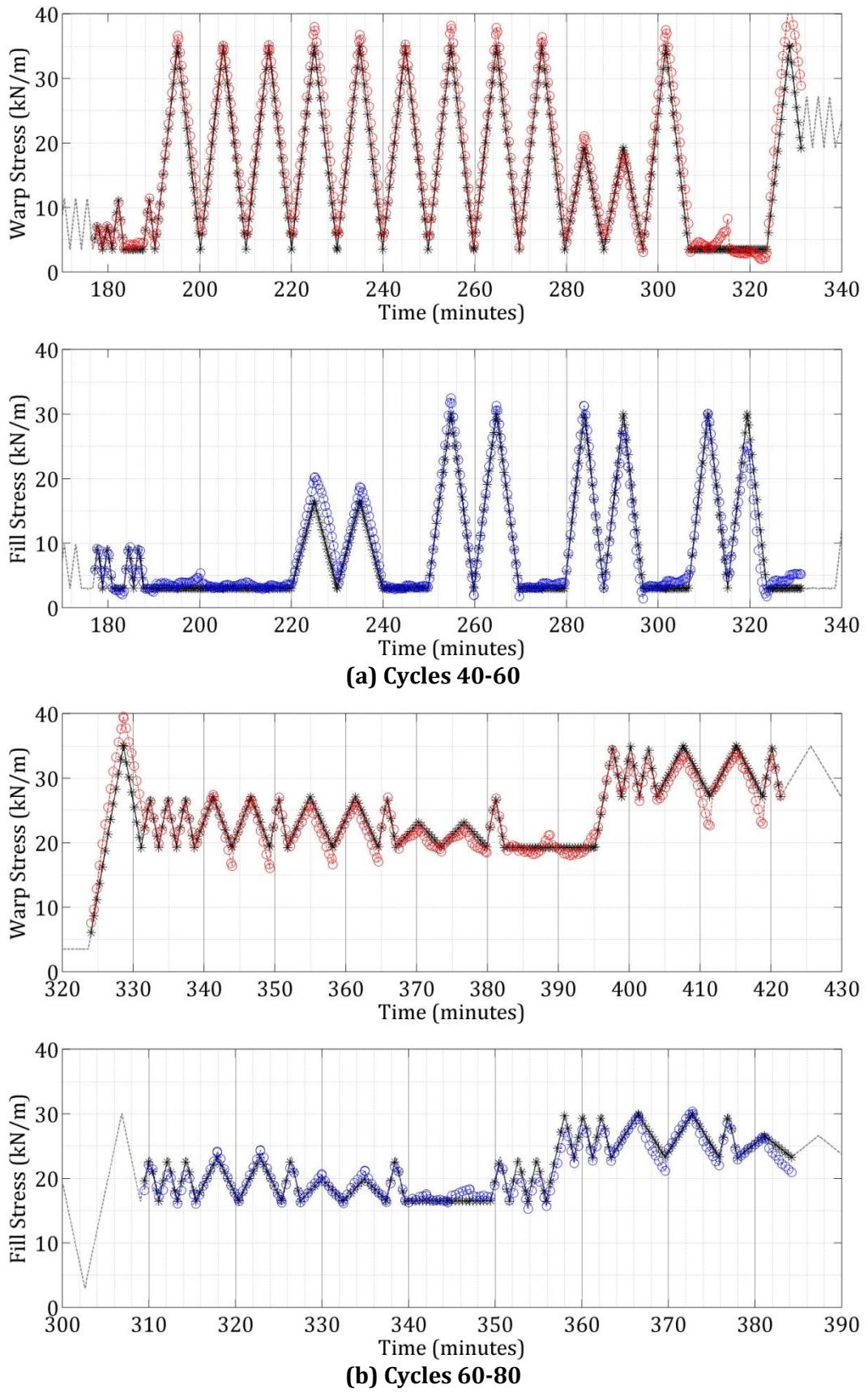
**Figure B.33: PTFE network tested with 0:1 profile in recurrent mode**  
**(— warp, — fill, \* target, ° network output)**



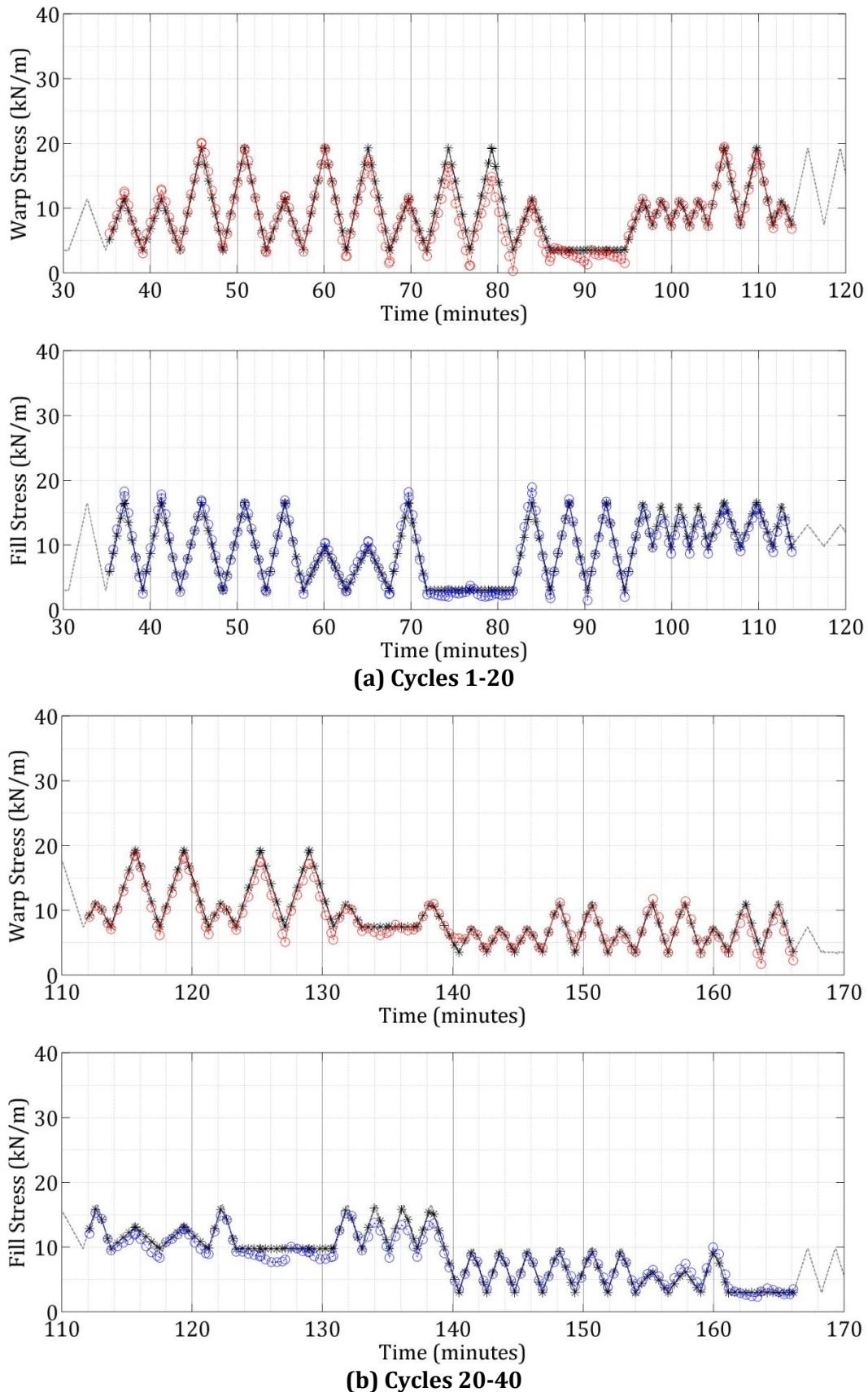
**Figure B.34: PTFE network tested with 0:1 profile in recurrent mode**  
**(— warp, — fill, \* target, ○ network output)**



