

UNIVERSITY OF CINCINNATI

December 3, 19 92

I, William J. Stevenson,

hereby submit this as part of the requirements for the degree of:

Doctor of Philosophy

in Mechanical Engineering

It is entitled The Use of Neural Nets in Rotor Balancing

Approved by:

Robert W. Rost
Randall J. Allemang
David L. Brown

# **THE USE OF NEURAL NETS IN ROTOR BALANCING**

A Dissertation submitted to the

Division of Graduate Studies and Research  
of the University of Cincinnati

in partial fulfillment of the  
requirements for the degree of

**DOCTOR OF PHILOSOPHY**

in the Department of Mechanical,  
Industrial, and Nuclear Engineering

1993

by

**WILLIAM J. STEVENSON**

BSc. BEng., University of Stellenbosch, 1971

M. Eng. University of Stellenbosch, 1983

Committee Chair: Robert W. Rost

## ABSTRACT

This dissertation deals with the balancing of high speed rotors whose design speed is at least above the first rotor bending frequency. The technique developed applies to the repeated balancing of the same rotor system, or balancing of similar rotors.

Current methods of high speed or flexible balancing may require many trial runs to determine influence coefficients in order to calculate the corrective mass distribution. Particularly if the system contains multiple unbalance planes or is non-linear, this process becomes an iterative trial and error procedure to reduce the unbalance response to within the allowed limits.

Artificial neural nets are a non-parametric, non-linear technique which has been used for pattern analysis in medical, business, and sonar data. To date there have been relatively few applications in mechanical engineering. In this dissertation a multi-layer feed-forward neural net is taught to utilize the patterns in the unbalance response of a flexible multi-disk shaft which behaves non-linearly, enabling the net to produce the unbalance distribution without any trial runs directly from the unbalance response.

The net was trained to associate a small set of experimentally measured

acceleration responses with the unbalances that caused them. Testing showed that the net was able to indicate the location of unbalance within reasonable tolerances for small unbalances within the trained unbalance envelope, but poorly at the edges and beyond the unbalances trained on. Results from a standard multi-speed multi-plane balancing technique were even further from correct. A simulation with a considerably extended training set produced much improved results compared to those from the original training set, showing that neural net techniques may be used for rotor balancing, but at a considerable investment in acquiring training information.

As a result of the large data requirement, two applications of this technique are possible. In a manufacturing environment with good quality control, and requiring 100% inspection of rotating equipment, a small data set would be adequate to speed up balancing considerably. Alternately, starting with a small training set, those unbalances which cannot be balanced immediately by the net could be added to the training set and the net retrained. Additionally, this technique could be used as a condition monitor.

This method could be extended to non-linear control of rotating systems.

© 1993

William John Stevenson

## ACKNOWLEDGEMENTS

I wish to thank the many individuals who have contributed, directly or indirectly, to this dissertation.

In particular I would like to thank my advisor, Dr. Robert Rost, for his assistance. I would also like to express my gratitude to the other members on my dissertation committee, Dr. Randall Allemang and Dr. David Brown, who freely gave their time and effort in supporting me through this program. Others, like Dr. Ivan Morse, provided useful insights and discussion in the setting up of these experiments. Dr. Timothy Grogan was willing to cross over to the world of mechanical engineering, and assist me in the reverse process of entering the world of computer engineering and artificial intelligence.

I must also thank the people in the UC Structural Dynamics Research Laboratory for creating an enjoyable working environment, and their enthusiasm and willingness to help.

I have appreciated the willingness of the neural net community to provide advice and software (generally via *Internet*). In particular, I have used software from Dr. Etienne Barnard of Pretoria University (conjugate gradient program *optb*), Dr. Scott

Fahlman of Carnegie Mellon University (*Cascade Correlator*), and MITRE Corporation of McLean Virginia (*Aspirin/MIGRAINES*).

Last, but certainly not least, I must thank my wife Isobel for her patience, support, encouragement and understanding during this program. She held the family together while my attention was elsewhere.

This work was made possible by the granting of study leave and sponsorship by the Council for Scientific and Industrial Research, Pretoria, South Africa.

