

Bangor University

DOCTOR OF PHILOSOPHY

The application of neural network techniques to magnetic and optical inverse problems

Jones, Huw Vaughan

Award date:
2000

Awarding institution:
Bangor University

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

The Application of Neural Network Techniques to Magnetic and Optical Inverse Problems

Huw Vaughan Jones

Thesis submitted to the University of Wales, Bangor
in candidature for the degree of Doctor of Philosophy.

School of Informatics, University of Wales, Bangor, Wales, U.K.

December 2000



Supervisors: R.W. Chantrell and A.W.G. Duller



To my family, for their continued support.

Summary

The processing power of the computer has increased at extraordinary rates over the last few decades. However, even today's fastest computer can take several hours to find solutions to some complex mathematical problems, and there are instances when a high powered supercomputer may be impractical, with the need for near instant solutions just as important. One approach to this kind of problem is through machine learning. This research investigates the application of various neural network techniques, and aims to solve known complex inverse problems in the field of magnetic and optical recording.

Investigations were conducted to determine anisotropy distributions from transverse susceptibility data (an area of magnetism which is proving to be very important, as it gives information about the overwrite characteristics of the media). The neural networks produced results which were very impressive, and compared well with other more conventional methods (solutions were obtained in a fraction of the time). More importantly, it suggested that this technique was indeed feasible, and could be used to solve a series of similar inverse problems.

The neural network was then modified to investigate diffraction patterns from a compact disc. This was an interesting application which had practical uses in industry, as the idea of reducing the testing time for each disc was very attractive. Simulations on theoretical data were successful, and suggested that this method could be carried out experimentally on an online testing system.

The final part of the research involved the extraction of important features from magnetisation maps and magnetic force microscopy images. The use of neural networks to study general image analysis is well established, but its use in magnetism was quite novel. The results obtained were surprisingly good as the images investigated hardly contained enough data for the human eye to observe, and features such as percolation and vortices were observed on a variety of samples. This success has prompted suggestions as to ways in which this approach can be expanded to solve similar problems.

Acknowledgements

I would like begin expressing thanks to my supervisors, Roy Chantrell and Andy Duller for taking me on as a PhD student, and for their continued guidance throughout the course of the work. I would also like to thank Adam Hoare for being my external supervisor and giving me an insight into the world of industry.

I would like to extend a special thank you to the School of Informatics for their continued support during my period here, and for letting me do both my undergraduate and postgraduate study in a very friendly environment.

My thanks also go to the Engineering and Physical Sciences Research Council for my studentship grant and to Aerosonic Ltd. for my case award, without which I couldn't afford all that beer for three years.

I would like to express my gratitude to Roy for giving me the opportunity to attend several conferences, both in the UK and abroad (and to Jackie for organising the travel arrangements). They were very enjoyable, and I will always look back at them.

Many thanks must also go to Kevin O'Grady and the rest of the experimental magnetism group for their help. I very much enjoyed all the seminars and conferences we went to over the last three years. I am very grateful to Gareth Roberts too, for helping me through the final stages of my study, and for sending me to Gregynog every year.

I would also like to thank the lads at the computational magnetism group (both past and present) for dragging me out for a pint and thus helping me spend my grant. On the work side, I would like to extend a special thanks to Nick Walmsley and Simon Clarke letting me use their data and models.

Finally, I would like to thank my family, especially my parents, who have helped me both financially and morally throughout my time at University. I bet they'll be happy now that I've finished though!

Contents

1 Introduction To Magnetism and Magnetic Recording

1.1 Introduction	1
1.2 The Basics of Magnetism	2
1.2.1 Magnetic Fields	2
1.2.2 Magnetic Moments	3
1.2.3 Magnetic Susceptibility	3
1.2.4 Magnetic Domains	4
1.3 Anisotropy	6
1.3.1 Crystal Anisotropy	6
1.3.2 Shape Anisotropy	7
1.3.3 Exchange Anisotropy	7
1.4 Magnetic Materials	8
1.4.1 Particulate Media	8
1.4.2 Thin Film Media	10
1.5 Stoner-Wohlfarth Theory	11
1.6 The Magnetic Recording Process	12
1.6.1 AC and CD Erasure	13
1.6.2 Recording Media	14
1.7 Magnetic Force Microscopy	16
1.7.1 Scanning Probe Microscope	17
1.7.2 Magnetic Force Microscope	18
1.7.2.1 Applications in Data Storage Methods	19

2 Introduction To Neural Networks

2.1 Introduction	21
2.2 The Model of a Neuron	22
2.3 Network Architectures	23
2.3.1 Single-layer feedforward network	24
2.3.2 Multi-layer feedforward network	24
2.3.3 Recurrent networks	25

2.3.4 Lattice structures	26
2.4 Learning Processes	27
2.4.1 Supervised Learning	29
2.4.1.1 Backpropagation Learning Algorithm.	30
2.4.1.2 Conjugate-Gradient Method	32
2.4.2 Unsupervised Learning	34
2.4.2.1 Kohonen Self-Organising Network	35
2.4.3 Pattern Recognition	37
2.5 Knowledge Representation	38
2.6 The Advantages of Neural Networks	39
3 Anisotropy Distributions in Transverse Susceptibility Data	
3.1 Introduction	41
3.2 Theoretical Determination of the Transverse Susceptibility	43
3.3 Experimental Determination of the Transverse Susceptibility	47
3.4 The Inverse Problem	49
3.5 The Maximum Entropy Approach	50
3.5.1 Introduction	50
3.5.2 Practical Considerations	51
3.5.3 Simulated Annealing.	52
3.5.4 The Deconvolution of H_K Distributions	52
3.6 Application of Neural Networks	53
3.6.1 Introduction	53
3.6.2 Multi-layer Perceptron.	54
3.7 Data Preparation and Network Optimisation	56
3.7.1 Introduction	56
3.7.2 Data Preparation	56
3.7.3 Training Set Size	57
3.7.4 Input Data Points	60
3.7.5 Stopping Criteria and Choice of Learning Algorithm.	61
3.7.6 Optimal Network Size and Parameters.	63

3.8 Results	67
3.8.1 Using Theoretical Data	67
3.8.2 Using Experimental Data	70
3.9 Summary	73
4 Analysis of Compact Disc Pit Geometries Using Optical Diffraction Methods	
4.1 Introduction	74
4.2 Diffraction Theory	78
4.2.1 Theoretical Measurement Techniques	80
4.3 Experimental Measurement Techniques	86
4.3.1 Three Detector Arrangement	86
4.3.2 Multi-element Array Arrangement	87
4.3.3 CCD Camera Arrangement	87
4.4 Application of Neural Networks	88
4.5 Data Preparation and Network Optimisation	91
4.5.1 Introduction	91
4.5.2 Data Preparation	91
4.5.3 Input Data Points	92
4.5.4 Training Set Size	93
4.6 Results	94
4.6.1 Using Theoretical Data	94
4.6.2 Using Simulated Experimental Data	96
4.7 Summary	99
5 Analysis of Magnetisation Maps and MFM Images of Thin Film Media	
5.1 Introduction	100
5.2 Thin Film Media Model	101
5.2.1 Magnetisation Maps	103
5.2.2 Samples Used	103
5.3 Application of Neural Networks	107
5.3.1 Unsupervised Learning	107
5.3.2 Self-Organising Maps	107

5.3.2.1 The SOM Algorithm	108
5.3.2.2 Visualising the SOM Using U-Matrix Diagrams	111
5.3.3 Image Coding Techniques	112
5.4 Feature Extraction From Magnetisation Maps	114
5.4.1 Pre-processing Techniques	114
5.4.1.1 Moment Analysis	115
5.4.1.2 Divergence	116
5.4.1.3 Curl	117
5.4.2 Data Preparation	118
5.4.3 Network Optimisation	120
5.4.4 Results	123
5.5 The MFM Model	133
5.5.1 MFM Sample Images.	136
5.6 Investigation of Optimal MFM Scanning Height.	137
5.6.1 Network Optimisation	137
5.6.2 Results	138
5.7 Feature Extraction From MFM Images	140
5.7.1 Method 1.	140
5.7.2 Method 2.	141
5.7.2.1 Data Preparation	141
5.7.2.2 Network Optimisation.	143
5.7.2.3 Results	145
5.8 Summary	152
6 Conclusions and Further Work	
6.1 Conclusions	153
6.2 Further Work	158
References	159
Appendices	
Appendix A: Published Papers	164

List of Figures

1.1	The magnetic field of a magnet	2
1.2	A typical hysteresis loop	4
1.3	Domain walls within a sample	5
1.4	Particulate recording media	8
1.5	Simulated particulate media (a) non-aligned (b) aligned	9
1.6	Cross-section through a typical hard disk	10
1.7	Hysteresis loops with applied field (a) perpendicular (b) parallel to particle	11
1.8	The recording process	12
1.9	Typical minor hysteresis loops with DC erasure	13
1.10	Typical hysteresis loop with AC erasure	14
1.11	Modes of recording (a) longitudinal (b) transverse (c) perpendicular	15
1.12	A close-up of a SPM tip	17
1.13	Schematic of a scanning probe microscope	17
1.14	Lift-Mode technique	19
1.15	Tracks on a hard disk (a) topography (b) magnetic force image	19
2.1	Model of a neuron	22
2.2	Single-layer feedforward network	24
2.3	Multi-layer feedforward network	25
2.4	Recurrent network with hidden layers	25
2.5	Two-dimensional lattice network	26
2.6	The learning process	27
2.7	Supervised learning	29
2.8	Unsupervised learning	34
2.9	Kohonen model	35
2.10	Neighbourhood decay function	37
2.11	Adaptive pattern classifier using LMS algorithm	37
3.1	The consequences of varying anisotropy on a magnetic material	42
3.2	Variation of χ_t with different degrees of texture	45
3.3	Effect of integrating the χ_t curve with a distribution function	47
3.4	Experimental apparatus to measure the transverse susceptibility of a sample	47