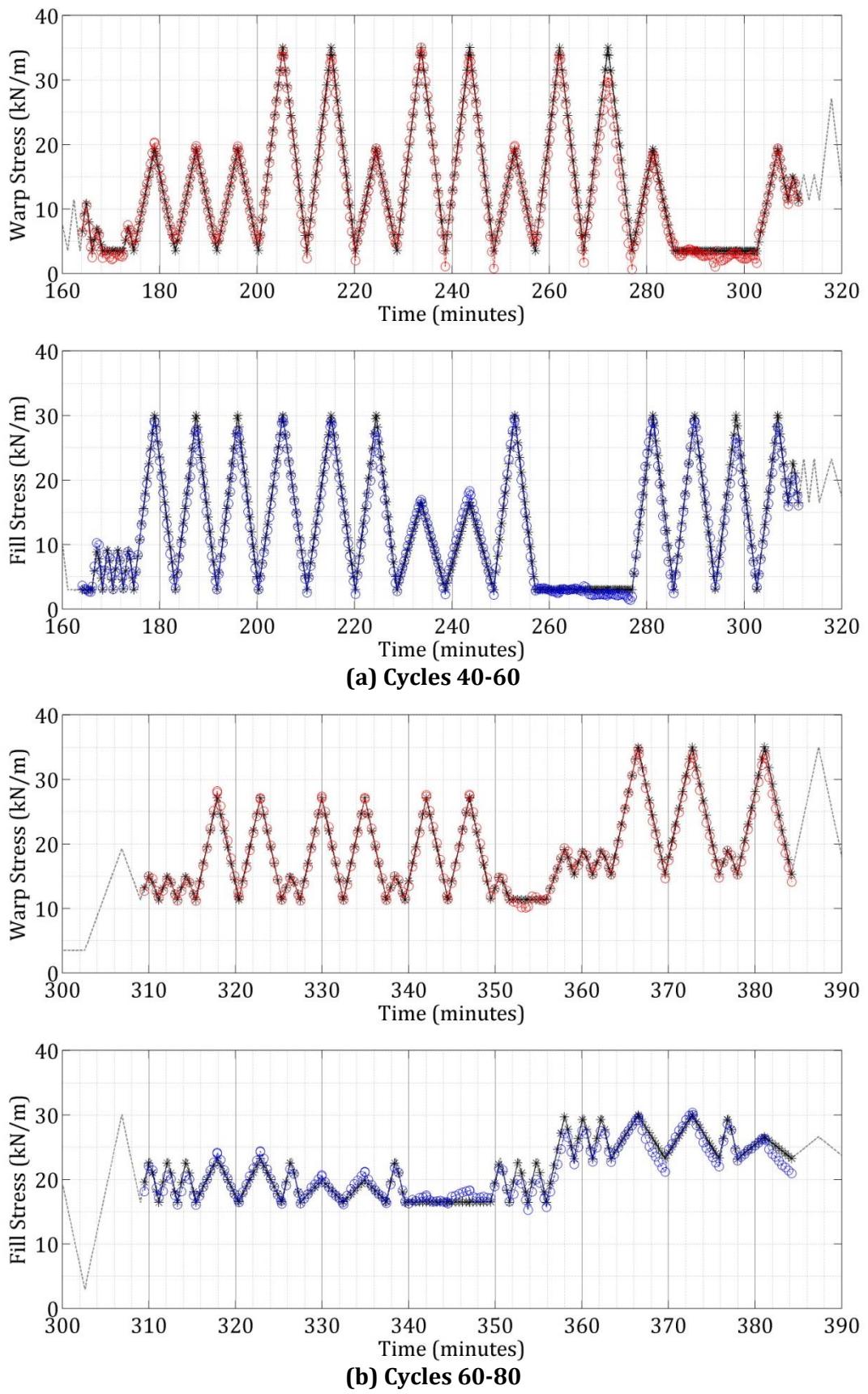
**Figure B.35: PTFE network tested with 1:0 profile in recurrent mode**  
 (— warp, — fill, \* target, ° network output)



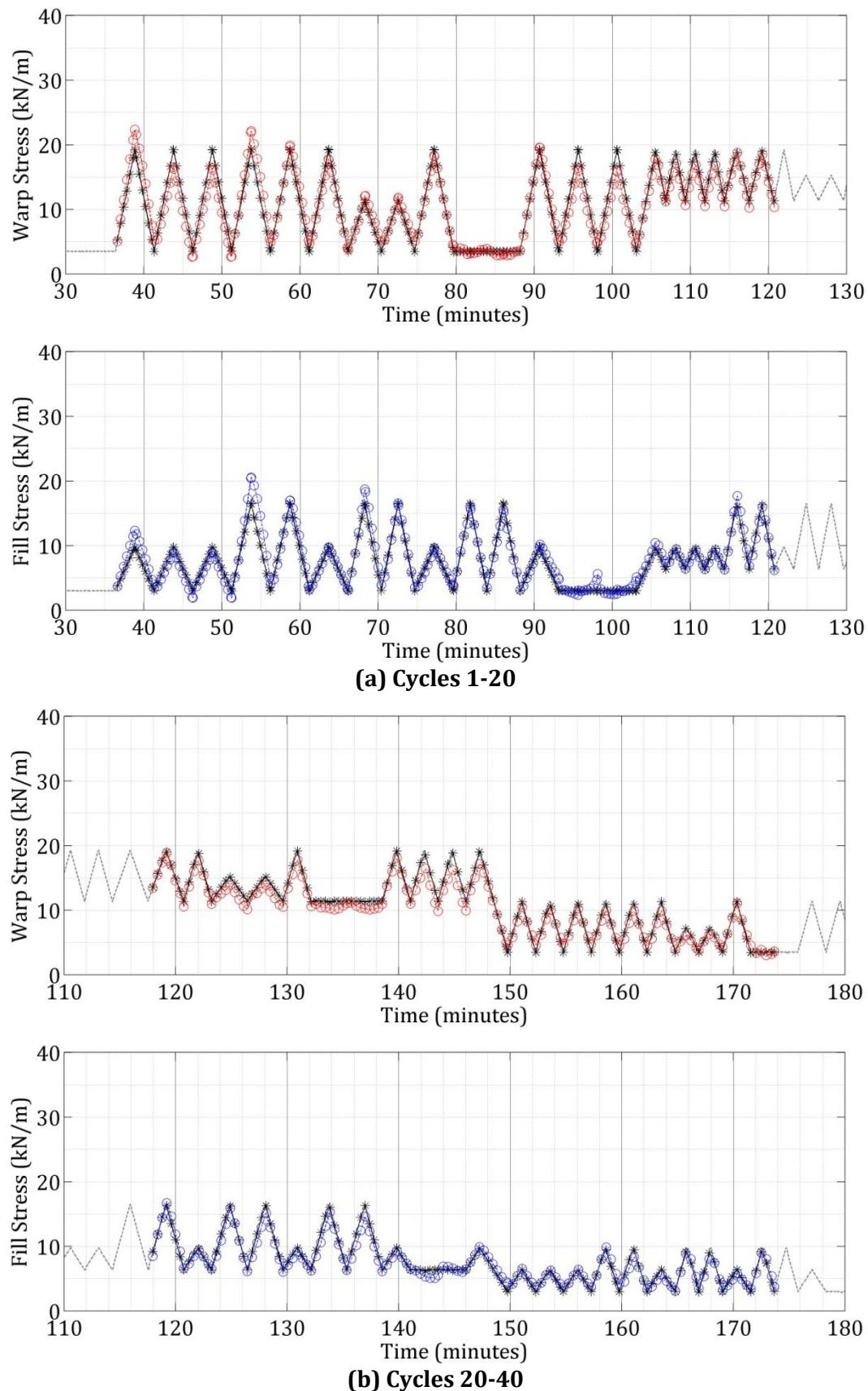
**Figure B.36: PTFE network tested with 1:0 profile in recurrent mode**  
**(— warp, — fill, \* target, ° network output)**