# TABLE OF CONTENTS

|  | Page |
|--|------|
| <b>ABSTRACT</b> .....  | iii  |
| <b>ACKNOWLEDGEMENTS</b> .....                                    | vi   |
| <b>LIST OF FIGURES</b> .....                                     | xi   |
| <b>LIST OF TABLES</b> .....                                      | xiii |
| <b>NOMENCLATURE</b> .....  | xiv  |
| <b>1 INTRODUCTION</b> .....                                      | 1    |
| 1.1 BACKGROUND .....   | 1    |
| 1.2 PRESENT BALANCING METHODS .....                              | 2    |
| 1.3 PROPOSED METHOD .....  | 5    |
| 1.4 SUMMARY OF PROPOSED METHOD .....                             | 6    |
| <b>2 NEURAL NET BACKGROUND</b> .....                             | 8    |
| 2.1 INTRODUCTION .....   | 8    |
| 2.2 GENERAL DESCRIPTION .....                                    | 8    |
| 2.3 THE HISTORY OF NEURAL NETS .....                             | 12   |
| 2.4 THE FUNCTION OF NEURAL NETS .....                            | 13   |
| 2.5 TYPES OF NETS .....  | 14   |
| 2.6 PROBLEMS WITH THE USE OF NEURAL NETS .....                   | 20   |
| 2.7 ADVANTAGES OF NEURAL NETS .....                              | 23   |
| 2.8 SUITABLE TASKS FOR NEURAL NETS .....                         | 25   |
| 2.8.1 Applications in condition monitoring and diagnostics ..... | 26   |
| 2.8.2 Applications in quality control .....                      | 27   |
| 2.8.3 Applications in control and system identification .....    | 27   |
| 2.9 BACKPROPAGATION THEORY .....                                 | 28   |
| 2.9.1 Learning equations .....                                   | 29   |
| 2.9.2 Output equations .....                                     | 39   |
| 2.10 PRACTICAL COMMENTS ON THE BACKPROPAGATION EQUATIONS .....   | 40   |
| 2.10.1 Choice of activation function .....                       | 40   |
| 2.10.2 Speed of learning .....                                   | 41   |
| 2.10.3 Weight initialization .....                               | 44   |



|   |     |
|---|-----|
| <b>3 ROTOR EQUATIONS</b> .....                            | 45  |
| 3.1 EQUATIONS FOR AN UNBALANCED ROTOR .....               | 45  |
| 3.2 EXPERIMENTAL DETERMINATION OF TRANSFER FUNCTION ..... | 49  |
| 3.3 SIMILARITY TO NEURAL NET EQUATIONS .....              | 52  |
| <br>  |     |
| <b>4 EXPERIMENTAL DESCRIPTION AND RESULTS</b> .....       | 53  |
| 4.1 EXPERIMENTAL RIG .....                                | 53  |
| 4.2 NEURAL NET IMPLICATIONS AND PARAMETERS .....          | 58  |
| 4.3 NEURAL NET ARCHITECTURE .....                         | 60  |
| 4.4 RESULTS .....   | 62  |
| 4.5 NET PARAMETERS .....                                  | 66  |
| 4.5.1 Error criterion .....                               | 66  |
| 4.5.2 Training time and gradient descent parameters ..... | 67  |
| 4.5.3 Sigmoidal look-up or calculation .....              | 68  |
| 4.5.4 Features used .....                                 | 68  |
| 4.5.5 Number of hidden neurodes .....                     | 70  |
| 4.5.5.1 Auto-associative network .....                    | 71  |
| 4.5.5.2 Singular value decomposition .....                | 72  |
| 4.5.6 Number of hidden layers .....                       | 74  |
| 4.6 DATA SET CONFIGURATION .....                          | 75  |
| 4.6.1 Size of data set .....                              | 75  |
| 4.6.2 Improving generalization .....                      | 77  |
| 4.6.3 Four layer net .....                                | 78  |
| 4.7 DISCUSSION OF RESULTS .....                           | 79  |
| 4.7.1 Results .....                                       | 79  |
| 4.7.1.1 Previously seen data .....                        | 80  |
| 4.7.1.2 Similar data .....                                | 84  |
| 4.7.1.3 Data not in the training set at all .....         | 87  |
| 4.7.1.4 Location of the error .....                       | 90  |
| 4.7.1.5 Conclusions .....                                 | 91  |
| 4.8 COMPARISON WITH STANDARD BALANCING TECHNIQUES ..      | 91  |
| 4.9 DATA SET REDUCTION .....                              | 94  |
| 4.9.1 Logical reduction .....                             | 95  |
| 4.9.2 Karhunen-Loève Compression .....                    | 95  |
| 4.9.3 Singular Value Decomposition .....                  | 96  |
| 4.9.3.1 Details .....                                     | 97  |
| 4.9.3.2 Usage .....                                       | 98  |
| 4.9.4 QR Reduction .....                                  | 99  |
| 4.9.4.1 Details .....                                     | 100 |
| 4.9.4.2 Usage .....                                       | 100 |
| 4.9.5 Energy content .....                                | 104 |