3.5	The inverse problem (applied to the transverse susceptibility problem).	50
3.6	Fits to log-normal distribution using simulated annealing	53
3.7	A basic multi-layer perceptron network	54
3.8	Two basic signal flows in a multi-layer perceptron.	55
3.9	Normalisation of training curve	57
3.10	Comparison of the training time required with various data set sizes.	59
3.11	Comparison of the distribution function errors with various data set sizes	59
3.12	Comparison of the distribution function errors with various sample points	60
3.13	Illustration of overtraining a network	62
3.14	Mean-squared error for various η and α using back-propagation learning	64
3.15	Mean-squared error for various η and α using conjugate-gradient learning	65
3.16	Comparison of the distribution function errors with 3 test sets	66
3.17	The determination of the value and type of distribution using 1 network	67
3.18	Errors in the distribution function and classification accuracy	68
3.19	The determination of the value and type of distribution using 2 networks	68
3.20	Errors in the distribution function and classification accuracy	69
3.21	Plot of σ values for experimental test runs	70
3.22	Plot of σ values and distribution type for experimental test runs	71
3.23	Comparison of experimental curve with the neural network prediction	72
4.1	Physical specifications and construction of a compact disc	76
4.2	The pit and lands on a compact disc	77
4.3	Compact disc pit surface from an SEM image	77
4.4	The Airy pattern obtained when reading from a compact disc	79
4.5	The CD pit surface acting as a diffraction grating	80
4.6	The relationship between the pit width normalised by the track pitch and the diffracted light intensity ratio (I_{2n}/I_n)	83
4.7	The intensity distribution at various diffraction angles	85
4.8	Three detector arrangement	86
4.9	Multi-element array arrangement	87
4.10	CCD camera arrangement	88
4.11	(a) Conventional testing method.	89

4.11 (b) Neural network testing method	90
4.12 Theoretical diffraction pattern from a CCD camera device	92
4.13 Average errors for P, W and D with various data set sizes	93
4.14 The determination of P, W and D using 1 network	94
4.15 Errors in the values of P, W and D for various network configurations	95
4.16 The determination of P, W and D using 3 networks	95
4.17 Errors in the values of P, W and D for various network configurations.	96
4.18 The application of random noise to simulate experimental data	97
4.19 Errors in determining P, W and D for sample with 5% random noise	98
4.20 Errors in determining P, W and D for sample with 25% random noise	98
5.1 Parameters for the Stoner-Wohlfarth particle.	101
5.2 Magnetisation map orientation colour scale	103
5.3 Magnetisation map of a system with 4 written bits	105
5.4 Magnetisation map with $h_i=0.1$ and $C^*=0.1$	105
5.5 AC erased state with $h_i=0.5$ and $C^*=0.1$	106
5.6 DC erased state with $h_i=0.5$ and $C^*=0.1$	106
5.7 Various SOM arrangements	108
5.8 Neighbourhood function	110
5.9 Relationship between the map and the weight of the best-matching neuron	110
5.10 U-matrix representation of the self-organising map	111
5.11 Original pixelised image	112
5.12 The masks which were learned by the network	113
5.13 Coding of image using learned masks	113
5.14 Moment analysis of a sample with 4 written bits	115
5.15 Divergence of a sample with 4 written bits	117
5.16 Curl of a sample with 4 written bits	117
5.17 Arrangement of grains into an orderly grid	118
5.18 Sampling a magnetisation map with $2 \times 2, 4 \times 4$ and 8×8 grids	119
5.19 The features present in the 8 groups in the SOM (using 2×2 grids)	119
5.20 Replotting the SOM groups in their original positions	120
5.21 Effect of varying the number of classification groups	121

5.22	The error values w.r.t training and the outputs obtained	122
5.23	Neural network results with various grid sizes	124
5.24	Comparison of results (a) neural network (b) moment analysis	125
5.25	Confirmation of results with original data	125
5.26	Neural network results with various grid sizes	126
5.27	Comparison of results (a) neural network (b) moment analysis	126
5.28	Confirmation of results with original data	127
5.29	Neural network results with various grid sizes	127
5.30	Comparison of results (a) neural network (b) moment analysis	128
5.31	Confirmation of results with original data	128
5.32	Neural network results with various grid sizes	129
5.33	Comparison of results (a) neural network (b) moment analysis	130
5.34	Confirmation of results with original data.	130
5.35	Neural network results with various grid sizes	131
5.36	Comparison of results (a) neural network (b) moment analysis	131
5.37	Comparison of results with original data set	132
5.38	(a) The stray field above a medium (b) The force exerted on the tip	134
5.39	Interactions between cubes in the tip with cubes in the sample	135
5.40	Effect of applying an MFM simulation to a sample with 4 written bits	136
5.41	MFM images produced at different scanning heights	138
5.42	The accuracy obtained at various scanning heights	139
5.43	MFM image of a vortex structure	141
5.44	Arrangement of MFM pixels into an orderly grid.	142
5.45	Sampling an MFM image with 2×2, 8×8 and 16×16 grids	142
5.46	The features present in the 8 groups in the SOM (using 2×2 grids)	143
5.47	Replotting the SOM groups in their original positions	143
5.48	The error values w.r.t training and the outputs obtained	144
5.49	Neural network results with various grid sizes	146
5.50	Comparison of results (a) neural network (b) original data	146
5.51	Neural network results with various grid sizes	147
5.52	Comparison of results (a) neural network (b) original data	148

5.53 Neural network results with various grid sizes 148
5.54 Comparison of results (a) neural network (b) original data 149
5.55 Neural network results with various grid sizes 149
5.56 Comparison of results (a) neural network (b) original data 150
5.57 Neural network results with various grid sizes 150
5.58 Comparison of results (a) neural network (b) original data 151

List of Tables

3.1 Variable parameters for use with the multi-layer perceptron 64
3.2 Variable parameters for use with the multi-layer perceptron 66
3.3 Summary of the network parameters 69
4.1 The compact disc specification (Red Book) 75
4.2 Summary of the network parameters 96
4.3 Average errors observed for the unknown parameters 98
5.1 Optimal neural network parameters 123
5.2 Optimal neural network parameters 145

Chapter 1

Introduction To Magnetism and Magnetic Recording

1.1 Introduction

The principles of basic magnetism can be traced back before the birth of Christ to a mineral called magnetite, which was found to attract iron and also magnetise a piece of iron which was rubbed against it. This later found its use as a sailor's compass, which was made from a shaped piece and allowed to turn on a pivot. However, it wasn't until centuries later that the first scientific study of magnetism was carried out, by William Gilbert (1540-1603). He experimented with lodestones and iron magnets, and explained a clear picture for the earth's magnetic field. This was to be the last significant discovery for more than a century and a half, until 1825, when the first electromagnet was made by Hans Christian Oersted.

In 1898, the first evidence of magnetic recording appeared with the invention of the telegraphone, by Valdemar Poulsen. Nowadays, magnetic recording plays a vital role in modern society and its success is partly due to its versatility, and its applicability in many applications, such as audio and video recording and the storage of digital information on a computer hard disk.

1.2 The Basics of Magnetism

1.2.1 Magnetic Fields

A material which has been magnetised will have a force associated with it which appears at the poles. The poles always come in pairs and are located near the ends of the magnet. Around each pole there is a magnetic field, and calculations show that this field is proportional to the product of the pole strength and the field strength.

Therefore, the field created by a unit pole will have an intensity of 1Oe at a distance of 1cm from the pole, and can be calculated using:

$$H = \frac{p}{d^2} \quad (1.1)$$

where p is the pole strength and d is the distance from the pole. The directions of the fields arising from the magnetic material were first introduced by Michael Faraday, and they show the fields leaving one pole (north) towards the other (south). This is illustrated in Figure 1.1.

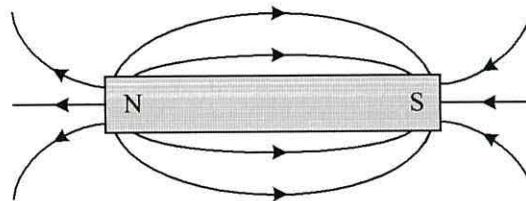


Figure 1.1: *The magnetic field of a magnet*

1.2.2 Magnetic Moments

The magnetic moment is a very important quantity in all types of magnets, and when the magnet is placed at an angle θ to a uniform field H a couple will act on it which causes the magnet to rotate parallel to the field. This moment can be defined using equation 1.2 (when the field is 1Oe and θ is 90°):

$$m = pl \quad (1.2)$$

where p is the strength of the poles at each end of the magnet and l is the separation.