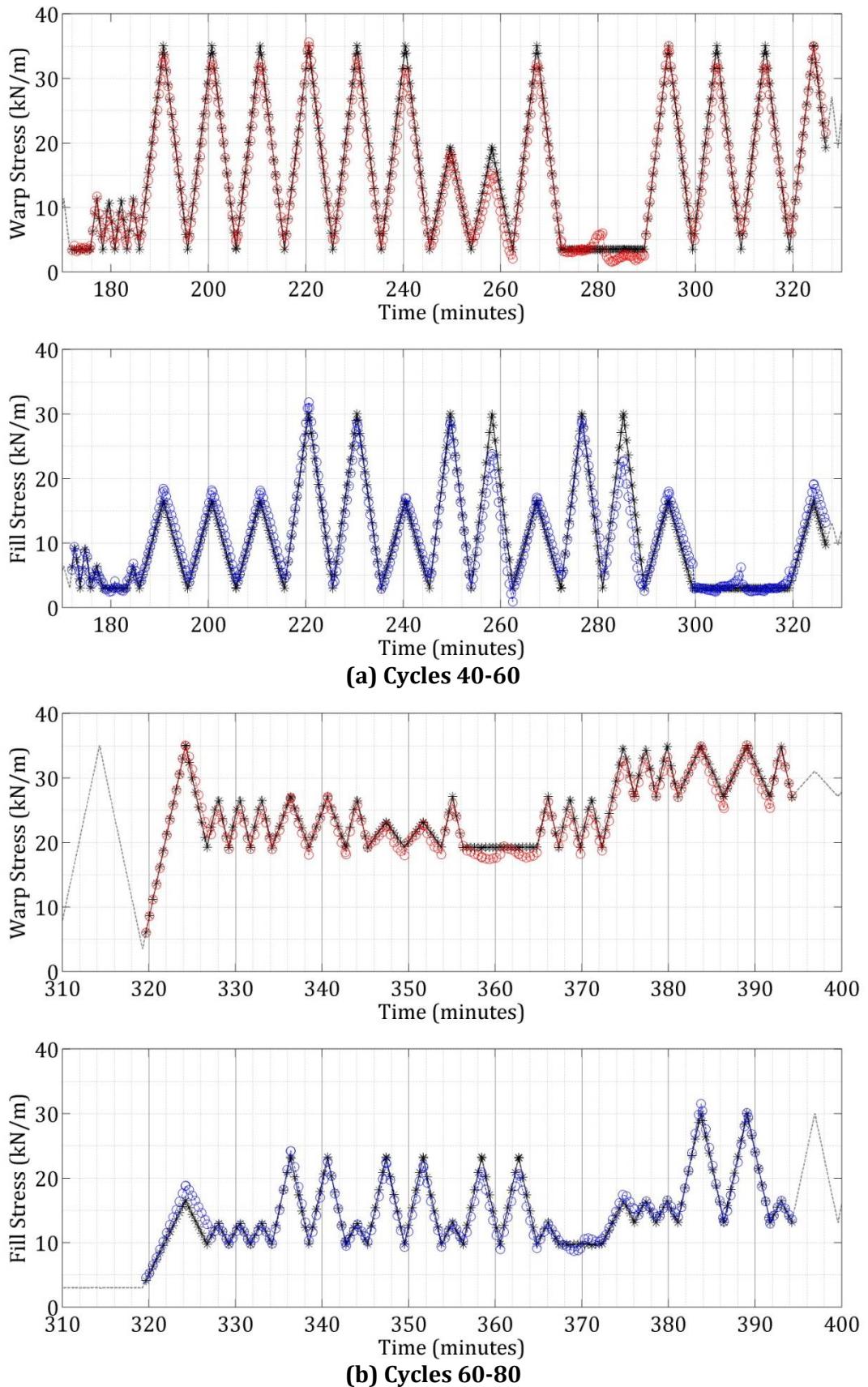
**Figure B.37: PTFE network tested with 1:2 profile in recurrent mode  
(— warp, — fill, \* target, ° network output)**



**Figure B.38: PTFE network tested with 1:2 profile in recurrent mode  
(— warp, — fill, \* target, ○ network output)**



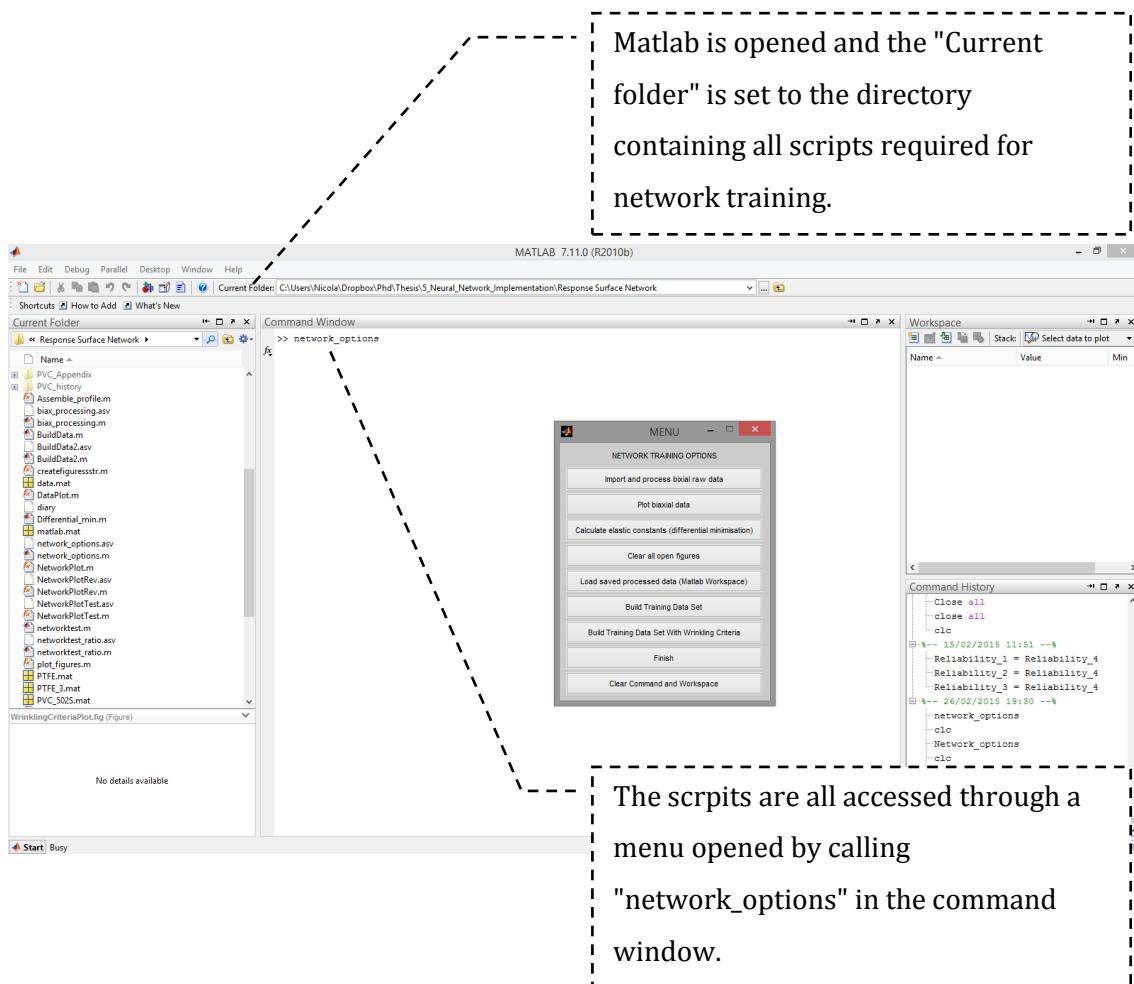
**Figure B.39:** PTFE network tested with 2:1 profile in recurrent mode  
 (— warp, — fill, \* target, ° network output)

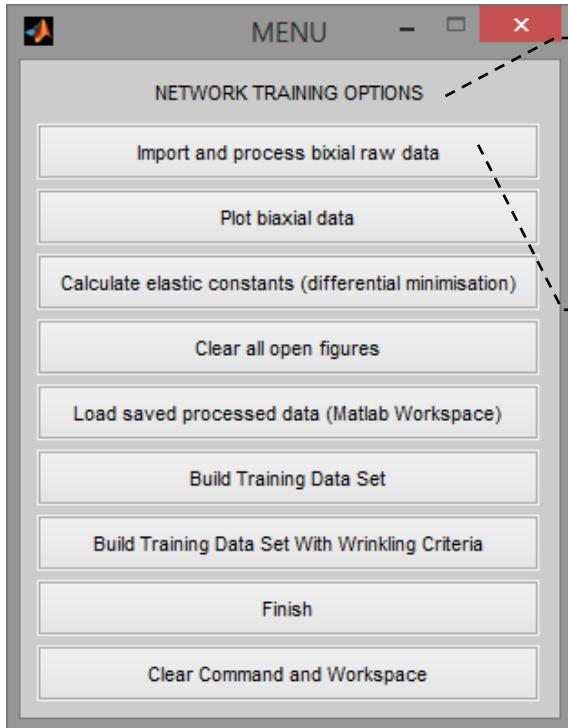


**Figure B.40: PTFE network tested with 2:1 profile in recurrent mode  
 (— warp, — fill, \* target, ° network output)**

## Appendix C. Neural Network Training Tool

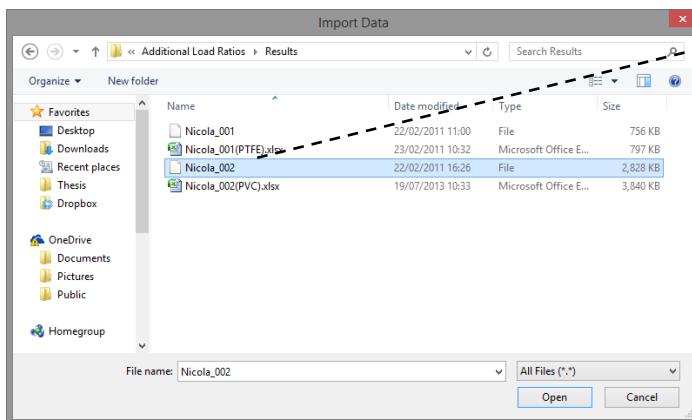
MATLAB® is a high level language and interactive environment that provides a great many tools for the generation of GUI, processing of raw data, graph plotting and neural network training. The Neural Network Toolbox™ (<http://uk.mathworks.com/products/neural-network/>) contains all of the neural network development tools available within the Matlab environment. A number of scripts have been developed in order to process raw experimental data and build training data sets. Network training and testing is also performed through scripts which call the functions of the Neural Network Toolbox. The internal variables of the networks may be saved in a .csv file used as input when implementing the network in the Fortran FE code. Several useful tools for reporting the network training have also been developed including numerical performance output and data plotting. The full response surface network training procedure is documented here.