|  |            |
|--|------------|
| 4.10 SIMULATION .....                                | 105        |
| 4.10.1 Short simulation .....                        | 107        |
| 4.10.2 Extended simulation .....                     | 113        |
| 4.11 COMPARISON WITH OTHER NEURAL NET SOFTWARE ..... | 120        |
| 4.12 TRAINING TIME .....                             | 121        |
| 4.13 SUMMARY OF RESULTS .....                        | 122        |
| <b>5 CONCLUSIONS AND RECOMMENDATIONS .....</b>       | <b>127</b> |
| 5.1 CONCLUSIONS .....                                | 127        |
| 5.2 RECOMMENDATIONS .....                            | 130        |
| 5.2.1 Practical applications .....                   | 130        |
| 5.2.2 Further applications .....                     | 131        |
| 5.2.3 Applications in Mechanical Engineering .....   | 131        |
| <b>REFERENCES .....</b>                              | <b>132</b> |

## LIST OF FIGURES

|   |     |
|---|-----|
| Figure 1 Simplified sketch of biological neuron . . . . .                               | 9   |
| Figure 2 Neurode . . . . .  | 10  |
| Figure 3 Three possible transfer functions . . . . .                                    | 10  |
| Figure 4 Three layer neural net . . . . .   | 18  |
| Figure 5 Sketch of shaft . . . . .  | 45  |
| Figure 6 Unbalance rig . . . . .  | 54  |
| Figure 7 Unbalance rig: accelerometer placement . . . . .                               | 54  |
| Figure 8 Horizontal response at drive end bearing . . . . .                             | 57  |
| Figure 9 Vertical response at drive end bearing . . . . .                               | 57  |
| Figure 10 Equivalence of mass vectors . . . . .   | 59  |
| Figure 11 Acceleration spectra for test F15 at non-drive end bearing . . . . .          | 62  |
| Figure 12 Acceleration spectra for test F30 at non-drive end bearing . . . . .          | 63  |
| Figure 13 Acceleration spectra for test F30 at drive end bearing . . . . .              | 63  |
| Figure 14 Spectra and points chosen for first five accelerometers, test F26 . . . . .   | 64  |
| Figure 15 Spectra and points chosen for last five accelerometers, test F28 . . . . .    | 65  |
| Figure 16 Neural net architecture . . . . .   | 69  |
| Figure 17 Prediction error for different numbers of hidden neurodes . . . . .           | 72  |
| Figure 18 Normalized singular values, linear plot . . . . .                             | 73  |
| Figure 19 Normalized singular values, logarithmic plot . . . . .                        | 74  |
| Figure 20 Variation in error due to training set size . . . . .                         | 76  |
| Figure 21 Previously seen: data in training set, test F3 . . . . .                      | 80  |
| Figure 22 Previously seen: data in training set, test F4 . . . . .                      | 81  |
| Figure 23 Previously seen: data in training set, test F13 . . . . .                     | 81  |
| Figure 24 Previously seen: data in training set, test F210 . . . . .                    | 82  |
| Figure 25 Previously seen: data in training set, test F94 . . . . .                     | 82  |
| Figure 26 Previously seen: data in training set, test F31 . . . . .                     | 83  |
| Figure 27 Previously seen: data in training set, test F242 . . . . .                    | 83  |
| Figure 28 Not previously seen but similar data in training set, test F41 . . . . .      | 84  |
| Figure 29 Not previously seen but similar data in training set, test F25 . . . . .      | 85  |
| Figure 30 Not previously seen but similar data in training set, test F100 . . . . .     | 85  |
| Figure 31 Not previously seen but similar data in training set, test F56 . . . . .      | 86  |
| Figure 32 Not previously seen but similar data in training set, test F244 . . . . .     | 86  |
| Figure 33 Not previously seen: data not in training set, test F32 . . . . .             | 87  |