1.2.3 Magnetic Susceptibility

One of the most important characteristics of a magnetic material is its susceptibility. This is defined as the ratio of the magnetism M ($M = \frac{m}{v}$, where v is the volume) to the magnetic field strength H :

$$\chi = \frac{M}{H} \quad (1.3)$$

A magnetisation curve is a plot of M against H , and the susceptibility can be measured directly from it. Materials which have a linear M - H curve include diamagnets, paramagnets and antiferromagnets, and these retain no magnetism when the magnetic field is removed. Nonlinear M - H curves arise from ferromagnets and ferrimagnets, and the susceptibility varies with applied field, giving rise to a hysteresis loop (Figure 1.2).

The loop provides us with information about three important parameters: the remanent magnetisation M_r , the coercivity H_c and the squareness M_r/M_s .

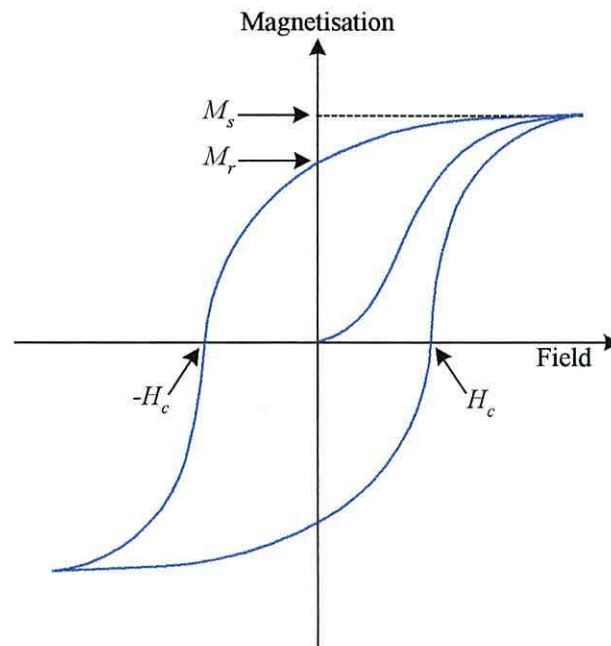
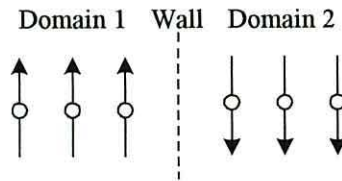


Figure 1.2: A typical hysteresis loop

The value of M_r gives information about the amount of magnetisation left in the media when the applied field is reduced to zero. It is dependent on the direction and magnitude of the previously applied field. The coercivity H_c is the field strength when the magnetisation is zero. The ratio of M_r/M_s (squareness) gives the degree to which the material retains its magnetisation in the presence of no applied field.

1.2.4 Magnetic Domains

A domain is a region throughout which the magnetisation is constant, and a domain wall is an interface between regions in which the spontaneous magnetisation has different directions. The magnetisation must at some point change direction, be it within or at the wall. For a sample with a single domain, the energy due to the demagnetising field is proportional to the volume of the specimen. Thus, as the volume of the sample becomes large, it becomes more favourable (in terms of energy) for domain walls to form (Figure 1.3).

Figure 1.3: *Domain walls within a sample*

The figure shows a rather oversimplified representation of the domain walls. In reality there is a change of magnetisation within the wall itself. The domain wall in the figure would have a large exchange energy associated with it (because the spins adjacent to the wall are anti-parallel). The exchange energy for a pair of atoms with the same spin can be calculated by:

$$E_{ex} = -2JS^2 \cos\theta \quad (1.4)$$

where J is the exchange integral and θ is the angle between the spins. This energy can decrease if the 180° change in spin direction takes place gradually over a number of atoms. However, the spins in the wall are now pointing towards noneasy directions. The crystal anisotropy energy within the wall is now higher than the adjoining domains. Whilst the exchange energy tries to make the wall as wide as possible, the anisotropy energy tries to make the wall thin (in order to reduce the number of spins pointing in noneasy directions). The thickness of the wall can be approximated by equation (1.5), and its derivation can be found in *Cullity* [1]:

$$\delta \approx \sqrt{T_c/K} \quad (1.5)$$

where δ is the wall thickness, T_c is the Curie temperature and K is the anisotropy constant. The smaller the anisotropy constant, the thicker the wall; therefore the wall thickness increases with temperature (as K decreases with temperature).

1.3 Anisotropy

Anisotropy is found in almost all magnetic materials, and it means that the magnetic properties depend on the direction in which they are measured, and it is of great practical interest in the choice of a magnetic material [2]. It is therefore necessary to have some understanding of the theory of anisotropy. There are several types of anisotropy, namely: crystal, shape, stress and exchange. There are other sources of anisotropy (such as those due to surface effects), but of all the anisotropies, crystalline anisotropy is the only intrinsic property of the material, as the others are extrinsic properties which are dependent on the preparation of the sample.

1.3.1 Crystal Anisotropy

Crystal anisotropy arises mainly due to spin-orbit coupling, which then gives rise to a preferred direction for the magnetic moment. This coupling is very strong, and keeps neighbouring spins parallel, or antiparallel to one another. However, the exchange energy associated with it is isotropic, and depends on the angle between adjacent spins (and not on the direction of the spin axis relative to the crystal lattice). In a system with uniaxial anisotropy the magnetocrystalline energy can be calculated using equation (1.6):

$$E_K = K_1 \sin^2 \phi + K_2 \sin^4 \phi \quad (1.6)$$

where K_i are the anisotropy constants (which are dependant on the material) and ϕ is the rotation of the moment away from the easy axis. The strength of the anisotropy in a crystal is measured by the magnitude of these anisotropy constants, which can be measured using torque curves (a torque magnetometer). These determine the torque which is required to rotate the magnetisation away from an easy direction as a function of the angle of rotation.

1.3.2 Shape Anisotropy

A polycrystalline sample which has no preferred orientation of its grains has no crystal anisotropy. It has a spherical shape and the applied field will magnetise it equally in any direction. However, if it is non-spherical then it will be easier to magnetise it along a long axis (rather than the short axis). The applied field along the short axis must be stronger in order to produce the same field inside the sample. Therefore, the shape of the sample must be a source of anisotropy - hence shape anisotropy.

The shape anisotropy is due to the shape dependence of the demagnetising field on the magnetic sample (which is due to the formation of uncompensated poles on its surface). The demagnetising field is proportional to and opposes the magnetisation that it creates, and is given by equation (1.7):

$$H_d = -N_d M \quad (1.7)$$

where N_d is the demagnetising factor (and is dependent on the shape of the sample); and the magnetostatic energy which is associated with the demagnetising field at saturation is given by:

$$E_{ms} = \frac{1}{2} N_d M^2 \quad (1.8)$$

1.3.3 Exchange Anisotropy

Exchange anisotropy was an effect discovered by Meiklejohn and Bean [3, 4], and is to be found in materials which consist of ferromagnetic and antiferromagnetic layers in contact; and is the result of exchange coupling between the spins at the interface of the two layers.

The energy of a ferromagnet due to exchange anisotropy is given by equation (1.9):

$$E = -K \cos(\theta) \quad (1.9)$$

where θ is the rotation of the magnetisation of the ferromagnet away from the easy direction and K is the anisotropy constant. The main requirements for a material to exhibit exchange anisotropy are (in an ideal situation, where crystal imperfections and domain wall formations are ignored):

- The sample should be field-cooled through the Néel temperature (the temperature at which a sample changes from paramagnetic to antiferromagnetic).
- There should be intimate contact between the ferromagnetic and antiferromagnetic materials.
- There should be strong crystalline anisotropy in the antiferromagnet.

1.4 Magnetic Materials

1.4.1 Particulate Media

Particulate materials are a very important type of magnetic recording media, and is used mainly in recording tape. The ideal particulate media has isolated long ellipsoidal particles suspended longitudinally in a matrix (as shown in Figure 1.4).

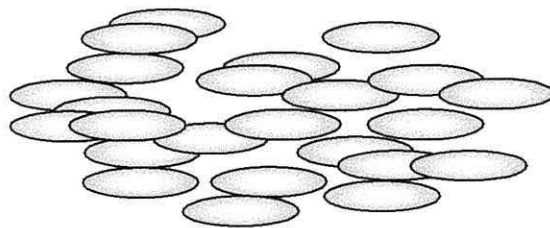


Figure 1.4: *Particulate recording media*

However, a real magnetic material is never as perfect as this in reality. Figure 1.5 shows an image of a simulated particulate medium [5] with the non-aligned particles (a) and the aligned particles (b).