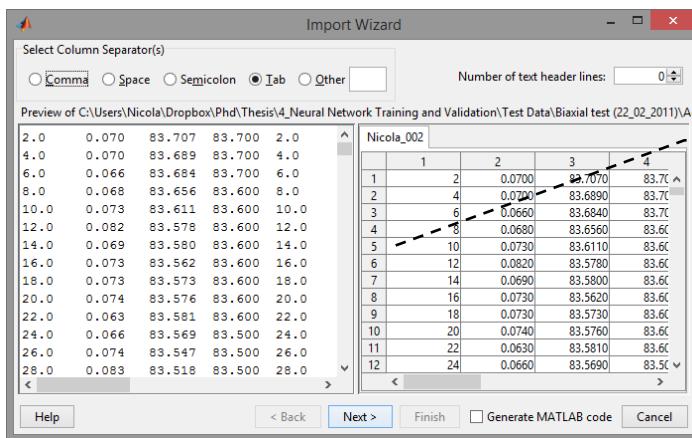


Initially the menu provides limited options. Further options become available as the training process proceeds.

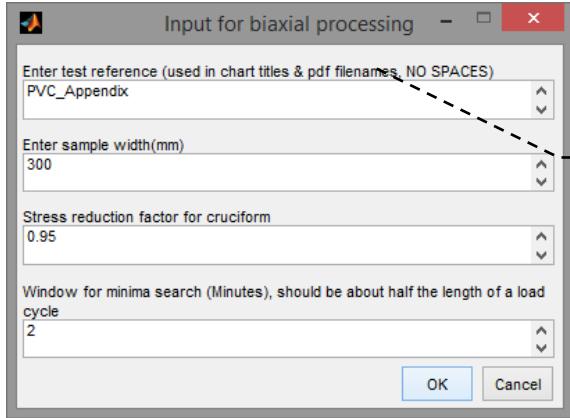
To initiate network training select "Import and process biaxial raw data". The raw data processing scripts have been adapted from in house biaxial data processing scripts developed by Dr. Ben Bridgens for the calculation of elastic constants.



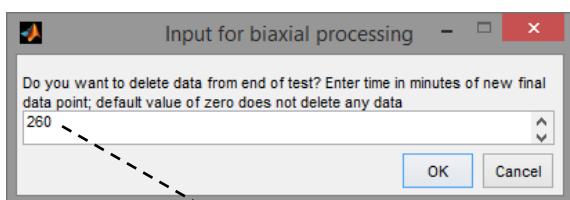
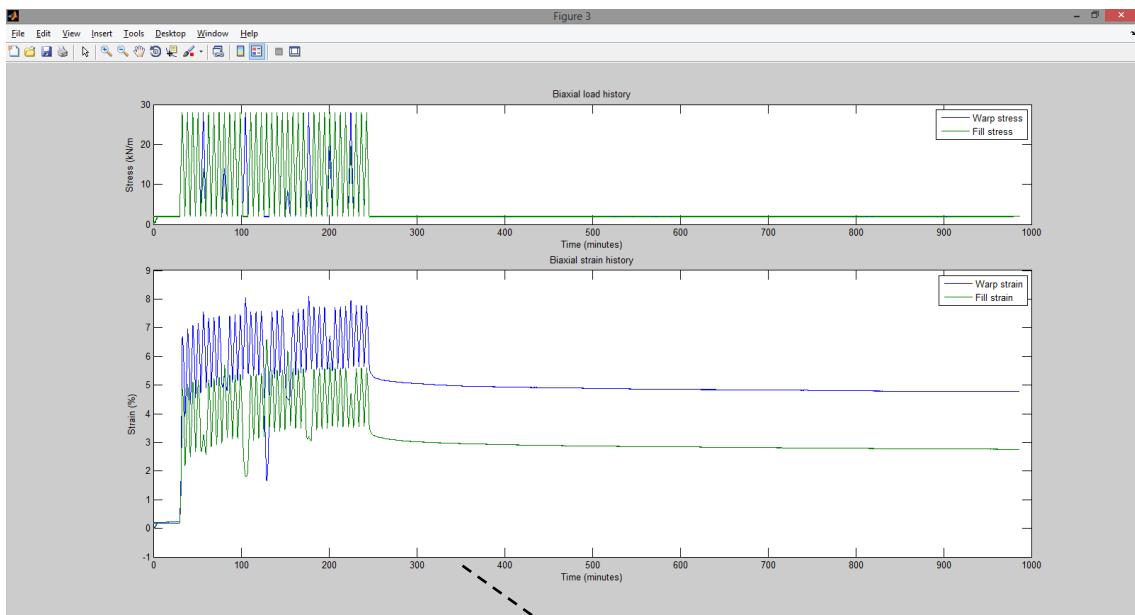
Upon selecting "Import and process biaxial raw data" a file explorer window is opened to allow you to navigate to the raw data file generated during a biaxial test.



The raw data is imported via the Matlab Import Wizard.

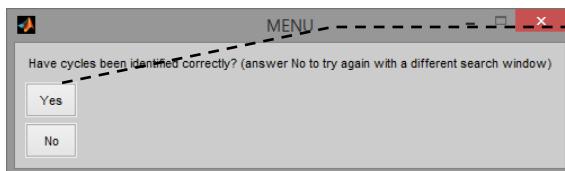
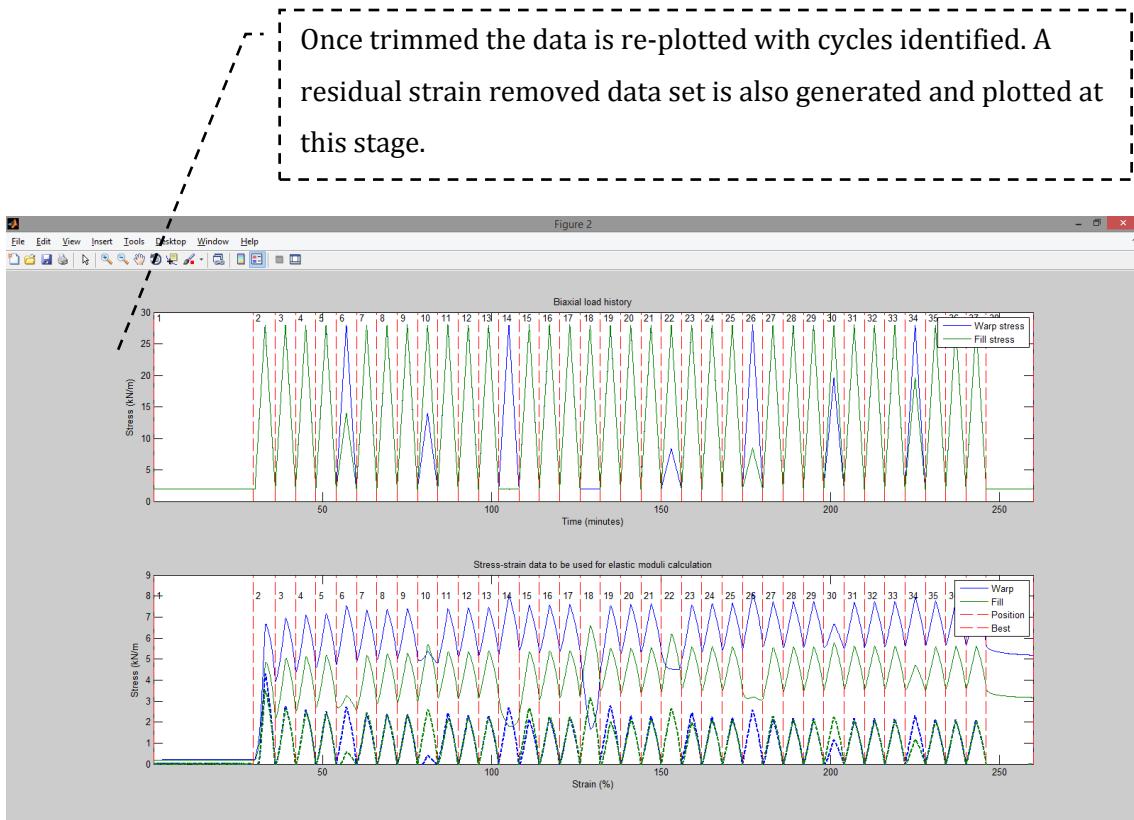


A GUI is displayed in which the test data is named and variables used in the conversion of load and displacement data to equivalent stress and strain data are set.