| Figure 34 Not previously seen: data not in training set, test F26 . . . . .             | 88  |
| Figure 35 Not previously seen: data not in training set, test F84 . . . . .             | 88  |
| Figure 36 Not previously seen: data not in training set, test F28 . . . . .             | 89  |
| Figure 37 Not previously seen: data not in training set, test F30 . . . . .             | 89  |
| Figure 38 Not previously seen: data not in training set, test F83 . . . . .             | 90  |
| Figure 39 Comparison of linear balancing and neural net predictions, test F32 . . . . . | 92  |
| Figure 40 Comparison of linear balancing and neural net predictions, test F26 . . . . . | 93  |
| Figure 41 Prediction error for different training set configurations . . . . .          | 99  |
| Figure 42 Diagonal value of R-matrix for training data . . . . .                        | 101 |
| Figure 43 Comparison of error for full and QR reduced training sets . . . . .           | 102 |
| Figure 44 Comparison of error for full and QR reduced testing sets . . . . .            | 103 |
| Figure 45 Imbalance prediction using QR reduced set, test F83 . . . . .                 | 103 |
| Figure 46 Comparison of error for short simulated training set and actual set . . . . . | 108 |

|   |     |
|---|-----|
| Figure 47 Prediction trained using short simulated training set: data in training set, test F3 .....      | 108 |
| Figure 48 Prediction trained using short simulated training set, data in training set, test F210 .....    | 109 |
| Figure 49 Prediction trained using short simulated training set: data in training set, test F41 .....     | 109 |
| Figure 50 Prediction trained using short simulated training set: data not in training set, test F32 ..... | 110 |
| Figure 51 Prediction trained using short simulated training set: data not in training set, test F26 ..... | 110 |
| Figure 52 Prediction trained using short simulated training set: data not in training set, test F84 ..... | 111 |
| Figure 53 Prediction trained using short simulated training set: data not in training set, test F28 ..... | 111 |
| Figure 54 Prediction trained using short simulated training set: data not in training set, test F30 ..... | 112 |
| Figure 55 Prediction trained using short simulated training set: data not in training set, test F83 ..... | 112 |
| Figure 56 Singular values for the extended simulated training data .....                                  | 114 |
| Figure 57 Error distribution for large simulated training set and actual set .....                        | 115 |
| Figure 58 Error distribution for large simulated test set and actual set .....                            | 115 |
| Figure 59 Prediction trained using large simulated training set: data in training set, test F3 .....      | 116 |
| Figure 60 Prediction trained using large simulated training set: data in training set, test F210 .....    | 116 |
| Figure 61 Prediction trained using large simulated set: data in training set, test F41 .....              | 117 |
| Figure 62 Prediction trained using large simulated training set: data not in training set, test F32 ..... | 117 |
| Figure 63 Prediction trained using large simulated training set: data not in training set, test F26 ..... | 118 |
| Figure 64 Prediction trained using large simulated training set: data not in training set, test F84 ..... | 118 |
| Figure 65 Prediction trained using large simulated training set: data not in training set, test F28 ..... | 119 |
| Figure 66 Prediction trained using large simulated training set: data not in training set, test F30 ..... | 119 |
| Figure 67 Prediction trained using large simulated training set: data not in training set, test F83 ..... | 120 |
| Figure 68 Prediction error for different training set configurations including simulations .....          | 123 |

