NOTE TO USERS

This reproduction is the best copy available.

UMI[®]

APPLICATION OF ARTIFICIAL NEURAL NETWORKS IN VIBRATION-BASED DAMAGE IDENTIFICATION

By

Hongpo Xu

B. Eng., M. Eng., Ocean University of China, Qingdao, China

A thesis submitted to the
Faculty of Graduate Studies and Research
In partial fulfillment of the requirements for the degree of

Doctor of Philosophy

In
Engineering

Carleton University, Ottawa, Canada
September 2005

The Doctor of Philosophy in Civil Engineering is a joint program with the University of
Ottawa, administered by the Ottawa-Carleton Institute for Civil Engineering.

© Copyright, Hongpo Xu, 2005

Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-494-13404-6

Our file Notre référence

ISBN: 0-494-13404-6

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

Early detection of structural damage has recently received much attention because of the promise that it could lead to economies in the maintenance of structure and provide warning of a later catastrophic structural collapse and the serious consequences that may follow. Researchers in the field have attempted to identify physical or vibration characteristics of the structure that could provide reliable indications of the health of a structure. The traditional methods of damage detection include visual inspection or instrumental evaluation. A comparatively recent development in the health monitoring of civil engineering structures is vibration-based damage detection. It offers several advantages. However, in practice there exist a number of challenges in vibration-based damage identification. In fact, most of the available damage identification algorithms fail when applied to practical civil engineering structures.

In this thesis, a new vibration based structural damage identification algorithm is proposed that may be able to overcome some of the difficulties. The proposed method combines structural dynamics and artificial neural network techniques. It splits the problem into two distinct sub-problems. In the first step, the damage location is determined by using the strain energy technique. In the second step the damage magnitude is evaluated by applying artificial neural network technique. It is expected that the proposed method will be both practical and effective.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to my supervisor, Professor J. L. Humar, for his constant encouragement, for having shown immense patience in solving all my questions and doubts, and for his guidance during the course of my thesis. His continuous availability along with many open-minded discussions, are greatly appreciated. Thanks also to Prof. Humar for the financial support I received through the graduate research assistantship. I would also like to thank Dr. H. Khoo and Dr. D. Lau for serving on my advisory committee

Many other persons have supported me in many different ways to make this thesis possible; I am truly grateful for their encouragement and help, and would like to thank them all.

I am especially thankful to Dr. A. Bagchi for providing many useful suggestions, for having many fruitful discussions, and for sharing his expertise and friendship.

Finally, I wish to thank my wife, Yanjun, for her love and support during my Graduate studies. Her understanding and sacrifice have made this thesis a reality. Very special thanks to my daughter, Anran Xu, for the joyfulness she brought to us during my thesis work.

Table of Contents

	Page
Abstract.....	ii
Acknowledgments.....	iii
Table of Contents.....	iv
List of Tables.....	ix
List of Figures.....	x
Notations.....	xxxi
Chapter 1 Introduction	1
1.1 General.....	1
1.2 Review of literature on vibration based damage detection algorithms	4
1.2.1 Laboratory and field testing.....	5
1.2.2 Vibration based damage identification algorithms.....	8
1.2.3 Challenges associated with vibration based damage detection....	17
1.3 Review of literature on Artificial Neural Network applications in damage detection.....	22
1.4 The proposed approach for damage detection.....	30
1.5 Objectives of the present study.....	33
1.6 Layout of the thesis.....	34

Chapter 2 Vibration Based Damage Identification Techniques 36

2.1 Introduction.....	36
2.2 Eigenvalue equation.....	36
2.3 Algorithms for vibration based damage detection.....	38
2.3.1 Methods based on changes in truncated frequencies.....	40
2.3.2 Methods based on mode shape changes.....	40
2.3.3 Mode shape curvature method.....	41
2.3.4 Methods based on change in flexibility matrix.....	45
2.3.5 Methods based on changes in uniform flexibility shape curvature.....	47
2.3.6 Strain energy method.....	49
2.3.7 Method based on modal residual vector.....	55
2.3.8 Matrix update methods.....	60
2.4 The proposed approach.....	66

Chapter 3 Application of Artificial Neural Networks in Damage

Detection 94

3.1 Introduction.....	94
3.2 Artificial neural network architectures.....	95
3.2.1 The weights and biases.....	97

3.2.2 The transfer function.....	97
3.2.3 Hidden layer.....	100
3.2.4 Layer neurons.....	100
3.3 Artificial neural network learning algorithms.....	101
3.3.1 Perceptron neural network and learning rule.....	102
3.3.2 Backpropagation neural network and training algorithm.....	104
3.3.3 Generalization of the trained neural network.....	111
3.4 The design of the proposed artificial neural network.....	112
3.4.1 Creating the training samples.....	112
3.4.2 Pre-processing.....	115
3.4.3 Choosing the proper learning algorithm.....	115
3.4.4 Determining the network architecture.....	116
 Chapter 4 Case study--Damage Detection in a Girder Bridge	 124
4.1 Introduction.....	124
4.2 Modeling of the Crowchild Bridge.....	125
4.3 Modal Reduction and Expansion Techniques.....	128
4.3.1 Static condensation.....	130
4.3.2 System equivalent reduction expansion process (SEREP).....	132
4.4 Localization of the simulated damage in Crowchild Bridge.....	134

4.4.1 Use of damage indices based on expansion of incomplete mode shapes.....	134
4.4.2 Damage indices based on curvatures obtained from translational measurements.....	135
4.4.3 Damage indices based on curvatures obtained from strain measurements.....	137
4.4.4 Substructure method.....	139
4.4.5 Application of substructure method to Crowchild Bridge model.....	141
4.4.6 Perceptron neural network method.....	145
4.5 Evaluation of the simulated damage extent in Crowchild Bridge.....	150
4.5.1 One-element damage case.....	150
4.5.2 Two-element damage case, damage index vectors based on curvature obtained from translational mode shapes.....	154
4.5.3 Two-element damage case, damage index vectors based on curvature obtained from strain measurements.....	163

Chapter 5 Cast study – Damage Detection in a Space Truss 248

5.1 Introduction.....	248
5.2 Modeling of the 3D space truss.....	249
5.3 Localization of the simulated damage.....	250

5.3.1 Damage index based structural health monitoring.....	250
5.3.2 Sub-structuring method.....	254
5.3.3 Perceptron neural network method for locating damage.....	256
5.4 Quantifying the simulated damage extent in the 3D space truss.....	260
5.4.1 One-element damage case.....	262
5.4.2 Two-element damage case.....	264
 Chapter 6 Summary and Conclusions	 337
6.1 Summary.....	337
6.2 Conclusions.....	339
6.3 Work needed to be done in the future research.....	343
 References	 345
 Appendix A	 355
 Appendix B	 410

List of Tables

	Page
Table 4.1: Measured and analytically obtained frequencies of vibration modes.....	170
Table 4.2: Material properties of the girder model.....	170
Table 4.3: Sectional properties of the girder model.....	170
Table 4.4: Training of neural networks for damage extent in the Crowchild Bridge	
(a) Damage index vectors based on curvature from translational mode shapes...	171
(b) Damage index vectors based on curvature mode shapes from strain measurements.....	172
Table 5.1: Properties of the space truss components.....	275
Table 5.2: 3D truss element connectivity	275
Table 5.3: Training of neural networks for damage extent in the 3D space truss.....	277

List of Figures

	Page
Figure 2.1: Finite element model of a beam.....	69
Figure 2.2: Translational mode shapes for the exact mode No. 1 of healthy and damaged beam, 50% damage in element No. 5.....	70
Figure 2.3: Modal curvatures for the exact mode No. 1 of healthy and damaged beam, 20% damage in element Nos. 5 and 12 simultaneously, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements.....	71
Figure 2.4: Difference in the modal curvatures of damaged and healthy beam, exact mode No. 1, 20% damage in element Nos. 5 and 12 simultaneously, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements.....	72
Figure 2.5: Modal curvatures for exact mode No. 6 of healthy and damaged beam, 20% damage in element Nos. 5 and 12 simultaneously, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements.....	73
Figure 2.6: Difference in the modal curvatures of damaged and healthy beam for exact mode No. 6, 20% damage in element Nos. 5 and 12 simultaneously, (a)	

modal curvatures from translational mode shape, (b) modal curvatures from strain measurements.....74

Figure 2.7: Difference in the modal curvatures of damaged and healthy beam for mode No. 1 with 2% measurement error, 50% damage in element Nos. 5 and 12 simultaneously, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements.....75

Figure 2.8: Difference in the modal curvatures of damaged and healthy beam for mode No. 6 with 2% measurement error, 50% damage in element Nos. 5 and 12 simultaneously, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements.....76

Figure 2.9: Maximum differences in the columns of flexibility matrices of damaged and healthy beam, 70% damage in element No. 5; accurate mode shapes.....77

Figure 2.10: Maximum differences in the columns of flexibility matrices of damaged and healthy beam, 70% damage in element No. 5; 2% error in mode shapes.....77

Figure 2.11: Maximum differences in the columns of flexibility matrices of damaged and healthy beam, 70% damage in element Nos. 5 and 12; accurate mode shapes.....78

Figure 2.12: Maximum differences in the columns of flexibility matrices of damaged and healthy beam, 70% damage in element Nos. 5 and 12; 2% error in mode

shapes.....	78
Figure 2.13: Differences in uniform flexibility curvatures of healthy and damaged beams with 50% damage in each of Element Nos. 5 and 12; accurate mode shapes.....	79
Figure 2.14: Differences in uniform flexibility curvatures of healthy and damaged beams with 50% damage in each of Element Nos. 5 and 12; 2% error in mode shapes.....	79
Figure 2.15: Damage indices calculated from 6 modes of the beam, damage 20% in Element Nos. 5 and 12, accurate mode shapes, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements.	80
Figure 2.16: Damage indices calculated from 6 modes of the beam, damage 20% in Element Nos. 5 and 12, errors 2% in mode shapes, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements.....	81
Figure 2.17: Damage factors calculated from 6 modes of the beam, damage 20% in Element Nos. 5 and 12, accurate mode shapes, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements.	82
Figure 2.18: Damage factors calculated from 6 modes of the beam, damage 20% in Element Nos. 5 and 12, errors 2% in mode shapes, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain	

measurements.....	83
Figure 2.19: Damage indices calculated from 6 modes of the beam, damage 50% in	
Element Nos. 5 and 12, errors 2% in mode shapes, (a) modal curvatures	
from translational mode shape, (b) modal curvatures from strain	
measurements.....	84
Figure 2.20: Damage indices calculated from 6 modes of the beam, damage 70% in	
Element Nos. 5 and 12, errors 2% in mode shapes, modal curvatures from	
translational mode shape.....	85
Figure 2.21: Damage factors calculated from 6 modes of the beam, damage 70% in	
Element Nos. 5 and 12, errors 2% in mode shapes, modal curvatures from	
translational mode shape.....	85
Figure 2.22: Damage indices calculated from six modes of the beam; damage 20% in	
Element No. 5 and 70% in Element No. 12, accurate mode shapes, (a) modal	
curvatures from translational mode shape, (b) modal curvatures from strain	
measurements.....	86
Figure 2.23: Damage indices calculated from six modes of the beam; damage 20% in	
Element No. 5 and 70% in Element No. 12, 2% error in mode shapes, (a)	
modal curvatures from translational mode shape, (b) modal curvatures from	
strain measurements.....	87
Figure 2.24: Absolute sums of modal residuals corresponding to degrees of freedom	

from six modes of a beam; complete and accurate modes, damage 20% in element No. 5 and 30% in element No. 12.....	88
Figure 2.25: Absolute sums of the modal residuals of 6 beam modes, accurate but incomplete modes; damage 20% in element No. 5 and 30% in element No. 12.....	88
Figure 2.26: Absolute sums of modal residuals from six beam modes; incomplete modes with measurement errors; damage 20% in Element No. 5 and 30% in element No. 12, 2% error in mode shapes.....	89
Figure 2.27: Perturbations in the stiffness matrix obtained from six complete and accurate modes of a beam, damage 30% in element 5 and 20% in element 12.....	89
Figure 2.28: Perturbations in the stiffness matrix obtained from six incomplete modes of the beam, 2% random errors in mode shapes, damage 30% in element 5 and 20% in element 12.....	90
Figure 2.29: Damage factors for a beam; damage 30% in Element No. 5 and 20% in Element No. 12; six full and exact mode shapes used in the non-negative least squares optimization.....	90
Figure 2.30: Damage factors for the beam; damage 30% in Element No. 5 and 20% in Element No. 12; six exact but incomplete mode shapes used in the non-negative least square optimization.....	91

Figure 2.31: Damage factors for the beam; damage 30% in element No. 5 and 20% in element No. 12; six complete mode shapes with 2% error used in the non-negative least square optimization.....	91
Figure 2.32: Damage factors for the beam; damage 30% in element No. 5 and 20% in element No. 12; six incomplete mode shapes used in the non-negative least squares optimization, mode shape errors 2% random.....	92
Figure 2.33: Damage factors for a beam; damage 30% in Element No. 5 and 20% in Element No. 12; six exact but incomplete mode shapes used in the optimization, constraint Eq. 37.....	92
Figure 2.34: Damage factors for a beam; damage 30% in Element No. 5 and 20% in Element No. 12; six exact but incomplete mode shapes used in the optimization, constraint Eq. 40.....	93
Figure 2.35: Damage factors for a beam; damage 20% in Element No. 5 and 30% in Element No. 12; six incomplete mode shapes used in the optimization, mode shape errors 2% random.....	93
Figure 3.1: Biological neurons.....	118
Figure 3.2: Three layer artificial neural network.....	119
Figure 3.3: Three layer artificial neural network, abbreviated notation.....	120
Figure 3.4: Hard limit transfer function.....	121
Figure 3.5: Pure linear transfer function.....	121

Figure 3.6: Log-sigmoid transfer function.....	122
Figure 3.7: Tan-sigmoid transfer function.....	122
Figure 3.8: Architecture of perceptron neural network.....	123
Figure 3.9: An example of back-propagation artificial neural network.....	123
Figure 4.1: Crowchild Bridge – West elevation.....	173
Figure 4.2: Cross section of the Crowchild Bridge.....	174
Figure 4.3: Two-dimensional beam element (3 DOFs per node).....	175
Figure 4.4: Girder model of the Crowchild Bridge.....	175
Figure 4.5: Damage indices calculated from three modes of the girder model of the Crowchild Bridge, incomplete but accurate mode shapes expanded by SEREP, (a) damage 50% in element number 3, (b) damage 50% in each of element numbers 3 and 8.....	176
Figure 4.6: Damage indices based on curvatures from translational components of three modes of the girder model of Crowchild Bridge, 30% damage in element 3, (a) accurate modes, (b) modes with $\pm 2\%$ random error, (c) modes with $\pm$ 5% random error.....	178
Figure 4.7: Damage indices based on curvature from translational component of three mode shapes of the girder model, damage 50% in element number 3, (a) modes with $\pm 2\%$ random error, (b) modes with $\pm 5\%$ random error.....	179

Figure 4.8: Damage indices based on curvature from translational component of three mode shapes of the girder model, damage 30% in element number 8, (a) accurate modes, (b) modes with $\pm 2\%$ random errors, (c) modes with $\pm 5\%$ random errors.....181

Figure 4.9: Damage indices based on curvature from translational component of three mode shapes of the girder model, damage 50% in element number 8, (a) mode shapes with $\pm 2\%$ random errors, (b) mode shapes with $\pm 5\%$ random errors.....182

Figure 4.10: Damage indices based on curvature from translational component of three mode shapes of the girder model, damage 30% in each of element numbers 3 and 8, (a) accurate modes, (b) modes with $\pm 2\%$ random error, (c) modes with $\pm 5\%$ random error.....184

Figure 4.11: Damage indices based on curvature from translational component of three mode shapes of the girder model, damage 50% in each of element numbers 3 and 8, (a) modes with $\pm 2\%$ random error, (b) modes with $\pm 5\%$ random error.....185

Figure 4.12: Damage indices calculated from curvature based on strain measurement of three mode shapes of the girder model, damage 30% in element number 3, (a) modes with $\pm 2\%$ random error, (b) modes with $\pm 5\%$ random error.....186

Figure 4.13: Damage indices calculated from curvature based on strain measurement of three mode shapes of the girder model, damage 50% in element number 3, (a) modes with $\pm 2\%$ random error, (b) modes with $\pm 5\%$ random error.....	187
---	-----

Figure 4.14: Damage indices calculated from curvature based on strain measurements of three mode shapes of the girder model, damage 30% in element number 8, (a) modes with $\pm 2\%$ random error, (b) modes with $\pm 5\%$ random error.....	188
--	-----

Figure 4.15: Damage indices calculated from curvature based on strain measurement of three mode shapes of the girder model, damage 50% in element number 8, (a) curvature mode shapes with $\pm 2\%$ error, (b) curvature mode shapes with $\pm 5\%$ error.....	189
---	-----

Figure 4.16: Damage indices calculated from curvature based on strain measurements of three lowest mode shapes of the girder model, damage 30% in each of element numbers 3 and 8 simultaneously, (a) curvature mode shapes with $\pm 2\%$ error, (b) curvature mode shapes with $\pm 5\%$ error.....	190
---	-----

Figure 4.17: The 1 st step of substructure method -- Damage indices calculated from six curvature modes of the 60m beam, damage 20% and 30% in original element numbers 14 and 36 respectively, (a) curvature mode shapes with $\pm 2\%$ error, (b) curvature mode shapes with $\pm 5\%$ error.....	191
--	-----

Figure 4.18: Substructure 2 of 60m beam.....	192
Figure 4.19: The 2 nd step of substructure method -- Damage indices calculated from six curvature modes of the 60m beam, damage 20% and 30% in original element numbers 14 and 36 respectively, (a) curvature mode shapes with $\pm$ 2% error, (b) curvature mode shapes with $\pm 5\%$ error.....	193
Figure 4.20: Substructure 1 of girder model of the Crowchild Bridge for translational measurement.....	194
Figure 4.21: The 1 st step of substructure method -- Damage indices calculated from three translational modes of the girder model of the Crowchild Bridge, damage 30% in original element number 3, (a) accurate modes, (b) modes with $\pm 2\%$ error, (c) modes with $\pm 5\%$ error.....	196
Figure 4.22: The 1 st step of substructure method -- Damage indices calculated from three translational modes of the girder model of the Crowchild Bridge, damage 50% in original element number 3, (a) modes with $\pm 2\%$ error, (b) modes with $\pm 5\%$ error.....	197
Figure 4.23: Substructure 2 of girder model of the Crowchild Bridge for translational measurements.....	198
Figure 4.24: The 2 nd step of substructure method -- Damage indices calculated from three translational modes of the girder model of the Crowchild Bridge, damage 30% in original element number 3, (a) accurate modes, (b) modes	

with $\pm 2\%$ error, (c) modes with $\pm 5\%$ error.....200

Figure 4.25: The 2nd step of substructure method -- Damage indices calculated from

three translational modes of the girder model of the Crowchild Bridge,
damage 50% in original element number 3, (a) modes with $\pm 2\%$ error, (b)
modes with $\pm 5\%$ error.....201

Figure 4.26: The 1st step of substructure method -- Damage indices calculated from

three translational modes of the girder model of the Crowchild Bridge,
damage 30% in original element number 3 and 8, (a) accurate modes, (b)
modes with $\pm 2\%$ error, (c) modes with $\pm 5\%$ error.....203

Figure 4.27: The 1st step of substructure method -- Damage indices calculated from

three translational modes of the girder model of the Crowchild Bridge,
damage 50% in original element number 3 and 8, (a) modes with $\pm 2\%$ error,
(b) modes with $\pm 5\%$ error.....204

Figure 4.28: The 2nd step of substructure method -- Damage indices calculated from

three translational modes of the girder model of the Crowchild Bridge,
damage 30% in original element number 3 and 8, (a) accurate mode shapes,
(b) modes with $\pm 2\%$ error, (c) modes with $\pm 5\%$ error.....206

Figure 4.29: The 2nd step of substructure method -- Damage indices calculated from

three translational modes of the girder model of the Crowchild Bridge,
damage 50% in original element number 3 and 8, (a) modes with $\pm 2\%$ error,

(b) modes with $\pm 5\%$ error.....	207
Figure 4.30: Substructure 1 of girder model of the Crowchild Bridge for strain measurements.....	208
Figure 4.31: The 1 st step of substructure method -- Damage indices based on three curvature mode shapes of the girder model from the strain measurements, damage 30% in element number 3, (a) modes with $\pm 2\%$ error, (b) modes with $\pm 5\%$ error.....	209
Figure 4.32: The 1 st step of substructure method -- Damage indices based on three curvature mode shapes of the girder model from strain measurements, damage 50% in element number 3, (a) modes with $\pm 2\%$ error, (b) modes with $\pm 5\%$ error.....	210
Figure 4.33: Substructure 2 of girder model of the Crowchild Bridge for strain measurements.....	211
Figure 4.34: The 2 nd step of substructure method -- Damage indices based on three curvature mode shapes of the girder model from the strain measurements, damage 30% in element number 3, (a) modes with $\pm 2\%$ error, (b) modes with $\pm 5\%$ error.....	212
Figure 4.35: The 1 st step of substructure method -- Damage indices based on three curvature mode shapes of the girder model from the strain measurements, damage 30% in element number 3 and 8 simultaneously, (a) modes with $\pm$	

2% error, (b) modes with $\pm 5\%$ error.....	213
Figure 4.36: The 2 nd step of substructure method -- Damage indices based on three curvature mode shapes of the girder model from the strain measurements, damage 30% in element numbers 3 and 8 simultaneously, (a) modes with $\pm 2\%$ error, (b) modes with $\pm 5\%$ error.....	214
Figure 4.37: The perceptron neural network for damage localization.....	215
Figure 4.38: Prediction of damage location using perceptron network, damage in element No. 1, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.....	218
Figure 4.39: Prediction of damage location using perceptron network, damage in element No. 3, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.....	221
Figure 4.40: Prediction of damage location using perceptron network, damage in element No. 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.....	224
Figure 4.41: The designed artificial neural network for the evaluation of damage extent in Crowchild Bridge.....	225

Figure 4.42: Training and validation errors of the network for the damage factor of element No. 3 trained using damage index vectors from translational mode shapes both with and without error, one-element damage case, (a) stage 1, (b) stage 2, (c) stage 3.....	227
Figure 4.43: Prediction of damage extent using backpropagation network, damage in element No. 3, one-element damage case, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.....	230
Figure 4.44: Training and validation errors of the network for the damage factor of element No. 8 trained using damage index vectors from translational mode shapes both with and without error, one-element damage case, (a) stage 1, (b) stage 2, (c) stage 3.....	232
Figure 4.45: Prediction of damage extent using backpropagation network, damage in element No. 8, one-element damage case, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.....	235
Figure 4.46: Training and validation errors of the network for the damage factor of element No. 3 trained using damage index vectors from translational mode	

shapes both with and without error and the resilient backpropagation
training algorithm (a) stage 1, (b) stage 2, (c) stage 3.....237

Figure 4.47: Training and validation errors of the network for the damage factor of
element No. 8 trained using damage index vectors from translational mode
shapes both with and without error and the resilient backpropagation
training algorithm (a) stage 1, (b) stage 2, (c) stage 3.....239

Figure 4.48: Prediction of damage extent using backpropagation network of
architecture 15-30-1 and smaller training data set, damage index vectors
based on curvature obtained from translational mode shapes, damage in
element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage
between 10% and 20%, (c) damage between 20% and 30%, (d) damage
between 30% and 40%, (e) damage between 40% and 50%.....242

Figure 4.49: Training process of the network with architecture of 15-30-30-2 for the
damage factor of elements Nos. 3 and 8 trained using damage index vectors
based on the curvature from the rotational component of incomplete mode
shapes both with and without error, resilient backpropagation training
algorithm, (a) stage 1, (b) stage 2, (c) stage 3.....244

Figure 4.50: Prediction of damage extent using backpropagation network with
architecture of 15-30-30-2, damage in element Nos. 3 and 8, logistic
sigmoid transfer function used in the output layer, damage index vectors

based on the curvature obtained from strain measurements, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.....	247
Figure 5.1: Details of the Meroform strut and nodes.....	278
Figure 5.2: Finite element model of the eight-bay space truss.....	278
Figure 5.3: Modal strain energies contributed by the elements of the 3D truss in the first six modes, (a) 1st mode, (b) 2nd mode, (c) 3rd mode, (d) 4th mode, (e) 5th mode, (f) 6th mode, (g) total of six modes.....	282
Figure 5.4: Damage in chord members, 2% error in incomplete mode shapes, (a) 30% damage in chord element No. 1, (b) 30% damage in chord element No. 2, (c) 30% damage in chord element No. 30, (d) 20% damage in each of the chord elements Nos. 18 and 28.....	284
Figure 5.5: Damage 30% in vertical element No. 45, (a) accurate incomplete modes corresponding to 36 x degrees of freedom, (b) accurate incomplete modes corresponding to 36 y degrees of freedom, (c) accurate incomplete modes corresponding to 36 z degrees of freedom, (d) accurate complete modes, (e) incomplete modes corresponding to 36 x degrees of freedom with 2% error (f) incomplete modes corresponding to 36 y degrees of freedom with 2% error, (g) incomplete modes corresponding to 36 z degrees of freedom with	

2% error, (h) complete modes with 2% error.....288

Figure 5.6: Damage 30% in horizontal element No. 63 in z direction, (a) accurate

incomplete modes corresponding to 36 x degrees of freedom, (b) accurate

incomplete modes corresponding to 36 y degrees of freedom, (c) accurate

incomplete modes corresponding to 36 z degrees of freedom, (d) accurate

complete modes, (e) incomplete modes corresponding to 36 x degrees of

freedom with 2% error (f) incomplete modes corresponding to 36 y degrees

of freedom with 2% error, (g) incomplete modes corresponding to 36 z

degrees of freedom with 2% error, (h) complete modes with 2%

error.....292

Figure 5.7: Damage 30% in horizontal element No. 55 in z direction, (a) accurate

incomplete modes corresponding to 36 x degrees of freedom, (b) accurate

incomplete modes corresponding to 36 y degrees of freedom, (c) accurate

incomplete modes corresponding to 36 z degrees of freedom, (d) accurate

complete modes, (e) incomplete modes corresponding to 36 x degrees of

freedom with 2% error, (f) incomplete modes corresponding to 36 y degrees

of freedom with 2% error, (g) incomplete modes corresponding to 36 z

degrees of freedom with 2% error, (h) complete modes with 2%

error.....296

Figure 5.8: Damage 30% in diagonal element No. 74 in x - y plane, (a), (b), (c) accurate

incomplete modes corresponding to 36 x , y , and z degrees of freedom, respectively, (d), (e), (f) accurate incomplete modes corresponding to 72 xy , yz , and zx degrees of freedom, respectively, (g) accurate complete modes, (h), (i), (j) incomplete modes with 2% error corresponding to 36 x , y , and z degrees of freedom, respectively, (k), (l), (m) incomplete modes with 2% error corresponding to 72 xy , yz , and xz degrees of freedom, respectively, (n) complete modes with 2% error.....303

Figure 5.9: Damage 30% in diagonal element No. 107 in y - z plane, accurate modes, (a),

(b), (c) incomplete modes corresponding to 36 x , y , and z degrees of freedom, respectively, (d), (e), (f) incomplete modes corresponding to 72 xy , yz , and zx degrees of freedom, respectively, (g) complete modes....307

Figure 5.10: Damage 30% in diagonal element No. 94 in x - z plane, (a), (b), (c) accurate

incomplete modes corresponding to 36 x , y , and z degrees of freedom, respectively, (d), (e), (f) accurate incomplete modes corresponding to 72 xy , yz , and zx degrees of freedom, respectively, (g) accurate complete modes, (h), (i), (j) incomplete modes with 2% error corresponding to 36 x , y , and z degrees of freedom, respectively, (k), (l), (m) incomplete modes with 2% error corresponding to 72 xy , yz , and xz degrees of freedom, respectively, (n) complete modes with 2% error.....314

Figure 5.11: 30% damage in chord elements No. 18 connecting nodes 7 and 11 and

No. 1 connecting nodes 1 and 5, incomplete mode shapes corresponding to x degrees of freedom with 2% error.....	315
Figure 5.12: 30% damage in chord elements No. 18 connecting nodes 7 and 11 and No. 28 connecting nodes 16 and 20, incomplete mode shapes corresponding to x degrees of freedom with 2% error.....	315
Figure 5.13: 30% damage in elements No. 21 connecting nodes 19 and 23 and No. 31 connecting nodes 28 and 32, incomplete mode shapes corresponding to x degrees of freedom with 2% error.....	316
Figure 5.14: 30% damage in elements No. 18 connecting nodes 7 and 11 and No. 30 connecting nodes 24 and 28, incomplete mode shapes corresponding to x degrees of freedom with 2% error.....	316
Figure 5.15: The first step of the substructure method, 30% damage in original element 18 and 28 (substructures 1 and 2) simultaneously, 6 incomplete modes with, (a) 2% error, (b) 4% error.....	317
Figure 5.16: The second step of the substructure method, 30% damage in elements 10 and 16 (original elements 18 and 28) simultaneously, 6 incomplete modes with, (a) 2% error, (b) 4% error.....	318
Figure 5.17: Damage indices for 30% damage in element 10 connecting nodes 7 & 11, and 30% damage in element 16 connecting nodes 16 & 20 obtained by sub-structuring; 6 modes with, (a) 2% random errors, (b) 4% random	

errors.....	319
Figure 5.18: The perceptron neural network for damage location in 3D space truss...	318
Figure 5.19: Prediction of damage location using perceptron network, damage in element No. 17, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%.....	322
Figure 5.20: Prediction of damage location using perceptron network, damage in element Nos. 17 and 18, (a) damage between 0% and 10%, (b) damage between 20% and 30%, (c) damage between 30% and 40%.....	324
Figure 5.21: The initial backpropagation neural network designed for evaluating damage extent in 3D space truss.....	325
Figure 5.22: Prediction of damage extent using backpropagation network, damage in element No. 18, one-element damage case, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.....	328
Figure 5.23: Training and validation errors of the network with the architecture of 32-64-1 for the damage factor of element No. 18 trained using damage index vectors from incomplete mode shapes both with and without error, (a) stage 1, (b) stage 2, (c) stage 3.....	330
Figure 5.24: Prediction of damage extent using backpropagation network with the	

architecture of 32-64-1, damage in elements 18 and 28, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%, (f) damage between 10% and 20% in element 18 and damage between 40% and 50% in element 28.....333

Figure 5.25: Prediction of damage extent using backpropagation network with the architecture of 32-64-2 and smaller training data set, damage in elements 18 and 28, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.....336

Notations

$\mathbf{a}(k)$	Output of the neural network at iteration k
$\mathbf{a}^n$	Output vector of n th layer of the artificial neural network
a_i^m	Output of i th neuron in the m th layer
A	Cross-sectional area
$\mathbf{b}$	Bias vector
b_i^m	Bias of i th neuron in the m th layer
$\mathbf{B}$	Modal residuals in matrix form
E	Modulus of elasticity or the performance function in back-propagation training algorithm
$\mathbf{F}$	Flexibility matrix of the undamaged structure
$\mathbf{F}_d$	Flexibility matrix of the damaged structure
F_{ij}	Fraction of total strain energy contributed by element j of the healthy structure
F_{ij}^d	Fraction of total strain energy contributed by element j of the damaged structure
$\Delta \mathbf{F}$	Difference between the flexibility matrices of the damaged and the healthy structures
$\mathbf{g}_k$	Gradient of the performance function at point $\mathbf{w}_k$
G	Rigidity modulus

I	2 nd moment of inertia of the section about the x-y plane
I_a	Identity matrix of size a by a
I_p	2 nd polar moment of area about the x axis
J	Torsional constant
k_j	Stiffness matrix of member j of the healthy structure
k_j^d	Stiffness matrix of member j of the damaged structure
K	Stiffness matrix of the healthy structure
K^d	Stiffness matrix of the damaged structure
L	Length of the beam
M	Mass matrix
n_i^m	Net input to the i th neuron in the m th layer
p	Input vector to the artificial neural network
p_k	Search direction
R_i	Modal residual vector for the i th mode
s_i^m	Sensitivity of performance function E to the change of the net in-put to the i th neuron in the layer m
S_{ij}	Strain energy contributed by the j th element of the healthy structure
S_i	Modal strain energy in the i th mode of the healthy structure
S_{ij}^d	Strain energy contributed by the j th element of the damaged structure
S_i^d	Modal strain energy in the i th mode of the damaged structure

$\mathbf{t}(k)$	Corresponding target output at iteration k
$\mathbf{T}$	Transformation matrix.
$\mathbf{U}$	Vector of nodal displacements
w_{ij}^m	Weight from j th neuron to i th neuron in the m th layer
$\mathbf{W}^n$	Weight matrix of n th layer of the artificial neural network
$\mathbf{Y}$	Modal coordinates
y_i	Mode shape lateral displacement at location i along a beam
α	Learning rate which is a positive constant used to determine the length of the step
β	Vector of the unknown changes in element stiffness matrices
γ_{ij}	Damage index for the j th element in mode No. i
δ	Row vector whose j th element is equal to the element with the largest absolute value in the j th column of $\Delta \mathbf{F}$
$\delta \lambda$	Vector of measured eigenvalue changes
λ_1	Eigenvalue (squared natural frequency) of the healthy structure
λ_{di}	i th eigenvalue of the damaged structure
Λ	Diagonal matrix of squared frequencies
ρ	Density
ϕ_i	i th mass-orthonormal mode shape
ϕ_{di}	i th mode shape of the damaged structure

Φ	Modal matrix of the mass-normalized mode shapes ϕ_j
Φ_a	Rectangular modal matrix of size a by m
Φ_d	Modal matrix of the damaged structure
Φ_{da}	Incomplete mode shapes measured in the damaged structure
Ψ_j	j th curvature shape obtained from F
Ψ_{dj}	j th curvature shape obtained from F_d
$\psi''(x_i)$	Curvature at location i along a beam

Chapter 1

Introduction

1.1 General

The process of condition/damage detection and monitoring for aerospace, civil and mechanical engineering infrastructure is commonly referred to as structural health monitoring (SHM). The SHM process involves the observation of a system over time using periodically sampled dynamic or/and static response measurements from an array of sensors, the extraction of damage-sensitive features from these measurements and the analysis of these features to determine the current state of system health (Farrar and Sohn, 2001).

The monitoring of civil structures has received increasing interest in the research community during the last several decades. There have been increased economic and societal demands to periodically monitor the safety of structures against long-term deterioration, and to immediately assess the condition after extreme events such as earthquakes. The condition assessment of civil structures after extreme events is of great importance to the emergency management officials for them to properly allocate resources to prompt emergency response. An equally important task is

continuous/periodic monitoring of civil structures to ensure the safety and adequate performance during the life span of such structures.