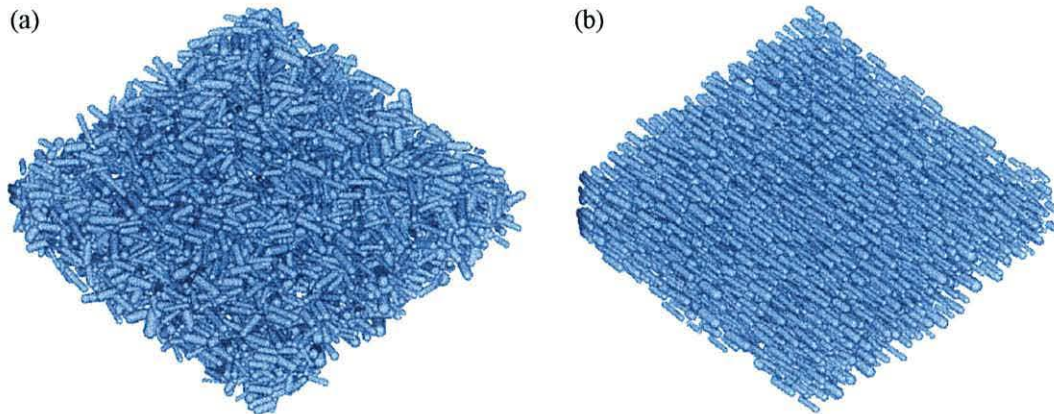


Figure 1.5: *Simulated particulate media (a) non-aligned (b) aligned*

As particulate magnetic media are most commonly used in audio tape applications, it is important to ensure recorder and reader compatibility; and the AC bias, signal current and frequency equalisation must be standardised. Improvements in performance include such things as particle alignment and morphology.

The most common magnetic material used for particulate media is an oxide of iron called gamma ferric oxide (Fe_2O_3). However, modern high coercivity particulate media are formed of metal rather than oxide particles (Fe particles with a small amount of Co to help against oxidation). Although elemental metals or alloys can be used, it is more difficult to control the morphology of metals than oxides. Elemental metals and alloys are better suited for film deposition.

1.4.2 Thin Film Media

Thin film media usually consist of cobalt based alloys, which often possess a hexagonal crystal structure in which magnetocrystalline anisotropy aids in achieving a high coercivity. Hence, thin film media have been thoroughly studied for their application in hard disks.

A hard disk consists of several layers of varying thickness. The substrate is usually an aluminium alloy with some sort of overcoat to increase surface hardness, reduce corrosion, and improve the adhesion of the metal film. The undercoat is followed by a thin coat of the magnetic material. This coat is followed by a protective overcoat. In order to minimise the transition width, and hence the minimum separation required between two successive tracks, the magnetic layer is made as thin as possible. Figure 1.6 shows a cross-section through a typical hard disk.

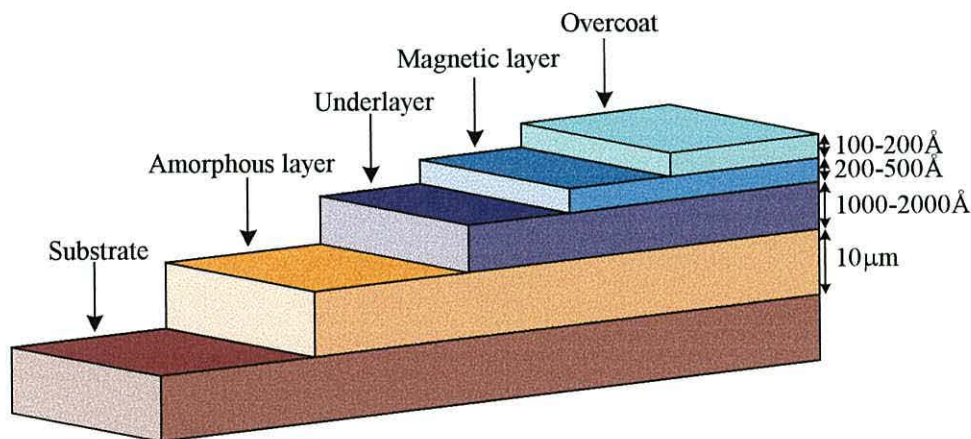


Figure 1.6: Cross-section through a typical hard disk

The magnetic layer and the underlayer are the two most important layers in the media. The data is recorded onto the magnetic layer (which is usually a combination of cobalt and nickel), and has a typical thickness of about 200-500Å. The underlayer (usually chromium) has a typical thickness of about 1000-2000Å, and is the layer onto which the magnetic layer is sputtered. The properties of the magnetic layer are dependent on the thickness of the underlayer; and as its thickness increases, the coercivity increases and the remanence decreases.

1.5 Stoner-Wohlfarth Theory

In 1955 a single-domain particle model was proposed by Stoner and Wohlfarth [6]. The particle is assumed to be ellipsoidal in shape, and the shape anisotropy is the determining force in the orientation of the particle's magnetisation, overriding exchange and crystalline anisotropies. It is possible to represent the angle of orientation of the magnetisation with the uniaxial anisotropy easy direction by θ , and the anisotropy energy by:

$$E_{an} = -K \sin^2 \theta \quad (1.10)$$

where K is the anisotropy constant. The two cases that arise are when the particle is either perpendicular or parallel to the field.

- If the particle is perpendicular to the field, then it will be rotated away from the easy axis and into the direction of the field (then this gives rise to completely reversible changes in the magnetisation). This is illustrated in Figure 1.7(a).
- If the particle is parallel to the field, then the magnetisation will remain in its original direction. When the field is strong enough in the opposite direction, the magnetisation will line up parallel, and can be made to follow the direction of the applied field (Figure 1.7(b)).

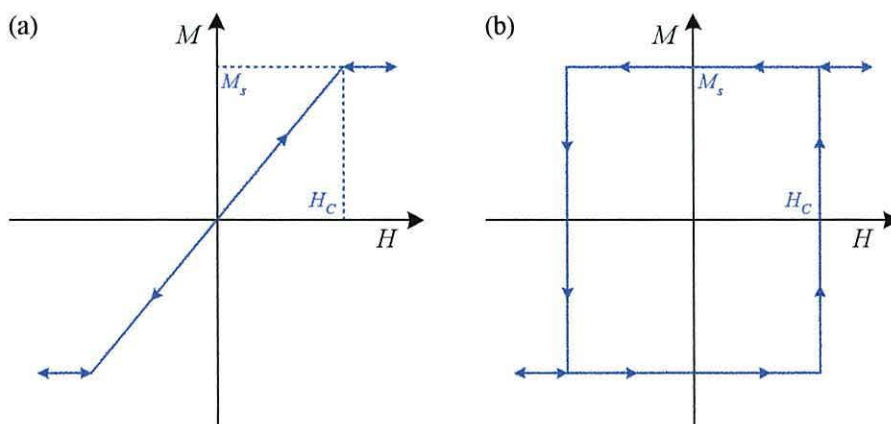


Figure 1.7: Hysteresis loops with applied field (a) perpendicular (b) parallel to particle

1.6 The Magnetic Recording Process

The recording process is the method in which information is stored on a magnetic medium (this is illustrated in Figure 1.8). Information is written by applying a current through the coil windings, which has the effect of inducing a magnetic fringing field across the gap. The field direction depends on the direction of the current flowing through the windings, and thus determines the direction of the recorded information. The recorded information will be retained in the medium after the medium has passed the head gap.

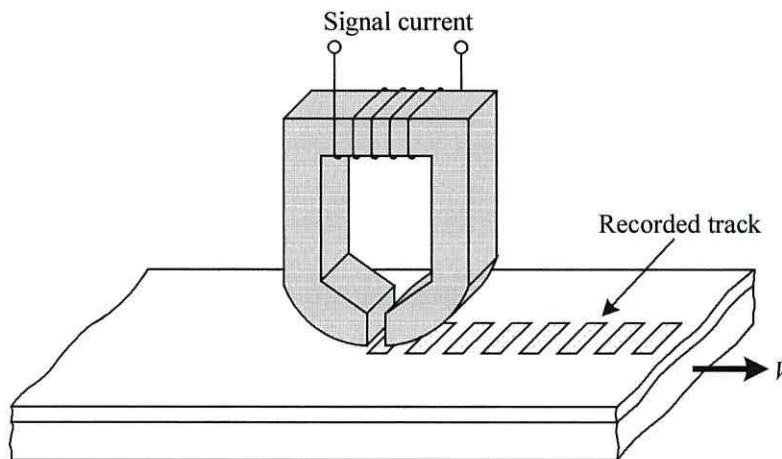


Figure 1.8: *The recording process*

Information can also be read from the recorded media in a similar manner, by basically doing the reverse of the writing process. As the medium passes beneath the head gap, the stray field from the magnetisation in the medium induces a field in the head. This then induces a current in the coil windings, which is amplified and converted back into its original form. The voltage can be determined by making use of the reciprocity theorem, which states that the magnitude of the induced flux in the head by the recording medium passing by the head is proportional to the magnetisation of the medium and is also proportional to the field which would be induced in the head if it was excited by a unit current.

1.6.1 AC and DC Erasure

The erasure of a magnetic sample is of great importance, as it leads to further areas of study, such as overwrite. There are several methods available to demagnetise a sample; the two important ones being AC and DC erasure.