## LIST OF TABLES

|   |     |
|---|-----|
| <b>Table 1:</b> Normalized root mean square error for original data . . . . . | 91  |
| <b>Table 2:</b> Training times for various configurations . . . . .           | 122 |
| <b>Table 3:</b> Normalized root mean square error for training set . . . . .  | 125 |
| <b>Table 4:</b> Normalized root mean square error for testing set . . . . .   | 126 |

## NOMENCLATURE

Roman variables:

- A** acceleration response matrix
- $a_i$  distance from bearing 1 to the  $i$ 'th disk
- E** error
- E** pivot matrix in QR transformation
- $f$  neural net activation function
- H** transfer function matrix, with complex elements  $H_{ij}$
- $l$  distance between bearings
- M** mass matrix (forcing function), a column matrix with complex elements  $M_{i,j}$  which contain the product  $m_i$  and  $r_i$  at angle  $\theta_i$
- $m_i$  mass of effective imbalance on the  $i$ 'th disk
- N** Number of disks on the shaft
- net** net input to a layer
- o** net output from a layer
- Q** unitary orthogonal matrix from QR transformation
- R** bearing force column matrix, containing complex elements  $R_1$  and  $R_2$  which are the forces at bearing 1 and 2.
- R** upper triangular matrix from QR transformation
- R** rank of a matrix
- $r$  number of significant columns in input matrix
- $r_i$  radius of effective imbalance mass on disk  $i$
- ret** percentage retained energy
- S** singular values matrix from singular value decomposition
- t** target (desired output at final layer)
- t** reduced target vector
- U** eigenvector matrix from singular value decomposition
- u** reduced eigenvector matrix
- V** eigenvector matrix from singular value decomposition

- $v$  reduced eigenvector matrix
- $W_j$  neural net weight matrix at layer  $j$
- $w$  weight element (for a particular neurode in a layer)
- $X$  input matrix
- $x$  reduced input matrix
- $X_1$  'horizontal' force at bearing 1
- $X_2$  'horizontal' force at bearing 2
- $Y_1$  'vertical' force at bearing 1
- $Y_2$  'vertical' force at bearing 2

Greek variables:

- $\alpha$  momentum constant
- $\delta$  partial derivative of error
- $\eta$  learning constant or step size
- $\sigma$  singular value
- $\omega$  rotational speed
- $\theta_i$  angle between mass arm and reference position
- $\theta$  threshold unit in neural net architecture

Subscripts:

- $i$  disk number
- $i, j, k$  net layer index
- $p$  pattern index

resonance speeds are usually called *critical speeds* due to the tendency to large deflections when unbalanced. Shafts running above the first critical speed may be said to be running at *supercritical* speeds.

The vibration due to unbalance can cause rough running which in turn can cause poor quality manufactured goods, noise and discomfort. The vibration will reduce not only the life of the bearings in the equipment, but may lead to consequent damage of the equipment itself if the bearings fail unexpectedly.

Unbalance in rotating machinery is a very common cause of unserviceability. For example, according to Lifson *et al*<sup>[2]</sup>, unbalance is the most common cause of unserviceability in power generation. Similarly Thompson<sup>[3]</sup> states that one of the leading causes of engine removal for the LM2500 gas turbine in US Navy ships has been unbalance in the gas generator rotor.

## **1.2 PRESENT BALANCING METHODS**

Darlow<sup>[1, 4]</sup> presents a detailed study of high speed (flexible shaft) balancing. He describes three methods of balancing.

*Modal balancing* relies on the orthogonality of vibration modes of the rotor to



minimize the effects of interference between modes. Modes are balanced starting at the lowest speed mode, and then going to successively higher modes. By knowing the shape of the modes, masses may be placed to balance each mode without affecting the balance condition of other modes. This method requires a good knowledge of the structure and a skilled operator. It cannot easily be automated.