Damage refers to changes introduced into a system, either intentionally or unintentionally, which may adversely affect the current or future performance of that system. The concept of damage is not meaningful without a comparison between two different states of the system, one of which is assumed to represent the initial undamaged state (Doebeling et al., 1996).

A generally accepted classification system for damage identification defines four levels of damage identification as follows (Rytter, 1993):

Level 1: Determination that damage is present in the structure

Level 2: Level 1 plus determination of the geometric location of the damage

Level 3: Level 2 plus quantification of the severity of the damage

Level 4: Level 3 plus prediction of the remaining service life of the structure

Damage may be detected by global or local techniques. Local condition evaluation methods include visual inspection or localized instrumental evaluation. The reliability of visual inspection is dependent to a great extent on the experience and knowledge of the field inspector; it requires extensive time in searching for damage and does not always reveal structural damage. In practice, the time span between successive inspections may be long enough for minor structural damage to initiate and evolve into

major damage, and eventually cause collapse of the structure. Another drawback of visual inspection is that all portions of the structure need to be accessible, which may be impractical, particularly when the structure is complex and/or large in size. Certain concealed damage existing inside the structure cannot be detected. Also, visual inspection does not provide a quantitative value for the damage of the structure. Because of these constraints the process of visual inspection is time consuming, labour-intensive, and expensive.

A variety of localized damage detection techniques based on the use of specialized equipment have been proposed to provide detailed information about specific elements of a structure. These include acoustic or ultrasonic methods, magnetic field methods, radiography, X-ray, and eddy current methods. Application of such methods requires *a-priori* knowledge of the possible damage sites and access to such sites. Also, the results of instrumental evaluation are often inconclusive or difficult to evaluate. Instrumental evaluation may at times involve measurement of local stress and strains and accelerations through sensors installed on some critical elements and components. Changes in stress and strain states or accelerations can provide warning of possible damage. However, it may be too expensive or impractical to instrument all elements and components that are possibly critical.

In view of the limitations cited above, vibration based damage detection techniques have been developed for health monitoring of civil structures by measuring global parameters

at a few easily accessible points on the structure.

The basic idea of vibration based damage detection is that vibration characteristics of a structure, namely, its frequencies, mode shapes, and damping are directly affected by the physical characteristics of the structure including its mass and stiffness. Damage reduces the stiffness of the structure and hence changes its vibration characteristics, which are global properties of the structure. Therefore, changes in the dynamic characteristics of a structure can be used to detect, locate, and quantify the structural damage suffered by the structure.

Vibration based damage identification offers several advantages, one of which is that the location of damage need not be known *a-priori*. Often, the sensors required to measure the vibration characteristics need not be located in the vicinity of the damage. In addition, a limited number of sensors can at times provide sufficient information to locate the damage and determine its severity, even in a large and complex structure. Vibration measurements do not need bulky equipment, except when forced vibration tests are carried out.

1.2 Review of literature on vibration based damage detection algorithms

A large volume of research has been carried out in the past several decades on vibration

based damage detection. The research includes laboratory and field testing, development of damage detection algorithms, and study of the challenges associated with vibration based damage identification (VBDI). The literature associated with the various aspects of VBDI is briefly reviewed in the following sections.

1.2.1 Laboratory and field testing

Mazurek and DeWolf (1990) used resonant frequencies and displacement mode shapes to monitor a two-span aluminum plate-girder bridge in the laboratory by conducting ambient vibration tests. They found that support failures and crack propagation could cause changes in certain natural frequencies and displacement mode shapes, although significant degradation was needed to cause a measurable change. They concluded that it was feasible to detect failures well in advance of subsequent entire structural collapse.

Hearn and Testa (1991) tested in the laboratory a welded four-member steel frame to detect the progressive fatigue cracks by applying a perturbation method. They demonstrated the effectiveness of this method and found that changes in natural frequency and damping could be good damage indicators.

Salawu and Williams (1995) used four accelerometers to measure the dynamic response of a six-span, two-lane reinforced concrete highway bridge in England before and after repairs when excited by a hydraulic actuator. They found that structural repairs did not

change significantly the natural frequencies of the bridge, but the modal assurance criterion (MAC) and the coordinate modal assurance criterion (CMAC) values were good indicators of the presence and location of repairs.

Farrar et al. (1994, 1996) conducted extensive testing of damage detection methods on the Alamosa Canyon Bridge in New Mexico to study the variability of various modal parameters and attempted to quantify the statistical significance of parameter changes. The Alamosa Canyon Bridge was out of service, allowing the researchers to perform various detailed testing on the undamaged and damaged structure with controlled damage levels before demolition. Both forced and ambient vibration responses were measured using 26 different kinds of accelerometers. Modal parameters were extracted using linear signal processing techniques and compared with those obtained with a detailed finite element model. The results showed that all methods identified the damage location correctly for a cut completely through the bottom flange. The methods were inconsistent and did not clearly identify the damage location when they were applied to the less severe damage cases.

Farrar and Jauregui (1996) conducted a series of forced vibration tests on the I-40 Bridges over the Rio Grande. The type of bridge tested was made up of a concrete deck supported by two welded-steel plate girders and three steel stringers. The first three modes of the bridge in the undamaged state were obtained by forced excitation tests with coarse sets of accelerometers and random excitation tests with refined sets of

accelerometers. Four different levels of damage were then introduced into the middle span of the north girder to simulate fatigue cracking in the plate-girder. Tests similar to those carried out on the undamaged bridge were repeated after each level of introduced damage. The test results showed that changes in natural frequencies or mode shapes were not sensitive to the damage in a bridge structure. Researchers applied five damage identification methods to the experimental data to try and obtain a Level 2 assessment of damage, that is, to identify that the bridge was damaged and to locate the damage. It was observed that all methods identified the damage location correctly for a cut completely through the bottom flange. However, the methods did not clearly identify the damage location when they were applied to the light damage cases. Results also showed that the damage index method performed the best.

Some collaborative projects on vibration based damage detection have been organized at an international level, such as the BRITE-EURAM project BE96-3157 SIMCES (System Identification to Monitor Civil Engineering Structures), which is one of the EU cooperative research projects. The SIMCES project focused on structural health monitoring of civil engineering structures, mainly on damage detection in bridges by using structural dynamics models (Auweraer et al., 2003). Extensive tests were carried out on the Z-24 Bridge, which is a classical post-tensioned concrete box girder bridge located in Switzerland over-passing the national highway A1 between Bern and Zurich. A large amount of experimental data was obtained after different levels of damage had

been introduced in the bridge. Testing on a real bridge allowed the researchers to better understand the dynamic behaviour under the influence of various environmental conditions, to optimize the dynamic test procedures, to explore the sensitivity and variability of different dynamic parameters, and to verify the robustness of the various vibration based damage detection algorithms.

Additional references related to field tests are provided in the following sections.

1.2.2 Vibration based damage identification algorithms

Research on vibration based damage identification has been rapidly expanding over the last few years, and a variety of vibration based damage identification algorithms have been developed. Doebling et al. (1996) and Shen et al. (1998) have presented an excellent and extensive review of the research in this area. Based on this literature review, a few of commonly used vibration based damage identification algorithms are described in this section.

Method based on frequency changes

The presence of damage or deterioration in a structure causes changes in the natural frequencies of the structure. Frequency measurements can be obtained quite economically and frequency tests can be quickly conducted and are often reliable.

Damage detection using shifts in natural frequencies was therefore the subject of a large volume of the early research. Salawu (1997) presents an excellent review of the use of natural frequency changes for damage identification.

Cawley and Adams (1979) assumed that the change in the natural frequency of a specific mode of a structure due to localized damage is a function of the position vector of the damage and the reduction in stiffness caused by the damage. They concluded that the ratio of the frequency changes in two modes was a function of only the damage location. In the method proposed by Cawley and Adams to locate the damage, theoretical frequency shifts due to damage at selected positions on the structure are calculated and compared with measured values. Positions where the theoretically determined ratio equals the experimentally measured value are therefore possible damage sites. The shortcoming of the method is that erroneous results are obtained for types of damage involving creation or alteration of mode shapes and no indication is given as to the accuracy of the predicted damage location. Application of this method to reinforced concrete and steel beams indicated that at least nine mode shapes should be considered in the computations if the damage is to be located with reasonable accuracy.

Ju and Mimovich (1986) used changes in natural frequencies to successfully locate damage occurring at sections of a beam to within 3% of the length. They found that the accuracy of the damage localization was improved to less than 1% of the length if the built-in end of the experimental beam was represented by a torsional spring.

Based on the works by Cawley and Adams, Stubbs and Osegueda (1990) developed a damage detection method using the sensitivity of modal frequency changes. In their method, they assume that only one member is damaged. An error function for each mode and each structural member is computed. The structural member that minimizes this error is determined to be the damaged member. This method has been demonstrated to produce more accurate results than the method suggested by Cawley and Adams in the case where the number of members is much greater than the number of measured modes.

It should be noted that damage detection by using natural frequency changes has several limitations. The natural frequencies vary proportionally with the square root of the stiffness, thus large stiffness changes are needed before significant frequency changes can be detected. Also, natural frequency changes alone may not be sufficient for a unique determination of the location of structural damage. This is because damage states associated with similar damage extent but at two different locations may cause the same amount of frequency change.

Method based on changes of mode shape and its derivative

There are a number of different ways in which mode shape information can be processed to detect structural damage. Fox (1992) considered experimentally a uniform one metre long, 12 mm deep free-free rectangular steel beam with a defect, which was a simple saw cut whose depth was zero, 1mm, 3mm, or 6 mm, located about 200 mm from one

end. Research results showed that changes in single-number parameters like natural frequency and MAC were reasonable indicators of the presence of damage, but were not in themselves sufficient to locate the damage. Therefore, Fox introduced the concept of Node Line MAC, a MAC based on measurement points close to a node point for a particular mode, based on the idea that changes in mode shape will be most noticeable in the vicinity of node points or nodal lines (Wolff et al., 1989). It was found that the Node Line MAC was a more sensitive indicator of the changes in the mode shape caused by damage. By graphical comparisons of relative changes in mode shapes, the damage location could be detected with reasonable accuracy when only the resonant frequencies and mode shapes were examined.

An alternative to the use of displacement mode shapes to obtain information about sources of vibration changes is to use mode shape derivatives, such as curvature mode shapes. This method, proposed by Pandey et al. (1991), is based on the observation that the damage in beam type structures would reduce the flexural rigidity EI of the beam. Since the curvature of the beam is inversely proportional to the flexural stiffness, EI , a reduction of stiffness EI associated with damage will, in turn, lead to an increase in the magnitude of the curvature at the damaged section. The pre- and post-damage mode shapes are first extracted from an experimental analysis. Curvature of the mode shapes for the undamaged and damaged conditions can then be estimated numerically from the displacement mode shapes using any of the mathematical means of differentiation, such

as central difference approximation. Theoretically, the difference between the pre- and post-damaged curvature mode shapes will be largest in the damaged region. Chance et al. (1994) found that numerically calculating the curvature from measured mode shapes resulted in unacceptable errors. They used measured strains directly instead of measuring curvature mode shape, which improved the results.

Stubbs and Kim (1995) presented an algorithm called Damage Index to identify damage in a linearly elastic beam-type structure. They defined the damage index in terms of the change in the strain energy content in the structure when it deforms in its particular mode shapes. This requires the determination of more complete mode shapes than a limited number of sensors can provide. To determine mode shape values between sensor locations, an interpolation using cubic polynomials can be used. Statistical methods are then used to examine changes in the damage index and to associate these changes with possible damage locations. This algorithm has the advantage that an analytical model need not be developed and only a few lower order modes are enough for locating the damage. Also, it is not necessary to mass normalize the modes and any consistent normalization procedure may be used. The main limitation of the method is that, if the damaged member makes very little contribution to the strain energy of the measured modes, the damage may not be detected.

Methods based on dynamically measured flexibility

Pandey and Biswas (1994) developed a damage detection approach based on changes in the measured flexibility of the structure. The measured flexibility matrix can be estimated from the mass-normalized measured mode shapes and the natural frequencies. In fact, the flexibility matrix can be approximated using the data from only the first few modes of the structure. Each column of the flexibility matrix represents the displacement pattern of the structure associated with a unit force applied at the corresponding degree of freedom. The difference between the flexibility matrices obtained from the undamaged and damaged mode shapes can be used to locate the damage. Pandey and Biswas used three types of beams as examples. A 50% reduction of the modulus of elasticity was imposed in turn on all 32 elements of the three beams. Additional numerical evaluations were carried out with a reduction in the modulus of elasticity over the range of 0.1—0.9 in steps of 0.2 imposed only on element number 20 of each of the three beams. Results showed that estimates of the damage condition and the location of the damage could be obtained from just the first two measured modes of the structure.

Aktan et al. (1994) used measured flexibility as a condition index to evaluate the health of a bridge. They applied this technique to two bridges and analyzed the accuracy of the flexibility obtained from the displacement mode shapes by comparing it with the static deflections measured from a set of truckload tests.

Toksoy and Aktan (1994) tested a continuous three-span reinforced concrete skewed slab

decommissioned bridge through serviceability and damage limit states to failure by simulating a progressively increasing stationary truck load using four hydraulic actuators. In all they conducted four modal tests using vibrations induced by impact at various stages of damage. The measured flexibilities were computed, and then the deflection profiles were obtained. The researchers concluded that anomalies in the deflection profile could indicate damage even when a baseline was not available.

Zhang and Aktan (1995) used the flexibility and its derivative, the curvature of the uniform load surface, which is defined as the deformed shape of the structure when subjected to a uniform load, as indices of the structural status. The authors stated that the absolute difference in the uniform load surface, which is constructed by summing the columns of the measured flexibility matrix, could serve as an indicator of damage both under uniform deterioration and in an initial stage. The absolute difference in uniform load surface curvature, which is calculated from the uniform load flexibilities using the central difference approximation technique, could highlight an individual location where damage started.

He and Ewins (1986) presented the stiffness error matrix, which is a function of the flexibility change in the structure and the undamaged stiffness matrix, as an indicator of errors between measured parameters and analytical stiffness and mass matrices. The stiffness matrix generally provides more information than the mass matrix, so it is more widely used in the error matrix method. Gysin (1986) has demonstrated that the

performance of this method depends on the type of matrix reduction technique used and the number of mode shapes used to calculate the flexibility matrices.

Lim (1990) proposed the unity check method, which defines an error matrix to measure the degree to which the pseudo-inverse relationship between the dynamically measured flexibility matrix and the structural stiffness matrix is satisfied. The location of the entry with maximum magnitude in each column of the error matrix determines the error location. Later, Lim extended the unity check method for application to the problem of damage detection (Lim, 1991). He defined a least-squares problem for the elemental stiffness changes, which were consistent with the unity check error, in potentially damaged members.

Methods based on updating physical property matrices

These methods rely on updating the physical property matrices, usually the stiffness matrix, so that the frequencies and mode shapes of the modified system closely match the measured characteristics of the damaged structure. Different mathematical algorithms are used to solve for the update matrices (or perturbations to the nominal model that produce the updated matrices). Most methods rely on the solution of a non-linear optimization problem based on the structure's equations of motion, properties of the nominal model, and the measured dynamic characteristics. The methods use a common basic set of equations, and the difference in the various algorithms can be

related to the choice made for the following:

1. Objective function to be minimized;
2. Constraints placed on the problem; and
3. Numerical scheme used to implement the optimization.

Most of the objective functions are formulated to minimize the Frobenius norm of the changes in the property matrices or the perturbations to the nominal model that produce the updated matrices. The associated equality and inequality constraints are derived from the measured data and aim to preserve the symmetry, sparsity and connectivity of the stiffness matrix. Constraining the sparsity of the updated matrix to be the same as the original matrix ensures that the updated model generates no new load paths. Finally, a number of alternative numerical schemes are available for solving the optimization problem.

Baruch and Bar Itzhack (1978), Berman and Nagy (1983), and Kabe (1985) have proposed a common formulation of the optimal update problem that is basically a minimization of the Frobenius norm of the global parameter matrix perturbations subject to modal force error and orthonormal constraints. Chen and Garba (1988) have established a method which estimates the parameters of a finite element model capable of providing modal characteristics consistent with those measured in test. Using these estimated parameters, the perturbed mass and/or stiffness matrices are constructed. The

optimization problem reduces to minimizing the norm of the model property perturbations with a zero modal force error constraint and a connectivity constraint. Another approach to this problem, used by Kammer (1988), is to seek a minimization of the modal force error subject to a property matrix symmetry constraint. Smith and Beattie (1991) have extended the formulation of Kabe (1985) to include the preservation of the structural connectivity, and have formulated the problem as minimization of both the perturbation matrix norm and the modal force error norm subject to the symmetry and connectivity constraint.

1.2.3 Challenges associated with vibration based damage detection

Many of the existing damage detection and monitoring algorithms (Smith, 1988, Hemez, 1993, Kaouk et al., 1994) were originally developed in the field of aerospace industry for the monitoring of space-station-like structures such as truss structures. These algorithms do not fully address the issues that arise in the monitoring of civil structures. The main challenges for the development of robust damage detection and monitoring system for civil structures are as follows:

1. Effect of Factors Other than Damage

Global vibration characteristics are often affected by a significant amount of uncertainty produced by factors other than damage, which may prevent the accurate

detection of small levels of damage. These factors include test conditions and environmental effects, such as change of mass caused by humidity, water waves and snow accumulation. For example, an experimental study showed that several concrete bridges in the United Kingdom absorbed a considerable amount of moisture during damp weather with a consequent increase in their mass (Wood, 1992).

The boundary conditions of the structural system always affect its dynamic response, especially when the system is constrained or indeterminate. Thermal effects caused by temperature variation often introduce axial stresses in the structural elements. The presence of such axial stresses changes the stiffness of the structure and may alter its vibration characteristics. When boundary conditions change with the age of the structure, this may lead to a change in the vibration characteristics even when there is no damage in the structure. Experimental modal analysis conducted on the Alamosa Canyon Bridge demonstrated that the variation in ambient temperature caused more than 5% variation in the estimated fundamental frequency within a 24-hour period (Hemez, 1993). These changes tend to make the identified modal parameters less representative of the true dynamic properties of the structure.

2. Difficulty in precise modeling of civil structures

Civil structures typically possess complicated geometry, consist of many different materials such as steel, concrete, and asphalt, and involve greater redundancy in the design. These issues make the accurate modeling of civil structures very difficult.

Model updating and refinement techniques can be employed prior to damage detection. However, the practical applications of these techniques remain limited primarily to research activities. Model updating and refinement techniques assume that the degrees of freedom (DOFs) of the measured modal vectors match identically to the DOFs of the analytical model. This assumption can be met only when measurements are conducted at all DOFs of the analytical model, or when mode shape expansion or model reduction techniques are employed. However, the mode shape expansion and model reduction techniques introduce significant errors into the model refinement procedure if the measurements are limited to a small number of locations. Furthermore, many model updating techniques utilize the connectivity information of the structural members to obtain a better baseline model. On the other hand, the connectivity information is lost when model reduction is applied. Model reduction techniques are commonly employed for the dynamic analysis of civil structures since the modeling of civil structures produces large system matrices and the rotational DOFs are typically condensed out from dynamic analysis DOFs.

3. Effect of incomplete modes, measurement errors and errors caused in modal extraction and mode truncation.

The size of civil structures does not permit the instrumentation of all or even a majority of the degrees of freedom with sensors and actuators, nor the excitation of higher modes. Furthermore, the application of forced vibration tests, which are

commonly used for system identification, is difficult for civil structures in service because of the economic and social consequence of service interruption due to road closure and evacuation of buildings. Ambient vibration tests are more suitable for civil structures since such tests can be conducted under normal operation of structures and can be easily repeated to collect additional modal data sets. One difficulty with ambient tests is in exciting higher modes. Most damage detection techniques for civil structures suffer from lack of data because only a small number of measurement points and a few fundamental modes are available.

4. Low Sensitivity to Damage

Vibration characteristics are global properties of the structure, whereas damage typically is a local phenomenon. Local response is captured by higher frequency modes whereas lower frequency modes tend to capture the global response of the structure and are less sensitive to local changes in a structure. It is more difficult to excite the higher frequency response of a structure, as more energy is required to produce measurable response at these higher frequencies than at the lower frequencies. As a result, the change in the global properties may be difficult to identify unless the damage is very severe or the measurements are very accurate. Accurate vibration measurements on civil engineering structures are difficult because such structures are large, so that controlled force excitation is often impractical. As a result, reliance must be placed on measurement of vibrations

induced by ambient forces, but precise measurement of vibration characteristics under such excitation is difficult.

5. Complexity and Limitation of the Damage Identification Algorithms

The identification of a possible damage site and severity of damage on the basis of a change in global properties derived from measurements at a limited number of sensor locations is a problem that has a non-unique solution. Sophisticated and complex mathematical techniques, including non-linear programming, need to be employed to obtain the most probable solution.

Almost all of the current damage identification methods rely on linear structural models and cannot deal with situations where the damage introduces non-linearity in the structure. Such non-linearity may, for example, result from the presence of cracks. Closing and opening of the cracks alters the stiffness of the structure, introducing non-linearity in its behaviour. Non-linearity may also result from loose connections that slip under load.

6. Need for analytical models or test data for healthy structure

Many algorithms depend on prior analytical models and/or prior test data for the detection and location of damage. Often, the lack of availability of this type of data or the lack of an accurate model can make a method impractical for certain applications. While it is doubtful that all dependence on prior models and data can be

eliminated, certainly steps can and should be taken to minimize the dependence on such information.

1.3 Review of literature on Artificial Neural Network applications in damage detection

Most techniques proposed so far for damage detection use mathematical models to describe structural behaviour and approximate the relationship between the specific damage condition and changes in the structural response. Damage detection in a structure based on its dynamic response is an inverse problem; the causes must be determined from the effects. A unique solution often does not exist for an inverse problem. Generally, damage detection can be viewed as a pattern recognition problem to effectively classify the changes in structural properties due to a specific damage. Neural network techniques are quite effective in dealing with pattern recognition problems. Consequently, during recent years, there has been a growing interest in using artificial neural network technique in damage detection for civil engineering.

Neural networks, originally developed to simulate the function of the human brain or neural system, are computational models. The pioneering work in this field is usually attributed to McCulloch and Pitts (1943), who showed that networks of artificial neurons could, in principle, compute any arithmetic or logical function, and who developed the first simplified model of a neuron. The main feature of their neuron model is that a

weighted sum of input signals is compared to a threshold to determine the neuron output. When the sum is greater than or equal to the threshold, the output is 1. When the sum is less than the threshold, the output is 0.

Rosenblatt (1958) invented the perceptron neural network and associated learning algorithm, which is the first practical application of artificial neural networks. Rosenblatt built the network and demonstrated its ability to perform pattern recognition. This success triggered a great deal of interest in neural network research. At almost the same time, Widrow and Hoff (1960) developed a new learning algorithm, which was similar in structure and capability to Rosenblatt's perceptron except that its transfer function was linear instead of hard-limiting. Unfortunately, later studies showed that both Rosenblatt's and Widrow's networks could solve only a limited class of problems. Many people believed that further research on neural networks would lead to a dead end because of the lack of new ideas and powerful computers on which to experiment.

During the 1980s, research in neural networks increased dramatically with the development of the powerful computer. The invention of the backpropagation algorithm is most responsible for the rebirth of artificial neural networks. There are now many kinds of neural networks, and new ones (or at least variations of old ones) are invented every week. Neural networks have been widely applied to diverse fields ranging from biology, medicine, business, finance and to many engineering fields.

In recent years, there has been an increasing interest in using artificial neural networks to

detect damage in complex structures based on the fact that changes in the physical properties, such as reductions in stiffness resulting from the onset of cracks or loosening of a connection, will cause detectable changes in modal properties. Changes in modal properties or other properties derived from them serve as indicators of damage. Thus the process of vibration based damage detection eventually reduces to some form of pattern recognition problem (Doebeling et al., 1996). Pattern recognition is one of the main problems for whose solution an artificial neural network is suitable.

The type of networks frequently used for the detection, quantification and classification of damage is the backpropagation neural network. In the backpropagation neural network, the outputs of one layer, multiplied by weights, summed, and then shifted by a bias are used as the inputs to the next layer. Once an architecture for the network is chosen, the weights and biases are adjusted according to the backpropagation learning algorithm by minimizing the error between the predicted and target outputs.

Kudva et al. (1991) used a backpropagation neural network to identify different hole locations and sizes introduced in a plate stiffened with a 4 by 4 array of bays under uni-axial compression. The structure of the network was designed to have two hidden layers with the same number of hidden nodes per layer as the number of inputs. To demonstrate the approach, the authors ran several examples. In all of the examples, the damage was in the form of circular holes of various diameters (from 12.7 mm to 63.5 mm) in the plate at the centres of the different bays, each of which was 305 mm by

203 mm. The neural network was trained using simulated strain gage readings on elements in the bays obtained by finite element analysis. The trained neural network was used to identify the mapping from the analytical strain gage data to the location and size of the hole. In different examples, for different neural networks, 8, 20, and 40 strain gages were used as input. The networks were trained with 3, 12, or 32 patterns, and the outputs were damage radius, and damage location, or the quadrant in which the damage was located. The networks converged from within a few seconds to around ten minutes on the 386 PC, depending on the example. It should be noted that in one example, the neural network failed to converge, and the authors were forced to modify their procedure to a two-step algorithm, which first predicted the hole quadrant, then the correct bay within the quadrant. The authors found that it was easier for the neural networks to predict the damage location than to predict the hole size, and hierarchical neural networks were more efficient than one big network. In the cases where the neural network successfully determined the hole size, typical errors were on the order of 50%.

Chen and Shah (1992) studied a pier in an actual bridge structure by a backpropagation neural network. The inputs to the network were accelerations obtained from four accelerometers fixed on either the top or the bottom of the pier with two in the transverse and two in the longitudinal direction of the bridge. The outputs for the network included the frequencies, displacements, and displacement mode shape. The network was trained with the data obtained under two different conditions: before an

explosion for demolishing three nearby buildings, and after the explosion. By comparing the results, they concluded that the network successfully detected the changes in the parameters of the pier after the explosion. They also noted that more accurate results would be obtained if more training data were used to train the network.

Wu, et al. (1992) used a backpropagation neural network to identify damage in a simple three-story building, modeled as a two-dimensional shear building with three degrees of freedom, driven by earthquake excitation. The damage was modeled by reducing column member stiffness by 50% to 75%. The neural network was used to identify the mapping from the Fourier spectra of the relative acceleration time histories to the level of damage in each of the members. The first 200 points in the fast Fourier transform (0 Hz to 20 HZ) were used as network inputs. A network architecture with one hidden layer of 10 hidden nodes and the output layer consisting of 3 neurons was selected, and 42 training cases were used. The authors did not give the training time required for the neural net to converge. The first attempt relied on using only acceleration data from the third or top floor with only one column member damaged at a time. Using the test data, the neural network was only able to identify the damage of column member connected to the third floor. The authors attributed this inability of the network to not being provided with enough information. To increase the information on which the damaged column member identification was based, a second network, which consisted of 400 node input sets and two hidden layers consisting of eight units in each layer, was implemented. The network

was designed to use acceleration data from two floor levels (the second and the third floors). This network was able to diagnose damage in the column members connected to the first or third floor to an accuracy of about 25% but was still unable to predict damage in the column member between the first and the second floor. The authors also identified further research needed in this area, including investigating the role of noise in actual measurements.

Elkordy et al. (1993) used backpropagation neural networks to identify damage in five-story buildings. Damage was simulated by reducing member stiffness in the bottom two stories by 10% to 70%. The authors chose a network with a single hidden layer of fourteen nodes. The mode shapes were used as input, the percent change in member stiffness was the output of the network. The two networks were trained on two mathematical models and verified with experimental data. Both models were two dimensional, finite element representations of different complexity. The first network was trained using eleven training patterns created from the first model; the second network was trained using nine training patterns created from the second model. In general, the neural network trained on the first model of lower complexity made poorer predictions; the network trained on the second model gave reasonable diagnosis and the absolute prediction errors of damage extent were generally within 10%.

Leath and Zimmerman (1993) used a perceptron neural network based on a training method they developed to identify damage in a four-element cantilevered beam. The

idea involved the creation of a network that would exactly fit the data with a minimal number of hidden nodes. The damage in the beam was modeled by reducing Young's modulus by up to 95%. The neural network was used to identify the mapping from the first two bending frequencies to the level of damage in each member. The algorithm was able to identify damage to within 35%.

Spillman et al. (1993) used a neural network with 14 input nodes, a hidden layer with 20 neurons, and 3 outputs, one for each of the possible damage states, to identify damage in a scale model steel bridge element. The element was 4.58 m in length. Damage was introduced by cutting through the lower three of the four wall sections and bolting a gusset plate across the cut section. With the plate attached loosely, the element was considered to be semi-damaged. Three sensors were mounted on the element. The simply supported beam was struck at four equally spaced points using the impact hammer technique. A total of eleven impact tests were performed under three states (no cut, loosely attached plate and no plate). Each sensor output was analyzed to identify the first two frequency peaks; their height and frequency were used as inputs to the neural network. The hammer impact location and calibrated hammer intensity were also provided as inputs. The backpropagation algorithm was used to train the network. The results were moderately successful. Using all three sensors, the authors found the proportion of correct diagnoses to be 58%.

Kirkegaard and Rytter (1994) used the backpropagation neural network with two hidden

layers of five neurons each to identify damage in a 20-m high steel lattice mast subject to wind excitation. The eight lower diagonals were cut and provided with bolted joints, each of which consisted of 4 splice plates. The damage states were simulated by removing one or more plates in these joints. The neural network was used to identify the mapping from the simulated patterns of the relative changes in natural frequencies to the sizes and locations of damage. The results showed that, at 100% damage the trained neural network was able to locate and quantify the damage, at 50% the trained network was able to predict the existence of damage, but not the magnitude. Damage less than 50% was not identified.

Later, Chen and Kim (1995), used the backpropagation neural network as an information processing technique for a 3D steel bridge, instrumented with accelerometers and strain gauges. The vibration signals were measured from a series of experiments performed on the instrumented 21 ft long scale model bridge structure. Four networks were involved in their study: the first one identified the existence of the damage using acceleration data; the second one located the damage in the model bridge using strain data; the third one determined both the damage occurrence and the location; and the fourth one was trained with acceleration data produced by a finite element model. The damage was introduced through saw cuts of various lengths on the flanges and webs of the girders. An impact hammer was used for the applied excitation. Their study concluded that the recognition of damage severity was much less accurate than that of damage detection and location.

With sensor signals used as inputs to the networks, it was feasible to identify the damage and its location in this model bridge structure instrumented with accelerations and strain gauges.

Although a large number of publications already exist, showing promise in the application of artificial neural networks for structural health monitoring, the area is still in its infancy and much work is still needed. Most of the papers attempt to identify damage from information related to modal frequencies. To the knowledge of the author, none of the researchers has combined structural dynamics and artificial neural network techniques to take advantage of their respective merits, and presented a thorough analysis of its implementation when mode shape errors and natural frequency errors are introduced in the simulation.

1.4 The proposed approach for damage detection

In this thesis, a new vibration based structural damage identification method is developed. The proposed method combines structural dynamics and artificial neural network techniques. It divides the structural damage detection problem into two distinct sub-problems. The use of a perceptron network to find the location is also explored. In the first step, the damage location is obtained by using the strain energy method. In the second step, the damage magnitude is evaluated by applying an artificial neural network technique. It is shown that the proposed method is practical and effective.

The basic strategy for developing a neural network based approach to damage assessment of a structural system is to train a backpropagation neural network to recognize member damage severity from the measured response of the structure. This strategy is based on the fact that the measured response of any structure contains information regarding its natural frequencies and corresponding mode shapes. Structural damage affects these dynamic characteristics of the structure. Thus, in theory, the measured response of a damaged structure, when compared with the response of the undamaged structure, contains information regarding the location and the extent of damage. However, this information may be extremely difficult to extract by the traditional methods, because the rules governing the cause and effect relationships must be established explicitly. In the approach proposed in this work, the training process is used to extract the cause and effect relationships, which are then stored within the connection weights of an appropriate backpropagation neural network. In this way, the learning capability of the neural network is utilized to eliminate the need for explicitly extracting the cause and effect relationships. These relationships exist in a distributed way within the connection strengths of the neural networks but are not obvious or externally available to the user.

The training of the neural network with appropriate data containing the information regarding the cause and effect relationships is a critical part of the proposed method of damage assessment. Thus, the first step is to generate a data set that can be used in

training an appropriate neural network to recognize damage patterns from the response of the structure. Ideally, this data set should contain the response of the undamaged structure as well as the responses of the structure in various damaged states. This data can be generated through measurement of structural response, model test results or through numerical simulation, or a combination of all three types of data. In this study, numerical simulation is used to create the training samples.

In a typical application of backpropagation neural network, an appropriate network architecture is determined and a training algorithm selected at the very beginning. Next, the selected neural network is trained with the training data set. The trained network is then tested to verify how well it has learned the training cases. The last step is usually to test the generalization capability of the neural network. In this step, the trained network is tested with data which was not present in the training data set. This last step is actually the test of the performance capability of the network. How well a trained network is able to generalize depends very much on the adequacy of the selected network architecture and the richness of the training data set in the information needed for damage assessment. Often, changes in the network architecture and/or additions to training data set are needed. Such changes are followed by the repetition of the whole training and testing process until satisfactory performance is obtained.

In this study, the damage index vector is used as input to the neural network, which has one hidden layer containing twice the number of input neurons.

The proposed approach includes the following main steps:

Step 1: Using the damage index method to locate the damaged element. The damage index may be used directly, by a sub-structuring technique, or as input to a perceptron network. The perceptron neural network method is successful in some cases.

Step 2: Applying the artificial neural network technique to evaluate the severity of the damage. This involves the following actions,

- a. Identifying a set of structural signatures that can characterize the health condition of the structure. The selected signature is damage index vector.
- b. Generating simulated damage patterns by analytical modeling.
- c. Training properly designed artificial neural networks with the analytically produced damage patterns.

1.5 Objectives of the present study

The following objectives are identified for the present study,

1. Review and evaluate the currently available methods of vibration based damage detection.
2. Based on the review, select an appropriate method of damage localization. The output produced by the method would serve as an input to the neural networks described in steps 3 and 4.

3. Evaluate the potential of a specially designed network to predict the location of damage using the data generated in step 2.
4. Design and implement a neural network to assess the severity of the identified damage. This includes the following actions,
 - a. Explore the effect of measurement errors in mode shapes on the accuracy of the damage prediction of the neural network using computer simulation.
 - b. Evaluate the effect of incomplete mode shapes on the damage detection.
 - c. Find the optimal size of training sample set by adjusting the damage states and number of data samples corresponding to each damage state.
 - d. Assess the effect of the neural network architecture, namely, the number of layers and the number of neurons in each layer, on the performance of the neural network to optimize the network topology.
5. Apply the selected vibration based damage identification technique to identify simulated damage in a 3D space truss and a field girder bridge named Crowchild Bridge, which has been instrumented and is periodically monitored by ISIS, a Canadian Centre of Excellence.