In the DC erasure method a positive saturating field H_{sat} is applied, before the application of a negative field H_1 . The field is then reduced to zero and the remanent magnetisation measured. The sample is then subjected to H_{sat} again before the application of the next negative field H_2 . This sequence is repeated until M_{irr} has completely reversed. The DC demagnetised state occurs when the remanent magnetisation is zero. Figure 1.9 shows a M-H plot for a sample with DC erasure:

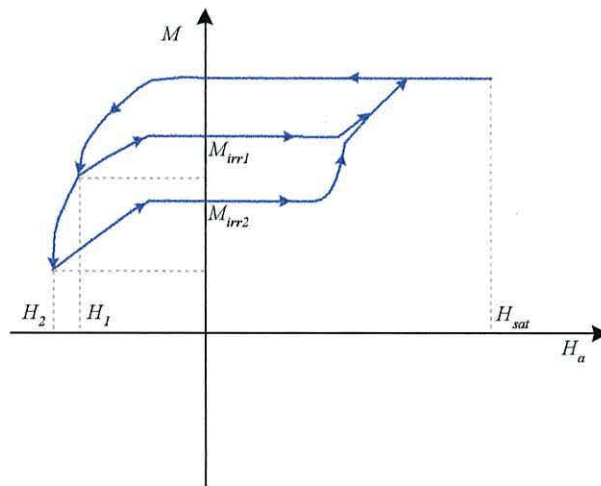


Figure 1.9: *Typical minor hysteresis loops with DC erasure*

In the AC erasure method the magnetic sample is placed in the vicinity of a large, alternating field. The magnitude of this field is much greater than the coercivity of the sample, thus causing the moments to align with the field direction. The alternating field is then reversed and the moments switch into the new field direction. This process is repeated, causing the sample to be rapidly taken around the hysteresis loop. The field is then slowly decreased or the sample is slowly removed from the influence of the field (both have the same effect and both are subjected to an alternating field of

decreasing amplitude). This has the effect of causing the sample to trace (decreasing) minor hysteresis loops, until the origin is reached and the sample has zero magnetisation in zero field (as illustrated in Figure 1.10).

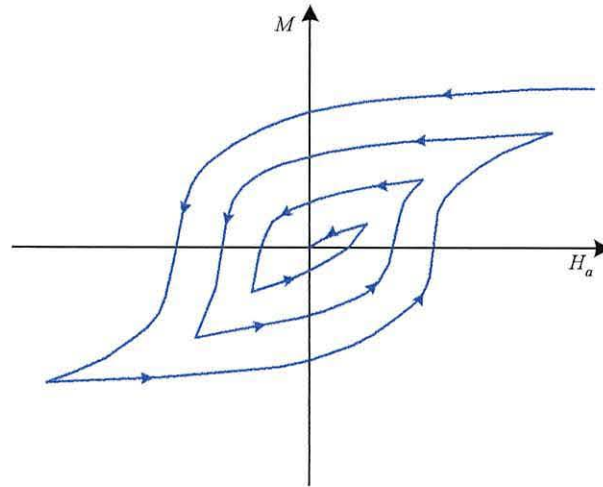


Figure 1.10: *Typical hysteresis loop with AC erasure*

1.6.2 Recording Media

Information is stored in a medium in the form of written bits, which are denoted by regions of constant magnetisation within the medium. The direction of recorded magnetisation is strongly influenced by the magnetic anisotropy in the medium. This gives rise to the terms *easy* and *hard* directions of magnetisation, and they relate to the energy which is required to induce a magnetisation along a specific axis. There are three modes of operation which can be used in thin film media: longitudinal, transverse and perpendicular recording (as illustrated in Figure 1.11).

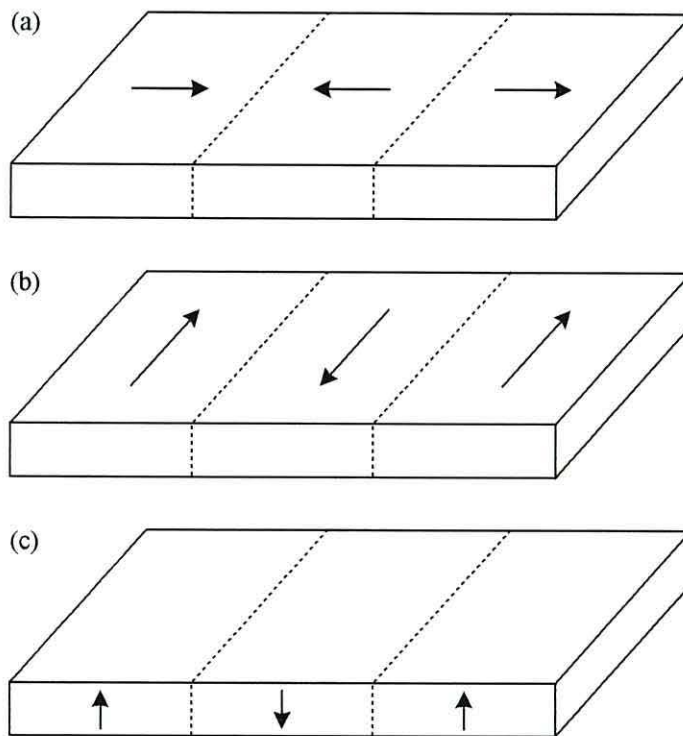


Figure 1.11: *Modes of recording (a) longitudinal (b) transverse (c) perpendicular*

Longitudinal recording media has the magnetisation of each particle parallel to the plane of the medium and the recorded tracks are parallel to the motion of the medium. In perpendicular recording the magnetisation of each particle is perpendicular to the plane of the media, which has the possibility of obtaining much higher recording densities. The disadvantage is that a very small readback voltage is produced in the reading head. Transverse recording has the recorded information written perpendicular to the direction of motion of the media (and parallel to the plane of the medium). The preferred current mode of recording is the longitudinal method, as the remanent magnetisation is much higher along the longitudinal direction.

The requirements for a recording medium can be investigated by considering the recording process itself. If a magnetic material were to be placed in the presence of an externally applied magnetic field, then when the strength of the field is at such a level that the moments of the magnetic particles are aligned with the field, then the sample is said to be at saturation magnetisation. Initially, as the field is reduced, the moments will undergo a reversible change. But if a negative field were applied (above a critical

level) the moments would undergo irreversible changes. These changes are illustrated by plotting the hysteresis loop for the sample.

The ideal magnetic medium should be a thin film of small thickness giving rise to a square hysteresis loop (with a high remanence M_r and high coercivity H_c). Materials with a high coercivity and high M_s are generally used in recording media (compounds are typically cobalt based).

1.7 Magnetic Force Microscopy

Magnetic force microscopy is a development of the non-contact mode of Atomic Force Microscopy [7], and is a technique used for studying micromagnetic features in magnetic media, which become increasingly difficult to measure as the bit densities of storage media continue to increase. Storage densities in the range of gigabit/square inch have produced magnetic features smaller than the wavelength of light. Extremely high-resolution magnetic imaging is becoming a requirement for the design and manufacture of most data storage devices. A typical example is for measuring the head performance on hard disks.

Magnetic force microscopy (MFM) utilises the power of the scanning probe microscope [8, 9], by scanning a tiny ferromagnetic probe over a sample. The MFM then maps the stray magnetic fields close to the sample surface. It boasts many capabilities that complement existing imaging methods; its resolution is consistently better than 50nm (which is superior to optical techniques, such as Kerr microscopy). The sensitivity is also sufficient to image individual submicron particles. Fields can be imaged through the nonmagnetic and opaque overcoats often applied to hard disks. The imaging is usually done under ambient conditions, and it requires little or no sample preparation, with results obtained in a few minutes.

1.7.1 Scanning Probe Microscope

In order to understand the principles of the MFM, it is best to look at the scanning probe microscope (SPM), from which the MFM technique is derived [10]. An SPM probe consists of a sharp tip (Figure 1.12) mounted on a weak cantilever spring.

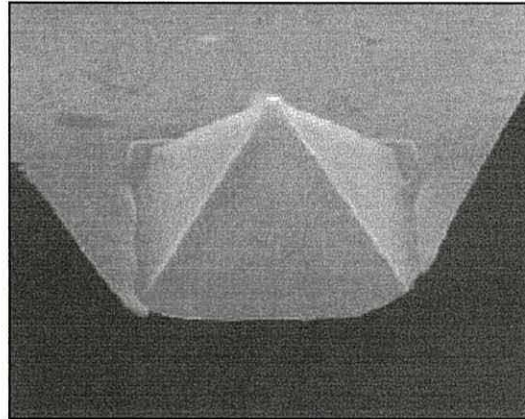


Figure 1.12: A close-up of a SPM tip

The tip is brought close to the sample and a piezoelectric scanner moves the probe in a raster pattern. The interactions between the tip and sample deflect the cantilever; and the deflection is then monitored by reflecting a laser beam off the cantilever into a segmented photodiode (as illustrated in Figure 1.13). The image is formed by mapping this laser-detected deflection during the scanning process.