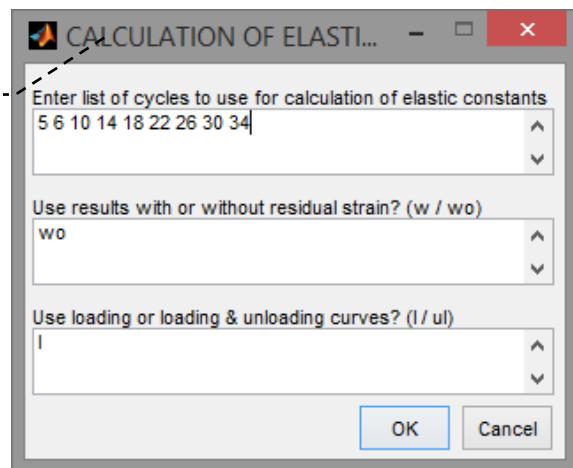
The resulting raw stress strain data is plotted.

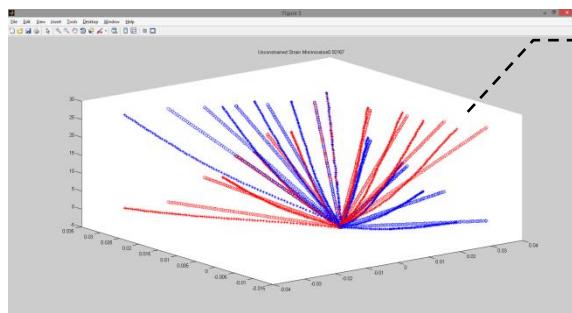
The option to trim unnecessary data from the end of the profile is given.



If any cycles have been incorrectly identified the turning point search window may be modified and the turning points are recalculated.

Once the cycles have been identified correctly elastic constants for the generation of additional data in the wrinkling and slack region are fit to selected cycles of the experimental data. The cycles, residual strain settings, and portion of the loading curve are each set via a GUI.

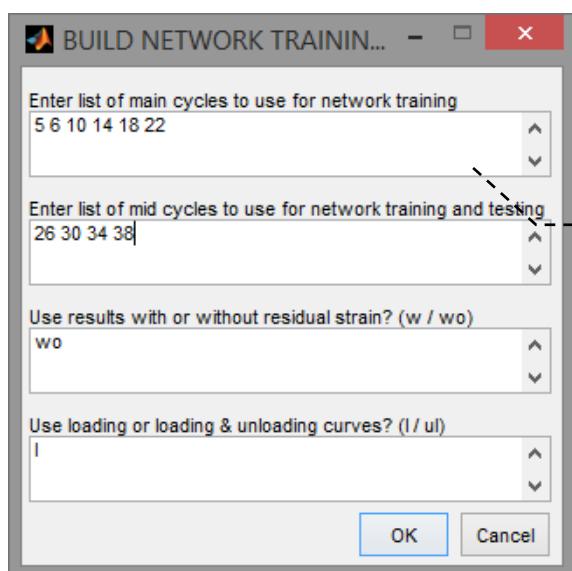
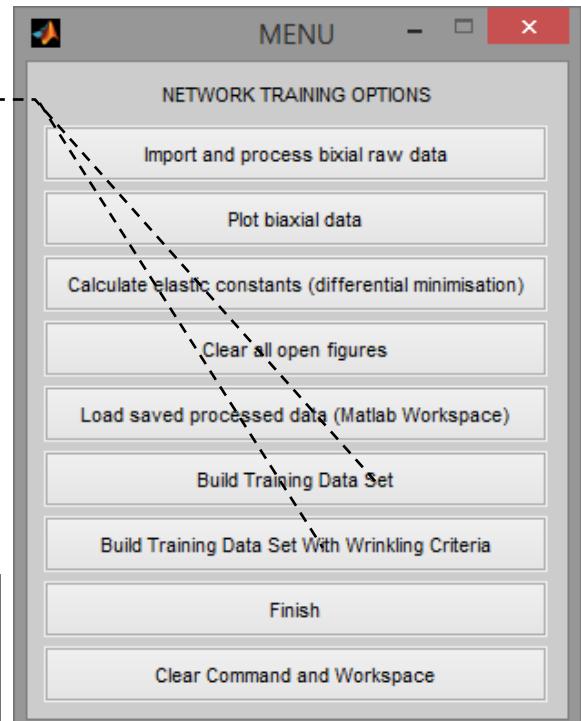




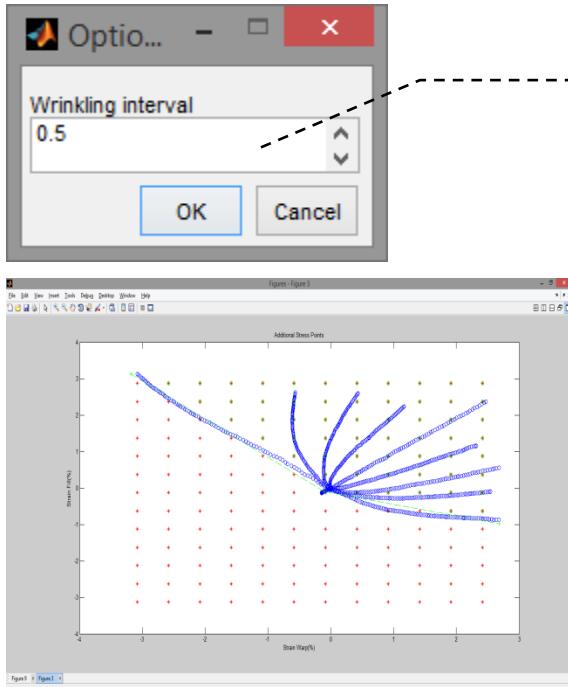
A 3D plot of the plane stress model fit to the data is automatically generated to allow inspection of the fit. The elastic constants are stored for use later and also output to the workspace

The raw biaxial data has now been processed and is output the Matlab workspace. The remaining scripts have been developed solely for the generation of network material models in this Thesis.

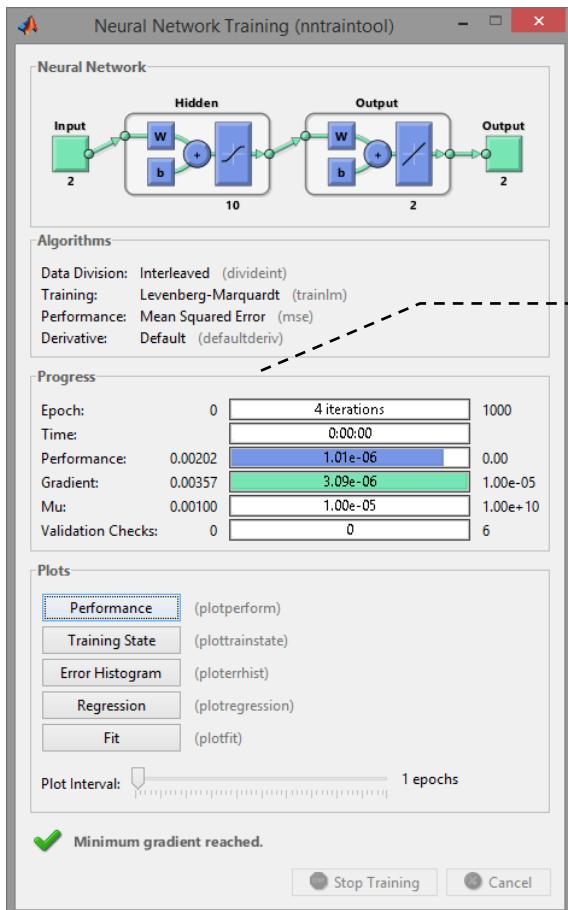
The next step in the process is the generation of training and testing data sets. Options to generate data sets with or without additional wrinkling training data are available. The following shows the generation of a data set with wrinkling data.