1.6 Layout of the thesis

An outline of this thesis is presented in the following.

Chapter 1 provides an overview of the existing vibration based damage identification algorithms and a review of Neural Network applications to damage detection. It also presents a review of the important literature on these topics and describes the challenges associated with damage detection.

Chapter 2 presents the theoretical background of vibration based damage detection, as well as an evaluation of the commonly used vibration based damage identification algorithms. The evaluation is carried out by applying the various algorithms to the detection of simulated damage in a beam model. Based on such an evaluation, the damage index method is selected for the approach used in this study.

Chapter 3 describes the theoretical background of neural network, and reviews the standard steepest gradient descent backpropagation training algorithm, which is the basis of a variety of other improved training algorithms.

Chapter 4 discusses a case study that applies the proposed vibration based damage identification technique to identify simulated damage in an ISIS bridge, the Crowchild Bridge.

Chapter 5 presents a case study that applies the proposed vibration based damage identification technique to identify simulated damage in a 3D space truss.

Chapter 6 presents the conclusion based on the study, and the further work still needed to be done.

Chapter 2

Vibration Based Damage Identification Techniques

2.1 Introduction

In this chapter, the theoretical background to the vibration based damage identification techniques is discussed. A beam model is used as an example for the evaluation of various conventional damage detection algorithms through computer simulation studies. By comparing the different vibration based damage identification algorithms, the most appropriate algorithms to locate the damage are selected and the parameters that contain sufficient structural information and can be used as input to artificial neural networks designed for either detecting the location of damage or the magnitude of damage or both are identified.

2.2 Eigenvalue equation

It is assumed that the damage results in a local decrease of the stiffness, which, in turn, affects the modal parameters, that is, the natural frequencies and mode shapes, of the

structure. The equation of motion representing the free vibration of an undamped structure is,

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = 0 \quad (2.1)$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{M}$ is the mass matrix, and $\mathbf{u}$ is the vector of nodal displacements.

The corresponding eigenvalue equation, for the healthy structure is given by:

$$\mathbf{K}\phi_i = \lambda_i \mathbf{M}\phi_i \quad (2.2)$$

where ϕ_i is the i th mass-orthonormal mode shape, and λ_i is the associated eigenvalue (squared frequency). Damage in the structure may change both the stiffness and the mass matrices, altering the frequencies as well as the mode shapes. The eigenvalue equation for the damaged structure is given by

$$(\mathbf{K} + \delta \mathbf{K})(\phi_i + \delta \phi_i) = (\lambda_i + \delta \lambda_i)(\mathbf{M} + \delta \mathbf{M})(\phi_i + \delta \phi_i) \quad (2.3)$$

Assuming that the system's mass is constant, and damage is associated with stiffness variations only, using Equation 2.2, we get:

$$\mathbf{K}\delta\phi_i + \delta\mathbf{K}\phi_{di} = \lambda_i\mathbf{M}\delta\phi_i + \delta\lambda_i\mathbf{M}\phi_{di} \quad (2.4)$$

where $\phi_{di} = \phi_i + \delta\phi_i$ is the i th mode shape of the damaged structure. Pre-multiplying both sides of Equation 2.4 by ϕ_i^T and using the transpose of Equation 2.2 we get:

$$\frac{\phi_i^T \delta\mathbf{K}\phi_{di}}{\phi_i^T \mathbf{M}\phi_{di}} = \delta\lambda_i \quad (2.5)$$

If the damage severity is quite small, $\phi_{di} \approx \phi_i$, and Equation 2.5 reduces to

$$\phi_i^T \delta K \phi_{di} = \delta \lambda_i \quad (2.6)$$

2.3 Algorithms for Vibration Based Damage Detection

Structural systems are susceptible to damage over their lives due to many factors, such as impact, operating loads, fatigue, and corrosion. Reliable and efficient techniques for damage detection and health monitoring are essential to their safe operation. Various analytical techniques of vibration-based damage detection have therefore been developed in the past decade or so. Early vibration-based structural health monitoring (SHM) research and applications were mainly focused on oil, aerospace and mechanical engineering areas. Recently, vibration-based assessment of the health of civil engineering structures has attracted a significant amount of research. Doebling et al (1996) provide an extensive bibliography related to this subject. Farrar et al. (1994) have reviewed the literature on vibration testing and damage detection in bridges.

The following is a list of the various techniques proposed. The list is not exhaustive but covers most of the methods that have been explored in some detail. As shown in the following study, not all methods can be considered efficient or effective, and there is need for further improvements.

1. Methods based on frequency changes

2. Methods based on mode shape changes
3. Mode shape curvature method
4. Methods based on change in flexibility matrix
5. Methods based on changes in uniform flexibility shape curvature
6. Damage index method
7. Method based on modal residual vector
8. Matrix update methods
9. Neural network methods

In this chapter, the above listed techniques, except the neural network methods, are applied to an example beam to evaluate their respective strengths and limitations through computer simulation studies. The beam, shown in Figure 2.1 is modeled by 20 one-metre long elements connected to the 21 nodes shown. Each intermediate node has two degrees of freedom, a vertical translation, and a rotation. The two support nodes have only the rotational degrees of freedom. The global stiffness matrix of a beam element shown in Figure 2.1 is given by,

$$[K] = EI_{zz}/L^3 \begin{matrix} & \begin{matrix} u_1 & u_2 & u_3 & u_4 \end{matrix} \\ \begin{bmatrix} 12 & -12 & 6L & 6L \\ -12 & 12 & -6L & -6L \\ 6L & -6L & 4L^2 & 2L^2 \\ 6L & -6L & 2L^2 & 4L^2 \end{bmatrix} \end{matrix}$$

where L = length of element

I_{zz} = 2nd moment of area about the z axis

E = Young's modulus of elasticity

The mass is assumed to be lumped along the translational degrees of freedom.

2.3.1 Methods based on changes in truncated frequencies

Damage reduces the magnitude of natural frequencies. However, the low sensitivity of frequency shifts to damage requires either very precise measurements or large levels of damage. Also, factors other than damage may cause frequency shifts. For example, in offshore platforms, damage induced frequency shifts are difficult to distinguish from shifts resulting from increased mass from marine growth.

Although higher modal frequencies are associated with local responses, in practice, the limitations in the excitation and identification of these higher modes make this technique difficult to implement and the results are unreliable.

As an example, consider the beam shown in Figure 2.1. Let it be assumed that damage reduces the 2nd moment of inertia of the section in element 5 by 30% or 50%, the corresponding frequencies are shown in Table 2.1. The results show that the changes in frequencies for 30% damage are insignificant and even with 50% damage they are small.

2.3.2 Methods based on mode shape changes

Like frequencies the mode shape changes induced by damage are small, consequently, such changes rarely provide a clear indication of damage (Zhang and Aktan 1995,

Pandey et al. 1991). As in Section 2.3.1, 50% damage is introduced in element 5 of the same beam; the first mode shape before and after damage is plotted in Figure 2.2. It is difficult to identify the damage from the information provided.

2.3.3 Mode shape curvature method

The reduction in stiffness caused by damage alters the mode shapes of the structure. In theory, changes in the mode shapes could be used to detect damage; however, as stated above, the change is usually so small that detection of damage is difficult. For a beam type structure, change in the modal curvature, rather than in the mode shape itself, can be used to detect damage. The instantaneous curvature at location x along the length of the beam, $\psi'(x)$, is given by

$$\psi'(x) = M(x)/(EI) \quad (2.7)$$

where E = the modulus of elasticity, and I = the moment of inertia of the section.

From the above equation, it is evident that the curvature is inversely proportional to the flexural stiffness of the beam. Consequently, reduction in the cross-sectional moment of inertia resulting from damage tends to increase the curvature of the mode shapes in the vicinity of the damage. In comparison to the mode shape itself, the mode shape curvature is more sensitive to a change in the moment of inertia. This implies that change in the mode shape curvature could be used to detect at least the location of the damage.

Based on the foregoing reasoning, Pandey et al. (1991) have suggested the application of mode shape curvature in detecting damage. The curvature is calculated from the lateral displacement components of the measured mode shapes by using a central difference expression. Thus, the curvature $\psi''(x_i)$ at location i along a beam is obtained from

$$\psi''(x_i) = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} \quad (2.8)$$

where it is assumed that modal displacements are measured at a number of equally spaced discrete points along the beam, y_i is the mode shape lateral displacement at location i , and $h = x_{i+1} - x_i$. An advantage in using Equation 2.8 is that when the healthy mode shapes are also obtained from measurement, an analytical model of the structure need not be developed. In addition, the rotational components of the mode shapes need not be measured, and the mode shapes can be normalized in any arbitrary manner, as long as the same procedure is used in the normalization for both the healthy and the damaged structure.

In this study, another method to calculate the modal curvature by using strain measurements is proposed. In practice, the curvature can be obtained by attaching strain gages capable of measuring dynamic strains at two points over the depth of the beam. If the depths of two points from an arbitrary datum are y_1 and y_2 , and the corresponding strains are ϵ_1 and ϵ_2 , the curvature is given by

$$\phi = \frac{\epsilon_2 - \epsilon_1}{y_2 - y_1} \quad (2.9)$$

For the computer simulations used in this study, the curvature for each element of the beam is derived from the rotational components of the full analytical mode shapes as follows,

$$\psi^*(i) = \frac{\theta_{i+1} - \theta_i}{l} \quad (2.10)$$

where, $\psi^*(i)$ is the constant curvature along the i th element, θ_{i+1} and θ_i are the rotational modal components at the ends of the i th element, and l is its length. It needs to be emphasized that in field application, Equation 2.9 would be used to obtain the curvature, but the results from Equation 2.10 would be similar to those obtained from Equation 2.9.

As an example of the application of mode shape curvature method, consider the beam shown in Figure 2.1. Let it be assumed that damage reduces the moment of inertia of elements 5 and 12 by 20%, simultaneously. Computer simulation based on the finite element model of the beam provides both the healthy and the damaged mode shapes of the beam. Assuming that the mode shapes have been correctly measured, Figure 2.3 shows the mode shape curvature for mode number 1 both before and after damage, while Figure 2.4 shows the difference between the two sets of curvature. It is quite evident from both plots, particularly from the plot in Figure 2.4, that the modal curvature has significantly increased between nodes 5 and 6 and between nodes 12 and 13 (Figure 2.4(a)), or at 5m and 12m from the left hand side end of the beam (Figure 2.4(b)), thus correctly identifying the damage at these sites. For a higher mode,

as shown in Figures 2.5 and 2.6, the mode shape curvature obtained from Equation 2.10 clearly identifies the damaged elements; however, the mode shape curvature obtained from Equation 2.8 does not.

In practice, measurement errors are bound to exist. To investigate the effect of errors, random errors up to a maximum of 2% were introduced in the elements of lateral and rotational mode shapes of the damaged beam. Thus, each element of the mode shape was varied by a randomly generated percentage value lying between $\pm 2\%$. The beam suffers 50% damage in element Nos. 5 and 12. Differences in modal curvature were calculated and are plotted in Figures 2.7 and 2.8 for mode Nos. 1 and 6, respectively. The mode shape curvature obtained from Equation 2.10 indicates the damage location clearly, but it is no longer possible to detect the damage using the mode shape curvature obtained from Equation 2.8. Evidently, modal curvatures calculated from lateral displacement by using Equation 2.8 are very sensitive to measurement errors, and damage cannot be detected with reliability on the basis of observed changes in the modal curvature. The sensitivity to measurement errors is to be expected because numerical second order differentiation required for obtaining the curvature by Equation 2.8 magnifies the effect of errors. The mode shape curvature from strain measurements can be used to identify the damage with greater confidence.

2.3.4 Methods based on change in flexibility matrix

Relying on the observation that the flexibility matrix of a structure can be derived from the measured mode shapes, Pandey and Biswas (1994) have proposed a method of detecting damage that is based on the difference between the flexibility matrices of the damaged and the healthy structure. The flexibility matrix $\mathbf{F}$ of the undamaged structure is given by

$$\mathbf{F} \approx \mathbf{\Phi} \mathbf{\Omega}^{-1} \mathbf{\Phi}^T = \sum_{j=1}^m \frac{1}{\lambda_j} \phi_j \phi_j^T \quad (2.11)$$

where $\mathbf{\Phi}$ is the modal matrix of the mass-normalized mode shapes ϕ_j , and m is the number of measured mode shapes. The relationship in Equation 2.11 is approximate only because m is less than the number of degrees of freedom, N . In a similar manner, the flexibility matrix of the damaged structure $\mathbf{F}_d$ is obtained from

$$\mathbf{F}_d \approx \mathbf{\Phi}_d \mathbf{\Omega}_d^{-1} \mathbf{\Phi}_d^T = \sum_{j=1}^m \frac{1}{\lambda_{dj}} \phi_{dj} \phi_{dj}^T \quad (2.12)$$

where $\mathbf{\Phi}_d$ is the modal matrix of the damaged structure.

The difference between the flexibility matrices of the damaged and the healthy structures is next obtained from

$$\Delta \mathbf{F} = \mathbf{F}_d - \mathbf{F} \quad (2.13)$$

Now let δ represent the row vector whose j th element is equal to the element with the largest absolute value in the j th column of $\Delta \mathbf{F}$.

$$\delta_j = \max |\Delta F_{ij}| \quad i = 1, \dots, N \quad (2.14)$$

Elements of δ_j that have a large value correspond to the location of damage. As in the case of the mode shape curvature method, development of an analytical model is not necessary provided the healthy mode shapes are obtained from measurements. However, the mode shapes used in Equations 2.11 and 2.12 must be mass normalized, which means that the mass matrix must be determined. In addition, if the mode shapes used in Equations 2.11 and 2.12 are incomplete, implying that measurements of modal displacements are made along only a limited number of degrees of freedom, the mass should be lumped along only the measured degrees of freedom, or the mass matrix must be reduced to a size equal to the number of measured degrees of freedom. Reduction could be achieved by using a standard method of reduction, such as for example the Guyan's reduction (Guyan, 1965).

As an example of the application of flexibility matrix method, consider again the beam shown in Figure 2.1. It is assumed that the mass of the beam is lumped along the translational degrees of freedom and mode shape measurements are available along the same set of degrees of freedom, for both the healthy and the damaged beam. The number of modes measured is 6. Damage in element No. 5 of the beam causes its stiffness to reduce by 70%. The flexibility differences δ_j obtained from Equation 2.14 are plotted in Figures 2.9 and 2.10 for the accurate mode shapes and the mode shapes with up to 2% measurement error, respectively. It is apparent that degrees of freedom 4 and 5, that is,

element No. 5, have been most affected by damage. Measurement errors cause the identification of spurious damage in element No. 6.

The problem is repeated with 70% damage in each of elements 5 and 12. The results are shown in Figures 2.11 and 2.12. This time it is possible to conclude that element No. 12 has suffered damage, but the damage in element 5 has been obscured.

2.3.5 Methods based on changes in uniform flexibility shape curvature

Each column of the flexibility matrix $\mathbf{F}$ obtained from Equation 2.11 represents the static displacement shape assumed by the healthy structure when a unit load is applied along the corresponding degree of freedom. Thus the j th column of $\mathbf{F}$ gives the displaced shape when a unit load is applied along degree of freedom j . In a similar manner, columns of $\mathbf{F}_d$ represent the displacements produced by the application of unit loads on the damaged structure. A displacement curvature shape can be obtained corresponding to each column of $\mathbf{F}$ and $\mathbf{F}_d$. Because those curvatures correspond to the application of a uniform load of unity, they are known as uniform load flexibility curvatures. Zhang and Aktan (1995) state that the changes in curvatures can be used to determine the location of damage. Denoting the j th curvature shape obtained from $\mathbf{F}$ and $\mathbf{F}_d$ by ψ_j and ψ_{dj} , respectively, we obtain global measure of the difference between the two sets of curvatures from

$$\delta = \sum_j |\psi_{dj} - \psi_j| \quad (2.15)$$

Elements of δ that have a large value correspond to the location of damage. Again,

development of an analytical model is not necessary provided the healthy mode shapes are obtained from measurements, but the mode shapes used in Equations 2.11 and 2.12 must be mass normalized.

The method is applied to the detection of damage in the beam shown in Figure 2.1. Again, it is assumed that the mass of the beam is lumped along the translational degrees of freedom and mode shape measurements are available along the same set of degrees of freedom, for both the healthy and the damaged beam. The number of modes measured is 6. Damage in element Nos. 5 and 12 of the beam causes each of their stiffness to reduce by 50%. The uniform flexibility shape curvature differences obtained from the 19 by 19 flexibility matrix using Equation 2.15 are plotted in Figures 2.13 and 2.14 for two different cases: (1) when the mode shapes are accurate, and (2) when random measurement error up to 2% is introduced in the elements of the mode shapes. When the mode shapes are accurate it is apparent that degrees of freedom 4, 5, 11 and 12 have been most affected by damage, which means that elements Nos. 5 and 12 have suffered damage. Measurement errors make the identification somewhat inaccurate. Element No. 5 is correctly identified, but element No. 12 associated with degrees of freedom 11 and 12 is not found to be damaged. Instead, degree of freedom 14 is observed as being affected. Again, the numerical differentiation involved in obtaining the curvature magnifies the effect of measurement errors making the method unreliable.

2.3.6 Strain energy method

The strain energy method for beam-like structures was originally developed by Stubbs and Kim (Stubbs et al, 1995). This method was then extended to plate-like structures (Cornwell et al, 1997). The basic approach of the strain energy method is the division of the structure into a series of beam or plate-like elements, and then estimation of the modal strain energy of each element before and after damage. The curvatures of the mode shapes are used to approximate the strain energy. Comparison of the two sets of strain energy provides the location of the damage.

Consider a linear elastic beam that has been divided into NE elements. Damage causes reduction in the flexural rigidity of one or more elements. Measurements of the lowest nm mode shapes have been made both on the healthy and the damaged structure. Modal curvatures can now be derived by a process of numerical differentiation, say the central difference operation. The modal strain energy in the i th mode is given by

$$S_i = \int_0^L EI(x) [\psi_i''(x)]^2 dx \quad (2.16)$$

where L is the length of the beam, EI is the flexural rigidity, and $\psi''(x)$ denotes modal curvature. In practice, the integration indicated in Equation 2.16 will be carried out by a numerical procedure. The strain energy contributed by the j th element lying between $x = a$ and b is

$$S_{ij} = \int_a^b EI_j [\psi_i''(x)]^2 dx \quad (2.17)$$

The ratio $F_{ij} = S_{ij}/S_i$ gives the fraction of total strain energy contributed by element j .

The corresponding quantities for the damaged structure are

$$S_i^d = \int_0^L EI^d(x) [\psi_i^{d*}]^2 dx \quad (2.18)$$

$$S_{ij}^d = \int_a^b EI_j^d [\psi_i^{d*}]^2 dx \quad (2.19)$$

and $F_{ij}^d = S_{ij}^d/S_i^d$. It is now assumed that damage is confined to a few elements, so that in Equation 2.18 the $EI^d(x)$ may be taken as approximately equal to $EI(x)$. It is further assumed that

$$F_{ij}^d \approx F_{ij} \quad (2.20)$$

Substitution of Equations 2.16 through 2.19 in Equation 2.20 gives

$$\gamma_{ij} = \frac{EI_j}{EI_i^d} = \frac{\int_a^b [\psi_i^{d*}(x)]^2 dx}{\int_0^L [\psi_i^{d*}(x)]^2 dx} \bigg/ \frac{\int_a^b [\psi_i^*(x)]^2 dx}{\int_0^L [\psi_i^*(x)]^2 dx} \quad (2.21)$$

Equation 2.21 provides the damage index γ_{ij} for the j th element in mode No. i . To use the information available from the nm measured modes the definition of damage index is modified as follows

$$\gamma_j = \frac{\sum_{i=1}^{nm} f_{ij}^d}{\sum_{i=1}^{nm} f_{ij}} \quad (2.22)$$

where

$$f_{ij} = \frac{\int_a^b [\psi_i(x)]^2 dx}{\int_0^L EI(x) [\psi_i(x)]^2 dx} \quad (2.23)$$

and a similar expression can be developed for f_{ij}^d . Elements for which γ_{ij} is relatively large are likely to have been damaged. This provides the location of damage. Theoretically, an estimation of the severity of damage can be obtained from

$$\beta_j = 1 - \frac{1}{\gamma_j} \quad (2.24)$$

Numerical problems may arise in the evaluation of Equation 2.21 or 2.22 when the denominator is very small in magnitude, which may be case when the strain energy contributed by the j th member in the modes being considered is very small. In such a case, Equation 2.22 is modified as follows

$$\gamma_j = \frac{1 + \sum_{i=1}^{nm} f_{ij}^d}{1 + \sum_{i=1}^{nm} f_{ij}} \quad (2.25)$$

The method as outlined above for a beam structure can be extended to a structure of a general type. Thus, for the healthy structure the strain energy of member j in mode i is given by

$$S_{ij} = \phi_i^T \mathbf{k}_j \phi_i \quad (2.26)$$

The total strain energy of the structure deforming in its mode i is obtained from

$$S_i = \sum_j \phi_i^T \mathbf{k}_j \phi_i = \phi_i^T \mathbf{K} \phi_i \quad (2.27)$$

where $\mathbf{k}_j$ is the stiffness matrix of member j and $\mathbf{K}$ is the stiffness matrix of the structure.

Similar expressions can be written for the damaged structure. It may be noted that in deriving S_i^d , the global stiffness matrix of the damaged structure $\mathbf{K}^d$ is taken as being approximately equal to $\mathbf{K}$. As before the fractional strain energy is expressed as $F_{ij} = S_{ij}/S_i$ and $F_{ij}^d = S_{ij}^d/S_i^d$. Assuming that $F_{ij}^d \approx F_{ij}$ and expressing the stiffness matrix of damaged member j as $\mathbf{k}_j^d = \alpha_j \mathbf{k}_j$ we obtain an expression similar to Equation 2.22 for the damage index

$$\gamma_j = \frac{1}{\alpha_j} = \frac{\sum_{i=1}^{nm} f_{ij}^d}{\sum_{i=1}^{nm} f_{ij}} \quad (2.28)$$

where

$$f_{ij} = \frac{\phi_i^T \mathbf{k}_j \phi_i}{\phi_i^T \mathbf{K} \phi_i} \quad (2.29)$$

And

$$f_{ij}^d = \frac{\phi_i^{dT} \mathbf{k}_j^d \phi_i^d}{\phi_i^{dT} \mathbf{K}^d \phi_i^d} \quad (2.30)$$

Equations 2.24 and 2.25 apply to this case too.

The damage index method is applied to detect damage in the beam shown in Figure 2.1.

It is first assumed that element Nos. 5 and 12 have suffered damage causing a reduction

in their flexural rigidities by 20%. Six mode shapes are available for both the healthy and the damaged structure. Modal curvatures are calculated in two different ways, by Equation 2.8 and Equation 2.10. Using these modal curvatures and Equations 2.22 and 2.23, damage indices are obtained for the 20 elements and are plotted in Figures 2.15 and 2.16 for two different cases: (1) when the measured mode shapes are exact, and (2) when random errors up to 2% exist in elements of the mode shapes. The damage indices based on the curvature mode shapes from strain measurements correctly identify the damage location. However, measurement errors affect the prediction results when the damage indices are based on curvatures derived from translational mode shapes. The region of damage is not predicted exactly, but is still in the vicinity of the actual damage site.

The severity of damage obtained from Equation 2.24 is plotted in Figures 2.17 and 2.18. The estimates are completely inaccurate and the method cannot be used to find the damage severity. Calculations are repeated with damage represented by 50% reduction in the flexural rigidity of both element Nos. 5 and 12; the measurement errors in mode shapes are up to 2%. The damage indices are shown in Figure 2.19. The damage location is predicted accurately by the damage indices based on directly measured curvatures. The damage extents are further increased to 70% in each of the elements 5 and 12. The damage indices based on the curvatures derived from translational mode shapes with 2% measurement errors are shown in Figure 2.20. This time, the damage states are correctly

located. However, the damage extent prediction shown in Figure 2.21 is still not correct. Finally, damage indices are obtained for the case when element 5 suffers 20% reduction in the flexural rigidity while element 12 suffers a 70% reduction. The results presented in Figures 2.22 and 2.23 show that while damage to element 12 is clearly predicted, damage to element 5 is obscured.

The results tend to show that the damage index method is fairly successful in predicting damage. The damage indices based on directly measured curvatures always accurately locate the damage. For cases of low damage, measurement errors may cause the prediction of damage region to be somewhat inaccurate when the damage indices are obtained from the modal curvatures derived from translational mode shapes, but for practical purposes the inaccuracy may be acceptable. In the case of multiple damage sites, the sites with lower damage may not be identified. Also, if the damaged member makes very little contribution to the strain energy of the measured modes, the damage will not be detected. Finally, although the method is generally successful in locating the damage region, the estimates of damage severity are not accurate. The advantages of the method are: (1) an analytical model need not be developed, (2) only a few lower order modes are adequate for locating the damage, (3) it is not necessary to mass normalize the modes and any consistent normalization procedure may be used, and (4) measurement errors in the mode shapes have minor effect on the ability of the method to predict damage.

2.3.7 Method based on modal residual vector

This method relies on the measurement of both the frequencies and mode shapes of the damaged structure. The eigenvalue equation for the damaged structure can be written as:

$$(\mathbf{K} + \delta\mathbf{K})\boldsymbol{\phi}_{di} - \lambda_{di}\mathbf{M}\boldsymbol{\phi}_{di} = 0 \quad (2.31)$$

where λ_{di} is the i th eigenvalue of the damaged structure and it is assumed that damage does not cause a change in the mass. In Equation 2.31, all quantities other than $\delta\mathbf{K}$ are known or have been determined through modal tests.

Equation 2.31 can be written in the following alternative form:

$$\mathbf{K}\boldsymbol{\phi}_{di} - \lambda_{di}\mathbf{M}\boldsymbol{\phi}_{di} = \mathbf{R}_i = -\delta\mathbf{K}\boldsymbol{\phi}_{di} \quad (2.32)$$

Evaluation of the left hand side of Equation 2.32 provides the modal residual vector $\mathbf{R}_i$ for the i th mode. Matrix $\delta\mathbf{K}$ will have non-zero terms corresponding to only those degrees of freedom that are connected to the damaged elements. Correspondingly, the non-zero terms in $\mathbf{R}_i$ will also lie along the same degrees of freedom. Knowledge of the affected degrees of freedom and the connectivity relation between the elements and the degrees of freedom allows the determination of the damage location. When more than one mode is determined, an absolute sum of the modal residuals can be used to determine the damage location. When using the absolute sum, the mode shapes should be appropriately normalized so that individual elements of the various mode shapes have

the same relative order of magnitude. This can be achieved by using a consistent method of normalizing the mode shapes.

Having identified the damaged elements, it is possible to express $\delta\mathbf{K}$ as the weighted sum of the stiffness matrices of the damaged elements. The weighting factors, which are the unknowns in the problem, define the severity of damage in the affected elements. For example, if the reduction in the stiffness of element j is expressed as $\beta_j \mathbf{k}_j$, we have

$$\delta\mathbf{K} = -\sum_j \beta_j \mathbf{k}_j \quad (2.33)$$

where the summation is carried out over all the damaged elements. Equation 2.32 can now be expressed as

$$\left(\sum_j \beta_j \mathbf{k}_j \right) \phi_{di} = \mathbf{R}_i \quad (2.34)$$

or as

$$\mathbf{H}_i \boldsymbol{\beta} = \mathbf{R}_i \quad (2.35)$$

where the column vectors of matrix $\mathbf{H}_i$ are $\mathbf{k}_j \phi_{di}$, and vector $\boldsymbol{\beta}$ is of size ne , where ne is the number of elements likely to be damaged. If ne includes all of the structural elements, $\mathbf{H}$ will be of size N by ne while $\mathbf{R}_i$ will be of size N by 1. However, if an examination of the modal residuals shows that only a subset of the elements is likely to have suffered damage, only the degrees of freedom connected to the identified elements need be included in Equation 2.35. Thus, if the number of elements likely to have been damaged is le and the number of degrees of freedom connected to such elements is nd , $\mathbf{H}_i$ will be

of size nd by le , while β is of size le and R_i is of size nd . When more than one mode is determined, the columns of H corresponding to the additional modes may be strung below those corresponding to the previous modes. Vector R may be augmented in a similar manner. The augmented H matrix will be of size na by le where $na = nd \times m$ and m is the number of modes included. The augmented vector R will be of size na . Equation 2.35 or its augmented form can be solved for β by using a least squares method.

Kaouk and Zimmerman (1994) have suggested an alternative method of evaluating the damage extent from the modal residuals. The method is based on a generalized minimum rank perturbation theory. In matrix form, the modal residuals B can be expressed as

$$B = K\Phi_d - M\Phi_d\Lambda \quad (2.36)$$

where Λ is the diagonal matrix of squared frequencies. When matrix B is of full column rank, the perturbation to the original stiffness matrix caused by damage is given by

$$\delta K = B(B^T\Phi_d)^{-1}B^T \quad (2.37)$$

The perturbation matrix δK will have significant values corresponding to the degrees of freedom connected to the damaged elements. The extent of damage to such elements can be derived from the significant non-zero elements of δK . When B is not of full column rank, a subset of B with full rank and the corresponding subset of Φ_d should

be used in Equation 2.37.

In practice, errors introduced by mode shape expansion and measurement errors in the frequencies and modes may make the evaluation of the damage factors unreliable. The best one can hope to achieve is to determine the presence of damage and the location of damage. Even there practical difficulty exists, as will be seen from the examples that follow.

For the example beam, element No. 5 is assumed to have a damage of 20% and element No. 12 a damage of 30%. Let it be assumed first that exact and complete mode shapes are available for the first six mode shapes of the damaged structure. The absolute sums of the modal residuals of the first six modes are shown in Figure 2.24. It is apparent that translational degrees of freedom 4, 5, 11, 12 are affected. Element No. 5 is connected to degrees of freedom 4 and 5, while Element No. 12 is connected to degrees of freedom 11 and 12. Modal residuals thus indicate that these two elements are damaged. This is confirmed by the residuals against the rotational degrees of freedom 24, 25, 31 and 32, the first two of which connect to Element No. 5 while the last two connect to Element No. 12. The damage factors obtained by solving Equation 2.35 or its augmented form, which contains all six modes, are exact.

The next case of the example beam has the same level of damage. However, this time modal measurements are made only along the translational degrees of freedom. The mode shapes are expanded to the 40 degrees of freedom by using the SEREP method

(O'Callahan et al., 1989). The absolute sums of the modal residuals for six modes are shown in Figure 2.25. It is now difficult to ascertain the true location of damage, although damage is indicated around degrees of freedom 3 to 6 and 10 to 13. The damage factors obtained from the augmented form of Equation 2.35 in which all six modes are accounted for but only the translational degrees of freedom are measured are almost meaningless. Evidently, the expansion of the incomplete modes using SEREP introduces significant errors in mode shape elements corresponding to the rotational degrees of freedom, which in turn makes the elements of $\mathbf{H}$ incorrect.

It is next assumed that not only are the measured modes incomplete but there also exist up to 2% random errors in the mode shapes. The modal residuals obtained from such modes are shown in Figure 2.26. Prediction of the damage location now becomes very uncertain.

We next use the minimum rank perturbation theory to evaluate the extent of damage. When six complete and exact mode shapes are used, the calculated perturbations in the stiffness matrix are quite accurate. The perturbations along the 19 translational degrees of freedom are shown in Figure 2.27. The perturbations are all zero except along degree of freedom 4, 5, 11, and 12, indicating that elements 5 and 12 that are connected to these degrees of freedom have suffered damage. Perturbations in the stiffness matrix elements (11,11) and (12,12) are 1.3248×10^9 , while the Perturbations in the stiffness matrix elements (4,4) and (5,5) are 1.9872×10^9 . The damage factors for elements 5 and 12

calculated from these values are 30% and 20% respectively, which are, of course, exact. However, when the modes are incomplete and errors exist in the mode shapes, the damage extent is not predicted at all, as will be seen from Figure 2.28. The perturbation in element (4,4), (5,5), (11,11) and (12,12) are 6.5×10^9 , 5.9641×10^9 , 4.2586×10^9 and 1.946×10^9 , respectively. All of them are quite inaccurate.

2.3.8 Matrix update methods

The matrix update methods constitute the largest class of methods developed for the identification of damage from measured vibration properties. The methods are based on the determination of perturbations in the property matrices, such as $\delta \mathbf{K}$, that will satisfy the eigenvalue equation (Equation 2.5). The identified perturbation provides both the location and severity of the damage. In general, the number of unknown parameters in $\delta \mathbf{K}$ is significantly greater than the number of measured frequencies and mode shapes. The problem is therefore underdetermined and has an infinite number of solutions. A unique solution can be obtained through the minimization of an objective function subject to some specified constraints. The matrix update methods can thus be classified on the basis of the objective function selected and the constraints used.

Many different objective functions may be used. They include the norm of the estimated solution, rank of the perturbation matrix $\delta \mathbf{K}$, norm of the modal residual vector $\mathbf{R}_i$ and others. The constraints may include satisfaction of the eigenvalue equation,

and preservation of the sparsity and connectivity of the property matrices. In addition, the value of β_j (Equation 2.33) should be greater than or equal to zero but no greater than 1.

Because of the many different ways in which the objective function and constraints can be defined and the different numerical techniques that can be used in the solution of the resulting optimisation problem, a large number of methods have been developed. Here we present only a few such methods for the purpose of illustration.

Using Equation 2.33, Equation 2.5 can be expressed as

$$\delta\lambda_i = \frac{-\sum_{j=1}^{ne} \phi_i^T \mathbf{k}_j \phi_{di} \beta_j}{\phi_i^T \mathbf{M} \phi_{di}} \quad (2.38)$$

or

$$\mathbf{D}\boldsymbol{\beta} = -\delta\boldsymbol{\lambda} \quad (2.39)$$

where ne is the number of elements, $\mathbf{D}$ is an m by ne matrix whose elements are $d_{ij} = \phi_i^T \mathbf{k}_j \phi_{di} / \phi_i^T \mathbf{M} \phi_{di}$, $\boldsymbol{\beta}$ is the ne -vector of the unknown changes in element stiffness matrices, and $\delta\boldsymbol{\lambda}$ is the m -vector of measured eigenvalue changes. For a damage of low severity, the expression d_{ij} can be simplified as $d_{ij} = \phi_i^T \mathbf{k}_j \phi_i$ provided ϕ_i are mass normalized.