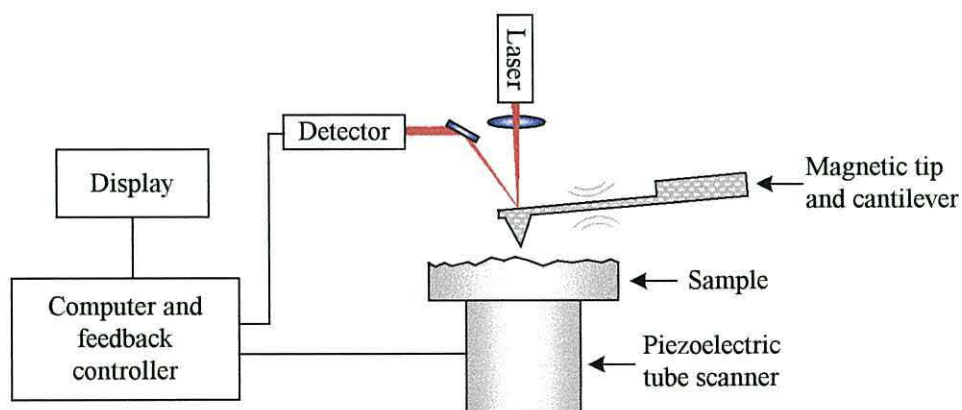


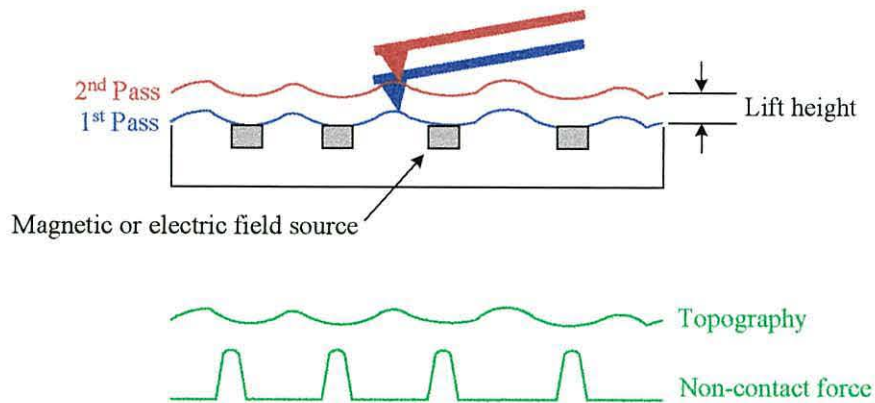
Figure 1.13: Schematic of a scanning probe microscope

The SPM brings the tip into direct contact with the sample, and the vertical position of the sample is adjusted using feedback to keep the cantilever deflection at a constant value while scanning. The resulting vertical offset is displayed as a three-dimensional image of the surface topography.

1.7.2 Magnetic Force Microscope

In an MFM the tip is magnetically sensitised by coating it with a ferromagnetic material. The tip is scanned several tens or hundreds of nanometres above the sample, avoiding any contact with the sample. Magnetic field gradients exert a force on the tip's magnetic moment, and monitoring the tip/cantilever response gives a magnetic force image. In order to increase the sensitivity, the cantilever is oscillated near its resonant frequency ($\sim 100\text{kHz}$) with a piezoelectric element. The gradients in the magnetic forces on the tip shift the resonant frequency of the cantilever. A magnetic force image is then created by monitoring the changes in oscillation amplitude or phase.

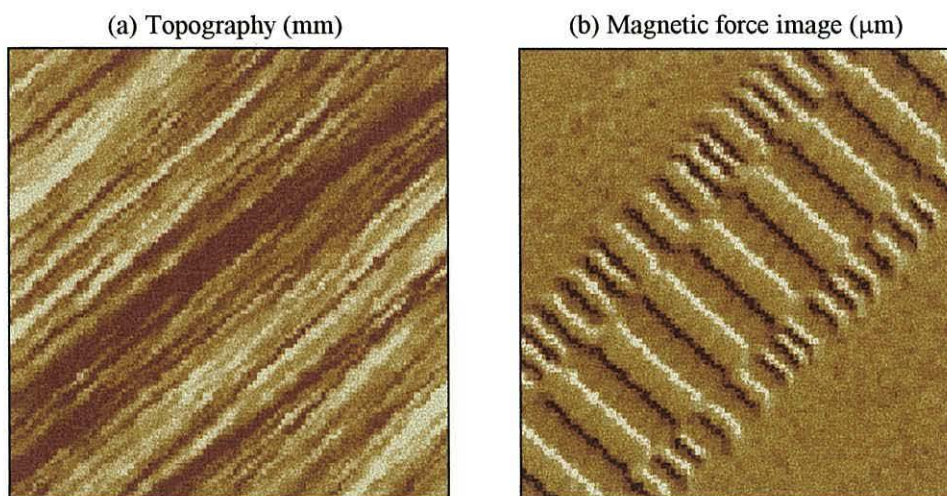
A major problem with the MFM is the influence of surface topography on the detection of magnetic data. The magnetic and topographic data should ideally be kept separate in order to give a more accurate magnetic image, and reveal correlations between magnetic and structural features. One method of achieving this is using a technique called lift-mode, which scans the surface twice (see Figure 1.14). On the first pass, topographical information is recorded using tapping-mode (where the oscillating cantilever lightly taps the surface). Magnetic force data is acquired during the second pass, for which the tip is raised to a pre-selected height (about 20-200nm).

Figure 1.14: *Lift-Mode technique*

The lift height is added on a point-by-point basis to the stored topographical data, thus keeping the tip-sample separation constant and preventing the tip from interacting with the surface. These two-pass measurements are taken for every scan line to produce separate topographic and magnetic force images of the same area.

1.7.2.1 Applications in Data Storage Methods

An MFM can provide valuable information about almost any recording medium. In a hard disk, the width, skew and spacing of the tracks can be measured very accurately to evaluate head performance (as shown in Figure 1.15).

Figure 1.15: *Images of tracks on a hard disk*

In a particulate media, such as digital audio tape, the separation of the topographic and magnetic information can show how particle characteristics affect the recorded information. Orientation of the particles becomes visible, and the magnetic signal weakens considerably as the bit spacing approaches the particle size. This demonstrates that an MFM can image magnetic structure in recording media.

However, the imaging of magnetically 'soft' materials (such as those used in thin-film inductive and magnetoresistive heads) can be more difficult, as the domains in magnetically soft materials are easily perturbed by any stray fields produced by the magnetised tip. Thick (several micron) films such as those used in head yokes can also be imaged, where weak stray fields cause a cantilever frequency shift of as small as 0.1Hz (relative to an 80kHz resonance). Such small signals put an emphasis on precision, low-noise scanning and signal detection.

Chapter 2

Introduction To Neural Networks

2.1 Introduction

The idea of an artificial neural network came about by the better understanding of the human brain, and the recognition that the brain works in a completely different way to a computer. In 1911 Ramón y Cajál [11] suggested that the brain consisted of neurons as its basic building blocks. These can be compared to silicon logic gates in a computer, but whereas logic gates function in the nanosecond region, the neurons in the human brain function in the microsecond region. This difference in speed is compensated for by the brain, by the sheer number of neurons present. Shepherd and Kock [12] estimated that there are on average about 10 billion neurons and 60 trillion connections in the human brain. Basically, a neural network is a machine which has been designed to model the way in which a human learns, and is described by Aleksander and Morton [13] as being:

A massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use, by which the knowledge has been acquired by the network through a learning process

The strengths of the connections between neurons (known as synaptic weights) are adjusted in an orderly fashion using a learning algorithm so as to store knowledge. This idea is the basis of computational neural networks. In the following sections a computational model of the neuron is outlined, and some typical network architectures and learning processes are described.

2.2 The Model of a Neuron

As mentioned, the neuron is the fundamental building block for any neural network. Each one consists of three elements (which are also illustrated in Figure 2.1):

- A series of connecting links, which can be altered individually by changing its weight.
- An adder which sums the input signals.
- An activation function, where a threshold can be applied to limit the output of the model neuron.

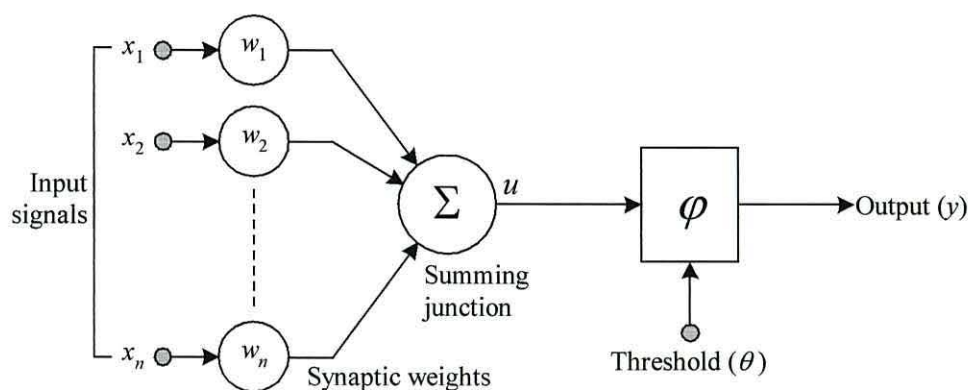


Figure 2.1: *Model of a neuron*

Mathematically, the summing action of the neuron can be described by equation (2.1), whilst the overall output of the neuron by equation (2.2).

$$u = \sum_{j=1}^n w_j x_j \quad (2.1)$$

$$y = \varphi (u - \theta) \quad (2.2)$$

where

x_1, x_2, \dots, x_n = input signals

w_1, w_2, \dots, w_n = synaptic weights

θ = threshold

φ = activation function

y = output of the neuron

The activation function (φ) describes the output of the neuron with respect to the activity level (v) at the input, given by equation (2.3).

$$y = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \quad (2.3)$$

where v is the internal activity level of the neuron, given by equation (2.4).

$$v = \sum_{j=1}^n w_j x_j - \theta \quad (2.4)$$

From equation (2.4), the output of the neuron gives a value of 1 if the total internal activity level of that neuron is not negative and a value of 0 otherwise. This model is sometimes referred to as the McCulloch-Pitts model of the neuron [14].