As with the fitting of elastic constants the network training data sets are made up of selected cycles. Two lists of cycles are required. These sets of cycles provide 'unseen' data for network testing as well as training. Again residual strain may be included or removed and the loading and / or unloading portions may be selected.

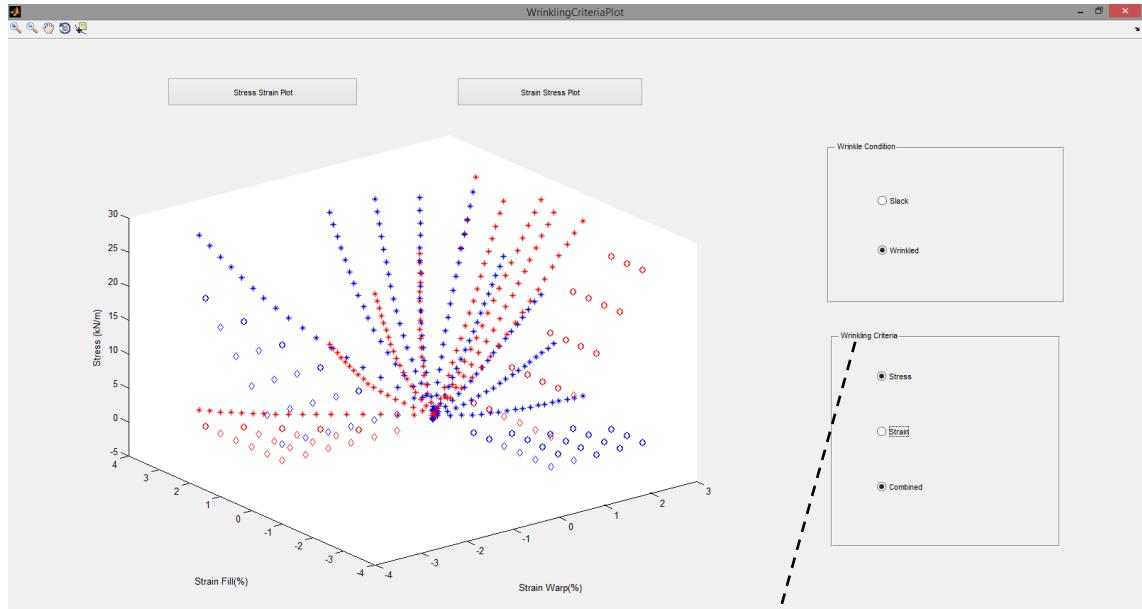


The density of a grid of additional stress data points is set. These points are used in the generation of additional data in the wrinkled region using a plane stress model. A stress to strain network is also used to generate additional data in the experimental data region. These points are plotted to allow the user to check the grid



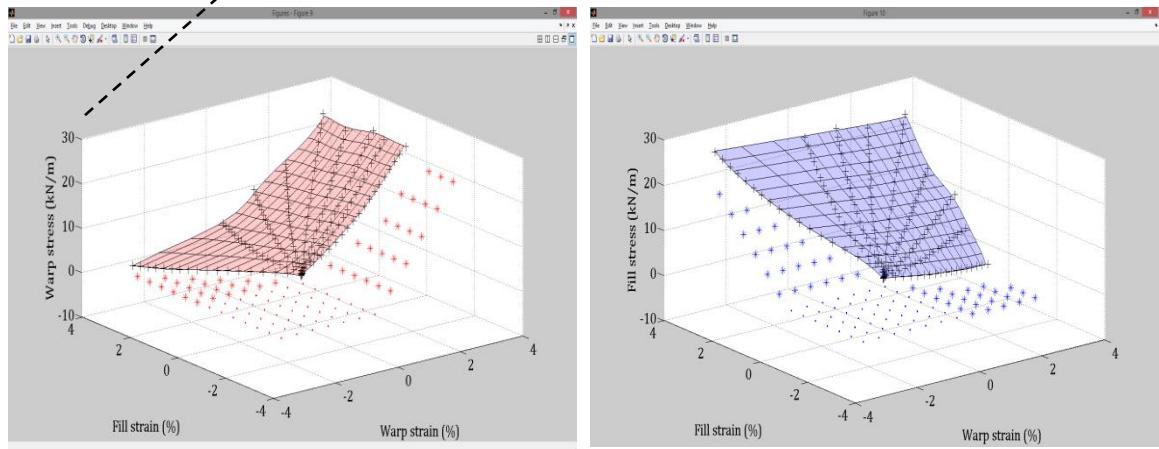
A set of 10 Stress to Strain networks are trained and tested. The network demonstrating the best fit is selected to generate the additional data set.

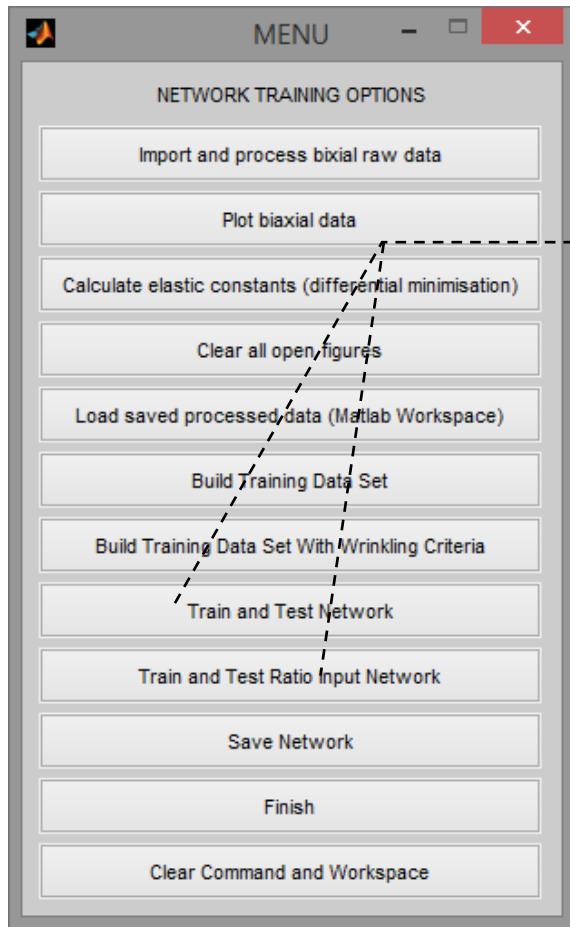
The Matlab training GUI allows network training to be supervised. Various plots may be generated.



An interactive plot of the wrinkling data is produced. This allows the user to plot the different wrinkling criteria (stress, strain and combined).

Plots of the final training data set are generated and saved as jpegs in an automatically generated network directory.





The network training and testing options are now available. Either a 2 input network or a 3 input network with an additional ratio input may be selected.

Various network options are set via a GUI.

The Network set is given a name and the number of networks to be trained is set.

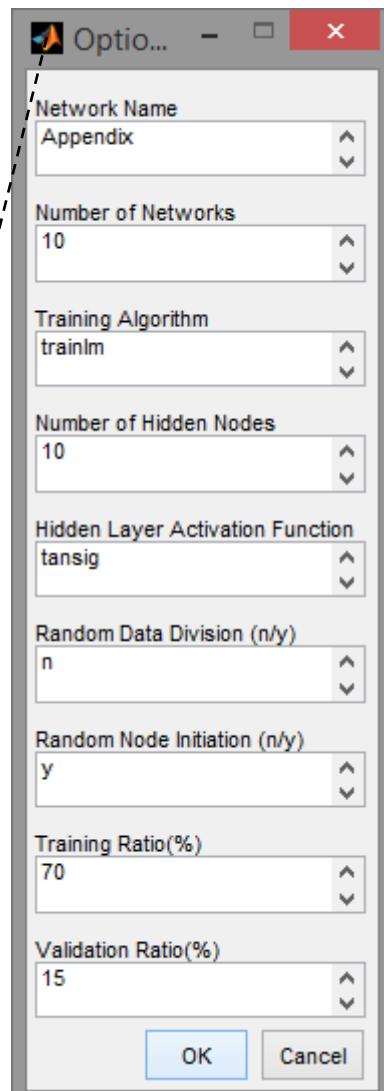
The training algorithm is set by default to "trainlm" but any training algorithm available in the Neural Network Toolbox may be used.

The number of hidden nodes is selected.