Equation 2.39 defines a set of constraints on the solution vector $\boldsymbol{\beta}$. An alternative set of constraints can be obtained by pre-multiplying Equation 2.32 by ϕ_{di}^T , which gives

$$-\phi_{di}^T \delta K \phi_{di} = \phi_{di}^T K \phi_{di} - \lambda_i = P_i \quad (2.40)$$

Substitution of Equation 2.33 in Equation 2.40 yields

$$\sum_{j=1}^{ne} \phi_{di}^T k_j \phi_{di} \beta_j = P_i \quad (2.41)$$

or

$$S\beta = P \quad (2.42)$$

where $s_{ij} = \phi_{di}^T k_j \phi_{di}$. The formulations presented in Equations 2.38 through 2.42 automatically ensure that the sparsity of the stiffness matrix and connectivity of the structure are preserved. It may be noted that the use of Equation 2.38 requires that the healthy mode ϕ_i be correlated to the damaged mode ϕ_{di} , which is not always straightforward, particularly if some modes switch positions following damage. On the other hand, Equation 2.41 uses only the damaged modes, and no correlation is required.

If $m = ne$, Equation 2.39 or Equation 2.42 can be directly solved for β . In general, however, m will be significantly smaller than ne so that the problem defined by Equation 2.39 or Equation 2.42 is underdetermined and has an infinite number of solutions. In order to obtain a unique solution an optimisation problem needs to be solved, in which a selected objective function is minimised subject to a specified set of constraints. It may be noted that a solution that leads to the minimisation of an objective function is not necessarily the correct solution.

Least squares solution

As stated above, when the problem represented by Equation 2.42 is undetermined, that is when the number of unknowns is larger than the number of equations or $m < ne$, Equation 2.42 has an infinite number of solutions. However, in the case of damage, an additional constraint exists on the damage factors, which must now lie between 0 and 1.

Thus

$$0 \leq \beta \leq 1 \quad (2.43)$$

In some cases this additional constraint may provide a unique solution. However, this is not always so.

Using Equation 2.39 the non-negative least squares routine of software Matlab 6.5 (The MathWorks Inc.) is used to estimate the damage in the beam shown in Figure 2.1. It is assumed that damage has caused a 30% reduction in the stiffness of element 5 and 20% reduction in the stiffness of element 12, and six complete or incomplete modes, both with and without error, of the damaged beam are available. The calculated damage factors are shown in Figure 2.29 through Figure 2.32. In Figure 2.29, it is found that the identification is quite accurate if the mode shapes are accurate and complete. When only the 19 translational degrees of freedom are measured, the incomplete mode shapes are expanded to the full 40 degrees of freedom by using SEREP. The damage factors, obtained by the least squares technique, are shown in Figure 2.30. The damage is not identified any more. Figure 2.31 shows the case when the mode shapes are complete but

contaminated by measurement error. The damage location is accurately determined in this case. Finally, assume that the modes are measured only along the 19 translational degrees of freedom; in addition there exist up to 2% random errors in the elements of mode shapes. The calculated damage factors are now shown in Figure 2.32. They are not accurate. From the results presented in Figures 2.29 to 2.32, it seems that the error due to modal expansion has more effect on the accuracy of damage prediction than the measurement error.

Minimization of the quadratic norm of damage vector

Here we choose to minimize the quadratic norm of the stiffness changes given by

$$J = \beta^T \beta \quad (2.44)$$

with the constraint that Equation 2.39 must be satisfied. If these are the only constraints, the problem has a closed-form solution given by

$$\beta = -D^T (DD^T)^{-1} \delta \lambda \quad (2.45)$$

This solution is useful in refining the analytical model of the undamaged structure so as to ensure that the analytically determined frequencies and mode shapes match those obtained by modal testing. The refined analytical model then serves as the baseline model in the subsequent determination of damage.

For a damaged structure, the additional constraint given by Equation 2.43 must be satisfied. The problem now becomes a nonlinear optimisation problem with the objective function defined by Equation 2.44, equality constraints given by Equation 2.39 and inequality constraints given by Equation 2.43. The optimisation problem can be solved by using any one of the several algorithms available for the solution of constrained quadratic nonlinear optimisation problems, for example, those in the computer software Matlab 6.5. The optimisation routine used in the present work is based on a sequential quadratic programming (SQP) method. The SQP methods represent state-of-the-art in nonlinear programming. Fletcher (1980) and Gill et al (1981) provide an overview of the SQP methods. As stated earlier, the minimization of an objective function does not necessarily produce the exact answer.

The method is used to predict simulated damage in the beam shown in Figure 2.1. Element 5 has a damage of 30% and element 12 has a damage of 20%. Six complete mode shapes of both the healthy and damaged structure are used in the prediction. The healthy mode shapes can be obtained from a correlated finite element model, while the damaged mode shapes are obtained through testing. It is first assumed that all mode shapes are exact. Estimates of the damage factors for elements of the beam are obtained by the nonlinear constrained optimisation method. The results are identical to those shown in Figure 2.29. The damage is predicted accurately in this case.

Next, it is assumed that the six available mode shapes and frequencies are accurate, but

mode shape measurements have been made only along the 19 translational degrees of freedom. The damaged mode shapes are expanded to the full 40 degrees of freedom using SEREP and the available analytical mode shapes of full size. The revised estimates of damage are shown in Figure 2.33. Damage is not correctly predicted in elements 5 and 12. Damage estimation is repeated using the constraint Equation 2.42 in place of Equation 2.39. The results presented in Figure 2.34, are only marginally better than those shown in Figure 2.33.

Finally, it is assumed that the six damaged mode shapes are incomplete, with only the translational degrees of freedom being measured; at the same time random errors of up to 2% are introduced in the mode shape elements of the damaged structure. These errors are representative of those inherent in measurements. The estimates of damage are shown in Figure 2.35. They are found to be quite inaccurate. In fact, the optimiser could not find a feasible solution in this case.

2.4 The proposed approach

Other researchers (Jauregui and Farr, 1996) have also examined the performance of different commonly used vibration based damage identification methods, which have been reported in the technical literature to be relatively successful. They have applied these methods to the same problem to compare and evaluate their relative merits. On the basis of such study and that presented in this chapter, the strain energy method is found

to perform the best. This method is capable of detecting and locating damage at almost all damage levels. The mode shape curvature method also performs well, although not as well as the damage index method. The change in flexibility method appears to have problems in identifying damage for situations when damage is not severe. The change in uniform flexibility curvature method performs satisfactorily only when accurate modal data are available.

All of the cited research indicates that no one universal vibration based damage identification algorithm has been shown to be always successful in detecting, locating and quantifying the damage at the same time, especially with incomplete mode shapes and measurement errors in the mode shapes.

Based on the previous research and the example cases presented in this chapter, it is observed that the damage index method is relatively effective in locating the damage. This method is therefore used for further work in this study, both for locating the damage and for input to the neural network designed to assess the severity of damage.

Table 2.1: The changes in frequencies due to 30% and 50% damage
in element 5 of the beam

Mode No.	Frequencies of Healthy beam (Hz)	Frequencies of Damaged beam (Hz)		Percentage Change	
		30%	50%	30%	50%
1	3.82	3.80	3.75	0.52%	1.83%
2	15.31	15.01	14.64	1.96%	4.37%
3	34.46	33.97	33.43	1.42%	2.98%
4	61.26	61.11	60.94	0.24%	0.52%
5	95.70	95.34	94.89	0.38%	0.85%

Figure 2.1: Finite element model of a beam

Figure 2.2: Translational mode shapes for the exact mode No. 1 of healthy and damaged beam, 50% damage in element No. 5

Figure 2.3: Modal curvatures for the exact mode No. 1 of healthy and damaged beam, 20% damage in element Nos. 5 and 12 simultaneously, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements

Figure 2.4: Difference in the modal curvatures of damaged and healthy beam, exact mode No. 1, 20% damage in element Nos. 5 and 12 simultaneously, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements

Figure 2.5: Modal curvatures for exact mode No. 6 of healthy and damaged beam, 20% damage in element Nos. 5 and 12 simultaneously, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements

Figure 2.6: Difference in the modal curvatures of damaged and healthy beam for exact mode No. 6, 20% damage in element Nos. 5 and 12 simultaneously, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements

Figure 2.7: Difference in the modal curvatures of damaged and healthy beam for mode No. 1 with 2% measurement error, 50% damage in element Nos. 5 and 12 simultaneously, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements

Figure 2.8: Difference in the modal curvatures of damaged and healthy beam for mode No. 6 with 2% measurement error, 50% damage in element Nos. 5 and 12 simultaneously, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements

Figure 2.9: Maximum differences in the columns of flexibility matrices of damaged and healthy beam, 70% damage in element No. 5; accurate mode shapes.

Figure 2.10: Maximum differences in the columns of flexibility matrices of damaged and healthy beam, 70% damage in element No. 5; 2% error in mode shapes.

Figure 2.11: Maximum differences in the columns of flexibility matrices of damaged and healthy beam, 70% damage in element Nos. 5 and 12; accurate mode shapes.

Figure 2.12: Maximum differences in the columns of flexibility matrices of damaged and healthy beam, 70% damage in element Nos. 5 and 12; 2% error in mode shapes.

Figure 2.13: Differences in uniform flexibility curvatures of healthy and damaged beams with 50% damage in each of Element Nos. 5 and 12; accurate mode shapes.

Figure 2.14: Differences in uniform flexibility curvatures of healthy and damaged beams with 50% damage in each of Element Nos. 5 and 12; 2% error in mode shapes.

Figure 2.15: Damage indices calculated from 6 modes of the beam, damage 20% in Element Nos. 5 and 12, accurate mode shapes, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements

Figure 2.16: Damage indices calculated from 6 modes of the beam, damage 20% in Element Nos. 5 and 12, errors 2% in mode shapes, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements

Figure 2.17: Damage factors calculated from 6 modes of the beam, damage 20% in Element Nos. 5 and 12, accurate mode shapes, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements

Figure 2.18: Damage factors calculated from 6 modes of the beam, damage 20% in Element Nos. 5 and 12, errors 2% in mode shapes, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements.

Figure 2.19: Damage indices calculated from 6 modes of the beam, damage 50% in Element Nos. 5 and 12, errors 2% in mode shapes, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements.

Figure 2.20: Damage indices calculated from 6 modes of the beam, damage 70% in Element Nos. 5 and 12, errors 2% in mode shapes, modal curvatures from translational mode shape.

Figure 2.21: Damage factors calculated from 6 modes of the beam, damage 70% in Element Nos. 5 and 12, errors 2% in mode shapes, modal curvatures from translational mode shape.

Figure 2.22: Damage indices calculated from six modes of the beam; damage 20% in Element No. 5 and 70% in Element No. 12, accurate mode shapes, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements.

Figure 2.23: Damage indices calculated from six modes of the beam; damage 20% in Element No. 5 and 70% in Element No. 12, 2% error in mode shapes, (a) modal curvatures from translational mode shape, (b) modal curvatures from strain measurements.

Figure 2.24: Absolute sums of modal residuals corresponding to degrees of freedom from six modes of a beam; complete and accurate modes, damage 20% in element No. 5 and 30% in element No. 12

Figure 2.25: Absolute sums of the modal residuals of 6 beam modes, accurate but incomplete modes; damage 20% in element No. 5 and 30% in element No. 12

Figure 2.26: Absolute sums of modal residuals from six beam modes; incomplete modes with measurement errors; damage 20% in Element No. 5 and 30% in element No. 12, 2% error in mode shapes.

Figure 2.27: Perturbations in the stiffness matrix obtained from six complete and accurate modes of a beam, damage 30% in element 5 and 20% in element 12.

Figure 2.28: Perturbations in the stiffness matrix obtained from six incomplete modes of the beam, 2% random errors in mode shapes, damage 30% in element 5 and 20% in element 12.

Figure 2.29: Damage factors for a beam; damage 30% in Element No. 5 and 20% in Element No. 12; six full and exact mode shapes used in the non-negative least squares optimization.

Figure 2.30: Damage factors for the beam; damage 30% in Element No. 5 and 20% in Element No. 12; six exact but incomplete mode shapes used in the non-negative least square optimization.

Figure 2.31: Damage factors for the beam; damage 30% in element No. 5 and 20% in element No. 12; six complete mode shapes with 2% error used in the non-negative least square optimization.

Figure 2.32: Damage factors for the beam; damage 30% in element No. 5 and 20% in element No. 12; six incomplete mode shapes used in the non-negative least squares optimization, mode shape errors 2% random.

Figure 2.33: Damage factors for a beam; damage 30% in Element No. 5 and 20% in Element No. 12; six exact but incomplete mode shapes used in the optimization, constraint Eq. 2.39.

Figure 2.34: Damage factors for a beam; damage 30% in Element No. 5 and 20% in Element No. 12; six exact but incomplete mode shapes used in the optimization, constraint Eq. 2.42.

Figure 2.35: Damage factors for a beam; damage 20% in Element No. 5 and 30% in Element No. 12; six incomplete mode shapes used in the optimization, mode shape errors 2% random.

Chapter 3

Application of Artificial Neural Networks in Damage Detection

3.1 Introduction

The brain consists of a large number of highly connected elements called neurons. These neurons have three principal components: the dendrites, the cell body and the axon. The dendrites are tree-like receptive networks of nerve fibers that carry electrical signals into the cell body. The cell body processes the incoming signals and produces an output. The axon is a single long fiber that carries the signal from the cell body out to other neurons. The point of contact between an axon of one cell and a dendrite of another cell is called a synapse. The total number of synapses in the brain exceeds 10^{14} . It is the arrangement of neurons and the strengths of the individual synapses, determined by a complex chemical process, that establishes the function of the neural network (Figure 3.1). Although each neuron is a relatively slow information processing system, the massive parallelism of information processing at many synapses simultaneously leads to an effective processing power which greatly exceeds that of the most powerful present

computer. Some of the neural structure is defined at birth, other parts are developed through learning, and these later changes consist of strengthening or weakening of synaptic junctions.

The conventional approach for computing is based on a series of explicit programmed instructions. Artificial neural networks represent an alternative computational technique in which the solution to a problem is learned from a set of examples. Artificial neural networks are developed based on the biological neural networks. They do not approach the complexity of the brain. However, there are two important similarities between biological and artificial neural networks. First, the building blocks of both networks are simple computational devices that are highly interconnected (although artificial neurons are much simpler than biological neurons). Second, the connections between neurons determine the function of the network.

3.2 Artificial neural network architectures

Figure 3.2 shows the general architecture of a multiple layer artificial neural network, Figure 3.3 describes the architecture through a set of abbreviated notations (Hagan et al, 1995).

Artificial neural networks are constructed imitating the human nervous system. They consist of a complex interconnection of simple units, called neurons. The artificial neural network handles information in such a way that when it receives an input it produces the

target output.

Neurons are usually organized into groups called layers. A typical layer of neurons includes the weight matrix, the summers, the bias vector, the transfer function and the output vector. Different layers can have different numbers of neurons. Each neuron behaves as an independent element of processing with very simple functioning. It carries out the weighted addition of the signals received from other neurons connected to it. If this sum is above a certain threshold the neuron activates its output that is connected to another set of neurons. For the artificial feedforward neural network shown in Figures 3.2 and 3.3, the input vector $\mathbf{p}$ or the output vector $\mathbf{a}^{n-1}$ from the immediately previous layer is multiplied by the weight matrix $\mathbf{W}^n$ to form $\mathbf{W}^1\mathbf{p}$ or $\mathbf{W}^n\mathbf{a}^{n-1}$, which is one of the terms sent to the summer. The other input vector, whose elements are all 1, is multiplied by a bias vector $\mathbf{b}$ and then passed to the summer. The summation of these two terms, often referred to as the net input to the transfer function, produces the output vector $\mathbf{a}^n$. The network of Figure 3.2 has one input layer, one output layer and 2 hidden layers. The number of inputs to the network is R , namely P_1 through P_R . There are S^1 neurons in the first hidden layer, S^2 neurons in the second hidden layer, and S^3 neurons in the output layer. The weight matrix $\mathbf{W}^1$ is of size $S^1 \times R$. The remaining weight matrices, denoted by $\mathbf{W}^n$, have a size S^n by S^{n-1} . Correspondingly, the bias vector $\mathbf{b}^n$ is of size $S^n \times 1$. The output vectors are denoted by $\mathbf{a}^n$ and have size $S^n \times 1$. The summers are denoted by Σ and the processing functions by f^n .

Relating to the biological neuron, the weight matrix in Figures 3.2 and 3.3 corresponds to the strength of the synapses, the summation and the transfer function represent the cell body, and the neuron output a represents the signal on the axon.

3.2.1 The weights and biases

The weights are used to scale the inputs received by a neuron from other neurons or from input to the network. The bias is much like a weight, except that it has a constant input of 1. One can choose neurons with or without biases; however, the networks with biases will be more powerful than those without biases. For example, a neuron without a bias will always have a net input n of zero when the network input $\mathbf{p}$ is zero; this may not be desirable and can be avoided by the use of a bias. The artificial neural network learns by adjusting both the weight matrix and bias vector according to a specified learning algorithm.

3.2.2 The transfer function

The transfer function may be a linear or a non-linear function of the net input. The transfer functions used in this research are described below.

Hard limit transfer function

The hard limit transfer function, shown in Figure 3.4, sets the output of the neuron to 0 if

the function input is less than 0, or to 1 if its input is greater than or equal to 0.

$$a(n) = \begin{cases} 0, & \text{if } n < 0 \\ 1, & \text{if } n \geq 0 \end{cases} \quad (3.1)$$

Neurons with a hard limit transfer function are used to classify inputs into two distinct categories.

Pure linear transfer function

The pure linear transfer function, shown in Figure 3.5, gives an output that is equal to its input.

$$a(n) = n \quad (3.2)$$

Logistic sigmoid function

The logistic sigmoid transfer function, shown in Figure 3.6, takes the form of

$$a(n) = 1/(1+e^{-n}) \quad (3.3)$$

The logistic sigmoid transfer function takes the input, which may have any value between plus and minus infinity, and squashes the output into the range 0 to 1. It is commonly used in multi-layer networks that are trained using the backpropagation algorithm, partially because this function is differentiable, and its derivative can easily be expressed in terms of the function itself.

$$a'(n)=a(n)[1-a(n)] \quad (3.4)$$

Tangent sigmoid function

The tangent sigmoid transfer function, shown in Figure 3.7, is another function often used for the hidden units of a multi-layer network. There are some small practical advantages in using a ‘tanh’ activation function of the form (Bishop, 1995)

$$g(a) \equiv \tanh(a) \equiv \frac{e^a - e^{-a}}{e^a + e^{-a}} \quad (3.5)$$

The above equation differs from the logistic sigmoid function only through a linear transformation. Specifically, an activation function $\tilde{g}(\tilde{a}) = \tanh(\tilde{a})$ is equivalent to an activation function $g(a) = 1/(1 + e^{-a})$ if we apply a linear transformation $\tilde{a} = a/2$ to the input and a linear transformation $\tilde{g} = 2g - 1$ to the output. Thus, a neural network whose hidden units use the Tan-Sigmoid activation function is equivalent to one with hidden units using the logistic sigmoid function but having different values for the weights and biases. Empirically, it is often found that ‘tanh’ activation functions give rise to faster convergence of training algorithms than logistic functions (Bishop, 1995).

The Tan-Sigmoid activation function also has a simple derivative expression, which is useful in training the neural network.

$$g'(a) = 1 - g^2(a) \quad (3.6)$$

3.2.3 Hidden layer

A layer whose output is the network output is called an output layer; other layers are called hidden layers. When using a backpropagation network, one of the most important configuration issues is to select the optimal number of hidden layers. Two hidden layers are recommended for most structural problems by Masri et al. (1996). Hecht-Nilsen (1987) proved that a network with one hidden layer could compute any arbitrary function of its inputs. Most practical neural networks have just two or three layers, four or more layers are used rarely.

3.2.4 Layer neurons

Another problem is the selection of the number of neurons in various layers. In fact, there are few problems for which one can predict the optimum number of neurons needed in a hidden layer. This problem is an active research area. The number of inputs to the network is determined by the problem and the number of neurons in the output layer will be coincident with the number of problem outputs. Choosing the right number of hidden nodes is the most interesting and complicated aspect of network configuring. Different theories have been proposed such as (Hervas Martinez et al, 1992):

1. Hecht-Nielsen ---- use $2N + 1$ hidden nodes, where N is the number of elements in the input vector.
2. Lippmann ---- the maximum number of hidden nodes is $T * (N+1)$, where T is the

number of output neurons and N is the number of elements in the input vector.

3. Kudrychi ---- the maximum number of hidden nodes is $T * 3$.
4. The number of hidden nodes is equal to the geometric mean of the number of elements in the input vector and the number of neurons in the output layer.

3.3 Artificial neural network learning algorithms

The most commonly used feedforward neural network can be regarded as a non-linear mathematical function, which transforms a set of input variables into a set of output variables. The transformation is governed by a set of parameters, namely, the weight matrix and the bias vector. The process of determining the values of these parameters is often called learning or training. The manner in which the parameter changes take place determines the type of learning. A neural network must be trained by a 'learning algorithm'. Once the weights and the biases have been appropriately fixed, new data can be processed by the network and a reasonable output will be given.

There are two main kinds of learning algorithms depending on the a-priori knowledge of the training patterns: supervised and unsupervised learning. In supervised learning, the target outputs are known and are given to the neural network during training so that the neural network can adjust its weight matrix to make its outputs close to the target values. After training, the neural network is tested by giving it only input values, to see how close its output is to the desired values. In unsupervised learning, the neural network is

not provided with the correct target outputs during training. The networks used in this study employ supervised learning.

The network learns the training set after thousands of repeated iterations of the input vectors and comparisons of the output vectors of the training set. Each presentation of one training case and subsequent modification of connection strengths is called a cycle. For back-propagation incremental training algorithm, each cycle actually includes three sub-steps: (1) for the training case to be learned, the network is presented with the input data which then pass on through the hidden processing neurons to the output neurons; (2) the error at the output neurons is then propagated back to the hidden processing neurons; and (3) connections between the neurons in different layers then modify their connection strengths using this back-propagated error. A set of cycles, made up of one cycle for each training case, is called an epoch. The training process for a network may require several hundreds or thousands of epochs for all of the training cases to be learned within a specified error tolerance.

The perceptron learning rule and the backpropagation training algorithm used to train the feedforward neural network in this study are described in the following sections.

3.3.1 Perceptron neural network and learning rule

The perceptron network consists of a single layer of S perceptron neurons, each of which uses a hard-limit transfer function, connected to R inputs through a set of weights W_{ij} , as

shown in Figure 3.8. Each external input is thus weighted with an appropriate weight, and the sum of the weighted external inputs and another input of 1 weighted with a bias b_i is sent to the hard-limit transfer function, which produces a 1 if the net input into the transfer function is equal to or greater than 0, otherwise a 0.

Perceptron network is trained on examples of desired behaviour, which can be represented by a set of input, output pairs

$$\{\mathbf{p}_1, t_1\}, \{\mathbf{p}_2, t_2\}, \{\mathbf{p}_3, t_3\}, \dots, \{\mathbf{p}_q, t_q\}$$

After each input vector $\mathbf{p}$ is presented to the network, the network output is compared to the target. The weight $\mathbf{W}$ and bias vectors $\mathbf{b}$ of the network are updated according to the following expressions in order to move the network output closer to the target.

$$\Delta \mathbf{W} = (\mathbf{t} - \mathbf{a})\mathbf{p}^T = \mathbf{e}\mathbf{p}^T \quad (3.7)$$

and

$$\Delta \mathbf{b} = \mathbf{t} - \mathbf{a} = \mathbf{e} \quad (3.8)$$

where the perceptron error $\mathbf{e}$ is the difference between the target $\mathbf{t}$ and the actual output $\mathbf{a}$ of the network.

The perceptron learning rule can now be written as,

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \mathbf{e}(k)\mathbf{p}^T \quad (3.9)$$

$$\mathbf{b}(k+1) = \mathbf{b}(k) + \mathbf{e}(k) \quad (3.10)$$

where k represents the iteration number.

The process of updating the weights and biases is repeated until the errors approach the

prescribed error goal.

In the present work, perceptron network is used to provide a better estimate of the damage location using damage index as the input.

3.3.2 Backpropagation neural network and training algorithm

Single layer networks are not capable of solving non-linear problems, and multi-layer networks must be used in such cases. The back-propagation neural network is one of the most fully developed and commonly used multi-layer networks. It consists of one output layer of neurons and one or more intermediate layers, referred to as hidden layers. The number of neurons in individual layers may be different, and they may each have a different transfer function. Neurons in each layer are connected to all the neurons in the previous and the subsequent layer. An example of the back-propagation network is shown in Figure 3.9. The network shown has two layers, namely, one hidden layer and one output layer. Each input vector has 15 elements. The hidden layer has 30 neurons and the output layer has 1 neuron. The transfer function in hidden layer is the tangent sigmoid function, given by

$$a = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (3.11)$$

where n is the input and a is the output. The transfer function in the output layer is a pure linear function for which $a = n$.

The training of a neural network is a process to optimize the weight matrix and the bias

vector in order to minimize some specified error function, also called performance function. A commonly used performance function is the sum-of-squared-error function, which is also used in the back-propagation training algorithm. The performance function can be regarded as a performance surface over weight and bias space. The neural network training is equivalent to the search for the global minimum point of this performance surface. For multi-layer neural networks, the performance function is a highly non-linear function of the weights and biases, and the search for the global minimum point is usually done in an iterative manner, starting from some randomly set point W_0 and b_0 in weight and bias space, and then updating it by iteration.

A large number of training algorithms have been developed for feedforward neural networks. They include the perceptron learning algorithm, Hebbian learning algorithm, back-propagation algorithm, Widrow-Hoff learning algorithm and so on. Among these algorithms, the back-propagation algorithm is capable of solving non-linearly separable pattern classification problems; it was first popularized and widely used by Hagan et al. (1995). There exist a variety of training algorithms for back-propagation neural networks. The scaled conjugate gradient algorithm is used in this thesis. Hagan et al. (1995) have described the theoretical background for the back-propagation algorithm in detail. A brief description is presented in the following.

Back-propagation adjusts the weight matrix according to the equation of the following form:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \alpha \mathbf{p}_k \quad \text{or} \quad \Delta \mathbf{W}_k = \mathbf{W}_{k+1} - \mathbf{W}_k = \alpha \mathbf{p}_k \quad (3.12)$$

where $\mathbf{p}_k$ represents a search direction, and the positive constant α is the learning rate used to determine the length of the step.

When the weight matrix or bias vector is updated iteratively, the performance function should decrease, that is,

$$E(\mathbf{W}_{k+1}) < E(\mathbf{W}_k) \quad (3.13)$$

In the back-propagation training algorithm, the performance function is given by,

$$E(\mathbf{W}_k) = (\mathbf{t}(k) - \mathbf{a}(k))^T (\mathbf{t}(k) - \mathbf{a}(k)) \quad (3.14)$$

where $\mathbf{a}(k)$ is the output of the neural network, and $\mathbf{t}(k)$ is the corresponding target output at iteration k .

Applying the Taylor series expansion to the performance function, and only considering the first-order term, we have,

$$E(\mathbf{W}_{k+1}) = E(\mathbf{W}_k + \Delta \mathbf{W}_k) \approx E(\mathbf{W}_k) + \mathbf{g}_k^T \Delta \mathbf{W}_k \quad (3.15)$$

where $\mathbf{g}_k$ is the gradient of the performance function at point $\mathbf{w}_k$, so that

$$\mathbf{g}_k = \nabla E(\mathbf{w}) \big|_{\mathbf{w}=\mathbf{w}_k} \quad (3.16)$$

To make sure that $E(\mathbf{w}_{k+1})$ is less than $E(\mathbf{w}_k)$, the second term in the Taylor series expansion must be negative, that is,

$$\mathbf{g}_k^T \Delta \mathbf{W}_k = \alpha \mathbf{g}_k^T \mathbf{p}_k < 0 \quad (3.17)$$

$$\mathbf{g}_k^T \mathbf{p}_k < 0 \quad (3.18)$$

when $\mathbf{p}_k$ points in the steepest descent direction, namely,

$$\mathbf{p}_k = -\mathbf{g}_k \quad (3.19)$$

Therefore, Equation 3.12 becomes,

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \alpha \mathbf{g}_k \quad (3.20)$$

Similarly, for the bias vector, we have,

$$\mathbf{b}_{k+1} = \mathbf{b}_k - \alpha \mathbf{g}_k \quad (3.21)$$

For a specific weight or bias in the m th layer, the above equations will reduce to,

$$w_{ij}^m(k+1) = w_{ij}^m(k) - \alpha \frac{\partial E}{\partial w_{ij}^m} \quad (3.22)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial E}{\partial b_i^m} \quad (3.23)$$

For multiple layer neural networks, we use chain rule to find the derivatives in the above equations.

$$\frac{\partial E}{\partial w_{ij}^m} = \frac{\partial E}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial w_{ij}^m} \quad (3.24)$$

$$\frac{\partial E}{\partial b_i^m} = \frac{\partial E}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial b_i^m} \quad (3.25)$$

where n_i^m is the net input to the i th neuron in the m th layer given by,

$$n_i^m = \sum_{j=1}^{S^{m-1}} w_{ij}^m a_j^{m-1} + b_i^m. \quad (3.26)$$

Thus, we obtain,

$$\frac{\partial n_i^m}{\partial w_{ij}^m} = a_j^{m-1} \quad (3.27)$$

$$\frac{\partial n_i^m}{\partial b_i^m} = 1 \quad (3.28)$$

If we denote the sensitivity of the performance function E to change of the net input to the i th neuron in the layer m as s_i^m , we have,

$$s_i^m = \frac{\partial E}{\partial n_i^m} \quad (3.29)$$

Equations (3.24) and (3.25) will reduce to,

$$\frac{\partial E}{\partial w_{i,j}^m} = s_i^m a_j^{m-1} \quad (3.30)$$

$$\frac{\partial E}{\partial b_i^m} = s_i^m \quad (3.31)$$

Equations (3.22) and (3.23) become,

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha s_i^m a_j^{m-1} \quad (3.32)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha s_i^m \quad (3.33)$$

We express the above equations in matrix form,

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T \quad (3.34)$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m \quad (3.35)$$

For the multilayer neural network, we will calculate $\mathbf{s}^m$ from the Jacobian matrix given below,

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} = \begin{bmatrix} \frac{\partial n_1^{m+1}}{\partial n_1^m} & \frac{\partial n_1^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_1^{m+1}}{\partial n_{s^m}^m} \\ \frac{\partial n_2^{m+1}}{\partial n_1^m} & \frac{\partial n_2^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_2^{m+1}}{\partial n_{s^m}^m} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_1^m} & \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_{s^m}^m} \end{bmatrix} \quad (3.36)$$

For the i, j element, we have

$$\begin{aligned} \frac{\partial n_i^{m+1}}{\partial n_j^m} &= \frac{\partial (\sum_{r=1}^{s^m} w_{i,r}^{m+1} a_r^m + b_i^{m+1})}{\partial n_j^m} = w_{ij}^{m+1} \frac{\partial a_j^m}{\partial n_j^m} \\ &= w_{ij}^{m+1} \frac{\partial f^m(n_j^m)}{\partial n_j^m} = w_{ij}^{m+1} \dot{f}^m(n_j^m) \end{aligned} \quad (3.37)$$

The Jacobian matrix can be written in the matrix form as,

$$\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} = \mathbf{W}^{m+1} \dot{\mathbf{F}}^m(\mathbf{n}^m) \quad (3.38)$$

where $\dot{\mathbf{F}}^m(\mathbf{n}^m)$ is a diagonal matrix, whose elements are $\dot{f}^m(n_1^m) \quad \dot{f}^m(n_2^m) \quad \dots \quad \dot{f}^m(n_{s^m}^m)$

The sensitivity of the performance function to the change of the input to a neuron in the m th layer is thus given by,

$$\mathbf{s}^m = \frac{\partial E}{\partial \mathbf{n}^m} = \left(\frac{\partial \mathbf{n}^{m+1}}{\partial \mathbf{n}^m} \right)^T \frac{\partial E}{\partial \mathbf{n}^{m+1}} = \dot{\mathbf{F}}^m(\mathbf{n}^m) (\mathbf{W}^{m+1})^T \mathbf{s}^{m+1} \quad (3.39)$$

for $m = M-1, M-2, \dots, 2, 1$, where M is the total number of layers.

The sensitivity for the last layer of the neurons is derived from

$$\mathbf{s}^M = -2 \dot{\mathbf{F}}^M(\mathbf{n}^M) (\mathbf{t} - \mathbf{a}) \quad (3.40)$$

The sensitivities are then back propagated starting from Equation (3.40).

Using Equation (3.39), we can get the sensitivity of the performance function to the change of the input to a neuron in any layer. Hence, using equations (3.34) and (3.35) the weight matrix and bias vector can be updated iteratively.

Limitations of the backpropagation network include long training times for large

networks, and convergence to a local rather than the global minimum. The basic backpropagation algorithm has been modified in different ways in an attempt to circumvent these limitations.

The basic backpropagation algorithm adjusts the weights in the steepest descent direction (negative of the gradient). This is the direction in which the performance function is decreasing most rapidly. It turns out that, although the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence. In the conjugate gradient algorithms, a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions. In most of the training algorithms, a learning rate is used to determine the length of the weight update (step size). In the conjugate gradient algorithms, the step size is adjusted at each iteration. A search is made along the conjugate gradient direction to determine the step size, which minimizes the performance function along that line. Some search functions are best suited to certain training functions, although the optimum choice can vary according to the specific application. Four different conjugate gradient algorithms have been proposed. They are the Fletcher-Reeves algorithm, the Polak-Ribière algorithm, the Powell-Beale algorithm, and scaled conjugate gradient algorithms. All of the conjugate gradient algorithms start out by searching in the steepest descent direction (negative of the gradient) on the first iteration. After this, the former three conjugate gradient algorithms require a line search at each iteration to determine

the optimal distance to move along the current search direction. This line search is computationally expensive. The scaled conjugate gradient algorithm (SCG) used in this study, developed by Moller (1993), was designed to avoid the time-consuming line search. This algorithm is too complex to explain in a short section, but the basic idea is to combine the model-trust region approach (used in the Levenberg-Marquardt algorithm), with the conjugate gradient approach.

3.3.3 Generalization of the trained neural network

During training, the network self-adjusts its connection weights until it is able to reproduce the target diagnosis for the training samples with high accuracy. If the training set contains all the necessary information relevant to making a correct diagnosis, the network should be capable of not only reproducing the correct values for the damage cases used in training, but also giving reasonable outputs for new data unseen during training. This is called the generalization of the neural network.