2.3 Network Architectures

The organisation of the neurons in a neural network is closely linked with the learning algorithm that is used to train the network. The simplest form of neural network is called a *perceptron*, which consists of a single neuron. There are four main architectures available (which are all based on this principle):

2.3.1 Single-layer feedforward network

A single-layer feedforward network consists of two layers of neurons, the input layer and the output layer (Figure 2.2). The input layer is projected onto the output layer in the forward direction only (hence the name feedforward).

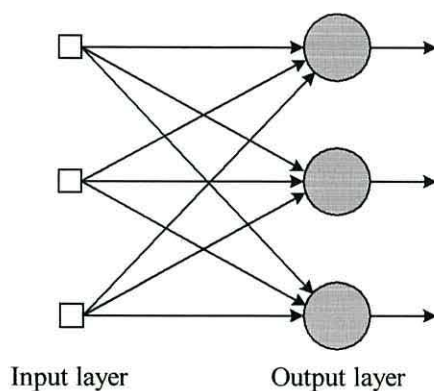


Figure 2.2: *Single-layer feedforward network*

The term ‘single-layer’ refers to the output layer of the network, as no computation is performed by the input layer (and hence it is not counted).

2.3.2 Multi-layer feedforward network

A multi-layer feedforward network (Figure 2.3) is similar to the single-layer network, but it consists of one or more hidden neurons. The function of the hidden neurons is to intervene between the external input neurons and the output of the network. By incorporating hidden layers the network is able to extract higher-order statistics, which becomes useful when the input layer becomes large. In Figure 2.3 the network is fully connected, since every node in each layer is connected to every other node in the adjacent forward layer. The outputs of the neurons in the input layer are used as inputs to the second layer (i.e. 1st hidden layer), and the output of the second layer provide the inputs to the third layer etc. for the whole of the network.

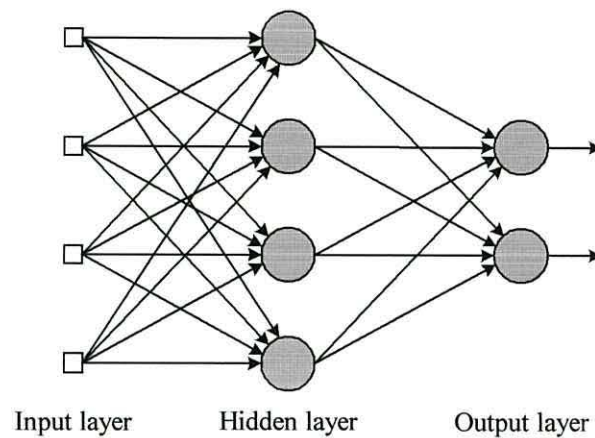


Figure 2.3: *Multi-layer feedforward network*

2.3.3 Recurrent networks

A recurrent network (Figure 2.4) differs from the previous two mentioned architectures in that it contains at least one feedback loop. A typical example would be a network with a single layer of neurons with each neuron feeding its output signal back to the inputs of all the other neurons.

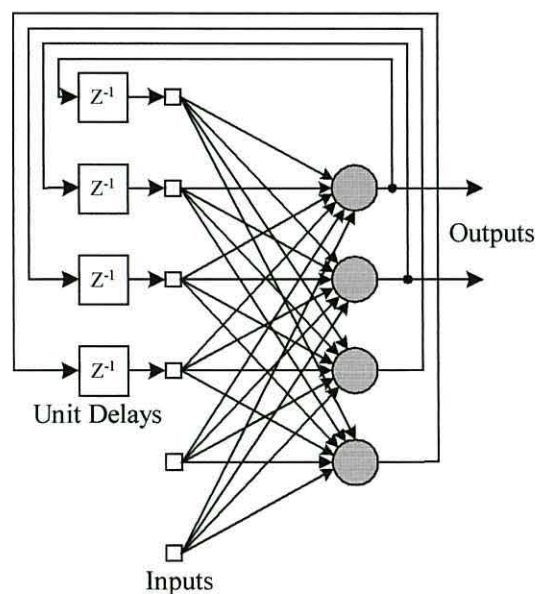


Figure 2.4: *Recurrent network with hidden layers*

Recurrent networks may or may not consist of hidden layers. The use of feedback in the network can have a large impact on the performance of the network.

2.3.4 Lattice structures

A lattice structure (Figure 2.5) consists of arrays of neurons which vary in dimensions. The dimension of the lattice refers to the number of dimensions of space in which the data lies. A lattice network can be compared to a feedforward network, but with the output layer arranged in rows and columns.

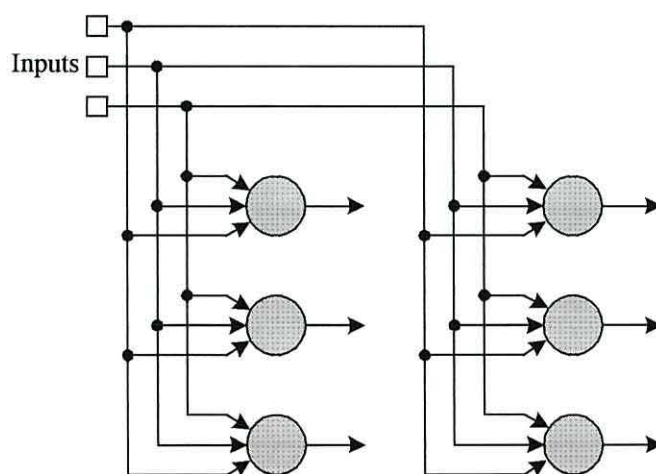


Figure 2.5: *Two-dimensional lattice network*

Each of the architectures are designed for a particular application and have their advantages and disadvantages. The single layer network is the simplest type and requires hardly any processing power, but may not be as accurate as the multi-layer network, whose hidden layers allow it to extract higher order statistics. Recurrent and lattice networks are more complex, as in most cases they involve the feedback of the output signals into the inputs (through delay operators). However, they are advantageous to use when the network is required to be continuously trained (as opposed to trained once).

2.4 Learning Processes

One of the most important features of a neural network is its ability to learn from its surroundings, thus greatly improving its performance. The network is able to learn by making slight adjustments to its synaptic weights, which occur after each cycle of the training data. In the ideal case, the neural network gains more knowledge about its environment after each iteration of the learning process. There are several ways of thinking about the learning process. The one proposed by Mendel and McLaren [15] is:

Learning is a process by which the free parameters of a neural network are adopted through a continuing process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place.

The process is summarised in Figure 2.6.

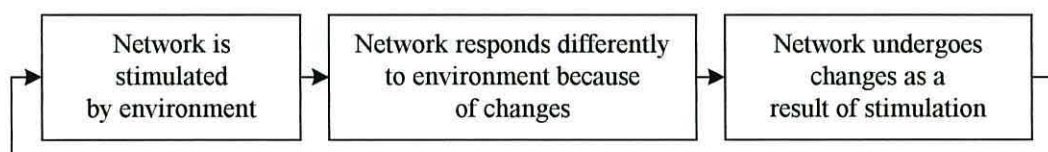


Figure 2.6: *The learning process*

The learning process is considered to consist of a set of learning algorithms and learning paradigms. Three of the most common learning algorithms are described briefly below.

1. Error-correction learning - The error signal is the difference between the required output value and the actual output value. The error signal is then used to minimise the difference between the required and actual output values.

2. Hebbian learning - can be summarised by two rules:
 - If two neurons on either side of a connection are activated simultaneously, then the strength of that connection is increased.
 - If two neurons on either side of a connection are not activated simultaneously, then the strength of that connection is decreased (or in some cases eliminated).

3. Competitive learning - The neurons in the output layer of the neural network compete to be activated, and only one neuron can be activated at any one time. A neuron is able to learn by moving weights from inactive to active input nodes. If a particular neuron does not respond to an input pattern, then no learning takes place in that neuron.

For this work the error-correction learning algorithm has been adopted, as the simplicity of our problem allows the use of such a basic algorithm. The error signal is the difference between the target response and the actual response; and is summarised by equation (2.5).

$$e_k(n) = d_k(n) - y_k(n) \quad (2.5)$$

where

$e_k(n)$ = error signal

$d_k(n)$ = desired response

$y_k(n)$ = actual response

The goal of the error-correction learning is to minimise a cost function based on the error signal $e_k(n)$, so that the actual response of each output neuron approaches the target response of that neuron. A criterion which is used for the cost function is the mean-square-error criterion (which is defined as the mean-square value of the sum of squared errors).

2.4.1 Supervised Learning

In this mode of learning the neural network is supervised by a teacher. The teacher will have knowledge about the environment (this environment is unknown to the neural network), in the form of input-output examples, which is used to train the network. When a neural network is presented with a set of training examples the teacher is able to provide the network with the desired output for particular examples. The network parameters are adjusted due to the influence of the training data and the error (the difference between the actual response and the desired response). The supervised learning method is illustrated in Figure 2.7.