*Influence coefficient balancing* is far more widely used because it does not need much operator skill, can be easily automated, and does not need much knowledge of the structure. In this method the rotor is first brought up to speed and responses from a number of sensors (either deflection probes or accelerometers) are recorded as the baseline response. A trial mass is placed at the first balancing plane, the rotor is brought up to speed and sensor readings recorded at as many speeds as required. The mass is removed and the process is repeated for as many balance planes as required. An influence coefficient (or transfer) matrix is constructed from the sensitivity of the structure to the addition of the trial mass. The advantage of this technique is that it is based entirely on empirical observations, and that the balancing calculation can be performed in a linear least squares fashion to optimize the behavior over the required speed range. However, once the rotor and its support system begin behaving non-linearly, or the rotor has to run supercritically, this process becomes more of an art, and readings from particular sensors have to be weighted or manipulated in some fashion in order to

reduce the vibration to the required levels. It often requires repeated iterative runs to accomplish this reduction.

The third method, called *unified balancing*, was developed by Darlow and combines modal and influence balancing to obtain the best of both methods, but at the expense of complex calculations and procedures.

When a rotor needs to be balanced again, or similar rotors need to be balanced, current methods attempt to speed up the balancing process by using previously determined influence coefficients. These are usually obtained by statistically averaging sets of influence coefficients over a number of rotors. This considerably reduces the number of trial mass runs<sup>[1, 3]</sup>. Despite this, there are cases where there can be several possible solutions to the balancing calculations, which appear "equally good or bad", usually involving an iterative process (which may occasionally diverge). Thompson<sup>[3]</sup> found that certain types of unbalance were associated with certain modes, and describes building up patterns of generic influence coefficients for each sensor and balance plane. His technique had a 70% success rate for balancing on the first attempt.

Because of the difficulties of balancing at high speeds, Speckhart and Euler<sup>[5]</sup> developed an inverse method which uses deflections at different planes on a rotor to determine the correct balance masses and locations. The method requires a

discrete linear analytical (mass and stiffness) model of the rotor and its supports. Their report describes analytical cases only, and states that its usefulness is dependent both on the accuracy of the analytical model and the accuracy with which the deflections can be measured.

### **1.3 PROPOSED METHOD**

It is clear that methods of improving and speeding up the balancing process for flexible shafts would be very welcome, particularly if they are able to cope quickly with non-linearity and remove the need for an iterative procedure.

The increasing need to perform 100% inspection of manufactured goods also requires a decrease in the time it takes to balance rotating equipment.

This investigation expands on the concept of finding patterns in the balancing process as Thompson<sup>[3]</sup> did, and uses an inverse procedure as did Speckhart and Euler<sup>[5]</sup>. However, this research will attempt to use the non-parametric and non-linear capabilities of artificial neural nets to extract a pattern in the balancing of a multi-plane flexible rotor. This work will apply only to balancing of similar rotors, as required, for example, in the inspection of a production line output, and is not an attempt at a general balancing algorithm.

Artificial neural networks attempt to mimic biological nervous systems and achieve good computation performance by dense interconnection of very simple computational elements, as described later in more detail. They are often used in pattern recognition, or in mapping an input from one domain to another.

Artificial neural nets have been used to exploit patterns in medical<sup>[6]</sup> and business<sup>[7]</sup> data. Another example of the use of the pattern recognition capabilities of neural nets is signature analysis for predictive maintenance by measuring the vibrations of shafts and flywheels<sup>[8]</sup>. Comparatively few applications in mechanical engineering are available; they are generally in the fields of condition monitoring and diagnostics, quality control, and initial attempts at control systems.

#### **1.4 SUMMARY OF PROPOSED METHOD**

This research investigates whether patterns in the accelerations or forces resulting from an unbalanced multi-plane rotor running in the flexible domain can be extracted by, and trained into, an artificial neural network. This method would have the advantage of coping with non-linearities without imposing a model on the rotor system. The learned patterns could then be used by the artificial neural net to predict the correct balance masses when previously unseen unbalance accelerations or forces are provided to it, without using any trial mass balances.