Over-fitting is a well-known phenomenon where a learning algorithm fits the parameters of a network model too close to the training set; the error on the training set is driven to a very small value. Thus, the network memorizes the training examples very well, but it has not learned to generalize for new situations, so that when new data is presented to the network the error is large.

To avoid over-fitting during training and to increase the generalization capability of the

proposed neural network, a validation data set is used in this thesis. The error on the validation set is monitored during the training process. The validation error will first decrease, as does the training set error. When the network begins to overfit the training data, the error on the validation set will become unstable. If the validation error keeps increasing for a specific number of consecutive iterations, the training will be stopped, and the weights and biases when the validation error is minimum are returned.

3.4 The design of the proposed artificial neural network

The perceptron neural network technique is one way to locate the damage by accepting the damage indices as input to the network in this study. Once the damage location has been determined either through a direct application of damage indices or by using a perceptron neural network, the backpropagation neural network is used to determine the magnitude of damage.

The neural network design is carried out using a series of steps: choosing of the neural network architecture, choosing the activation functions and choosing the learning algorithms. It is then trained, which requires the creation and pre-processing of data sets, and finally tested.

3.4.1 Creating the training samples

Generally, an ideal set of training samples should form a complete combination, i.e., be

able to cover all the possibilities. However, for the case that there are m parameters and each parameter may take n discrete values, the number of all the possible combinations is n^m . This number is prohibitively large and it is impractical to include all the samples for the damage identification problems, particularly when the measurement errors are taken into account. In this study, the training samples are generated from the analysis of a computer model in which simulated damage has been introduced. Such an analysis provides the damage indices that serve as input to the neural network. Input vectors are generated for a range of induced damage in selected element(s). The damage magnitude is varied from 0 to 60% in equal steps.

The samples used to train the neural networks are structure-dependent. They can be developed either naturally, as more damage states are experienced by the real structure, or by carrying out a test that records the variations in vibrational signatures resulting from various damage states introduced to a physical model of the structure. Both of these approaches are time consuming or costly, and sometimes may be difficult to implement. Analytical generation of training samples through analytical modeling of the structure is a viable alternative. Accurate prediction of the dynamic properties of an actual structure by any analytical model is a difficult task. However, for the purpose of damage diagnosis, only the changes in dynamic properties rather than their absolute values need to be considered. Therefore, generation of training samples through analytical studies of a mathematical forward model of a structure is a very attractive

approach. In this study, the neural networks were trained, and validated if necessary during training, with the sample cases generated from the finite element model of the structure.

It is widely recognized that the natural frequencies are least contaminated by measurement noise and can generally be measured with good accuracy. Messina et al. (1996) suggest a standard error of $\pm 0.15\%$ as a benchmark figure for natural frequencies measured in the laboratory with the impulse hammer technique. In contrast, modal damping and mode shape estimates have error levels as much as 20 times more than those in the corresponding natural frequency estimates (Farrar and Cone, 1995). In this study, noise-contaminated natural frequencies and mode shapes, obtained by adding some artificial random noise to the computer-generated natural frequencies and mode shapes, are used to simulate experimentally recorded data that are contaminated with measurement errors. Maximum measurement errors on the mode shape elements are assumed to be $\pm 5\%$. Also, it is assumed that only the translational mode shapes or part of translational mode shapes are measured, except in the case of a girder in which dynamic strain measurements are also considered. The training samples, and the validation samples if necessary, used during training of the artificial neural network are generated based on these inaccurate and incomplete mode shapes.

3.4.2 Pre-processing

In theory, a multilayered network should be able to produce the desired outputs from any input representation that contains the relevant information. In practice, careful pre-processing of the inputs is usually required to obtain an efficient network.

Different methods are available for removing the insignificant characteristics of learning examples. Selection of a particular method depends on the type of problem. In this study, a normalization technique was applied to pre-process the inputs of training data making the largest damage index to be 1.

3.4.3 Choosing the proper learning algorithm

For the perceptron neural network, the options on learning algorithm are limited. The perceptron learning rule is used in this study. For the backpropagation neural network, a large number of learning algorithms have been developed; it is very difficult to know which learning algorithm will be the most effective for a specific problem. Many factors influence the adequate choice of the learning algorithm, such as the complexity of the problem, the size of the training set, the number of parameters in the network, the error goal, the type of the problem (function approximation or pattern recognition), and so on. By taking six different problems as examples, Demuth et al. (2002) performed a number of comparisons of the various training algorithms that could be used for a backpropagation network. Based on these experiments, they found that the resilient

backpropagation training algorithm and the scaled conjugate gradient training algorithm were the two most suitable for the pattern recognition problems. The resilient backpropagation training algorithm is the fastest algorithm on pattern recognition problems. Its performance degrades as the error goal is reduced. This algorithm needs relatively small memory compared to other algorithms considered. The scaled conjugate gradient algorithm performs well for a wide variety of problems, in particular for the networks with a large number of parameters, namely weights and biases. Its speed to convergence is almost as fast as the resilient backpropagation algorithm, but its performance is better when the error is reduced. It needs relatively modest memory to execute training.

In view of the above conclusion, in this study, both the resilient backpropagation training algorithm and the scaled conjugate gradient algorithm are employed on each of the different models, and then the best one is selected by comparing their performance for each model.

3.4.4 Determining the network architecture

In addition to an adequate size for training samples and the selection of an efficient learning algorithm, the proper network architecture is another key factor in obtaining a reliable network model.

The perceptron neural network has only one layer, namely, the output layer with the

hard-limit transfer function. The number of neurons in the layer depends on the specific problem. The structure configuration of the backpropagation neural network is first assumed to be S-2S-1 or S-2S-2S-2 with the tangent sigmoid transfer function in hidden layers, and a pure linear transfer function in output layer. Here S represents the number of elements considered in the finite element model of the structure; it is also the number of elements in the input damage index vector. The input matrix to the network is formed by a large number of S-element input vectors. Corresponding to each individual input vector, the network produces a one- or two-element output, which represents the damage factor. 2S stands for the number of neurons in the hidden layer. This number is picked by trial and experience. If the network has trouble learning, then the architectural parameters of the network, such as the number of hidden layers, the number of neurons in each layer, and the transfer function can be adjusted.

Figure 3.1: Biological neurons

Figure 3.2: Three layer artificial neural network

Figure 3.3: Three layer artificial neural network, abbreviated notation

Figure 3.4: Hard limit transfer function

Figure 3.5: Pure linear transfer function

Figure 3.6: Log-sigmoid transfer function

Figure 3.7: Tan-sigmoid transfer function

Figure 3.8: Architecture of perceptron neural network

Figure 3.9: An example of back-propagation artificial neural network

Chapter 4

Case Study

Damage Detection in a Girder Bridge

4.1 Introduction

The application of a damage detection technique, which combines the use of damage indices to locate the damage and artificial neural networks to determine its magnitude, is presented in this chapter. The technique is applied to a simple girder model of an actual bridge named Crowchild Bridge, located in Calgary, Alberta, Canada. A finite element girder model of the Crowchild Bridge is developed assuming it is rigid in the transverse direction. The proposed method is applied to this model to verify its applicability to the detection of the various simulated states of damage. The required data are obtained through computer simulation by finite element analysis. In practice, the experimental modal data could be contaminated by measurement errors and modal extraction errors. Complete modal data generally cannot be obtained, because it would often require a large number of sensors. In addition, only a few of lowest modes can be measured. Consequently, in this study, measurement errors in mode shapes and limited modal data

are taken into account when the analytical data are used.

4. 2 Modelling of the Crowchild Bridge

Crowchild Bridge (Figures 4.1 and 4.2), rebuilt using the technology proposed by the Intelligent Sensing for Innovative Structures (ISIS) network of centres of excellence in Canada, is a two-lane one-way traffic overpass located in northwest Calgary, Alberta. It has three continuous spans whose lengths are 29.83 m, 32.818 m, and 30.23 m, respectively. The original superstructure of the bridge was found to be under-strength due to aging and increased traffic load. The new superstructure, completed in mid-August, 1997, is said to have the first continuous-span steel-free bridge deck in the world (Afhami and Cheng, 1999). Instead of internal steel reinforcement in the deck, evenly spaced transverse steel straps are placed across the top of the girders to provide lateral restraint to the supporting girders. The straps are integrated with the concrete slab and girders by stainless steel studs. The steel-free design for a concrete deck eliminates the problem of bridge deck deterioration due to corrosion of the reinforcing steel.

The 9030 mm wide concrete slab deck, reinforced with polypropylene fibres, is supported by five 900 mm deep longitudinal steel girders spaced 2 m apart. The slab overhang and barriers on each side are reinforced with glass fibre rods. Three types of girder sections are used in the bridge; two for the positive moment region, and one for the negative moment region on either side of the piers. There are four evenly spaced

cross-frames in each span and steel girder diaphragms at the supports to hold the main longitudinal girders in place. Five bearings at each abutment and two bearings at each pier support the superstructure of the bridge. The slab thickness is 275 mm along the girders and 185 mm elsewhere. To better understand the static and dynamic behaviour of the new steel-free bridge deck under vertical loads with time, a monitoring program has been developed at the University of Alberta in collaboration with ISIS Canada, City of Calgary, and SPECO consultant. The bridge was extensively instrumented by the University of Alberta and University of British Columbia in 1997 and 1998.

The Crowchild Bridge is modeled as a continuous girder. Using the 2D beam elements of the COSMOS (Structural Research & Analysis Corp., Santa Monica, CA. USA) finite element package, as shown in Figure 4.3, the bridge model shown in Figure 4.4 is obtained. The physical properties of the model are adjusted such that the dynamic characteristics of the bridge model match, as closely as possible, those obtained from the field ambient vibration measurements carried out by University of Alberta in August 1998 (Table 4.1) before the bridge was opened to traffic. The material properties and section properties of the girder, shown in Tables 4.2 and 4.3, are calculated under the assumptions that the barriers are continuous, and the slab deck acts compositely with the barriers and five girders. The barriers are included in the analysis because they act to stiffen the deck even though they are generally ignored in calculating the section properties for strength design. Two pin supports of the girder model represent the two

piers; the north and south abutments are modelled as roller supports. Five beam elements per span are used for a total of fifteen elements connected to the 16 nodes. Each intermediate node has three degrees of freedom: a horizontal translation, a vertical translation, and a rotation. Each pier node has only the rotational degree of freedom. Each abutment node has two degrees of freedom, a horizontal translation and a rotation. The stiffness matrix of the beam element shown in Figure 4.3 is given by,

$$[K] = \begin{matrix} & \begin{matrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{matrix} \\ \begin{bmatrix} EA/L & 0 & 0 & -EA/L & 0 & 0 \\ 0 & 12EI_z/L^3 & 6EI_z/L^2 & 0 & -12EI_z/L^3 & 6EI_z/L^2 \\ 0 & 6EI_z/L^2 & 4EI_z/L & 0 & -6EI_z/L^2 & 2EI_z/L \\ -EA/L & 0 & 0 & EA/L & 0 & 0 \\ 0 & -12EI_z/L^3 & -6EI_z/L^2 & 0 & 12EI_z/L^3 & -6EI_z/L^2 \\ 0 & 6EI_z/L^2 & 2EI_z/L & 0 & -6EI_z/L^2 & 4EI_z/L \end{bmatrix} \end{matrix}$$

where L = length of element

I_z = second moment of area about the z axis

A = cross-sectional area

E = elastic modulus

A lumped mass matrix, in which the mass is lumped along the translational degrees of freedom, is used. The dynamic characteristics of the girder model match the experimental data reasonably well, error in the first three frequency estimates being less than 0.96%. Since some matrices required for the VBDI technique are not available from

the COSMOS package, a Matlab 6.5 (The MathWork Inc.) program is written to calculate the damage index vectors corresponding to various simulated damage scenarios. These damage index vectors form the training data set of the neural network.

In this study, the developed finite element girder model is considered as the baseline reference of the healthy bridge. Finite element computer programs have been developed in MATLAB to implement the proposed VBDI algorithm and verify its efficiency in the detection of various simulated damage scenarios suffered by the bridge.

As mentioned earlier, Messina *et al.* (1996) suggest a standard error of $\pm 0.15\%$ as a benchmark figure for natural frequencies measured in the laboratory with the impulse hammer technique. In contrast, modal damping and mode shape estimates may have error levels as much as 20 times worse (Farrar *et al.* 1995) than those in the corresponding natural frequency estimates. So, in this study, the maximum error range in mode shape is considered as being $\pm 5\%$.

4.3 Modal Reduction and Expansion Techniques

Vibration based damage detection depends on the measurement of frequencies and mode shapes of both the undamaged and the damaged structure. Measurements on the undamaged structure are required for the development of a correlated finite element model. Measurements on the damaged structure are required to detect changes in the vibration properties caused by damage. A real structure will usually possess a large

number of degrees of freedom and hence a large number of mode shapes. However, the higher frequencies and mode shapes can rarely be measured with sufficient accuracy. Vibration based damage detection must therefore depend on the measurements of a limited number of the lowest vibration frequencies, or such frequencies and the associated mode shapes. In general, the measurement of mode shapes is more difficult than the measurement of frequencies. On the other hand, frequencies alone may not be sufficient for a reasonably accurate assessment of the location and severity of the damage. It is often very difficult or impracticable to measure the response along all of the degrees of freedom necessary for the complete definition of a given mode shape. Two different strategies are available to address this problem. Thus, the finite element model used in the damage detection may be reduced to correspond to the degrees of freedom along which the measurement of mode shapes is made by using any one of the standard methods of reduction. Alternatively, the partial mode shapes obtained from measurements may be expanded to the full set of the degrees of freedom in the finite element model by using a scheme of extrapolation or model expansion.

A number of techniques have been developed for model reduction or expansion. They include the static condensation, the system equivalent reduction expansion process (SEREP), the Guyan reduction, dynamic condensation, and improved reduction techniques. Here, only the first two methods are described briefly.

4.3.1 Static condensation

This method is possibly the most popular and the simplest. The basic assumption of static condensation is that the mass of the structure can be lumped at only some of the translational degrees of freedom that are preserved in the reduced model and the mass along the remaining translational degrees of freedom as well as the rotational degrees of freedom can be ignored because the kinetic energy component corresponding to those degrees of freedom is small in comparison with the included translational degrees of freedom. As the number of degrees of freedom at which the mass is lumped reduces, the computational effort is also reduced. However, at the same time it becomes more probable that the frequencies and mode shapes are predicted with less accuracy. The displacement vector is partitioned so that u_a represents the retained degrees of freedom and u_c represents the condensed degrees of freedom, and the stiffness and mass matrices are partitioned accordingly. Assuming no external load corresponding to the condensed degrees of freedom, the equation of un-damped free vibrations can then be written as

$$\begin{bmatrix} \mathbf{M}_{aa} & \mathbf{M}_{ac} \\ \mathbf{M}_{ca} & \mathbf{M}_{cc} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_a \\ \ddot{\mathbf{u}}_c \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{aa} & \mathbf{K}_{ac} \\ \mathbf{K}_{ca} & \mathbf{K}_{cc} \end{bmatrix} \begin{bmatrix} \mathbf{u}_a \\ \mathbf{u}_c \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad (4.1)$$

The relationship between the retained and the condensed degrees of freedom is expressed as follows.

$$\begin{bmatrix} \mathbf{u}_a \\ \mathbf{u}_c \end{bmatrix} = \mathbf{T}[\mathbf{u}_a] \quad (4.2)$$

Using this relationship in the generalized eigen-value problem and multiplying both sides by $\mathbf{T}^T$ we get the reduced stiffness matrix is

$$\mathbf{K}_r = \mathbf{T}^T \mathbf{K} \mathbf{T} \quad (4.3)$$

For the lumped mass matrix, in which the mass is lumped only along the retained degrees of freedom, the sub-matrices $\mathbf{M}_{ac}$, $\mathbf{M}_{ca}$, and $\mathbf{M}_{cc}$ are equal to zero, and accordingly the second set of equations of un-damped free vibration can be written as

$$\mathbf{K}_{ca} \mathbf{u}_a + \mathbf{K}_{cc} \mathbf{u}_c = \mathbf{0} \quad (4.4)$$

From the above equation, we can solve for the condensed degrees of freedom in terms of the retained degrees of freedom as follows.

$$\mathbf{u}_c = -\mathbf{K}_{cc}^{-1} \mathbf{K}_{ca} \mathbf{u}_a \quad (4.5)$$

Thus, the static transformation matrix $\mathbf{T}$ between the full displacement vector and the retained degrees of freedom is established as follows.

$$\begin{bmatrix} \mathbf{u}_a \\ \mathbf{u}_c \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{K}_{cc}^{-1} \mathbf{K}_{ca} \end{bmatrix} [\mathbf{u}_a] = \mathbf{T}[\mathbf{u}_a] \quad (4.6)$$

The reduced stiffness matrices can then be written explicitly as follows.

$$\mathbf{K}_r = \mathbf{T}^T \mathbf{K} \mathbf{T} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{K}_{cc}^{-1} \mathbf{K}_{ca} \end{bmatrix}^T \begin{bmatrix} \mathbf{K}_{aa} & \mathbf{K}_{ac} \\ \mathbf{K}_{ca} & \mathbf{K}_{cc} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ -\mathbf{K}_{cc}^{-1} \mathbf{K}_{ca} \end{bmatrix}$$

$$\begin{aligned}
&= \left(\mathbf{K}_{aa} - \mathbf{K}_{ac} \mathbf{K}_{cc}^{-1} \mathbf{K}_{ca} \right) \begin{bmatrix} \mathbf{I} \\ -\mathbf{K}_{cc}^{-1} \mathbf{K}_{ca} \end{bmatrix} \\
&= \mathbf{K}_{aa} - \mathbf{K}_{ac} \mathbf{K}_{cc}^{-1} \mathbf{K}_{ca}
\end{aligned} \tag{4.7}$$

4.3.2 System equivalent reduction expansion process (SEREP)

Avitabile et al. (1989) have provided a comparison of the various techniques for model reduction or expansion. Among all of the techniques, the SEREP method offers significant advantage when complete analytical mode shapes are available. A brief description of the method follows. For a more detailed description, reference may be made to O'Callahan et al. (1989).

The equation of undamped free vibrations of an N degree of freedom system is given by

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{0} \tag{4.8}$$

The displacements $\mathbf{u}$ can be expressed in terms of the m measured mode shapes and the corresponding modal coordinates as

$$\begin{bmatrix} \mathbf{u}_a \\ \mathbf{u}_r \end{bmatrix} = \begin{bmatrix} \Phi_a \\ \Phi_r \end{bmatrix} \mathbf{y} \tag{4.9}$$

where the displacements as well as the mode shapes have been partitioned along the active or measured degrees of freedom a and the deleted or unmeasured degrees of freedom r , and $\mathbf{y}$ are the modal coordinates. We assume that the complete analytical mode shapes of the healthy structure included in Eq. 4.9 are available. The first partition

in Eq. 4.9 gives

$$\mathbf{u}_a = \Phi_a \mathbf{y} \quad (4.10)$$

Matrix Φ_a is a rectangular matrix of size a by m , in which a is, in general, significantly larger than m . An inverse of the matrix therefore does not exist. However by taking a generalized inverse the modal coordinates $\mathbf{y}$ can be expressed in terms of $\mathbf{u}_a$.

$$\mathbf{y} = (\Phi_a^T \Phi_a)^{-1} \Phi_a^T \mathbf{u}_a = \Phi_a^g \mathbf{u}_a \quad (4.11)$$

Using Equation 4.11 we can express the inactive degrees of freedom in terms of the active degrees of freedom.

$$\mathbf{u}_r = \Phi_r \mathbf{y} = \Phi_r \Phi_a^g \mathbf{u}_a \quad (4.12)$$

The complete displacement vector is now given by

$$\begin{bmatrix} \mathbf{u}_a \\ \mathbf{u}_r \end{bmatrix} = \begin{bmatrix} \mathbf{I}_a \\ \Phi_r \Phi_a^g \end{bmatrix} \mathbf{u}_a = \mathbf{T} \mathbf{u}_a \quad (4.13)$$

where $\mathbf{I}_a$ is an identity matrix of size a by a , and $\mathbf{T}$ is a transformation matrix. By using $\mathbf{T}$, the incomplete mode shapes Φ_a measured in the damaged structure can be expanded to their full size.

The requirement of reducing finite element model degrees of freedom or extending the measured modal parameters may result in the loss of physical interpretability as well as errors, which affect the damage-induced localized changes in the stiffness matrix.

4.4 Localization of the simulated damage in Crowchild Bridge

Based on the finite element girder model of the Crowchild Bridge, three different methods, i.e., the damage index method, the substructure method, and the perceptron neural network method, are applied to determine the damage location by using the damage index vectors corresponding to various simulated damage scenarios. All the computer programs used in the computation are written in Matlab.

This research focuses on the identification of simulated damage in one element or simultaneously in two elements.

4.4.1 Use of damage indices based on expansion of incomplete mode shapes

Assuming that only translational mode measurements have been made along selected degrees of freedom, the damage indices are calculated, and plotted for the elements in the finite element girder model. Elements whose damage index is relatively large are likely to have been damaged. There are two ways to obtain the damage index; one is based on complete mode shapes obtained by expanding the measured degrees of freedom, and the other one is based on curvatures obtained from incomplete mode shapes. Figures 4.5 (a) and 4.5 (b) show the damage indices calculated from three modes of the girder model of the Crowchild Bridge, respectively for 50% damage in element 3 and 50% damage in each of elements 3 and 8. The damage indices are obtained by

expanding the incomplete mode shapes using SEREP. It is observed that expansion of translational modal measurements to get complete mode shapes is not entirely successful, even when the measurements are accurate. However, the predicted damage location does lie in the vicinity of the actual damage site.

4.4.2 Damage indices based on curvatures obtained from translational measurements

The method is first applied to localize simulated damage in only one element: element number 3 damaged, or element number 8 damaged. The damage indices are calculated from the strain energy based on the curvature. The curvature is derived from the vertical components of the measured mode shapes by using the central difference technique. To simulate the measurement error, random errors are introduced in the analytical incomplete mode shapes before the curvature calculation. The curvature $\psi'(i)$ at i th node along the finite element girder model of bridge is obtained from

$$\psi'(i) = \frac{y_{i+1} - 2y_i + y_{i-1}}{l^2} \quad (4.14)$$

where y_i is the mode shape vertical displacement at i th node and l is the length of an element.

The strain energy contributed by the j th element lying between nodes i and $i+1$ in m th mode is

$$S_{mj} = \int_a^b EI_j [\psi_{mj}''']^2 dx = EI_j \left[\frac{\psi_m''(i) + \psi_m''(i+1)}{2} \right]^2 l_j \quad (4.15)$$

where a and b are the x coordinates of nodes i and $i+1$, respectively.

Once the strain energies for every finite element before and after damage are obtained, the damage indices can be derived using Eqs. 2.21 and 2.22.

Based on the assumption that only the translational degrees are measured and three modes are available to calculate the damage index, Figures 4.6 through 4.9 present the prediction results for different errors introduced in the mode shapes and damage severities suffered by element 3 or 8. For example, Figure 4.6 (b) is a plot of damage index values for the case of 30% damage in element 3 when three incomplete modes are available and random errors between -2% and 2% are introduced in the elements of the mode shapes. Figures 4.6 (a) through 4.6 (c) show that the larger the random error, the more inaccurate the prediction will be. Figures 4.7 (a) and 4.7 (b) show that if the damage is significant, the random error does not affect the prediction accuracy too much. Again, comparison of Figures 4.6 (c) and 4.7 (b) indicates that the larger the damage, the more accurate the prediction will be. For element eight, Figures 4.8 and 4.9 show similar prediction results.

The damage index method is then applied to identify multiple damage locations. Assuming that element numbers 3 and 8 are damaged simultaneously, the damage indices are calculated from the strain energy based on the curvatures derived from incomplete translational mode shapes. The measurement errors are considered by

introducing random errors into the elements of the mode shapes. Figures 4.10 and 4.11 present the damage indices calculated from three vertical modes for different mode shape errors and damage severities suffered by elements 3 and 8 simultaneously. For example, Figure 4.11 (b) shows the damage indices for the case of $\pm 50\%$ damage in element numbers 3 and 8, calculated from mode shapes with 5% random error.

Similar to the one-element-damage case, it is observed that, the larger the random error, the more inaccurate the prediction would be. If the damage is significant, the prediction would be more accurate, and the random errors would not affect the prediction accuracy too much.

4.4.3 Damage indices based on curvatures obtained from strain measurements

As mentioned in Chapter 2, the curvature can be obtained by attaching strain gages capable of measuring dynamic strains at two points over the depth of the girder. If the depths of two points from an arbitrary datum are y_1 and y_2 and the corresponding strains are ϵ_1 and ϵ_2 , the curvature is given by

$$\phi = \frac{\epsilon_2 - \epsilon_1}{y_2 - y_1} \quad (4.16)$$

For the computer simulations used in this study, the curvature for each element of the finite element girder model is derived from the rotational components of the full analytical mode shapes as follows,

$$\psi^*(i) = \frac{\theta_{i+1} - \theta_i}{l} \quad (4.17)$$

where, $\psi^*(i)$ is the constant curvature along the i th element, θ_{i+1} and θ_i are the rotational modal components at the ends of the i th element, and l is its length.

To simulate measurement errors, random errors are introduced into the analytical mode shapes before the curvature calculation. The strain energy contributed by the j th element in the m th mode can be calculated by Equation 4.15. Once the strain energies for all finite elements before and after damage are obtained, the damage indices can be derived. The above procedure is first applied to localize simulated damage in only one element: element number 3 damaged, or element number 8 damaged. Based on the assumption that only three modes are available to calculate the damage index, Figures 4.12 through 4.15 present the prediction results for different errors introduced in the incomplete mode shapes and for different damage severities suffered by element 3 or 8. For example, Figure 4.12 (b) is a plot of damage index values for the case of 30% damage in element 3 when 3 incomplete modes are available and random errors between -5% and 5% exist in the curvature mode shapes. It is observed that the predictions of the damage location are very accurate even when the damage is not large, and the random errors do not affect the prediction accuracy at all.

The damage index method is then applied to identify multiple damage locations. Assuming that element number 3 and element number 8 are damaged simultaneously, the damage indices are calculated from the strain energy based on the lowest 3 curvature

mode shapes. The measurement errors are considered by introducing random errors into the elements of the mode shapes. Figures 4.16 present the damage indices calculated from three modes for different mode shape errors and 30% damage suffered by elements 3 and 8 simultaneously.

As in the one-element-damage case, it is observed that, the predictions of the damage location are very accurate even for slight damage, and the random errors do not affect the prediction accuracy.

4.4. 4 Substructure method

Civil engineering structures are often very large and are composed of a large number of elements or members. It may not be possible to make modal measurements along the large number of degrees of freedom of the structure. Structural health monitoring therefore becomes much more difficult and expensive both in terms of measurement and the following analysis. Generally, more sensors are needed to obtain meaningful mode shapes, even if the mode shapes include only the translational degrees of freedom in one direction or strain measurements about one axis. In view of the preceding challenges, a strategy that could be used is to divide the structure into a number of substructures, and then to measure the mode shape values along only those degrees of freedom that lie at the junctions of the individual substructures. If the damage indices calculated at the substructure level indicate there is possible damage in one or more substructures, a

second round of modal measurements can be carried out in which additional measurements are made along degrees of freedom that are internal to the affected substructures. This method is examined by taking a 60 m simply supported uniform beam as an example and then applied to the girder model of the Crowchild Bridge to show its practicability.

The unit mass of the 60 m beam is 1290 kg/m, its cross-sectional moment of inertia is 0.00276 m^4 , and the modulus of elasticity of the material is $200 \times 10^9 \text{ N/m}^2$. The beam is first divided into 60 equal elements. Assuming that element numbers 14 and 36 are damaged, the mode shapes corresponding to the healthy and damaged structures are calculated for different levels of damage. In the first step of the substructure method, it is assumed that the modal measurements of strains are made at the centres of 20 equal length elements. Element numbers 14 and 36 of the 60-element model lie in element numbers 5 and 12 of the smaller model, respectively. The curvature for each element can be obtained by using Equation 4.16 from the readings of the strain gauges. In computer simulation, the curvature for each element is calculated from the rotation at its two end nodes by using Equation 4.17. The rotational mode for the 20-element beam can be extracted from the mode shapes for the 60-element beam. To simulate the measurement error, random errors are introduced in the elements of the rotational mode. The damage indices are then derived from the curvatures. Assume that damage of 20% and 30% occur in original element numbers 14 and 36 respectively. The calculated damage

indices are shown in Figure 4.17. Element numbers 5 and 12 are correctly identified as being damaged.

To refine the damage localization, the second step of the substructure method is followed. More measurements are made in the area of initially determined damage location, ignoring most of the measurements at locations far from the damage site as shown in Figure 4.18. The element numbers 14 and 36 of the 60-element beam become element numbers 4 and 11, respectively, in Figure 4.18. The curvatures are obtained by using Equation 4.16 from the readings of the strain gauges or Equation 4.17 from the analytical rotational modes. Figure 4.19 shows that the damaged elements are correctly identified.

4.4.5 Application of substructure method to Crowchild Bridge model

Damage indices calculated from translational measurements

Assume that element number 3 in the girder model of Crowchild Bridge, as shown in Figure 4.4, is damaged. In the first step of the substructure method, the model is re-divided into 12 new elements as shown in Figure 4.20. Original elements 2, 3, and 4 in Figure 4.4 are merged into new element 2. Nine sensors are used to measure the vertical translational degrees of freedom at nine nodes. In this step, we put more sensors on the south span. The damage indices are calculated from the strain energy based on the curvatures obtained from the measured displacement mode shapes. Some results are

presented in Figure 4.21 and 4.22. From these results, it can be concluded that the south span is not damaged; the damage location can be narrowed to the new element No. 2, or to the middle span and the north span depending on the damage severity and the error in the mode shapes.

To accurately locate the damage, nine sensors are placed on the bridge, mainly on the north span and the middle span, and the beam is re-divided into 12 elements as shown in Figure 4.23. As in the first step, the damage indices are calculated. Figure 4.24 and 4.25 show the prediction results; Element No. 3 is clearly identified as being damaged. Spurious damage element is identified in No. 7 when the mode shape error is too large or when the damage is light.

A similar procedure is followed for the case when Element No. 8 is damaged, and similar prediction results are obtained. The results presented here show that the proposed substructure method is feasible. Because this method can greatly reduce the number of sensors, its advantage will be much clearer for complex structures.

The substructure method is next applied to the girder model of the Crowchild Bridge to show its ability to detect damage suffered by two elements simultaneously.

Assume that element numbers 3 and 8 in Figure 4.4 are damaged. As for the one-element-damage case, in the first step of the substructure method, the girder model is re-divided into 12 new elements as shown in Figure 4.20. Nine sensors are used to measure the vertical translational degrees of freedom at nine nodes. In this step, we put

more sensors on the south span. The damage indices are calculated from the strain energy based on the curvature obtained from the measured displacement mode shape. Results presented in Figures 4.26 and 4.27 show that the south span is not damaged; the damage location can be narrowed to the new element numbers 2 and 5, or to the middle span and the north span, depending on the damage severity and the error in the mode shapes.

To further refine the estimate of damage location, nine sensors are placed on the bridge mainly on the north span and the middle span as shown in Figure 4.23. As in the first step, the damage indices are calculated. Figures 4.28 and 4.29 show the prediction results; element No. 3 is clearly identified as being damaged, element No. 8 is also identified provided the mode shape error is not too large for the light damage case.

Damage indices calculated from curvature mode shapes

Assume that element No. 3 in Figure 4.4 is damaged. In the first step of the substructure method, the girder model is re-divided into 10 new elements as shown in Figure 4.30. Original elements 2, 3, and 4 in Figure 4.4 are merged into new element 2. Twenty strain sensors are used to measure the curvatures about Z-axis at the centres of the 10 elements. For computer simulation, the damage indices are calculated from the strain energy based on the curvatures obtained from the rotational mode shapes. Some results are presented in Figures 4.31 and 4.32. It is observed that the damage location can be clearly narrowed

to the new element No. 2.

To refine the location of the damage, twenty strain sensors are placed at centres of elements in the north span and the middle span as shown in Figure 4.33. As in the first step, the damage indices are calculated. Figure 4.34 shows the prediction results. Element No. 3 is clearly identified as being damaged. Results not presented here show that damage in element 8 is also identified correctly. It is evident that the proposed substructure method is feasible. Because this method can greatly reduce the number of sensors required, its advantage will be much clearer for complex structures.

The substructure method is next applied to the girder model of the Crowchild Bridge to show its effectiveness in detecting damage suffered by two elements.

Assume that elements 3 and 8 in Figure 4.4 are damaged. As for one-element-damage case, in the first step of the substructure method, the girder model is re-divided into 10 new elements as shown in Figure 4.30. Twenty sensors are used to measure the curvatures at the centres of 10 elements. For computer simulation, the damage indices are calculated from the strain energy based on the curvatures obtained from rotational mode shapes. Figure 4.35 shows that the damage location can be narrowed to the new element numbers 2 and 5.

To refine the prediction of damage location, twenty sensors are placed at centres of elements mainly on the north span and the middle span of the bridge as shown in Figure 4.33. As in the first step, the damage indices are calculated. Figure 4.36 shows

the prediction results. Element numbers 3 and 8 are accurately identified as being damaged. Compared to the results obtained from the damage index based on the curvature obtained from the translational degrees of freedom, the results shown here are much better.

4.4.6 Perceptron neural network method

Since perceptron neural networks have been found to be effective in solving pattern recognition problems, the feasibility of using such networks in detecting damage location is examined.

Generally, a perceptron neural network has relatively fixed architecture and transfer function. The status of any element in a structure has only two possibilities, healthy or damaged. Similarly, the output of neurons of the perceptron neural network is either 1 or 0. It would therefore appear that the perceptron neural network is suitable for damage localization.

The neural network designed for locating damage in the Crowchild Bridge model, shown in Figure 4.37, has only one layer, namely, the output layer, with 15 neurons, corresponding to the status of 15 finite elements. The damage indices are selected as the input to the network, and each input vector has 15 elements. The output of each neuron is either 1, which indicates damage in the numbered element or 0, which indicates no damage. The transfer function is a hard limit function, which outputs a 1 for an input that

is 0 or positive and a 0 for a negative input.