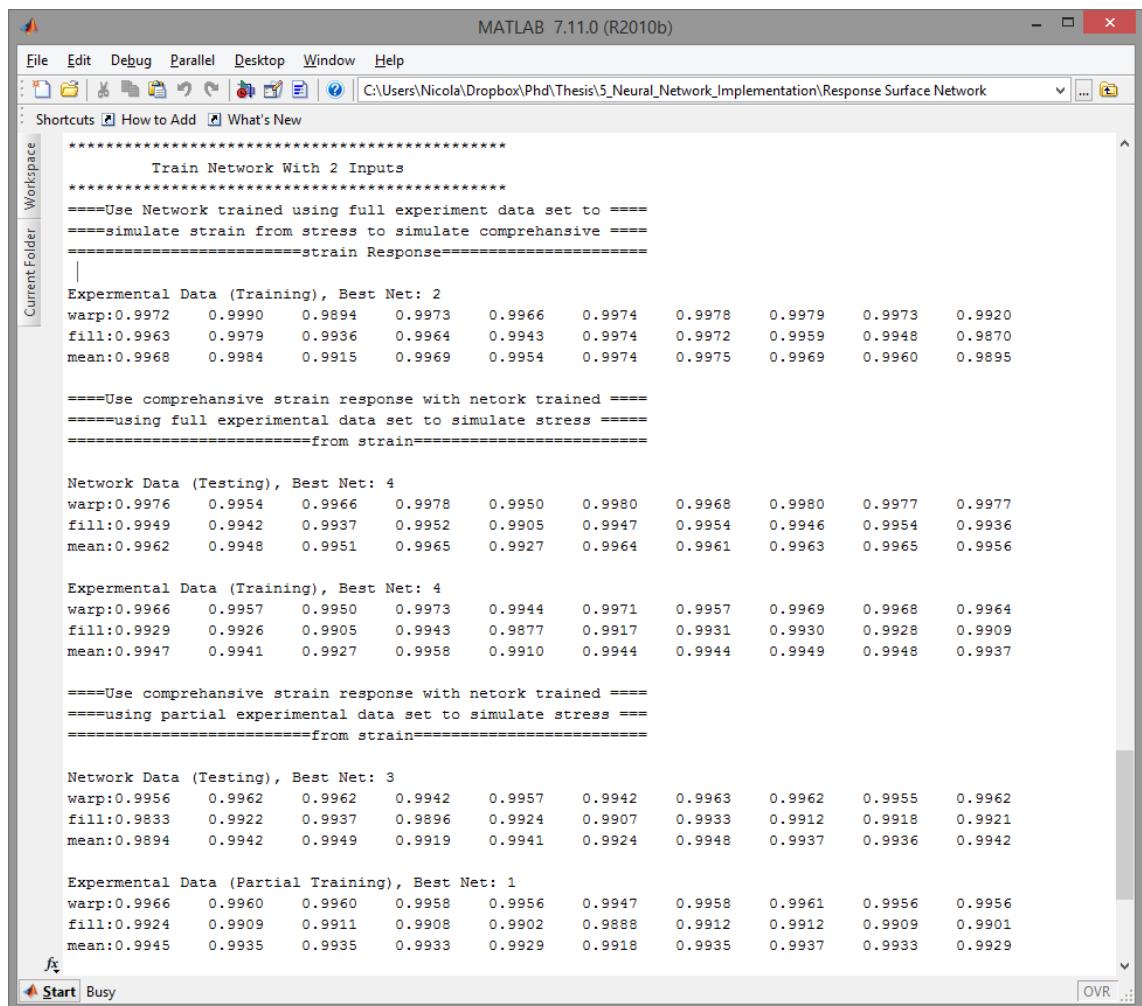
The hidden layer activation function is set by default to "tansig", again any available transfer function may be used.

Random data division and node initiation may be turned on or off.

Training data group sizes are set, these sets are used to implement early stopping.

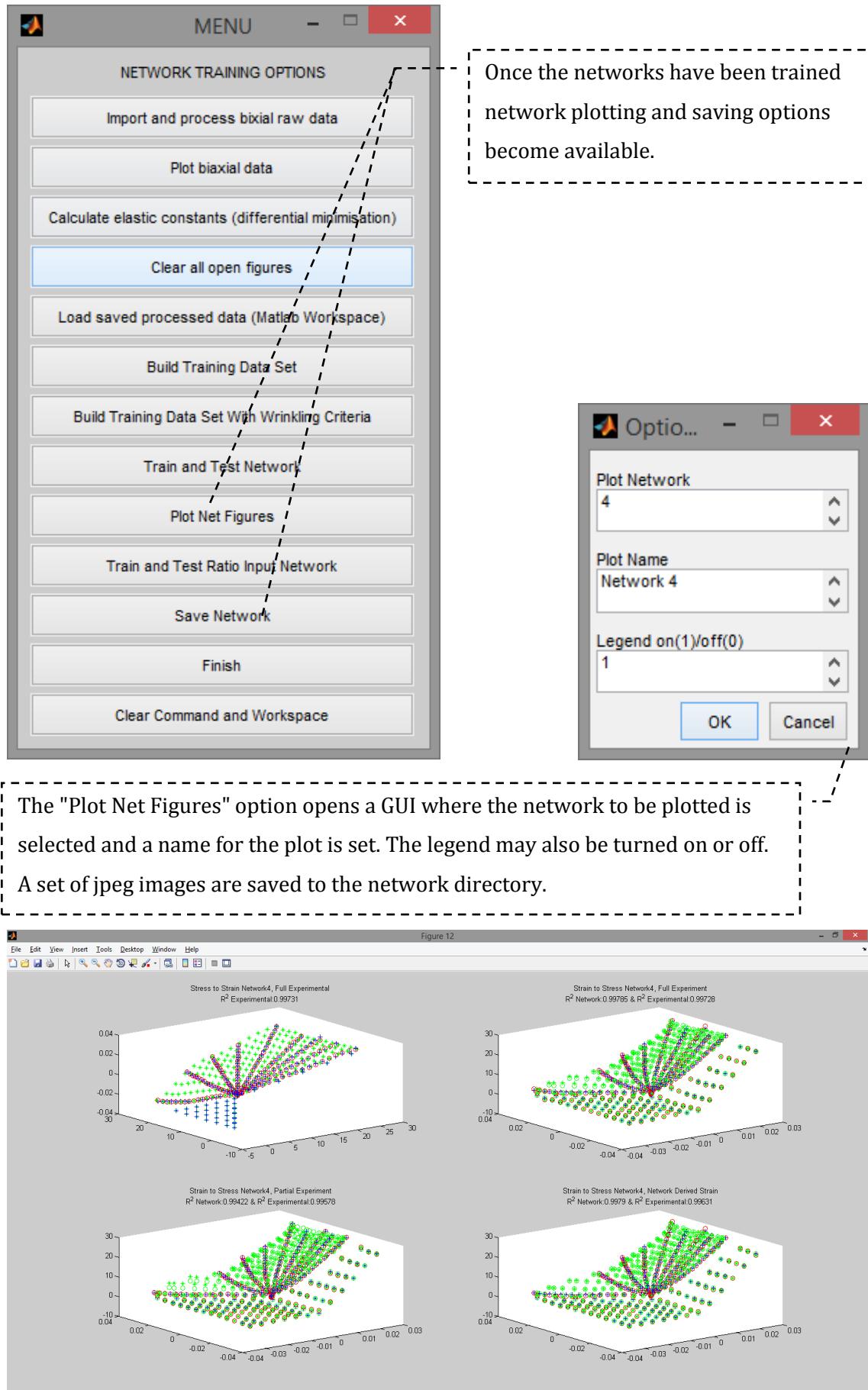


Once training and testing is complete the performance data for each set of networks is output to the command window. These values are also saved to the workspace and may easily be exported.



The screenshot shows the MATLAB 7.11.0 (R2010b) interface with the command window open. The window title is "MATLAB 7.11.0 (R2010b)". The command window displays the following text:

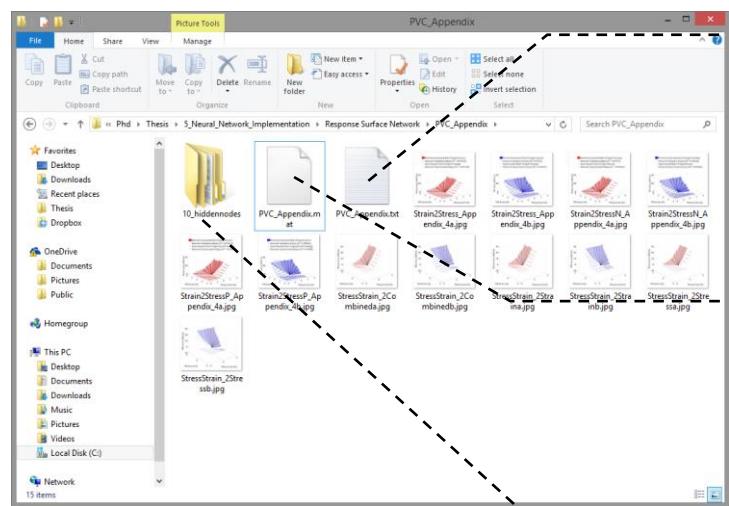
```
*****  
Train Network With 2 Inputs  
*****  
====Use Network trained using full experiment data set to ====  
====simulate strain from stress to simulate comprehensive ====  
=====strain Response=====  
  
Experimental Data (Training), Best Net: 2  
warp:0.9972 0.9990 0.9894 0.9973 0.9966 0.9974 0.9978 0.9979 0.9973 0.9920  
fill:0.9963 0.9979 0.9936 0.9964 0.9943 0.9974 0.9972 0.9959 0.9948 0.9870  
mean:0.9968 0.9984 0.9915 0.9969 0.9954 0.9974 0.9975 0.9969 0.9960 0.9895  
  
====Use comprehensive strain response with netork trained ====  
====using full experimental data set to simulate stress ====  
=====from strain=====  
  
Network Data (Testing), Best Net: 4  
warp:0.9976 0.9954 0.9966 0.9978 0.9950 0.9980 0.9968 0.9980 0.9977 0.9977  
fill:0.9949 0.9942 0.9937 0.9952 0.9905 0.9947 0.9954 0.9946 0.9954 0.9936  
mean:0.9962 0.9948 0.9951 0.9965 0.9927 0.9964 0.9961 0.9963 0.9965 0.9956  
  
Experimental Data (Training), Best Net: 4  
warp:0.9966 0.9957 0.9950 0.9973 0.9944 0.9971 0.9957 0.9969 0.9968 0.9964  
fill:0.9929 0.9926 0.9905 0.9943 0.9877 0.9917 0.9931 0.9930 0.9928 0.9909  
mean:0.9947 0.9941 0.9927 0.9958 0.9910 0.9944 0.9944 0.9949 0.9948 0.9937  
  
====Use comprehensive strain response with netork trained ====  
====using partial experimental data set to simulate stress ====  
=====from strain=====  
  
Network Data (Testing), Best Net: 3  
warp:0.9956 0.9962 0.9962 0.9942 0.9957 0.9942 0.9963 0.9962 0.9955 0.9962  
fill:0.9833 0.9922 0.9937 0.9896 0.9924 0.9907 0.9933 0.9912 0.9918 0.9921  
mean:0.9894 0.9942 0.9949 0.9919 0.9941 0.9924 0.9948 0.9937 0.9936 0.9942  
  
Experimental Data (Partial Training), Best Net: 1  
warp:0.9966 0.9960 0.9960 0.9958 0.9956 0.9947 0.9958 0.9961 0.9956 0.9956  
fill:0.9924 0.9909 0.9911 0.9908 0.9902 0.9888 0.9912 0.9912 0.9909 0.9901  
mean:0.9945 0.9935 0.9935 0.9933 0.9929 0.9918 0.9935 0.9937 0.9933 0.9929
```



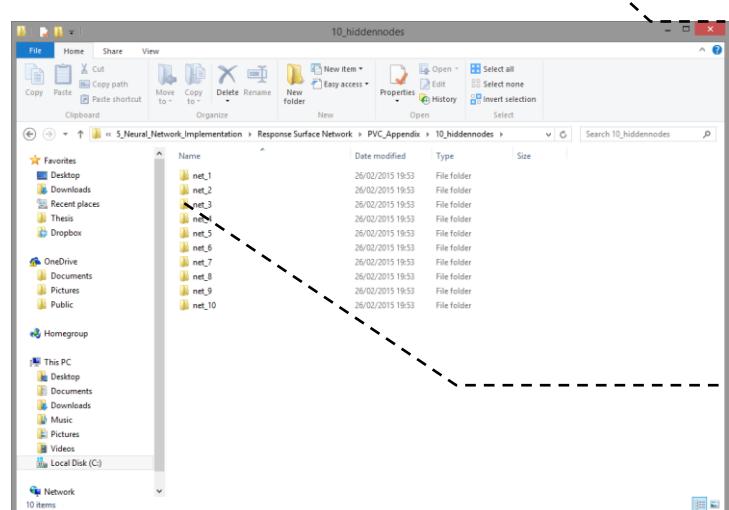
The "Save Network" option opens a GUI where the network set to be saved may be selected. The networks are saved in the automatically generated directory containing all the figures produced during the training process.



A .txt output from the command window is generated as a record of the training procedure.



The contents of the workspace is saved, this enables training to re-run or additional networks to be trained.



The networks are saved in a directory named for the number of hidden nodes allowing for sets with different hidden layer sizes.

Each network is saved in a separate directory.

The screenshot shows a Microsoft Excel spreadsheet titled "net.csv - Microsoft Excel". The data is organized into columns A through M. Column A contains row numbers from 1 to 35. Columns B through M contain numerical values representing weights, biases, and scaling factors. The data includes both positive and negative values, such as -2.5179 in cell A1 and 5.5335 in cell B1.

|    | A        | B        | C        | D        | E        | F        | G        | H        | I        | J       | K | L | M |
|----|----------|----------|----------|----------|----------|----------|----------|----------|----------|---------|---|---|---|
| 1  | -2.5179  | 5.5335   |          |          |          |          |          |          |          |         |   |   |   |
| 2  | -1.559   | -2.2763  |          |          |          |          |          |          |          |         |   |   |   |
| 3  | -2.6131  | 0.4089   |          |          |          |          |          |          |          |         |   |   |   |
| 4  | 1.0478   | 1.8348   |          |          |          |          |          |          |          |         |   |   |   |
| 5  | -4.6909  | -0.44012 |          |          |          |          |          |          |          |         |   |   |   |
| 6  | -6.0418  | -6.7712  |          |          |          |          |          |          |          |         |   |   |   |
| 7  | 0.86266  | -2.4217  |          |          |          |          |          |          |          |         |   |   |   |
| 8  | 1.737    | 2.4714   |          |          |          |          |          |          |          |         |   |   |   |
| 9  | 0.20114  | 10.049   |          |          |          |          |          |          |          |         |   |   |   |
| 10 | -3.9391  | -2.1582  |          |          |          |          |          |          |          |         |   |   |   |
| 11 | 0.83578  |          |          |          |          |          |          |          |          |         |   |   |   |
| 12 | 1.2049   |          |          |          |          |          |          |          |          |         |   |   |   |
| 13 | 1.9487   |          |          |          |          |          |          |          |          |         |   |   |   |
| 14 | -0.75806 |          |          |          |          |          |          |          |          |         |   |   |   |
| 15 | 1.0233   |          |          |          |          |          |          |          |          |         |   |   |   |
| 16 | 0.54754  |          |          |          |          |          |          |          |          |         |   |   |   |
| 17 | 1.6304   |          |          |          |          |          |          |          |          |         |   |   |   |
| 18 | -0.03731 |          |          |          |          |          |          |          |          |         |   |   |   |
| 19 | 3.7867   |          |          |          |          |          |          |          |          |         |   |   |   |
| 20 | -4.0653  |          |          |          |          |          |          |          |          |         |   |   |   |
| 21 | 0.0474   | -0.33723 | -0.67302 | -0.12714 | -0.28815 | -0.1584  | -0.01895 | -0.05518 | -0.03341 | 0.00387 |   |   |   |
| 22 | 0.03367  | 1.7276   | 0.08616  | 3.422    | -0.05065 | -0.16911 | -0.25589 | -0.75997 | -0.08951 | 0.01542 |   |   |   |
| 23 | 0.29071  |          |          |          |          |          |          |          |          |         |   |   |   |
| 24 | 0.237    |          |          |          |          |          |          |          |          |         |   |   |   |
| 25 | 0.02679  |          |          |          |          |          |          |          |          |         |   |   |   |
| 26 | 0.03094  |          |          |          |          |          |          |          |          |         |   |   |   |
| 27 | -0.03085 |          |          |          |          |          |          |          |          |         |   |   |   |
| 28 | -0.03129 |          |          |          |          |          |          |          |          |         |   |   |   |
| 29 | 27.959   |          |          |          |          |          |          |          |          |         |   |   |   |
| 30 | 27.974   |          |          |          |          |          |          |          |          |         |   |   |   |
| 31 | -0.03085 |          |          |          |          |          |          |          |          |         |   |   |   |
| 32 | -0.03129 |          |          |          |          |          |          |          |          |         |   |   |   |
| 33 |          |          |          |          |          |          |          |          |          |         |   |   |   |
| 34 |          |          |          |          |          |          |          |          |          |         |   |   |   |
| 35 |          |          |          |          |          |          |          |          |          |         |   |   |   |

The final output is a .csv containing all of the weights, biases and scaling data allowing the network to be implemented in the Fortran FE code.