Creation of the Training Data Set Using Analytical Model

Training data sets are obtained from the incomplete translational mode shapes both with and without error to simulate the actual measurement situation. For each damaged element, the selected damage extent is 0%, 10%, 20%, 30%, 40%, 50%, or 60% to obtain the damage index vectors with error. To increase the weight of the data without error in the training data set, the damage extent is chosen from 0% to 60% at the interval of 5% for the damage index vectors without error. Five error levels, -1% to 1% , -2% to 2% , -3% to 3% , -4% to 4% , and -5% to 5% are introduced in the mode shape elements at each damage scenario using a random number generator whose states are related to the clock. This generator can generate all the floating-point numbers in the closed interval $[2^{(-53)}, 1-2^{(-53)}]$. Theoretically, it can generate over 2^{1492} values before repeating itself. For each error level in the mode shapes, 50 different damage index data sets are created corresponding to each damage scenario. Individual data sets at the same damage scenario and modal error level are different from each other, because at each call to it the random number generator assigns different errors within the selected error range to the mode shape elements.

Training the Neural Network

To create a network that can handle an input obtained from the measured mode shapes, it is best to train the network on both accurate and noisy vectors. To do this, the neural

network is first trained using data set obtained from the simulated noisy mode shapes until it has a low sum-squared error. This forces the network to learn how to deal with noise, a common problem in the real world. The network is next trained with example data set obtained from incomplete accurate mode shapes. The purpose is to maintain the network's ability to deal with the accurate input vectors. After the first two steps of training, the network may have learned to manage some accurate vectors at the expense of properly managing a noisy vector. Therefore, the network is finally trained with data sets obtained from incomplete mode shapes both with and without error. This ensures that the network responds correctly when presented with either correct or noisy damage index vectors.

Testing of the Trained Neural Network

The reliability of the trained networks is measured by testing the networks with damage index vectors obtained for the various mode shape error levels and 6 damage severity ranges, that is, 0% to 10%, 10% to 20%, 20% to 30%, 30% to 40%, 40% to 50%, and 50% to 60%. Take damage severity range 0% to 10% as an example; the testing process is then as follows.

Choose 100 damage magnitudes randomly between 0% and 10%. Introduce random errors ranging from -1% to $+1\%$ in the elements of the structural mode shapes corresponding to the 100 damage magnitudes. Calculate 100 damage index vectors and feed them to the trained network. Compare the output of the network to the actual

damage element vector. The prediction result can be classified into one of the following four groups,

- 1) The prediction is exactly right.
- 2) Two elements are predicted as being damaged, but one of them is the actual damaged element or is adjacent to the latter.
- 3) One element is predicted as being damaged and is adjacent to actual damaged element.
- 4) The prediction is wrong.

The number of network outputs falling into each of the above four groups is summed up, and the statistics of the prediction results are obtained for the damage extent between 0% and 10% with 1% mode shape error.

Repeating the steps outlined above for 0%, 2%, 3%, 4%, and 5% error and the mode shapes corresponding to the 100 damage magnitudes between 0% and 10%, the statistical results for damage extents between 0 and 10% with 0%, 1% 2%, 3%, 4%, and 5% mode errors are obtained. A stack figure is then plotted for damage extent between 0% and 10%.

By adding 10% damage to each damaged element in the previously referred 100 damage magnitudes ranging from 0% to 10%, we obtain models with damage range between 10% and 20%. On following the steps outlined in the previous paragraphs, we can get the stack figure for damage extent between 10% and 20%. In a similar manner, we can

get the stack figures for damage extent ranges, 20% to 30%, 30% to 40%, 40% to 50%, and 50% to 60%.

Figures 4.38 to 4.40 show the statistical results for some cases. The results show that the network is not very successful in predicting low-level damage. The predictions become more reliable when the damage severity increases. As an example, in Figure 4.38 (e) for damage between 40% and 50% in element 1 and 4% mode shape error, out of a total of 100 damage magnitude cases, the trained network gives 67 accurate predictions of damage location, 11 predictions showing 1 element damaged adjacent to the actual damage, 2 predictions indicating 2 elements damaged, one of them adjacent to the real damage, and the remaining 20 giving wrong results. In general, the prediction and generalization capability of the trained neural network is quite good when the damage extent is significant, say greater than 30%.

An attempt was also made to use the perceptron neural network method to localize the damaged elements for two-element-damage case following a procedure similar to that used for the one-element-damage case. The training data set should include as many damage scenarios as possible. For the two-element-damage case, 105 damage scenarios are expected. The training data set will be huge if all scenarios are included, and the training time will be excessive. It therefore appears that the perceptron neural network method is not practical for such cases.

The prediction accuracy of the perceptron network for two-element-damage cases is not

as satisfactory as that for one-element-damage cases. Generally, the damage will occur first in one element, and then in two elements. If monitoring is continuous, damage could be identified confidently in the initial one-element-damage stage.

4.5 Evaluation of the simulated damage extent in Crowchild Bridge

4.5.1 One-element damage case

In this section, the performance of the neural network designed to evaluate the severity of damage suffered by only one element will be examined. The damage indices obtained from translational modes and the corresponding damage extent will be used as the training data pair. Once the neural network is trained, it should be able to output a reasonable estimate of the damage extent when a new damage index vector is presented.

The designed backpropagation neural network for a one-element damage case, shown in Figure 4.41, has two layers, namely, one hidden layer and one output layer. Each input vector has 15 elements, which are the damage indices corresponding to the 15 finite elements of the girder model of the Crowchild Bridge. The hidden layer has 30 neurons and the output layer has one neuron. The transfer function in the hidden layer is the tangent sigmoid function, while in the output layer it is a pure linear function.

An early stopping technique is used to prevent over-fitting in the process of training. Over-fitting the training data results in the loss of the generalization property of the network, so that it gives perfect results only for the training data, while producing

incorrect outputs for inputs that are not in the training set. To avoid over-fitting, the validation data are used to stop training early if further training on the training data will hurt generalization to the validation data. During the training process, the error on the validation set is monitored. The validation error will first decrease, as does the training error. When the network begins to over-fit the training data, the error on the validation set will become unstable. If the validation error keeps increasing for a specific number of consecutive iterations, the training will be stopped, and the weights and biases at the instant when the validation error was a minimum are returned. The other two control factors are the maximum number of training epochs and the performance goal. The performance in training is defined as the mean square error between the actual and target output of the network. Training will stop if a maximum number of epochs occurs, or the performance goal is met. In this study, unless otherwise stated, the maximum number of epochs is set to 5000, the performance goal is set to 0, and the maximum number of iterations during which validation fails is set to be 10.

Assuming that only element 3 has been identified as having been damaged, the damage index vectors are calculated from the strain energy based on the curvature. The translational components of the mode shapes, both with and without error, are used to simulate the actual measurement situation.

Creating the data set used to train the neural network

Training data set: For element number 3, damage extents from 0% to 60% at 5%

interval are selected to obtain the damage index vectors both with and without error. Five error levels, -1% to 1% , -2% to 2% , -3% to 3% , -4% to 4% , or -5% to 5% are introduced in the mode shape elements at each damage scenario using a random number generator whose states are related to the clock. For each error level in the mode shapes, 50 different damage index data sets are created corresponding to each damage scenario. The 50 vectors are different for the same reason as for the perceptron neural network.

Validation data set: To avoid over-fitting and to cause early stopping when necessary, the choice of the validation set is also important. The validation set should be representative of all points in the training set. For element No. 3, damage index vectors are calculated for damage extents that are slightly shifted from those used for training data set. Thus, 2% to 57% damage levels at interval of 5% are selected to calculate the damage index vectors.

Training the neural network

The designed back-propagation neural network is trained in three steps as in the case of perceptron network. The training algorithm is the scaled conjugate gradient algorithm. The training processes, shown in Figure 4.42, indicate the training and validation errors between the actual and target outputs, as a function of the numbers of epochs. The training epochs are 119, 92 and 50 for the 3 different training data sets, respectively. The total training time on a Pentium III computer is 53.5 seconds. From the training record, we can see that the training has been stopped by the validation data set.

Testing the trained neural network

The reliability of the trained networks is measured by testing the networks with damage index vectors obtained for the various mode shape error levels and 6 damage severity ranges, that is, 0% to 10%, 10% to 20%, 20% to 30%, 30% to 40%, 40% to 50%, and 50% to 60%. The creation of the test data set is similar to that in the damage localization for the one-damage-case. These data are fed to the trained network, the output of the network are obtained and compared to the actual damage extent, and the relative prediction errors (RPE) are calculated. The RPE is defined as the difference between the magnitude of predicted damage and the magnitude of actual damage as a percentage of the actual damage. The numbers of outputs within a specific range of RPE are shown in Figure 4.43. As an example, in Figure 4.43 (d) corresponding to damage cases between 30% and 40% in element 3 and 2% mode shape error, for every 100 test data, the trained network gives 72 predictions whose RPEs are less than 10%, 25 predictions with RPE between 10% and 20%, and 3 predictions with RPE between 20% and 30%.

The same procedure is applied to element number 8, the training and test results are shown in Figure 4.44 and Figure 4.45.

From the results presented, it is observed that the prediction and generalization capability of the trained neural network is quite good when the damage extent is significant, say greater than 20%.

4.5.2 Two-element damage case, damage index vectors based on curvature obtained from translational mode shapes

Assume that elements 3 and 8 have been identified as being damaged from the damage index method or substructure method. As in the one-element-damage case, an early stopping technique is used to prevent over-fitting during training. The data pair is composed of the damage index vector and the corresponding damage extent. The damage index vectors are calculated from the strain energy based on the curvature. The incomplete mode shapes, both with and without error, used to simulate the actual measurement situation are related to the 12 translational degrees of freedom.

To find the relatively optimal network architecture, effective size of the training data set, and the training algorithm and transfer function used, so as to improve the generalization capabilities of the trained network, the following cases are explored for optimization.

A. Neural network with architecture of 15-30-30-2

Generally, a larger number of hidden layers make the performance of the network more reliable. Thus, the initially proposed neural network used to evaluate the damage extent is designed with three layers: two hidden layers and one output layer. Each input vector has 15 elements, which are the damage indices corresponding to 15 finite elements. Each of the two hidden layers has 30 neurons and the output layer has two outputs, which are the predicted damage extents. The transfer functions in hidden layers are tangent

sigmoid functions, while in the output layer it is a pure linear function.

Creating the data set used to train the neural network

Training data set: For element number 3 and 8, damage extents from 0% to 60% at 5% interval are selected to obtain the damage index vectors both with and without error. Five error levels, -1% to 1% , -2% to 2% , -3% to 3% , -4% to 4% , or -5% to 5% are introduced in the mode shape elements at each damage scenario using a random number generator whose states are related to the clock. For each error level in the mode shapes, 50 different damage index data sets are created corresponding to each damage scenario. Individual damage index vectors are different for the same reason as for the perceptron neural network.

Validation data set: For element Nos. 3 and 8, damage index vectors corresponding to damage magnitudes that are slightly shifted from those used in the training data set, namely damage extent from 2% to 57% at interval of 5%, are selected to calculate the damage index vectors.

Training the neural network

The designed back-propagation neural network is trained in three steps, as before. The training algorithm is the scaled conjugate gradient algorithm. The training epochs are 225, 153 and 78 for the 3 different training data sets, respectively. The total training time on a Pentium III computer is 2317.2 seconds. The performance of the trained network is 0.003713. From the training record, it is observed that the training is stopped by the

validation data set.

Testing the trained neural network

The reliability of the trained networks is measured by testing the networks with damage index vectors obtained for the various mode shape error levels and 6 damage severity ranges, that is, 0% to 10%, 10% to 20%, 20% to 30%, 30% to 40%, 40% to 50%, and 50% to 60%. The creation of the test data set is similar to that in the damage localization for one-damage-case. These data are fed to the trained network, the output of the network are obtained and compared to the actual damage extent, and the relative prediction errors (RPE) are calculated for element number 3 and 8, respectively. The larger RPE is used in the following statistical process. The RPE is defined as the difference between the magnitude of predicted damage and the magnitude of actual damage as a percentage of the actual damage. The numbers of outputs within a specific range of RPE are summed up. The results are shown in Appendix A, Figure A1. They show that the prediction and generalization capabilities of the trained neural network are quite good when the damage extent is significant.

B. Neural network with architecture of 15-30-2

As the number of hidden layers is increased the training time increases, and the computations may become very expensive for complex structures, although use of more hidden layers makes the performance of the network more reliable. To assess the impact

of the number of hidden layers, the number of hidden layers is reduced by one. The neural network is trained in three steps to get the damage factors. The number of training epochs is 441, 182 and 185 for the 3 different training data sets, respectively. The total training time on a Pentium III computer is 2222 seconds. The performance of the trained network is 0.003687, which is better than that for the network with two hidden layers. Six hundred damage index vectors with different measurement errors in mode shapes are presented to the network. The prediction results are shown in Appendix A, Figure A2. The training time has increased by 4%, but the prediction and generalization capabilities of the trained neural network have improved slightly.

C. Neural network with architecture of 15-30-1

The number of outputs of the neural network used in section B is reduced to one, representing the damage extent of either element number 3 or 8. The training data set and the validation data set are split into two data sets. One is composed of the damage index vectors and corresponding damage extent of element 3; another one is composed of the damage index vectors and corresponding damage extent of element 8. The neural network is trained in the same three steps as in the preceding section using the data sets for element number 3. The trained network is saved and used to predict the damage extent of element number 3. The total training time on an Intel Pentium III computer is 783.25 seconds. Training is continued using the data set for element number 8. The

trained network is again saved and used to predict the damage extent of element number 8. The training time on a Pentium III computer is 538.86 seconds this time. The training process is stopped by the validation error for each of elements 3 and 8. The total training time is 1322.11 seconds, which is far less than the training time for the network in the preceding section.

Six hundred damage index vectors with different measurement errors in mode shapes were presented to the network. The prediction results, shown in Appendix A, Figure A3, have not deteriorated at all.

D. Neural network with architecture of 15-15-1 and 15-45-1

To reduce the training time, fewer neurons should be used in the hidden layer. Assuming that 15 neurons are used in the hidden layer, the neural network is trained with the data set for element number 3 in the same way as in section C. The trained network is saved and used to get the damage extent of element number 3. Training is then continued in a similar manner to get the damage factor of element number 8. The total training time on a Pentium III computer is 616.90 seconds. Six hundred damage index vectors with different measurement errors in mode shapes are presented to the network. The prediction results are shown in Appendix A, Figure A4.

Compared to the trained network with 30 neurons in hidden layer, the training time is reduced significantly, but the performance of the trained neural network is considerably

inferior.

The number of neurons in the hidden layer is next increased to 45. The network is trained in three steps using different data sets. The total training time on a Pentium III computer is 2249.72 seconds. Six hundred damage index vectors with different measurement errors in mode shapes are presented to the network. The prediction results are shown in Appendix A, Figure A5.

Compared to the network with 30 neurons in hidden layer, the training time has increased by 70%; however, the performance of the trained neural network has not improved except for the test data obtained from damage extent between 10% and 20% without error or with 1% random error in mode shapes.

E. Neural network with different validation parameter

The scaled conjugate gradient training algorithm is a training function that updates weight and bias values according to the scaled conjugate gradient method. In the foregoing analysis, the validation data set is used to stop training early if the network performance on the validation vectors fails to improve or remains the same for ten epochs in a row. It is normal for validation error to go up and down numerous times during training, and it is possible that the network validation error decreases again after this ten consecutive epochs.

Based on the above fact, the network with architecture of 15-30-1 is trained so that the

training stops when validation error has increased for fifteen consecutive epochs since the last time it decreased. The neural network is trained in the same way as before to get the damage factor of elements numbered 3 and 8. The total training time on a Pentium III computer is 2534.86 seconds. Six hundred damage index vectors with different measurement errors in incomplete mode shapes are presented to the network. The prediction results are shown in Appendix A, Figure A6.

Compared to the network with 30 neurons in hidden layer, the training time has increased by 92%, however, the prediction and generalization capabilities of the trained neural network have not improved.

F. Neural network with different learning algorithm

Demuth et al. (2002) show that the resilient backpropagation training algorithm and the scaled conjugate gradient training algorithm are the two most suitable for pattern recognition problems. Based on this fact, the network is also trained using the resilient backpropagation training algorithm. The training process, shown in Figures 4.46 and 4.47, is very unstable. The total training time on a Pentium III computer is 2912.88 seconds. Six hundred damage index vectors with different measurement errors in incomplete mode shapes are presented to the network. The prediction results are shown in Appendix A, Figure A7.

Compared to the network with 30 neurons in hidden layer trained with the scaled

conjugate gradient algorithm, the training time has increased by 120%, but the prediction and generalization capabilities of the trained neural network have become even worse.

G. Neural network with different transfer function in the output layer

The output of the logistic sigmoid function ranges from 0 to 1. The function is suitable for use as the transfer function to output the damage factor, which also varies from 0 to 1. The network used in Section C is redesigned by using the logistic sigmoid transfer function in the output layer. The network is trained in the same way as in Section C. The total training time on a Pentium III computer is 1615.16 seconds. Six hundred damage index vectors with different measurement errors in incomplete mode shapes are presented to the trained network. The prediction results are shown in Appendix A, Figure A8.

The training time has increased by 22%, but the prediction and generalization capabilities of the trained neural network have not improved at all.

H. Neural network trained with different size of the data sets

To explore the effect of the size of the data set on the performance of the trained network, the following two data sets of different size are employed. For each damage combination state and error level, 100 training example pairs are created. The network and the

training algorithm are the same as those in Section C. The total training time on a Pentium III computer is 3342.929 seconds. Six hundred damage index vectors with different measurement errors in incomplete mode shapes are presented to the network. The prediction results are shown in Appendix A, Figure A9.

The training time has increased by 153%, however, the prediction and generalization capabilities of the trained neural network are almost the same as those of the network trained with the data sets in Section C.

Next, assume that 20 training example pairs are created for each damage combination state and error level. The neural network and the training algorithm used in Section C are used again. The total training time on a Pentium III computer is 486.55 seconds. Six hundred damage index vectors with different measurement errors in incomplete mode shapes are presented to the network. The prediction results are shown in Figure 4.48.

Compared to the network trained in Section C, the training time has decreased by 63%, at the same time the prediction and generalization capabilities of the trained neural network have not suffered.

In conclusion, the optimal neural network has the architecture of 15-30-1, that is, the input damage index vector has 15 elements corresponding to 15 finite element, the network has one hidden layer with 30 neurons, and the output layer has one neuron to output the damage extent. The transfer function is the tangent sigmoid function in the

hidden layer and a pure linear function in the output layer. The data sets are created from the mode shapes both with and without error. For each damage status and error level, 20 data pairs are calculated. The training algorithm is the scaled conjugate gradient algorithm.

The performance of the network with different architectures are compared in Table 4.4 (a).

4.5.3 Two-element damage case, damage index vectors based on the curvature obtained from strain measurements

In this section, we assume again that elements 3 and 8 have been identified as being damaged using the damage index method or substructure method. The steps used in the identification are similar to those described in Section 4.5.2 except that we use the incomplete mode shapes related to the 16 rotational degrees of freedom to calculate the curvature. As noted earlier, in field application the curvatures would be derived from direct measurements of the strains.

A. Neural network with architecture of 15-30-30-2

A neural network with two hidden layers and one output layer is used. Each hidden layer has 30 neurons with sigmoid transfer function; the output layer has two neurons with pure linear transfer function. Fifty data sets are created at each error level for each

damage scenario. The scaled conjugate gradient algorithm is used to train the network; the maximum number of iterations up to which the validation error fails to improve is set to 10. The training epochs are 544, 76 and 270 for the 3 different training data sets, respectively. The total training time on a Pentium III computer is 6209.3 seconds. The performance of the trained network is 0.000336291. From the training record, it can be seen the training has been stopped by the validation data set.

The trained network is tested with six hundred damage scenarios with different simulated errors; the results are shown in Appendix A, Figure A10. The figures show that the prediction and generalization capabilities of the trained neural network are significantly better than when translations are used.

The following cases take the neural network used in the present section as reference.

B. Neural network with architecture of 15-30-2

To assess the impact of the number of hidden layers, the number is reduced to one. The training epochs are 249, 116 and 113 for the 3 different training data sets, respectively. The total training time on a Pentium III computer is 1546.9 seconds. The performance of the trained network is 0.000987878, which is worse than that of the network with two hidden layers. Six hundred damage index vectors with different measurement errors in mode shapes are presented to the trained network. The prediction results are shown in Appendix A, Figure A11.

Compared to the network in Section A, the training time has reduced by 75%, but the prediction and generalization capabilities of the trained neural network have deteriorated, especially for minor damage.

C. Neural network with architecture of 15-30-30-1

When one neuron is used in the output layer, the total training time on an Intel Pentium III computer is 6455.88 seconds. Six hundred damage index vectors with different measurement errors in mode shapes are presented to the network. The prediction results, shown in Appendix A, Figure A12, have not improved.

D. Neural network with architecture of 15-45-45-2 and 15-15-15-2

We next increase the number of the neurons in hidden layers of the network used in Section A to 45. The total training time on a Pentium III computer is 34368.3 seconds. Six hundred damage index vectors with different measurement errors in mode shapes are presented to the network. The prediction results are shown in Appendix A, Figure A13.

Compared to the network with 30 neurons in hidden layer, the training time has increased significantly, but the performance of the trained neural network has not improved too much.

Assume now that 15 neurons are used in the hidden layer. The neural network is again

trained in three steps using different data sets. The total training time on a Pentium III computer is 1596.0 seconds. Six hundred damage index vectors with different measurement errors in mode shapes are presented to the network. The prediction results are shown in Appendix A, Figure A14.

Compared to the network with 30 neurons in hidden layer, the training time has decreased by 74%, but the performance of the trained neural network has deteriorated considerably, especially for small damage extent

E. Neural network with different validation parameter

The network with architecture of 15-30-30-2 used in the Section A is trained so that the training stops when validation error has increased for fifteen consecutive epochs since the last time it decreased. The neural network is trained in the same way as before to get the damage factor of elements 3 and 8. The total training time on a Pentium III computer is 4330.88 seconds. Six hundred damage index vectors with different measurement errors in incomplete mode shapes are presented to the network. The prediction results are shown in Appendix A, Figure A15.

Compared to the trained network in Section A, the training time has decreased by 30%, but the prediction and generalization capabilities of the trained neural network become worse for minor damage.

F. Neural network with different learning algorithm

The network in Section A is trained with the resilient backpropagation training algorithm.

The training process, shown in Figures 4.49, is very unstable. The total training time on Pentium III computer is 6226.52 seconds. Six hundred damage index vectors with different measurement errors in incomplete mode shapes are presented to the network.

The prediction results are shown in Appendix A, Figure A16.

Compared to the network in Section A trained with the scaled conjugate gradient algorithm, the training time is almost the same, while the prediction and generalization capabilities of the trained neural network become slightly worse.

G. Neural network with different transfer function in the output layer

The network used in section A is redesigned by using the logistic sigmoid transfer function in the output layer. The network is trained in the same way as in Section A. The total training time on a Pentium III computer is 4017.2 seconds. Six hundred damage index vectors with different measurement errors in incomplete mode shapes are presented to the trained network. The prediction results are shown in Figure 4.50.

The training time has decreased by 35%, while at the same time the prediction and generalization capabilities of the trained neural network have obviously improved for minor damage scenarios.

H. Neural network trained with different size of the data sets

To explore the effect of the size of the data set on the performance of the trained network, the following two data sets of different sizes are employed. For each damage combination state and error level, 100 training example pairs are created. The network and the training algorithm are the same as those in Section G. The total training time on a Pentium III computer is 9391.91 seconds. Six hundred damage index vectors with different measurement errors in incomplete mode shapes are presented to the network. The prediction results are shown in Appendix A, Figure A17.

The training time has increased by 51%, however, the prediction and generalization capabilities of the trained neural network are almost the same as those of the network in Section G.

Next, assume that 20 training example pairs are created for each damage combination state and error level. The total training time on a Pentium III computer is 2912.65 seconds. Six hundred damage index vectors with different measurement errors in incomplete mode shapes are presented to the network. The prediction results are shown in Appendix A, Figure A18.

Compared to the network in section G, the training time has reduced by 53%, and the prediction and generalization capabilities of the trained neural network have slightly deteriorated.

In conclusion, the optimal neural network has the architecture of 15-30-30-2, that is, the input damage index vector has 15 elements corresponding to 15 finite elements, the network has two hidden layers each with 30 neurons, and the output layer has two neuron to output the damage extent. The transfer function is a tangent sigmoid function in the hidden layers and a logistic sigmoid function in the output layer. The data sets are created from the mode shapes both with and without error. For each damage status and error level, 50 data pairs are calculated. The training algorithm is the scaled conjugate gradient algorithm. The performances of different network architectures are compared in Table 4.4 (b).

From Table 4.4 and Figures 4.48 and 4.50, it is obvious that the prediction of damage extent of the network trained with the data sets based on the curvature mode shapes from the strain measurements is much more reliable.

To verify that the optimal architecture for the neural network is not dependent on the specific elements that are damaged, sets of calculations similar to those described in the earlier paragraphs were repeated for several different combinations of damaged elements, and the performance index and accuracy of damage magnitude predictions were compared. The results obtained are not presented here, but in general, the optimal architecture remained unchanged.

Table 4.1: Measured and analytically obtained frequencies of vibration modes
(University of Alberta, 1998)

Mode Shape No.	Measured Natural Frequencies	Calculated Natural Frequencies	Description
1	2.60 Hz	2.58 Hz	1 st vertical mode
2	2.90 Hz		1 st torsional mode
3	3.63 Hz	3.62 Hz	2 nd vertical mode
4	3.85 Hz		2 nd torsional mode
5	4.43 Hz	4.45 Hz	3 rd vertical mode
6	5.00 Hz		Torsion/Transverse mode
7	6.85 Hz		Torsion/Transverse mode
8	8.60 Hz		4 th vertical mode

Table 4.2: Material properties of the girder model

Element No.	E (N/m ²)	G (N/m ²)	Poison's Ratio μ	Mass per unit length (kg/m)
①—④	2.000e+011	7.700e+010	0.30	2.015e+004
⑤ ⑥ ⑩ ⑪	2.000e+011	7.700e+010	0.30	1.805e+004
⑦—⑨ ⑫—⑮	2.000e+011	7.700e+010	0.30	2.065e+004

Table 4.3: Section properties of the girder model

Element No.	Area(m ²)	I _y (m ⁴)	I _z (m ⁴)	J
①—④	0.450	0.145	0.145	1.000
⑤ ⑥ ⑩ ⑪	0.554	0.078	0.078	1.000
⑦—⑨ ⑫—⑮	0.430	0.096	0.096	1.000

Table 4.4: Training of neural networks for damage extent in the Crowchild Bridge
(a) Damage index vectors based on curvature from translational mode shapes

Neural networks	Training epochs			Training Time (s)	Performance (MSE)
	Stage 1	Stage 2	Stage 3		
15-30-30-2, trainscg, tansig, purelin, maxfail=10, data 50.	225	153	78	2317	0.003713
15-30-2, trainscg, tansig, purelin, maxfail=10, data 50.	441	182	185	2222	0.003687
15-30-1 , trainscg, tansig, purelin, maxfail=10, data 50.	113 (EL3)	57 (EL3)	97 (EL3)	1322	0.00241543 (EL3)
	110 (EL8)	23 (EL8)	35 (EL8)		0.00421358 (EL8)
15-45-1, trainscg, tansig, purelin, maxfail=10, data 50.	85 (EL3)	42 (EL3)	131 (EL3)	2250	0.00207867 (EL3)
	103 (EL8)	52 (EL8)	82 (EL8)		0.0040357 (EL8)
15-15-1, trainscg, tansig, purelin, maxfail=10, data 50.	79 (EL3)	87 (EL3)	66 (EL3)	617	0.0033257 (EL3)
	87 (EL8)	63 (EL8)	83 (EL8)		0.00489076 (EL8)
15-30-1, trainscg, tansig, purelin, maxfail=15, data 50.	110 (EL3)	55 (EL3)	92 (EL3)	2535	0.00244547 (EL3)
	343 (EL8)	64 (EL8)	125 (EL8)		0.00412888 (EL8)
15-30-1, trainrp, tansig, purelin, maxfail=10, data 50.	292 (EL3)	11 (EL3)	11 (EL3)	2913	0.0163986 (EL3)
	1083 (EL8)	11 (EL8)	11 (EL8)		0.0135938 (EL8)
15-30-1, trainscg, tansig, logsig, maxfail=10, data 50.	204 (EL3)	196 (EL3)	76 (EL3)	1615	0.00311872 (EL3)
	77 (EL8)	180 (EL8)	70 (EL8)		0.00486804 (EL8)
15-30-1, trainscg, tansig, purelin, maxfail=10, data 100.	42 (EL3)	56 (EL3)	165 (EL3)	3343	0.00210753 (EL3)
	130 (EL8)	78 (EL8)	107 (EL8)		0.00475846 (EL8)
15-30-1, trainscg, tansig, purelin, maxfail=10, data 20.	72 (EL3)	90 (EL3)	106 (EL3)	487	0.00227504 (EL3)
	119 (EL8)	41 (EL8)	39 (EL8)		0.0046038 (EL8)

Table 4.4 Training of neural networks for damage extent in the Crowchild Bridge
(b) Damage index vectors based on curvature mode shapes from strain measurements

Neural networks	Training epochs			Training Time (s)	Performance (MSE)
	Stage 1	Stage 2	Stage 3		
15-30-30-2, trainscg, tansig, purelin, maxfail=10, data 50.	544	76	270	6209	0.0003363
15-30-2, trainscg, tansig, purelin, maxfail=10, data 50.	249	116	113	1547	0.0009879
15-30-30-1, trainscg, tansig, purelin, maxfail=10, data 50.	203 (EL3)	20 (EL3)	290 (EL3)	6456	0.0001964 (EL3)
	259 (EL8)	52 (EL8)	105 (EL8)		0.0003350 (EL8)
15-45-45-2, trainscg, tansig, purelin, maxfail=10, data 50.	2923	36	41	34368	0.0001625
15-15-15-2, trainscg, tansig, purelin, maxfail=10, data 50.	329	74	87	1596	0.0008139
15-30-30-2, trainscg, tansig, purelin, maxfail=15, data 50.	438	136	130	4331	0.0003365
15-30-30-2, trainrp, tansig, purelin, maxfail=10, data 50.	1454	11	11	6227	0.0235059
15-30-30-2 , trainscg, tansig, logsig, maxfail=10, data 50.	249	48	272	4017	0.0001890
15-30-30-2, trainscg, tansig, logsig, maxfail=10, data 100.	526	23	77	9392	0.0001729
15-30-30-2, trainscg, tansig, logsig, maxfail=10, data 20.	650	80	299	2913	0.0001505

Note:

15-30-2: 15 elements in input vector, 30 neurons in hidden layer, 2 neurons in output layer

trainscg: the scaled conjugate gradient training algorithm

trainrp: the resilient backpropagation training algorithm

tansig: the tangent sigmoid transfer function

purelin: the pure linear transfer function

maxfail: the maximum number of iterations to validation error failure

data 20: 20 damage index vectors created corresponding to each damage extent and error level in the training data set.

The transfer function that appears first is for the hidden layer.

The transfer function that appears second is for the output layer.

Figure 4.1: Crowchild Bridge – West elevation

Figure 4.2: Cross section of the Crowchild Bridge

Figure 4.3: Two-dimensional beam element (3 DOFs per node)

Figure 4.4: Girder model of the Crowchild Bridge

Figure 4.5: Damage indices calculated from three modes of the girder model of the Crowchild Bridge, incomplete but accurate mode shapes expanded by SEREP, (a) damage 50% in element number 3, (b) damage 50% in each of element numbers 3 and 8

Figure 4.6: Damage indices based on curvatures from translational components of three modes of the girder model of Crowchild Bridge, 30% damage in element 3, (a) accurate modes, (b) modes with $\pm 2\%$ random error, (c) modes with $\pm 5\%$ random error

Figure 4.7: Damage indices based on curvature from translational component of three mode shapes of the girder model, damage 50% in element number 3, (a) modes with $\pm 2\%$ random error, (b) modes with $\pm 5\%$ random error

Figure 4.8: Damage indices based on curvature from translational component of three mode shapes of the girder model, damage 30% in element number 8, (a) accurate modes, (b) modes with $\pm 2\%$ random errors, (c) modes with $\pm 5\%$ random errors.

Figure 4.9: Damage indices based on curvature from translational component of three mode shapes of the girder model, damage 50% in element number 8, (a) mode shapes with $\pm 2\%$ random errors, (b) mode shapes with $\pm 5\%$ random errors

Figure 4.10: Damage indices based on curvature from translational component of three mode shapes of the girder model, damage 30% in each of element numbers 3 and 8, (a) accurate modes, (b) modes with $\pm 2\%$ random error, (c) modes with $\pm 5\%$ random error

Figure 4.11: Damage indices based on curvature from translational component of three mode shapes of the girder model, damage 50% in each of element numbers 3 and 8, (a) modes with $\pm 2\%$ random error, (b) modes with $\pm 5\%$ random error

Figure 4.12: Damage indices calculated from curvature based on strain measurement of three mode shapes of the girder model, damage 30% in element number 3, (a) modes with $\pm 2\%$ random error, (b) modes with $\pm 5\%$ random error

Figure 4.13: Damage indices calculated from curvature based on strain measurement of three mode shapes of the girder model, damage 50% in element number 3, (a) modes with $\pm 2\%$ random error, (b) modes with $\pm 5\%$ random error

Figure 4.14: Damage indices calculated from curvature based on strain measurements of three mode shapes of the girder model, damage 30% in element number 8, (a) modes with $\pm 2\%$ random error, (b) modes with $\pm 5\%$ random error

Figure 4.15: Damage indices calculated from curvature based on strain measurement of three mode shapes of the girder model, damage 50% in element number 8, (a) curvature mode shapes with $\pm 2\%$ error, (b) curvature mode shapes with $\pm 5\%$ error

Figure 4.16: Damage indices calculated from curvature based on strain measurements of three lowest mode shapes of the girder model, damage 30% in each of element numbers 3 and 8 simultaneously, (a) curvature mode shapes with $\pm 2\%$ error, (b) curvature mode shapes with $\pm 5\%$ error

Figure 4.17: The 1st step of substructure method -- Damage indices calculated from six curvature modes of the 60m beam, damage 20% and 30% in original element numbers 14 and 36 respectively, (a) curvature mode shapes with $\pm 2\%$ error, (b) curvature mode shapes with $\pm 5\%$ error

Figure 4.18: Substructure 2 of 60m beam
(Number in bold is for new element, number in parenthesis is for original element)

Figure 4.19: The 2nd step of substructure method -- Damage indices calculated from six curvature modes of the 60m beam, damage 20% and 30% in original element numbers 14 and 36 respectively, (a) curvature mode shapes with $\pm 2\%$ error, (b) curvature mode shapes with $\pm 5\%$ error

Figure 4.20: Substructure 1 of girder model of the Crowchild Bridge for translational measurement (Numbers in bold face for new elements, numbers in light face for original elements)

Figure 4.21: The 1st step of substructure method -- Damage indices calculated from three translational modes of the girder model of the Crowchild Bridge, damage 30% in original element number 3, (a) accurate modes, (b) modes with $\pm 2\%$ error, (c) modes with $\pm 5\%$ error

Figure 4.22: The 1st step of substructure method -- Damage indices calculated from three translational modes of the girder model of the Crowchild Bridge, damage 50% in original element number 3, (a) modes with $\pm 2\%$ error, (b) modes with $\pm 5\%$ error

Figure 4.23: Substructure 2 of girder model of the Crowchild Bridge for translational measurements (Numbers in bold face are for new elements; numbers in normal face are for original elements)

Figure 4.24: The 2nd step of substructure method -- Damage indices calculated from three translational modes of the girder model of the Crowchild Bridge, damage 30% in original element number 3, (a) accurate modes, (b) modes with $\pm 2\%$ error, (c) modes with $\pm 5\%$ error

Figure 4.25: The 2nd step of substructure method -- Damage indices calculated from three translational modes of the girder model of the Crowchild Bridge, damage 50% in original element number 3, (a) modes with $\pm 2\%$ error, (b) modes with $\pm 5\%$ error

Figure 4.26: The 1st step of substructure method -- Damage indices calculated from three translational modes of the girder model of the Crowchild Bridge, damage 30% in original element numbers 3 and 8, (a) accurate modes, (b) modes with $\pm 2\%$ error, (c) modes with $\pm 5\%$ error

Figure 4.27: The 1st step of substructure method -- Damage indices calculated from three translational modes of the girder model of the Crowchild Bridge, damage 50% in original element numbers 3 and 8, (a) modes with $\pm 2\%$ error, (b) modes with $\pm 5\%$ error

Figure 4.28: The 2nd step of substructure method -- Damage indices calculated from three translational modes of the girder model of the Crowchild Bridge, damage 30% in original element numbers 3 and 8, (a) accurate mode shapes, (b) modes with $\pm 2\%$ error, (c) modes with $\pm 5\%$ error

Figure 4.29: The 2nd step of substructure method -- Damage indices calculated from three translational modes of the girder model of the Crowchild Bridge, damage 50% in original element numbers 3 and 8, (a) modes with $\pm 2\%$ error, (b) modes with $\pm 5\%$ error

Figure 4.30: Substructure 1 of girder model of the Crowchild Bridge for strain measurements (Numbers in bold face are for new elements, numbers in normal face are for original elements)

Figure 4.31: The 1st step of substructure method -- Damage indices based on three curvature mode shapes of the girder model from the strain measurements, damage 30% in element number 3, (a) modes with $\pm 2\%$ error, (b) modes with $\pm 5\%$ error

Figure 4.32: The 1st step of substructure method -- Damage indices based on three curvature mode shapes of the girder model from strain measurements, damage 50% in element number 3, (a) modes with $\pm 2\%$ error, (b) modes with $\pm 5\%$ error

Figure 4.33: Substructure 2 of girder model of the Crowchild Bridge for strain measurements (Numbers in bold face are for new elements, numbers in normal face are for original elements)

Figure 4.34: The 2nd step of substructure method -- Damage indices based on three curvature mode shapes of the girder model from the strain measurements, damage 30% in element number 3, (a) modes with $\pm 2\%$ error, (b) modes with $\pm 5\%$ error

Figure 4.35: The 1st step of substructure method -- Damage indices based on three curvature mode shapes of the girder model from the strain measurements, damage 30% in element numbers 3 and 8 simultaneously, (a) modes with $\pm 2\%$ error, (b) modes with $\pm 5\%$ error.

Figure 4.36: The 2nd step of substructure method -- Damage indices based on three curvature mode shapes of the girder model from the strain measurements, damage 30% in element numbers 3 and 8 simultaneously, (a) modes with $\pm 2\%$ error, (b) modes with $\pm 5\%$ error.

Figure 4.37: The perceptron neural network for damage localization

(a)

(b)

(c)

(d)

(e)

Figure 4.38: Prediction of damage location using perceptron network, damage in element No. 1, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

Figure 4.39: Prediction of damage location using perceptron network, damage in element No. 3, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

(e)

Figure 4.40: Prediction of damage location using perceptron network, damage in element No. 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

Figure 4.41: The designed artificial neural network for the evaluation of damage extent in Crowchild Bridge

Figure 4.42: Training and validation errors of the network for the damage factor of element No. 3 trained using damage index vectors from translational mode shapes both with and without error, one-element damage case, (a) stage 1, (b) stage 2, (c) stage 3

(a)

(b)

(c)

(d)

(e)

Figure 4.43: Prediction of damage extent using backpropagation network, damage in element No. 3, one-element damage case, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

Figure 4.44: Training and validation errors of the network for the damage factor of element No. 8 trained using damage index vectors from translational mode shapes both with and without error, one-element damage case, (a) stage 1, (b) stage 2, (c) stage 3

(a)

(b)

(c)

(d)

Figure 4.45: Prediction of damage extent using backpropagation network, damage in element No. 8, one-element damage case, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

Figure 4.46: Training and validation errors of the network for the damage factor of element No. 3 trained using damage index vectors from translational mode shapes both with and without error and the resilient backpropagation training algorithm (a) stage 1, (b) stage 2, (c) stage 3

(a)

(b)

Figure 4.47: Training and validation errors of the network for the damage factor of element No. 8 trained using damage index vectors from translational mode shapes both with and without error and the resilient backpropagation training algorithm (a) stage 1, (b) stage 2, (c) stage 3

(a)

(b)

(c)

(d)

(e)

Figure 4.48: Prediction of damage extent using backpropagation network of architecture 15-30-1 and smaller training data set, damage index vectors based on curvature obtained from translational mode shapes, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

Figure 4.49: Training process of the network with architecture of 15-30-30-2 for the damage factor of elements Nos. 3 and 8 trained using damage index vectors based on the curvature from the rotational component of incomplete mode shapes both with and without error, resilient backpropagation training algorithm, (a) stage 1, (b) stage 2, (c) stage 3

(a)

(b)

(c)

(d)

(e)

Figure 4.50: Prediction of damage extent using backpropagation network with architecture of 15-30-30-2, damage in element Nos. 3 and 8, logistic sigmoid transfer function used in the output layer, damage index vectors based on the curvature obtained from strain measurements, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

Chapter 5

Case Study

Damage Detection in a Space Truss

5.1 Introduction

In this study, computer simulation is used to identify a number of potential damage patterns in a space truss and to evaluate the effects of measurement errors and incomplete mode shapes. A finite element model of the space truss is first developed. The damage is simulated by reducing the stiffness of one or more elements by a specified ratio. The corresponding mode shapes of the damaged structure are obtained, and then reduced in size based on a selected practical placement of sensors to simulate the incomplete mode shapes. Random errors are introduced in the incomplete modes to simulate the measurement errors in the mode shapes as necessary. The incomplete mode shapes are finally expanded to the full size using the System Equivalent Reduction Expansion Process (SEREP), which is a modal expansion technique. Alternatively, the stiffness matrix corresponding to the full set of degrees of freedom is reduced by static condensation to match the incomplete mode shape. Damage indices are calculated next

using the modal strain energies in selected modes, for both the healthy structure and the structure with simulated damage.

The algorithm proposed here divides the structural damage detection problem into two distinct sub-problems. The location of the damage is first obtained by using the damage index values or a perceptron neural network. A back-propagation neural network then evaluates the damage extent. It is shown that the proposed method is both practical and effective.

5.2 Modeling of the 3D space truss

For the present case study, the eight-bay space truss is assumed to be built from commercially available hardware (Meroform M12) produced by MeroForm System USA, Orange, California. The hardware consists of standardized aluminum nodes and tube struts of different sizes. Figure 5.1 shows the details of the nodes and struts. A finite element model of the space truss is shown in Figure 5.2, in which an individual element is represented by a 3D truss element. There are a total of 109 elements and 108 degrees of freedom in the model. The mass of the elements is lumped along the 108 translational degrees of freedom. The section and material properties of the space truss elements are listed in Table 5.1. Table 5.2 shows the element connectivity. A lumped mass of 0.44 kg is added at each node to represent the mass of the joint hardware, including the aluminum ball. Additional lumped masses, each of 1.75 kg, are added to nodes 4, 9, 25,

and 36. Masses of 2.75 kg are added at nodes 6, 17, and 30. These design features give well-separated modes to facilitate the evaluation of proposed damage detection methods. Analytical mode shapes are obtained for both the healthy and the damaged space truss with different levels of simulated damage.

5.3 Localization of the simulated damage

5.3.1 Damage index based structural health monitoring

We first examine various damage scenarios, in which mode shape measurements are taken either at the 36 degrees of freedom along x -axis only, or along all 108 degrees of freedom. Incomplete modes are expanded using the System Equivalent Reduction Expansion Process (SEREP) as described in Section 4.3.2. It is assumed that the mode shapes for the first six non-rigid-body modes are available.

The damage index method uses change in the modal strain energy to detect damage. This implies that if the strain energy contributed to the measured modes by a certain element is small, it will be difficult to detect damage in that element. For example, consider the 3D truss shown in Figure 5.2. If the full mode shapes are available, the relative strain energies contributed by the elements of the truss are shown Figure 5.3. Relative strain energy of an element in a given mode is defined as the ratio of the energy contributed by the element to the specified mode to the total energy contributed by all elements to the same mode. It will be observed that the energy contributed by the

verticals, horizontals in the z direction and the cross diagonals is relatively small; hence it will be difficult to detect damage in these elements

Damage in chord members

Assume that measurements are made along the 36 x degrees of freedom, and that measurement errors in modes are up to 2%. Also, assume that measurements are available for the first six non-rigid-body modes. The calculated damage indices are presented in Figure 5.4 for various damage scenarios; they clearly identify the damage location. The modal strain energy is comparatively small for chord 1 connecting nodes 1 and 5, hence measurement errors cause some spurious damage to be identified in other locations as well, but the magnitude is small. As strain energy increases, the results become quite accurate, for example, those for elements 2, 18, 28 and 30.

Damage in vertical members

The modal strain energy in the first six modes is quite small in vertical members. To identify the damage in these members becomes difficult. Figure 5.5 shows some results. It is noted that measurement of x or z degrees of freedom cannot be expected to reveal damage in a vertical member. It is possible to identify the damage only when measurements are taken along at least 36 degrees of freedom in y direction and the measurements are exact.

Damage in horizontal members along z -axis

Since the modal strain energies are small for horizontal members in the z direction, it will be difficult to detect damage. This is confirmed by the results presented in Figure 5.6 for 30% damage in Element 63. The x - and y -axis measurements cannot reveal damage in a member that is oriented along the z -axis. The damage can be identified only when the measurements along the z -axis are included in the modes and are accurate.

The computations are repeated for horizontal element 55. The strain energy in the element is larger. Thus, provided measurements along 36 z -axis degrees of freedom are included in the mode shapes, the damage location is clearly identified, as shown in Figure 5.7, even with 2% error in mode shapes.

Diagonal members in the x - y plane

Simulated damage of 30% is introduced in element No. 74 which is a diagonal member lying in the x - y plane. The calculated damage indices are presented in Figure 5.8. It is observed that as long as 36 y degrees of freedom are included in the measurement, damage location is well identified even in the presence of measurement errors. However, measurements along only the x , z , or xz degrees of freedom are not sufficient, even when the measurements are exact. The z displacements do not contribute to the modal strain

energy. The x direction motions, being caused by axial deformation of the truss, are small as compared to the y direction deformations that are caused by flexural action in the truss. This, presumably, is the reason why x direction measurements are not sufficient.

Diagonal members in y-z plane

Figure 5.9 shows the damage indices when 30% damage is introduced in the diagonal element No. 107 lying in the y - z plane. These members do not contribute much to the strain energy in the first six modes. It is therefore not possible to identify the damage even when all 108 degrees of freedom are measured and the measurements are exact.

Diagonal members in the x-z plane

A set of results is presented in Figure 5.10. It is observed that as long as 36 z degrees of freedom are included in the measurement, damage location is well identified even in the presence of measurement errors. However, measurements along only the x , y , or xy degrees of freedom are not sufficient, even when the measurements are exact.

Multiple damage sites

Results for damage at multiple sites are presented in Figures 5.11 through 5.14. When the strain energy in one element is significantly less than in other elements

simultaneously damaged, the damage in the lower strain energy element may not be clearly identified, as shown in Figure 5.11. In other cases, damage location is well estimated. The results presented in Figures 5.11 through 5.14 are based on the assumption that only the 36 x degrees of freedom are measured and the mode shapes have 2% error.

5.3.2 Sub-structuring method

As stated earlier, for complex large-scale structures it would be impracticable to instrument all or even a majority of degrees of freedom. In such a case, a sub-structuring technique is useful. In this technique, a coarse array of instruments is deployed in the first stage to determine the broad region of damage. In subsequent stages, instruments are concentrated in the identified region of damage in order to refine the determination of damage location.

In the present case study, the example truss is first divided into four equal length substructures; each of them has two bays. There are 60 degrees of freedom on the boundaries of the substructures. Two different methods are used to obtain the damage indices. In one case, the degrees of freedom that are internal to the substructures are condensed out from the finite element model leaving only the junction degrees of freedom along which modal measurements are made. In the other method, the standard modal expansion may be used to obtain mode shapes corresponding to all of the degrees

of freedom in the structural model.

Damage indices obtained by condensation of the degrees of freedom

Assume that simulated damage is introduced in elements No. 18 connecting nodes 7, 11 and No. 28 connecting nodes 16, 20 of the original space truss model. In the first step, the stiffness matrix of the four substructures used to calculate the damage index is obtained from the full stiffness matrix by static condensation, including only x degrees of freedom of the boundary nodes. The mode shapes for the healthy and damaged substructures are extracted from the corresponding full mode shapes, and the error is added to the damaged mode shapes if needed. As expected, the first and second substructures, in which the damaged elements No. 18 and No. 28 fall, are identified as being possibly damaged in this step, as shown in Figure 5.15.

In the second step, only the 57 elements in the first four bays, including the first and second substructures, are considered, and it is assumed that only the x degrees of freedom are measured for each of 20 nodes. The incomplete mode shapes are expanded by the SEREP procedure, and the damage indices of the 57 elements are calculated. As seen from Figure 5.16, element numbers 10 and 16 (that is, original elements 18 and 28) are clearly identified as being damaged.

Damage indices obtained by expansion of the degrees of freedom

Assume again that only 20 x degrees of freedom along the boundaries of the 4 substructures mentioned earlier are measured. For ease in interpreting the results, the elements are renumbered so that the 57 elements in the first four bays come first. Now let chord element 10 connecting nodes 7 and 11 and element 16 connecting 16 and 20 each suffer 30% damage. The damage indices are obtained by expanding the modes obtained from 20 degrees of freedom to the full 108 degrees of freedom and rearranged making the damage indices for the 57 elements in Substructures 1 and 2 coming first in Figure 5.17. The results show possible damage in Substructures 1 and 2.

The second step is the same as that in the last section. The mode shapes are measured only along the x degree of freedom lying on 20 nodes in the first four bays. The damage indices clearly show the location of damage (Figure 5.16).

5.3.3 Perceptron neural network method for locating damage

The perceptron network described in Chapter 4 was used to locate damage in the truss model. Making use of the 3D truss model, a Matlab 6.5 (The MathWork Inc.) program was written to calculate the damage index vectors corresponding to different simulated damage scenarios. The dimension of the damage index vector may be large due to the complexity of the truss model. Representative critical truss model elements which contain as much dynamic information as possible may be selected to constitute the input.

In this example, the damage indices corresponding to the 32 chord elements are extracted from the original damage index vector corresponding to all the elements to build a new damage index vector of smaller size. The training data set formed by collecting the new damage index vectors and the corresponding damage states are used to train the network. The trained neural network is then used to localize the damage.

As stated earlier, the perceptron neural network has a relatively fixed architecture and transfer function for a specific problem. The neural network designed for truss damage localization, shown in Figure 5.18, has only one layer, namely, the output layer, with 8 neurons, corresponding to the status of 8 chord elements 17, 18, 19, 20, 21, 22, 23, and 24 selected in the example as possible candidates for damage. The output of each neuron is either 1, which indicates damage in the numbered element, or 0, which indicates no damage. The transfer function is a hard limit function, which outputs a 1 for an input that is 0 or positive and a 0 for a negative input.

Creation of the Training Data Set Using Analytical Model

Training data sets are obtained from the incomplete mode shapes including only x degrees of freedom both with and without error to simulate the actual measurement situation. For each damaged element, the selected damage extent is from 0% to 60% at the interval of 6%. Five error levels, -1% to 1% , -2% to 2% , -3% to 3% , -4% to 4% , or -5% to 5% are introduced in the mode shape elements at each damage scenario using a random number generator whose states are related to the clock. Theoretically, it can

generate over 2^{1492} values before repeating itself. For each error level in the mode shapes, 20 different damage index data sets are created corresponding to each damage scenario. Individual data sets at the same damage scenario and modal error level are different from each other because at each call to it the random number generator assigns different errors within the selected error range to the mode shape elements.

Training the Neural Network

As in the training process for damage localization in Crowchild Bridge, the neural network here is first trained using data sets obtained from the simulated noisy mode shapes until it has a low sum-squared error, then trained with example data sets obtained from incomplete accurate mode shapes, and finally trained with data sets obtained from incomplete mode shapes both with and without error to ensure that the trained network responds correctly when presented with either correct or noisy damage index vectors.

Testing of the Trained Neural Network

The reliability of the trained networks is measured by testing the trained networks with damage index vectors obtained for the various mode shape error levels and 6 damage severity ranges, that is, 0% to 10%, 10% to 20%, 20% to 30%, 30% to 40%, 40% to 50%, and 50% to 60%. Taking the damage severity range 0% to 10% as an example, the testing process proceeds as follows.

One hundred damage magnitudes in the element for one-element-damage case or 10 damage magnitudes in each of the two elements for two-element-damage case between

0% and 10% are chosen randomly leading to 100 damage scenarios. Random errors ranging from -1% to $+1\%$ are introduced to the elements of mode shapes. One hundred damage index vectors are calculated and fed to the trained network. The outputs of the network are obtained, and compared to the actual damage element vector. The prediction result can be classified into one of the following four groups,

- 1) The prediction is exactly right
- 2) Two elements are predicted to be damaged, but one of them is or is adjacent to the actual damaged element,
- 3) One element is predicted to be damaged and is adjacent to actual damaged element,
- 4) The prediction is wrong.

The number of network outputs falling into each of the above four groups is summed up and the statistics of the prediction results are obtained for the damage extent between 0% and 10% with 1% mode shape error.

Introducing 0%, 2%, 3%, 4%, and 5% error to the mode shapes obtained from the previously mentioned 100 damage scenarios and following the above procedure, the statistical results for the damage extent between 0 and 10% with 0%, 2%, 3%, 4%, and 5% mode errors are obtained. A stack figure is then plotted for damage extent between 0% and 10%.

Adding 10% to each element of the referred 100 damage magnitudes randomly

generated for damage between 0% and 10%, and following steps that are similar to those mentioned earlier, we can get the stack figure for a damage extent between 10% and 20%. Similarly, we can get the stack figures for damage extent ranges, 20% to 30%, 30% to 40%, 40% to 50%, and 50% to 60%.

Figures 5.19 and 5.20 show the statistical prediction results for some cases. As an example, in Figure 5.20 (b) for damage between 20% and 30% in element 17 and 18, and 2% mode shape error, the trained network gives 96 accurate prediction of damage location out of 100, while 4 predictions show only one element damaged adjacent to the actual damage. It is observed that the prediction and generalization capability of the trained neural network is quite good when the damage extent is significant, say greater than 20%.

5.4 Quantifying the simulated damage extent in the 3D space truss

The back-propagation neural network designed to estimate the magnitude of damage is shown in Figure 5.21. It is assumed that the location has already been correctly determined. Results are obtained here for damage in element 18, or in both elements 18 and 28. The changes of the strain energy in the first 32 chord elements, whose orientations are parallel to the x -axis, are used to determine the damage severity. The initially designed network has two layers, namely, one hidden layer and one output layer. Each input vector has 32 elements corresponding to the damage indices for the 32

elements. The damage indices are calculated under the assumption that only incomplete mode shapes including just the 36 degrees of freedom in x direction are available. The hidden layer has 64 neurons and the output layer has one neuron to output the damage factor for both the one-element damage case and the two-element damage case. The transfer functions are the tangent sigmoid function in the hidden layer and a pure linear function in the output layer.

To avoid over-fitting during training and to increase the generalization capability of the proposed neural network, a validation data set is employed. Over-fitting the training data results in the loss of the generalization property of the network, so that it gives perfect results only for the training data, while producing incorrect outputs for inputs that are not in the training set. To avoid over-fitting, the validation data are used to stop training early if further training on the training data will hurt generalization to the validation data. During the training process, the error on the validation set is monitored. The validation error will first decrease, as does the training error. When the network begins to over-fit the training data, the error on the validation set will become unstable. If the validation error keeps increasing for a specific number of consecutive iterations (10 consecutive iterations are initially used in this study), the training is stopped, and the weights and biases at the instant when the validation error was minimum are returned. The other two control factors are the maximum number of training epochs and the performance goal. Training will also stop if a maximum number of epochs is reached, or the performance

goal is met. In this study, the maximum number of epochs is set to 5000, unless stated otherwise, and the performance goal set to 0.

5.4.1 One-element damage case

Assume that only element 18 has been damaged. The damage index vectors are calculated from the strain energy and random error is introduced into the mode shapes to simulate the actual measurement situation.

Creating the data set used to train the neural network

Training data set: For element number 18, damage extents from 0% to 60% at 5% intervals are selected to obtain the damage index vectors both with and without error. Five error levels, -1% to 1% , -2% to 2% , -3% to 3% , -4% to 4% , or -5% to 5% are introduced in the mode shape elements for each damage scenario using a random number generator whose states are related to the clock. For each error level in the mode shapes, 50 different damage index data sets are created corresponding to each damage scenario.

Validation data set: To avoid overfitting, the choice of validation set is also important. The validation set should be representative of all points in the training set. For element No. 18, damage extents that are slightly shifted from those used in the training data, namely, 2% to 57% at interval of 5%, are selected to calculate the damage index vectors. When error is introduced in the mode shapes, for each error level, 40 different damage

index data sets are created corresponding to each damage scenario.

Training the neural network

The designed back-propagation neural network is trained in three steps as in the case of perceptron network. The training algorithm used is the scaled conjugate gradient algorithm. The training processes, shown in Appendix B, Figure B1 indicates the training and validation error between the actual and target output as a function of the numbers of epochs. The training epochs are 99, 30 and 167 for the 3 different training data sets, respectively. The performance of the trained is 0.0007068. The total training time on a Pentium III computer is 176 seconds. From the training record, we can see that the training has been stopped by the validation data set.

Testing trained neural network

The reliability of the trained networks is measured by testing the networks with damage index vectors obtained for the various mode shape error levels and 6 damage severity ranges, that is, 0% to 10%, 10% to 20%, 20% to 30%, 30% to 40%, 40% to 50%, and 50% to 60%. The creation of the test data set is similar to that in the damage localization for one-damage-case. These data are fed to the trained network, the outputs of the network are obtained and by comparing them with actual damage extents, the relative prediction errors (RPE) are calculated. The RPE is defined as the difference between the magnitude of predicted damage and the magnitude of actual damage as a percentage of the actual damage. The numbers of outputs within a specific range of RPE are shown in

Figure 5.22. As an example, in Figure 5.22 (d) for damage between 30% and 40% in element 18 and 3% mode shape error, for every 100 test data, the trained network gives 96 predictions whose RPEs are less than 10%, and 4 predictions with RPE between 10% and 20%.

From the results presented, it is observed that the prediction and generalization capability of the trained neural network is quite good when the damage extent is significant, say greater than 20%.

5.4.2 Two-element damage case

Assume that elements 18 and 28 have been identified as being damaged using the damage index method or substructure method. The backpropagation network described earlier is used to determine the extent of damage. The training process is similar to that used in the one-element damage case and a validation data set is used to prevent overfitting and to stop the training process early when necessary. The data pair is composed of the damage index vectors and the corresponding damage extent. Incomplete mode shapes, both with and without error, are used to simulate the actual measurement situation.

To find a relatively optimal network architecture, effective size of the training data set, and the training algorithm and transfer function used so as to improve the generalization capability of the trained network, the following cases are explored.

A. Neural network with architecture of 32-64-1

The initially designed neural network for two-element-damage case has the structure of 32-64-1; the output of the neural network represents the damage extent of element number 18 or 28 depending on the training of the network. The following three steps are followed to evaluate the damage extent of elements 18 and 28.

Creating the data set used to train the neural network

Training data set: For element numbers 18 and 28, damage extents from 0% to 60% at 5% intervals are selected to obtain the damage index vectors, both with and without error. Five error levels, -1% to 1% , -2% to 2% , -3% to 3% , -4% to 4% , or -5% to 5% are introduced in the mode shape elements at each damage scenario using a random number generator whose states are related to the clock. For each error level in the mode shapes, 50 different damage index data sets are created corresponding to each damage scenario. The data sets are different for the same reason as for the perceptron neural network.

Validation data set: For element numbers 18 and 28, damage extents that are slightly shifted from the damage extents used for the training data set, namely 2% to 57% at intervals of 5%, are selected to calculate the damage index vectors. When error is introduced in the mode shapes, for each error level, 40 different damage index data sets are created corresponding to each damage scenario.

Training the neural network

The designed back-propagation neural network is trained in three steps as before. The training algorithm is the scaled conjugate gradient algorithm. The training data set and the validation data set are split into two data sets, respectively. One is composed of the damage index vectors and corresponding damage extent of element 18; another one is composed of the damage index vectors and corresponding damage extent of element 28. First, the neural network is trained using the data sets for element number 18. The trained network is saved and used to predict the damage extent of element number 18. The training process is shown in Figures 5.23 (a), (b) and (c), which indicate the training and validation error between the actual and target output as a function of the numbers of training epochs. Training is then continued using the data set for element number 28. The trained network is saved and used to predict the damage extent of element number 28. The total training time on a Pentium III computer is 6035 seconds. The performance of the trained network is 0.000849036. All of the training processes described above for element numbers 18 and 28 are stopped by the validation error.

Testing the trained neural network

The reliability of the trained networks is measured by testing the networks with damage index vectors obtained for the various mode shape error levels and 6 damage severity ranges, that is, 0% to 10%, 10% to 20%, 20% to 30%, 30% to 40%, and 40% to 50%. To test the effectiveness of the trained network to predict different damage extent levels in

elements, another damage scenario, that is, damage extent between 10% and 20% in element 18 and damage extent between 40% and 50% in element 28, is employed. The creation of the test data set is similar to that in the damage localization of Crowchild Bridge for one-damage-case. These data are fed to the trained network, the outputs of the network are obtained, and by comparing them with the actual damage extents, the relative prediction errors (RPE) are calculated for element numbers 18 and 28. The larger RPE is used in the following statistical process. The RPE is defined as the difference between the magnitude of predicted damage and the magnitude of actual damage as a percentage of the actual damage. The numbers of outputs within a specific range of RPE are summed up; the results are shown in Figure 5.24. The figure shows that, no matter whether the two elements are damaged at the same or different severity levels, the prediction and generalization capabilities of the trained neural network is quite good when the damage extent is significant.

B. Neural network with architecture of 32-64-2

In this section, the number of neurons in output layer is increased to two, so that the damage extents in the two damaged members are output simultaneously. The neural network is trained in three steps to get the damage factor. The training epochs are 769, 19 and 40 for the 3 different training data sets, respectively. The total training time on a Pentium III computer is 6815 seconds. The performance of the trained network is

0.000548858, better than the trained network with one output. Six hundred damage index vectors with different measurement errors in mode shapes are presented to the network. The prediction results are shown in Appendix B, Figure B2.

Compared to the neural network with the structure of 32-64-1, the training time is increased by 13%, but the prediction and generalization capabilities of the trained neural network have improved greatly.

C. Neural network with architecture of 32-64-64-2

Generally, more hidden layers make the performance of the network more reliable, so a network architecture with two hidden layers and one output layer is examined. Each input vector has 32 elements, which are the damage indices corresponding to 32 chord elements. Each of the two hidden layers has 64 neurons and the output layer has two outputs, which are the predicted damage extents. The transfer functions in hidden layers are tangent sigmoid functions, while in the output layer it is a pure linear function.

The neural network is trained in three steps to get the damage factor as before. The training epochs are 445, 58 and 361 for the 3 different training data sets, respectively.

The total training time on a Pentium III computer is 14145 seconds. The performance of the trained network is 0.000311068, slightly better than the trained network in Section B.

Six hundred damage index vectors with different measurement errors in mode shapes are presented to the network. The prediction results, shown in Appendix B, Figure B3, are

slightly better compared to the trained network with the structure of 32-64-1, but the training time has increased by 107%.

D. Neural network with architecture of 32-96-2 and 32-32-2

Assuming that 96 neurons being are used in the hidden layer, the neural network is trained in three steps using different data sets. The total training time on a Pentium III computer is 11788 seconds. Six hundred damage index vectors with different measurement errors in mode shapes were presented to the network. The prediction results are shown in Appendix B, Figure B4.

Compared to the trained network with 64 neurons in the hidden layer, the training time is increased by 73%; however, the performance of the trained neural network has not improved.

More neurons mean more training time, which maybe very expensive for complex structures, although more neurons may at times make the performance of the network more reliable. To reduce the training time, fewer neurons may be provided in the hidden layer. Assume that 32 neurons are used in the hidden layer. The neural network is now trained in three steps. The total training time on a Pentium III computer is 1260 seconds. Six hundred damage index vectors with different measurement errors in mode shapes were presented to the network. The prediction results are shown in Appendix B, Figure B5.

Compared to the trained network with 64 neurons in hidden layer, the training time is reduced significantly, but the performance of the trained neural network is significantly worse.

E. Neural network trained with the data sets of different size

To explore the effect of the size of the data set on the performance of the trained network, the following two data sets of different sizes are employed.

For each damage combination state and error level, 100 training example pairs are created. The network used here and the training algorithm are the same as those in Section B. The total training time on a Pentium III computer is 21518 seconds. Six hundred damage index vectors with different measurement errors in incomplete mode shapes were presented to the network. The prediction results are shown in Appendix B, Figure B6.

The training time has increased by 216%, but the prediction and generalization capabilities of the trained neural network become even worse than those of the network trained with the data sets described in Section B.

Next, assume that 20 training example pairs are created for each damage combination state and each error level. The neural network and the training algorithm used in Section B are used here, too. The total training time on a Pentium III computer is 3122 seconds. Six hundred damage index vectors with different measurement errors in

incomplete mode shapes are presented to the network. The prediction results are shown in Figure 5.25.

Compared to the network trained in Section B, the training time has reduced considerably, by as much as 54%, but the prediction and generalization capabilities of the trained neural network have not suffered.

F. Neural network with different transfer function in the output layer

The output from a logistic sigmoid function ranges from 0 to 1; it is suitable for use as the transfer function to output the damage factor, which also varies from 0 to 1. The network used in Section E is redesigned by using the logistic sigmoid transfer function in the output layer. The network is trained in the same way as in Section E using the smaller data set. The total training time on a Pentium III computer is 1020 seconds. Six hundred damage index vectors with different measurement errors in incomplete mode shapes are presented to the trained network. The prediction results are shown in Appendix B, Figure B7.

The training time has increased by 67%, but the prediction and generalization capabilities of the trained neural network become worse as compared to the network in Section E, trained with a smaller data set.

G. Neural network with Different learning algorithm

Demuth et al. (2002) show that the resilient backpropagation training algorithm and the scaled conjugate gradient training algorithm are the two most suitable for pattern recognition problems. Based on this fact, the network is also trained using the resilient backpropagation training algorithm. The training process, shown in Appendix B, Figure B8, is very unstable. The total training time on a Pentium III computer is 5208 seconds. Six hundred damage index vectors with different measurement errors in incomplete mode shapes are presented to the network. The prediction results are shown in Appendix B, Figure B9.

Compared to the network trained with smaller data sets using the scaled conjugate gradient backpropagation algorithm in Section E, the training time has increased by 67%, but the prediction and generalization capabilities of the trained neural network are even worse.

H. Neural network with Different validation parameter

The scaled conjugate gradient training algorithm is a training function that updates weight and bias values according to the scaled conjugate gradient method. The validation data set is used to stop training early if the network performance on the validation vectors fails to improve or remains the same for ten epochs in a row. In this section, the training parameter is set so that the training stops when validation error has

increased more than fifteen times since the last time it decreased. The neural network is trained in the same way as before. The total training time on a Pentium III computer is 2310 seconds. Six hundred damage index vectors with different measurement errors in incomplete mode shapes were presented to the network. The prediction results are shown in Appendix B, Figure B10.

Compared to the network in Section E trained with a smaller data size, the training time is reduced by 26%, but the prediction and generalization capabilities of the trained neural network become worse.

In conclusion, the optimal neural network has the architecture of 32-64-2, that is, one in which the input damage index vector has 32 elements corresponding to 32 finite elements, the network has one hidden layer with 64 neurons, and the output layer has two neurons to output the damage extent. The transfer functions are a tangent sigmoid function in the hidden layer and a pure linear function in the output layer. The data sets are created from the mode shapes both with and without error. For each damage status and error level, 20 data pairs are calculated. The training algorithm is the scaled conjugate gradient algorithm. The performances of different networks are compared in Table 4.4(b).

To verify that the optimal architecture for the neural network is not dependent on the specific elements that are damaged, sets of calculations similar to those described in the

earlier paragraphs were repeated for several different combinations of damaged elements, and the performance index and accuracy of damage magnitude predictions were compared. The results obtained are not presented here, but in general, the optimal architecture remained unchanged.

Table 5.1: Properties of the space truss components

Component	Dimensions (mm)	Cross-sectional area (mm ²)	Inertia (mm ⁴)	Unit mass (kg/m)
Aluminium node	46 diameter	-	-	-
Tubes in (x-y) planes	707×30×1.5	134.3	13673.7	0.3626
Tubes in (x-z) planes (other than those included in (x-y) planes)	707×22×1.0	65.97	3645.0	0.1781
Diagonal tubes in (x-y) planes	1000×30×1.5	134.3	13673.7	0.3626
Diagonal tubes in (y-z) planes	1000×22×1.0	65.97	3645.0	0.1781

Table 5.2: 3D truss element connectivity

Element No.	Incident Node i	Incident Node j	Element No.	Incident Node i	Incident Node j
Chords 1	1	5	56	21	22
2	5	9	57	25	26
3	9	13	58	29	30
4	13	17	59	33	34
5	17	21	Horizontals 60	3	4
6	21	25	61	7	8
7	25	29	62	11	12
8	29	33	63	15	16
Chords 9	2	6	64	19	20
10	6	10	65	23	24
11	10	14	66	27	28
12	14	18	67	31	32
13	18	22	68	35	36
14	22	26	Face diagonal 69	1	7
15	26	30	70	9	15
16	30	34	71	17	23
Chords 17	3	7	72	25	31
18	7	11	73	7	9
19	11	15	74	15	17
20	15	19	75	23	25

	21	19	23	76	31	33
	22	23	27	77	4	6
	23	27	31	78	12	14
	24	31	35	79	20	22
Chords	25	4	8	80	28	30
	26	8	12	81	6	12
	27	12	16	82	14	20
	28	16	20	83	22	28
	29	20	24	84	30	36
	30	24	28	Horizontal diagonals 85	2	5
	31	28	32	86	10	13
	32	32	36	87	18	21
Verticals	33	1	3	88	26	29
	34	5	7	89	5	10
	35	9	11	90	13	18
	36	13	15	91	21	26
	37	17	19	92	29	34
	38	21	23	93	3	8
	39	25	27	94	11	16
	40	29	31	95	19	24
	41	33	35	96	27	32
Verticals	42	2	4	97	8	11
	43	6	8	98	16	19
	44	10	12	99	24	27
	45	14	16	100	32	35
	46	18	20	Cross diagonals 101	2	3
	47	22	24	102	10	11
	48	26	28	103	18	19
	49	30	32	104	26	27
	50	34	36	105	34	35
Horizontals	51	1	2	106	5	8
	52	5	6	107	13	16
	53	9	10	108	21	24
	54	13	14	109	29	32
	55	17	18			

Table 5.3: Training of neural networks for damage extent in the 3D space truss

Neural networks	Training epochs			Training Time (s)	Performance (MSE)
	Stage 1	Stage 2	Stage 3		
32-64-1, trainscg, tansig, purelin, maxfail=10, data 50.	385 (EL18)	20 (EL18)	66 (EL18)	6035	0.000849036 (EL18)
	135 (EL28)	111 (EL28)	145 (EL28)		0.000477644 (EL28)
32-64-2, trainscg, tansig, purelin, maxfail=10, data 50.	769	19	40	6815	0.000548858
32-64-64-2, trainscg, tansig, purelin, maxfail=10, data 50.	445	58	361	14145	0.000311068
32-96-2, trainscg, tansig, purelin, maxfail=10, data 50.	705	15	238	11788	0.000545645
32-32-2, trainscg, tansig, purelin, maxfail=10, data 50.	201	108	90	1260	0.0010069
32-64-2, trainscg, tansig, purelin, maxfail=10, data 100.	557	240	214	21518	0.000591813
32-64-2 , trainscg, tansig, purelin, maxfail=10, data 20.	843	85	78	3122	0.000559252
32-64-2 , trainscg, tansig, logsig, maxfail=10, data 20.	195	106	103	1020	0.00121935
32-64-2, trainrp, tansig, purelin, maxfail=10, data 20.	2741	11	11	5208	0.144695
32-64-2, trainscg, tansig, purelin, maxfail=15, data 20.	622	85	59	2310	0.000752529

Note:

32-64-2: 32 elements in input vector, 64 neurons in hidden layer, 2 neurons in output layer

trainscg: the scaled conjugate gradient training algorithm

trainrp: the resilient backpropagation training algorithm

tansig: the tangent sigmoid transfer function

purelin: the pure linear transfer function

maxfail: the maximum number of iterations to validation error failure

data 20: 20 damage index vectors created corresponding to each damage extent and error level in the training data set.

The transfer function that appears first is for the hidden layer.

The transfer function that appears second is for the output layer.

Figure 5.1: Details of the Meroform strut and nodes (dimensions in mm)

Figure 5.2: Finite element model of the eight-bay space truss

Figure 5.3: Modal strain energies contributed by the elements of the 3D truss in the first six modes, (a) 1st mode, (b) 2nd mode, (c) 3rd mode, (d) 4th mode, (e) 5th mode, (f) 6th mode, (g) total of six modes.

Figure 5.4: Damage in chord members, 2% error in incomplete mode shapes, (a) 30% damage in chord element No. 1, (b) 30% damage in chord element No. 2, (c) 30% damage in chord element No. 30, (d) 20% damage in each of the chord elements Nos. 18 and 28

Figure 5.5: Damage 30% in vertical element No. 45, (a) accurate incomplete modes corresponding to 36 x degrees of freedom, (b) accurate incomplete modes corresponding to 36 y degrees of freedom, (c) accurate incomplete modes corresponding to 36 z degrees of freedom, (d) accurate complete modes, (e) incomplete modes corresponding to 36 x degrees of freedom with 2% error (f) incomplete modes corresponding to 36 y degrees of freedom with 2% error, (g) incomplete modes corresponding to 36 z degrees of freedom with 2% error, (h) complete modes with 2% error.

Figure 5.6: Damage 30% in horizontal element No. 63 in z direction, (a) accurate incomplete modes corresponding to 36 x degrees of freedom, (b) accurate incomplete modes corresponding to 36 y degrees of freedom, (c) accurate incomplete modes corresponding to 36 z degrees of freedom, (d) accurate complete modes, (e) incomplete modes corresponding to 36 x degrees of freedom with 2% error (f) incomplete modes corresponding to 36 y degrees of freedom with 2% error, (g) incomplete modes corresponding to 36 z degrees of freedom with 2% error, (h) complete modes with 2% error.

Figure 5.7: Damage 30% in horizontal element No. 55 in z direction, (a) accurate incomplete modes corresponding to 36 x degrees of freedom, (b) accurate incomplete modes corresponding to 36 y degrees of freedom, (c) accurate incomplete modes corresponding to 36 z degrees of freedom, (d) accurate complete modes, (e) incomplete modes corresponding to 36 x degrees of freedom with 2% error, (f) incomplete modes corresponding to 36 y degrees of freedom with 2% error, (g) incomplete modes corresponding to 36 z degrees of freedom with 2% error, (h) complete modes with 2% error.

Figure 5.8: Damage 30% in diagonal element No. 74 in x - y plane, (a), (b), (c) accurate incomplete modes corresponding to 36 x , y , and z degrees of freedom, respectively, (d), (e), (f) accurate incomplete modes corresponding to 72 xy , yz , and xz degrees of freedom, respectively, (g) accurate complete modes, (h), (i), (j) incomplete modes with 2% error corresponding to 36 x , y , and z degrees of freedom, respectively, (k), (l), (m) incomplete modes with 2% error corresponding to 72 xy , yz , and xz degrees of freedom, respectively, (n) complete modes with 2% error

Figure 5.9: Damage 30% in diagonal element No. 107 in y-z plane, accurate modes, (a), (b), (c) incomplete modes corresponding to 36 x , y , and z degrees of freedom, respectively, (d), (e), (f) incomplete modes corresponding to 72 xy , yz , and zx degrees of freedom, respectively, (g) complete modes

Figure 5.10: Damage 30% in diagonal element No. 94 in x - z plane, (a), (b), (c) accurate incomplete modes corresponding to 36 x , y , and z degrees of freedom, respectively, (d), (e), (f) accurate incomplete modes corresponding to 72 xy , yz , and xz degrees of freedom, respectively, (g) accurate complete modes, (h), (i), (j) incomplete modes with 2% error corresponding to 36 x , y , and z degrees of freedom, respectively, (k), (l), (m) incomplete modes with 2% error corresponding to 72 xy , yz , and xz degrees of freedom, respectively, (n) complete modes with 2% error

Figure 5.11: 30% damage in chord elements No. 18 connecting nodes 7 and 11 and No. 1 connecting nodes 1 and 5, incomplete mode shapes corresponding to x degrees of freedom with 2% error

Figure 5.12: 30% damage in chord elements No. 18 connecting nodes 7 and 11 and No. 28 connecting nodes 16 and 20, incomplete mode shapes corresponding to x degrees of freedom with 2% error

Figure 5.13: 30% damage in elements No. 21 connecting nodes 19 and 23 and No. 31 connecting nodes 28 and 32, incomplete mode shapes corresponding to x degrees of freedom with 2% error

Figure 5.14: 30% damage in elements No. 18 connecting nodes 7 and 11 and No. 30 connecting nodes 24 and 28, incomplete mode shapes corresponding to x degrees of freedom with 2% error

Figure 5.15: The first step of the substructure method, 30% damage in original element 18 and 28 (substructures 1 and 2) simultaneously, 6 incomplete modes with, (a) 2% error, (b) 4% error.

Figure 5.16: The second step of the substructure method, 30% damage in elements 10 and 16 (original elements 18 and 28) simultaneously, 6 incomplete modes with, (a) 2% error, (b) 4% error.

Figure 5.17: Damage indices for 30% damage in element 10 connecting nodes 7 & 11, and 30% damage in element 16 connecting nodes 16 & 20 obtained by sub-structuring; 6 modes with, (a) 2% random errors, (b) 4% random errors

Figure 5.18: The perceptron neural network for damage location in 3D space truss

(a)

(b)

(c)

Figure 5.19: Prediction of damage location using perceptron network, damage in element No. 17, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%.

(a)

(b)

Figure 5.20: Prediction of damage location using perceptron network, damage in element Nos. 17 and 18, (a) damage between 0% and 10%, (b) damage between 20% and 30%, (c) damage between 30% and 40%.

Figure 5.21: The initial backpropagation neural network designed for evaluating damage extent in 3D space truss

(a)

(b)

(c)

(d)

Figure 5.22: Prediction of damage extent using backpropagation network, damage in element No. 18, one-element damage case, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

Figure 5.23: Training and validation errors of the network with the architecture of 32-64-1 for the damage factor of element No. 18 trained using damage index vectors from incomplete mode shapes both with and without error, (a) stage 1, (b) stage 2, (c) stage 3

(e)

(f)

Figure 5.24: Prediction of damage extent using backpropagation network with the architecture of 32-64-1, damage in elements 18 and 28, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%, (f) damage between 10% and 20% in element 18 and damage between 40% and 50% in element 28.

(c)

(d)

(e)

Figure 5.25: Prediction of damage extent using backpropagation network with the architecture of 32-64-2 and smaller training data set, damage in elements 18 and 28, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%

Chapter 6

Summary and Conclusions

6.1 Summary

In the present study, the most commonly used vibration based damage algorithms are first reviewed by applying them to a simply supported beam. The modal energy based damage index is found to be a relatively efficient parameter for locating the damage. A two-step algorithm is then proposed to predict the damage location and severity. The method relies on the fact that damage would reduce the stiffness of damaged elements and is based on measurement of vibration modes in both the healthy and damaged states of the structure. It splits the problem into two distinct sub-problems. In the first step, the damage location is determined by using the strain energy technique. The damage index based on the modal strain energy is used either directly from the plots of damage indices for individual elements in the model, or as input to a perceptron neural network when only a single element is likely to be damaged. For complex structures, a sub-structuring technique is proposed to deal with the large number of degrees of freedom. Once the

location of damage has been determined, the magnitude of damage in the affected elements is evaluated by a backpropagation neural network in the second step. The designed neural network is trained on examples of simulated damage in elements and the corresponding damage indices from incomplete and truncated mode shapes and in the presence of measurement errors. The relatively optimal architecture of the neural network is obtained by evaluating a series of configurations and comparing the training time and the performance of the trained network.

The damage index is calculated in two different ways. It can be obtained from the modal strain energy either by expanding the measured mode shapes to full degrees of freedom using SEREP, or based on the modal curvature. The vibration modes could be translational or curvature based. A central difference expression is used to evaluate the curvature when only the translational components of the measured mode shapes are available. In the field, the curvature mode shapes would be obtained by attaching strain gages capable of measuring dynamic strains at two points over the depth of the element. For computer simulation, the curvature along an element is obtained by taking the difference in the modal rotations at the two ends of the element and dividing by the element length. Generally, the performance of the damage index based on directly measured curvature mode shapes is much better.

For large and complex structures, a sub-structuring technique is used. In this two-step process, a coarse array of sensors is installed in the first step. Modal measurements taken

from these sensors and the damage indices obtained from them would be able to identify the one or more substructures of the original structure that may contain the damage. In the second step, finer arrays of sensors are installed in the identified substructure, keeping only a few sensors away from the identified area. The step should provide a more refined prediction of the damage location.

The computer simulation studies show that the proposed sub-structuring and neural network techniques are successful and quite effective in localizing and quantifying the damage in a girder model of the Crowchild Bridge and the 3D space truss studied with reasonable accuracy, depending on the amount of the random error in the measurement data and provided the damage magnitude is not too small. The generalization capability of the trained network is reliable even for the measurement data with very large error.

6.2 Conclusions

The following important conclusions can be drawn from the computational studies in the present research:

1. The proposed two-step identification algorithm was developed to first identify the damaged elements using the modal strain energy based damage index either directly from the plots of the damage indices or as input to the perceptron neural work. The same indices were then applied as input to a backpropagation network to evaluate the damage magnitudes of the damaged elements. The computer

simulation studies show the method is quite successful even for complex structures and comparatively large measurement errors in mode shapes, especially when more than 20% damage is induced in one or two elements at a time. The method is shown to work for two different structural types: a beam type structure and a truss type structure.

2. Modal energy based damage indices are found to be quite effective in predicting the location of damage on the basis of comparison of the relative efficiency of various commonly used vibration based damage identification algorithms in locating the damage in a 20 m simply supported beam.
3. For a beam type structure, plots of modal energy based damage indices calculated from translational mode shapes can locate the damaged elements accurately only when the mode shapes are measured accurately. This may be attributed to the errors introduced by numerical differentiation involved in obtaining the curvatures. The numerical differentiation always magnifies the measurement errors and therefore introduces uncertainty in the predictions.
4. A perceptron neural network trained with simulated data from finite element analysis could be more effective in locating the damage in comparison to a direct application of damage indices calculated from translational modes. This demonstrates the effectiveness of neural networks in dealing with uncertainty. At the present stage of study, the perceptron network has been demonstrated to work

when a single element has been damaged.

5. The computer simulation studies also show that the modal energy based damage index vector is quite efficient as input to the backpropagation neural network to evaluate the damage extent. The trained network gives reasonable prediction even when the measurement errors ranging from -5% to $+5\%$ are added to the elements of the mode shapes. This also verifies the error-tolerant property of the artificial neural network again.
6. The backpropagation neural networks presented here are quite effective in predicting the damage magnitude after the damage location has been determined, provided the damage magnitude is not too small. The generalization capability of the trained networks is reliable even when the measured data have significant errors.
7. This study clearly shows that curvature modes obtained from measurements of dynamic strains provide a robust means of locating damage even when significant measurement errors exist. The backpropagation neural network trained with the data sets based on the directly measured curvature mode shapes performs much better when evaluating the damage severity.
8. The size of the optimal data set was obtained for the backpropagation neural network by a series of numerical experiments. The results show that the neural network trained with a larger data set is not necessarily better than that trained

with a smaller data set.

9. The performance of the trained backpropagation network does not always improve with the number of layers and neurons in each layer; in fact, sometimes it may get worse.
10. The scaled conjugate gradient training algorithm is found to be efficient for the neural network used in the damage evaluation of the Crowchild Bridge and the 3D space truss; the training process is smooth and the performance of the trained network is satisfactory. By contrast, the training process of the resilient backpropagation training algorithm is not stable, especially at the beginning of the training, and is time-consuming. The early stopping technique played an important role in stopping the training in time to maintain the generalization capability of the neural network.
11. A sub-structuring technique was proposed for dealing with real-world complex structures with a large number of degrees of freedom. The structure is first divided into a number of substructures. The mode shape values are measured along only those degrees of freedom that lie at the junctions of the individual substructures. If the damage indices calculated at the substructure level indicate there is possible damage in one or more substructures, a second round of modal measurements can be carried out in which additional measurements are made along degrees of freedom that are internal to the affected substructures. This process continues until

the damaged elements are accurately identified. The computer simulation studies on the Crowchild Bridge and the 3D space truss show that the sub-structuring technique is effective for locating the damage.

6.3 Work needed to be done in the future research

On the basis of the case studies presented in this work, it is clear that the use of neural networks for structural damage assessment is a promising subject of research. The results indicate that neural networks are capable of learning about the behaviour of undamaged and damaged structures and to identify the extent of the damage from response patterns of the structure in the frequency domain. However, as shown in this study, some issues remain to be resolved before this approach becomes a truly viable method of structural damage assessment. Firstly, there is a need for developing the artificial neural network to evaluate light damage in a structure. Secondly, at this time the perceptron neuron network is successful in locating the damage in only one-element damage cases. For two-element damage cases, a better input signature and an adequate size of the training data set should be found for the perceptron network. Thirdly, the information needed for training a neural network, such as how many, and at what locations sensor recordings must be obtained in order to reliably identify the damaged structural members, must be quantified. Finally, it will be useful to investigate the application of other types of artificial neural networks such as the multiplayer perceptron

neural network, the probabilistic neural network or the competitive neural network etc.

Much work has been dedicated to solving the problem of detecting the location and extent of damage in elastic structures by using the changes in their modal parameters; namely, the natural frequencies and the associated mode shapes. Further development of methods that are able to account for the effects of non-linear structural response is needed for damage identification by means of structural dynamics technique or artificial neural network techniques.

The modal energy based techniques do not work well when the damaged element contributes little energy in the measured mode shapes. Further work is needed to find alternative vibration parameters that may be effective in such cases.

It should be realized that the vibration characteristics could be altered by factors other than damage. For structures with redundant constraints, variations in temperature, and other environmental and boundary conditions change the natural frequencies and mode shapes. Further studies need to be carried out to apply the proposed two-step algorithm to account for the effects of environmental and boundary conditions and other uncertainties.

References

1. Afhami S., and Cheng J. J. R. 1999. Field instrumentation and monitoring of Crowchild Bridge in Calgary, Alberta. Report prepared for the City of Calgary.
2. Aktan A. E., Lee K. L., Chuntavan C., and Aksel T. 1994. Modal testing for structural identification and condition assessment of constructed facilities. Proceedings of 12th international modal analysis conference, Honolulu, Hawaii, 462-468.
3. Auweraer H. V., and Peeters B. 2003. International research projects on structural health monitoring: an overview. Structural health monitoring. Vol. 2, No. 4, 341-358.
4. Avitabile, P., O'Callahan, J.C., and Milani, J. 1989. Comparison of system characteristics using various model reduction techniques. 7th International Modal Analysis Conference, Las Vegas, Nevada, 777-785.
5. Baruch M., Bar Itzhack I. Y. 1978. Optimum weighted orthogonalization of measured modes. AIAA Journal, Vol. 16, No. 4, 346-351.
6. Berman A., and Nagy E. J. 1983. Improvement of large analytical model using test data, AIAA Journal, Vol. 21, No. 8, 1168-1173.
7. Bishop C. M. 1995. Neural networks for pattern recognition, Oxford University Press Inc., New York.

8. Cawley P., and Adams R. D. 1979. The location of defects in structures from measurements of natural frequencies. *Journal of strain analysis*, Vol. 14, No. 2, 49-57
9. Chance J., Tomlinson G. R. and Worden K. 1994. A simplified approach to the numerical and experimental modelling of the dynamics of a cracked beam. *Proceedings of the 12th international modal analysis conference, Honolulu, Hawaii*, 778-785.
10. Chen J. C., and Garba J. A. 1988. On-orbit damage assessment for large space structures. *AIAA Journal*, Vol. 26, No. 9, 1119-1126.
11. Chen J. C., and Garba J. A. 1980. Analytical model improvement using modal test results. *AIAA Journal*, Vol. 18, No. 6, 684-690.
12. Chen, S., and Shah, K. 1992. Neural networks in dynamic analysis of bridges. *Proceedings of the eighth annual conference of computing in civil engineering, ASCE, Dallas*, 1058-1065.
13. Chen, S., and Kim S. 1995. Neural network based sensor signal monitoring of instrumented structures. *Proceedings of structures congress XIII, Boston*.
14. Cornwell, P. J., Doebling S. W., and Farr C. R. 1997. Application of the strain energy damage detection method to plate-like structures. *Proceedings of the 15th international modal analysis conference, Orlando, FL*, 1312-1318.
15. Demuth, H., Beale M. 2002. *Neural network toolbox user's guide. Version 4. The MathWorks, Inc., Natick, MA.*

16. Doebling, S. W., Farrar, C. R., and Prime, M. B. 1998. A summary review of vibration based damage identification methods, *The Shock and Vibration Digest*, 30: 91-105
17. Doebling, S.W., Farrar, C.R., Prime, M.B., and Shevitz, D.W. 1996. Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review. Los Alamos National Laboratory, Los Alamos, New Mexico, Report No. LA 13070-MS.
18. Elkordy, M. F., K. C. Chang, and Lee G. C. 1993. Neural network trained by analytically simulated damage states. *ASCE journal of computing in civil engineering*, Vol. 7, No. 2, 130-145.
19. Farrar, C.R, Baker, W.E., Bell, T.M., Cone, K.M., Darling, T.W., Duffey, T.A., Eklund, A., and Migliori, A. 1994. Dynamic characterization and damage detection in the I-40 bridge over the Rio Grande. Los Alamos National Laboratory, Los Alamos, New Mexico, Report No. LA 12767-MS.
20. Farrar C. R., and Cone K. M. 1995. Vibration testing of the I-40 Bridge before and after the introduction of damage. *Proceedings of the 13th international modal analysis conference*, Nashville, Tennessee, 203-209.
21. Farrar C. R., and Jauregui D. 1996. Damage detection algorithms applied to experimental and numerical modal data from the I-40 bridge. Los Alamos National Laboratory, Los Alamos, New Mexico, Report No. LA 13074-MS..

22. Farrar, C. R., and Sohn H. 2001. Condition/damage monitoring methodologies. The consortium of organizations for strong motion observation systems workshop, invited talk, Emeryville, CA., LA-UR-01-6573.
23. Fletcher, R. 1980. Practical methods of optimisation, Vol. 1, Unconstrained optimisation, and Vol. 2 Constrained optimisation. John Wiley & Sons, Inc., New York.
24. Fox C. H. J. 1992. The location of defects in structures: A comparison of the use of natural frequency and mode shape data. Proceedings of the 10th international modal analysis conference, San Diego, 522-528
25. Gill, P.E., Murray, W., and Wright, M.H. 1981. Practical optimisation. Academic Press, London.
26. Guyan R. J. 1965. Reduction of stiffness and mass matrices. AIAA Journal, V3 No.2. p380-381
27. Gysin H. P. 1986. Critical application of an error matrix method for location of finite element modelling inaccuracies. Proceedings of the 4th international modal analysis conference, Los Angeles, 1339-1351
28. Hagan M. T., Demuth, H. B., and Beale, M. 1995. Chapter 11, Backpropagation, Neural network design, PWS publishing company, Boston, Massachusetts.
29. He J., Ewins D. J. 1986. Analytical stiffness matrix correction using measured vibration modes. Modal analysis: The international journal of analytical and

- experimental modal analysis, Vol. 1, No. 3, 9-14.
30. Hearn G, and Testa R. B. 1991. Modal analysis for damage detection in structures. *Journal of structural engineering*, ASCE, Vol. 117, No. 10, 3042-3063.
 31. Hecht-Nielsen, R. 1987. Proceedings of first IEEE international joint conference, Neural networks, San Diego, CA.
 32. Hemez, F. M. 1993. Theoretical and experimental correlation between finite element models and modal tests in the context of large flexible space structures. PhD thesis, Aerospace engineering sciences, University of Colorado, Boulder, CO.
 33. Hervas Martinez, C., Romero Soto, E. J., Garcia Pedrajas, N., and Medina Carnicer, R. 1992. Comparison between artificial neural networks and classical statistical methods in pattern recognition. Proceedings of 4th International Conference on Processing and Management of Uncertainty in Knowledge-Based Systems. Palma de Mallorca, Spain, 351-360.
 34. Jauregui, D. V., and Farr, C. R. 1996. Comparison of damage identification algorithms experimental modal data from a bridge. Proceedings of the 14th International Modal Analysis Conference, Dearborn, MI, 1423-1429.
 35. Ju, F. D., and Mimovich M. 1986. Modal frequency method in diagnosis of fracture damage in structures. Proceedings of 4th international modal analysis conference, Los Angeles, 1168-1174.
 36. Kabe, A. M. 1985. Stiffness matrix adjustment using mode data. *AIAA Journal*, Vol.

- 23, No. 9, 1431-1436.
37. Kammer, D. C. 1988. Optimal approximation for residual stiffness in linear system identification. *AIAA Journal*, Vol. 26, No. 1, 104-112.
 38. Kaouk, M., and Zimmerman, D. C. 1994. Structural damage assessment using a generalized minimum rank perturbation theory. *AIAA Journal*, Vol. 32, No. 4, 836-842.
 39. Kirkegaard P., and Rytter, A. 1994. Use of neural networks for damage assessment in a steel mast. Proceedings of the 12th international modal analysis conference, Honolulu, Hawaii, 1128-1134
 40. Kudva, J., Munir, N. and Tan, P. 1991. Damage detection in smart structures using neural networks and finite element analysis. Proceedings of ADPA/AIAA/ASME/SPIE conference on active materials and adaptive structures, Alexandria, Virginia, 559-562
 41. Leath, W. J., and Zimmerman, D. C. 1993. Analysis of neural network supervised training with application to structural damage detection. Damage and control of large structures, Proceedings of the 9th VPI & SU symposium, Blackburg, Virginia, 583-594.
 42. Lim, T. W. 1990. A submatrix approach to stiffness using modal test data. *AIAA journal*, Vol. 28, No. 6, 1123-1130.
 43. Lim, T. W. 1991. Structural damage detection using modal test data. *AIAA journal*,

Vol. 29, No. 12, 2271-2274.

44. Masri, S. F., Nakamura, M., Chassiakos, A. G., and Caughey, T. K. 1996. A neural network approach to the detection of changes in structural parameters. *Journal of engineering mechanics*, ASCE, Vol. 122, No. 5.
45. Matlab – The language of technical computing, Version 6.5. The MathWorks Inc., Natick, MA. USA.
46. Mazurek, D. F., and DeWolf, J. T. 1990. Experimental study of bridge monitoring technique. *Journal of structural engineering*, Vol. 116, No. 9, 2532-2569.
47. McCulloch, W., and Pitts W. 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical biophysics*, Vol. 5, 115-133.
48. Messina, A., Jones, I. A., and Williams, E. J. 1996. Damage detection and localization using natural frequency changes. *Proceedings of conference on identification in engineering systems*, U. K., 67-76.
49. Moller, M. F. 1993. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, Vol. 6, p525-533.
50. O'Callahan, J., Avitabile, P., and Riemer, R. 1989. System equivalent reduction expansion process (SEREP). 7th International Modal Analysis Conference, Las Vegas, Nevada, 29-37.
51. Pandey, A. K., and Biswas, M. 1994. Damage detection in structures using changes in flexibility. *Journal of Sound and Vibration*, Vol. 169, No. 1, 3-17.

52. Pandey, A.K., Biswas, M., and Samman, M.M. 1991. Damage detection from changes in curvature mode shapes. *Journal of Sound and Vibration*, Vol. 145, No. 2, 321-332.
53. Rosenblatt, F. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, Vol. 65, 386-408.
54. Rytter, A. 1993. Vibration based inspection of civil engineering structures. Ph. D. Dissertation. Department of building technology and structural engineering, Alborg University, Denmark.
55. Salawu, O. S. 1997. Detection of structural damage through changes in frequency: a review. *Engineering structures*, Vol. 19, No. 9, 718-723.
56. Salawu, O. S., and Williams, C. 1995. Bridge assessment using forced-vibration testing. *Journal of structural engineering*, ASCE, Vol. 121, No. 2, 161-173.
57. Shen, J. Y., Sharpe, L. Jr. and Jankovshy, A. L. 1998. An overview of vibration-based non-destructive evaluation techniques. *SPIE*, Vol. 3397, 117-128.
58. Smith, S. W., and Beatti C. A. 1991. Model correlation and damage location for large space truss structures: Secant method development and evaluation. NASA report NASA-CR-188102.
59. Smith, S. W. 1988. Damage detection and location in large space trusses. PhD thesis. Department of Aerospace and Ocean Engineering, Virginia Polytechnic Institute and State University, VA.

60. Spillman, W., Huston, D., Fuhr, P., and Lord, J. 1993. Neural network damage detection in a bridge element. SPIE smart sensing, processing, and instrumentation, Vol. 1918, 288-295.
61. Stubbs, N., Kim, J. -T., and Farrar, C.R. 1995. Field verification of a non-destructive damage localization and severity estimation algorithm. 13th International Modal Analysis Conference, Nashville, Tennessee, 210-218.
62. Stubbs, N., and Osegueda, R. 1990. Global non-destructive damage evaluation in solids. Modal analysis: The international journal of analytical and experimental modal analysis, Vol. 5, No. 2, 67-79.
63. Toksoy, T., and Aktan, A. E. 1994. Bridge-condition assessment by modal flexibility. Experimental mechanics, Vol. 34, 271-278.
64. Widrow, B., and Hoff, M. E. 1960. Adaptive switching circuits. IRE WESCON convention record (Part 4), New York, 96-104.
65. Wolff, T., and Richardson M. H. 1989. Fault detection in structures from changes in their modal parameters. Proceeding of the 7th international modal analysis conference, Las Vegas, Nevada, 87-94.
66. Wood, M. G. 1992. Damage analysis of bridge structures using vibrational techniques. PhD thesis, Department of mechanical and electrical engineering, University of Aston, Brimingham, UK.
67. Wu, X., Ghaboussi, J., and Garrett, J. H. Jr. 1992. Use of neural networks in

detection of structural damage. Computers and Structures, Vol. 42, No. 4, 649-659.

68. Zhang, Z., and Aktan, A. E. 1995. The damage indices for the constructed facilities. Proceedings of the 13th international modal analysis conference, Nashville, Tennessee, 1520-1529.

Appendix A

(a)

(b)

(c)

(d)

(e)

Figure A1: Prediction of damage extent using backpropagation network of architecture 15-30-30-2, damage index vectors based on curvature obtained from translational mode shapes, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

Figure A2: Prediction of damage extent using backpropagation network of architecture 15-30-2, damage index vectors based on curvature obtained from translational mode shapes, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

Figure A3: Prediction of damage extent using backpropagation network of architecture 15-30-1, damage index vectors based on curvature obtained from translational mode shapes, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

Figure A4: Prediction of damage extent using backpropagation network of architecture 15-15-1, damage index vectors based on curvature obtained from translational mode shapes, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

Figure A5: Prediction of damage extent using backpropagation network of architecture 15-45-1, damage index vectors based on curvature obtained from translational mode shapes, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

(e)

Figure A6: Prediction of damage extent using backpropagation network of architecture 15-30-1, damage index vectors based on curvature obtained from translational mode shapes, damage in element Nos. 3 and 8, validation failure for fifteen consecutive epochs, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

Figure A7: Prediction of damage extent using backpropagation network of architecture 15-30-1, damage index vectors based on curvature obtained from translational mode shapes, the resilient backpropagation training algorithm, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

Figure A8: Prediction of damage extent using backpropagation network of architecture 15-30-1 with logistic sigmoid transfer function in the output layer, damage index vectors based on curvature obtained from translational mode shapes, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

(e)

Figure A9: Prediction of damage extent using backpropagation network of architecture 15-30-1, larger training data set, damage index vectors based on curvature obtained from translational mode shapes, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

(e)

Figure A10: Prediction of damage extent using backpropagation network with architecture of 15-30-30-2, damage index vectors based on the curvature obtained from strain measurements, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

(e)

Figure A11: Prediction of damage extent using backpropagation network with architecture of 15-30-2, damage index vectors based on the curvature obtained from strain measurements, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

(e)

Figure A12: Prediction of damage extent using backpropagation network with architecture of 15-30-30-1, damage index vectors based on the curvature obtained from strain measurements, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

(e)

Figure A13: Prediction of damage extent using backpropagation network with architecture of 15-45-45-2, damage index vectors based on the curvature obtained from strain measurements, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

(e)

Figure A14: Prediction of damage extent using backpropagation network with architecture of 15-15-15-2, damage index vectors based on the curvature obtained from strain measurements, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

(e)

Figure A15: Prediction of damage extent using backpropagation network with architecture of 15-30-30-2, damage index vectors based on the curvature obtained from strain measurements, validation failure for fifteen consecutive training epochs, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

(e)

Figure A16: Prediction of damage extent using backpropagation network with architecture of 15-30-30-2, damage index vectors based on the curvature from the strain measurements, resilient backpropagation training algorithm, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

(e)

Figure A17: Prediction of damage extent using backpropagation network with architecture of 15-30-30-2 and larger training data set, damage index vectors based on the curvature obtained from strain measurements, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

(a)

(b)

(c)

(d)

(e)

Figure A18: Prediction of damage extent using backpropagation network with architecture of 15-30-30-2 and smaller training data set, damage index vectors based on the curvature obtained from strain measurements, damage in element Nos. 3 and 8, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%.

Appendix B

Figure B1: Training and validation errors of the network for the damage factor of element No. 18 trained using damage index vectors from translational mode shapes both with and without error, one-element damage case, (a) stage 1, (b) stage 2, (c) stage 3

(a)

(b)

(c)

(d)

Figure B2: Prediction of damage extent using backpropagation network with the architecture of 32-64-2, damage in elements 18 and 28, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%

(a)

(b)

(c)

(d)

Figure B3: Prediction of damage extent using backpropagation network with the architecture of 32-64-64-2, damage in elements 18 and 28, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%

(a)

(b)

Figure B4: Prediction of damage extent using backpropagation network with the architecture of 32-96-2, damage in elements 18 and 28, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%

(a)

(b)

(c)

(d)

(e)

Figure B5: Prediction of damage extent using backpropagation network with the architecture of 32-32-2, damage in elements 18 and 28, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%

(a)

(b)

(c)

(d)

Figure B6: Prediction of damage extent using backpropagation network with the architecture of 32-64-2, larger training data set, damage in elements 18 and 28, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%

(a)

(b)

(c)

(d)

Figure B7: Prediction of damage extent using backpropagation network with the architecture of 32-64-2, smaller training data set, logistic sigmoid transfer function in the output layer, damage in elements 18 and 28, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%

Figure B8: Training and validation errors of the network trained with the resilient propagation algorithm using damage indices from incomplete mode shapes, both with and without error, for the damage extent of elements 18 and 28, smaller data set, (a) stage 1, (b) stage 2, (c) stage 3

(a)

(b)

(c)

(d)

Figure B9: Prediction of damage extent using backpropagation network with the architecture of 32-64-2, resilient backpropagation training algorithm, smaller training data set, damage in elements 18 and 28, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%

(a)

(b)

(c)

(d)

Figure B10: Prediction of damage extent using backpropagation network with the architecture of 32-64-2, 15 epochs of maximum validation failure, smaller training data set, damage in elements 18 and 28, (a) damage between 0% and 10%, (b) damage between 10% and 20%, (c) damage between 20% and 30%, (d) damage between 30% and 40%, (e) damage between 40% and 50%