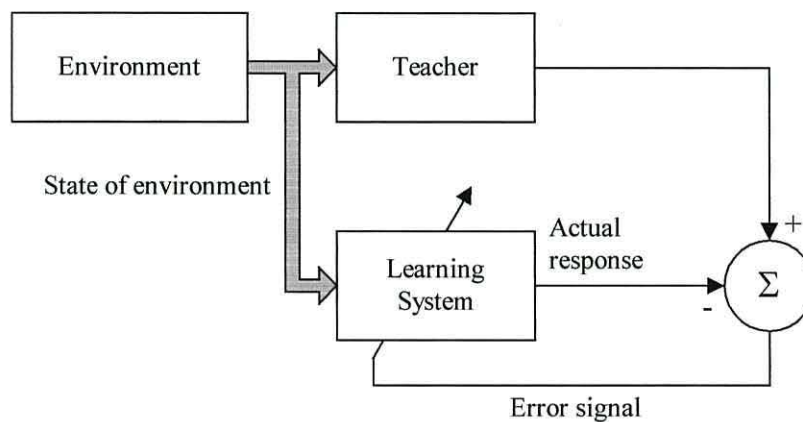


Figure 2.7: *Supervised learning*

The weights of the neurons in the neural network are adjusted as a result of the training examples and the calculated error; the aim being to be able to calculate the input-output relation without the presence of the teacher. This form of learning makes use of the error-correction learning rule. The performance of the network is measured in terms of the mean-squared error, which is averaged over all possible input-output examples. In order for the performance of the network to improve, the operating point has to move down successively towards a minimum point on the error surface, eventually towards the global minimum.

2.4.1.1 Backpropagation Learning Algorithm

Every learning process must have an associated learning algorithm. One such algorithm is the backpropagation learning algorithm [16], which derives its name from the fact that errors are back-propagated through the network on a layer-by-layer basis.

During the forward pass the synaptic weights of the network are all fixed, but during the backward pass, the synaptic weights are adjusted depending on the error signals, with the intention of making the actual response of the network move closer to the desired response. The correction applied to a particular weight in the network is given by the *delta rule* (equation (2.6)).

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \quad (2.6)$$

where

Δw_{ji} = weight correction

η = learning rate parameter

$\delta_j(n)$ = local gradient

$y_i(n)$ = input signal of neuron j

The rate at which the back-propagation algorithm learns depends mainly on two factors, the learning rate parameter η , and the momentum α . The smaller the learning parameter, the smaller will the changes be to the synaptic weights in the network. If the learning parameter is set too large, thus speeding the rate of learning, the network may become unstable.

The learning rate can be increased without making the network unstable by modifying equation (2.6) to include a momentum term, thus giving equation (2.7), which is known as the *generalised delta rule* (Rumelhard *et al.* [17]).

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n) \quad (2.7)$$

where

α = momentum constant

Δw_{ji} = weight correction

η = learning rate parameter

$\delta_j(n)$ = local gradient

$y_i(n)$ = input signal of neuron j

The incorporation of the momentum term represents a small modification to the update of the weights, but still has a great effect on the learning rate of the network. It can also prevent the learning process from going into a shallow local minimum on the error surface.

The back-propagation algorithm can be summarised by five main steps:

1. Initialisation - The synaptic weights of the network are randomised.
2. Presentation of training examples - The network is presented with a set of training examples.
3. Forward computation - The net internal activity and error signal are calculated.
4. Backward computation - The local gradient of the network is calculated by back-propagation, a layer at a time.
5. Iteration - The network is presented with new examples of the training data until the mean-squared error is reduced to about 1% of its initial value.

It has been suggested (Russo [18] and Guyon [19]) that the design of a neural network which uses the back-propagation learning algorithm is more down to trial

and error than any scientific rules. However, there are a few ways in which the algorithm can be improved:

1. The initialisation of the weights in the network should be uniformly distributed inside a small range. However, the range should not be made too small, as it can cause the learning to be very slow initially.
2. All neurons in the multi-layer perceptron should learn at the same rate. Neurons with many inputs should have a smaller learning rate parameter than neurons with few inputs.
3. For an on-line system, the weights should be adjusted in a pattern-by-pattern fashion instead of a batch updating method.
4. The order in which the training patterns are presented to the network should be randomised from one epoch (presentation) to the next.

2.4.1.2 Conjugate-Gradient Method

The conjugate-gradient method consists of choosing ‘conjugate directions’ of minimisation that do not interfere with each other, and performing a line minimisation along that direction. The conjugate gradient method will always locate the minimum of any quadratic function of N variables in at most N steps.

For a non-quadratic function (e.g. the training of a multi-layer perceptron) the process is iterative. The weight vector of the network is updated using equation (2.8).

$$w(n+1) = w(n) + \eta(n)p(n) \tag{2.8}$$

where

$w(n)$ = weight vector

$\eta(n)$ = learning rate parameter

$p(n)$ = direction vector

The direction vector $p(0)$ is initially set equal to the negative gradient vector $g(n)$ at the initial point, given by equation (2.9).

$$p(0) = -g(0) \tag{2.9}$$

Each successive direction vector is then calculated as a linear combination of the current gradient vector and the previous direction vector (equation (2.10)).

$$p(n+1) = -g(n+1) + \beta(n)p(n) \tag{2.10}$$

where

$\beta(n)$ = time-varying parameter

The time-varying parameter $\beta(n)$ can be calculated using two rules; the *Fletcher-Reeves* [20] formula, and the *Polak-Ribière* [21] formula. In the case of a quadratic function, both rules reduce to the same form.

Kramer and Sangiovanni-Vincentelli [22] studied the use of the conjugate-gradient method for the supervised training of multi-layer perceptrons. They showed that the conjugate-gradient method had advantages over the standard backpropagation method, mainly in that it required fewer epochs (and thus quicker to train the neural network); but the disadvantage being that the conjugate-gradient method proved to be computationally more complex.

2.4.2 Unsupervised Learning

Unsupervised learning (self-organising learning) differs from supervised learning in that there is no external teacher present to oversee the learning process, which is illustrated in Figure 2.8. A network which is being trained by unsupervised learning is not provided with any learning examples, instead it is presented with only the input patterns. Once the network has recognised any regularities within the input patterns, it can develop the ability to create internal representations for encoding specific features. From there the network can then create new classes automatically.

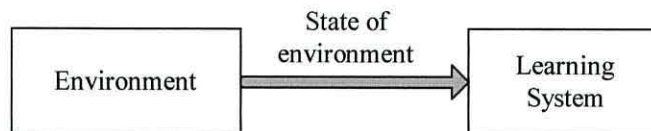


Figure 2.8: *Unsupervised learning*

The self-organising network, which is generally used for unsupervised learning is based on competitive learning. Only one neuron in the output (competitive) layer can be activated at any one time, and is thus called the *winner* neuron.

The neurons are normally situated at the nodes of a lattice that is normally of one or two dimensions. These neurons become selectively tuned to various input patterns, and their positions become ordered with respect to each other so that a meaningful co-ordinate system is created for the different input features.

The unsupervised learning approach is discussed in greater detail later in the work, in particular the network architecture and its ability to extract features from magnetic based images (such as magnetisation maps and MFM images).

2.4.2.1 Kohonen Self-Organising Network

One model of an unsupervised network was proposed by Kohonen [23], and is called the *Kohonen self-organising network*. The network is an example of a self-organising feature map, where neurons are placed at the nodes of a one or two dimensional lattice. The model tries to capture the important features of the computational maps in the brain and still remain computationally tractable.

A one dimensional lattice of neurons usually contains two different types of connections: Forward connections are arranged from the primary source of excitation, whilst the lateral feedback connections are internal to the network. Figure 2.9 illustrates the Kohonen network model.

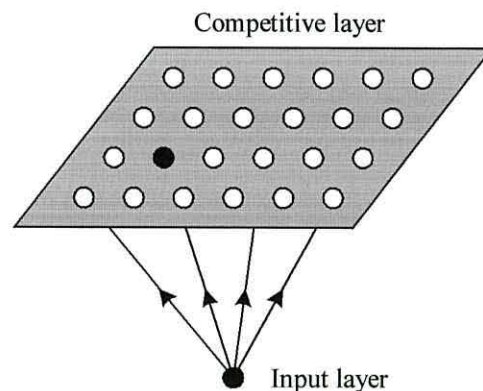


Figure 2.9: *Kohonen model*

In order to start the learning process, and the development of the self-organising feature map, the competitive layer must be initialised. The input pattern data is simultaneously presented to the neurons in the competitive layer in a parallel fashion, and the network is trained for a set number of epochs. During this period the neurons become tuned to particular input patterns, and their locations in the lattice become ordered with respect to each other.

The goal of the self-organising map algorithm which was developed by Kohonen [23] is to transform an input pattern of a particular dimension into a one or two dimension discrete map. This algorithm can be summarised as:

1. A one or two dimensional lattice of neurons whose purpose is to compute simple mathematical functions.
2. A mechanism to compare the functions and select a neuron with the largest discriminant function value.
3. A network that activates the selected neuron (and the neighbouring neurons).

The accuracy of the self-organising map is dependent on the number of iterations of the learning algorithm, the choice of values for the learning-rate parameter η and the neighbourhood function, which decides the number of neurons which surround the winning neuron (e.g. a neighbourhood value of 1 will include every neuron within a radius of 1 to the winning neuron). As there is no theoretical method for selecting these values, they have to be estimated on a trial and error basis.

However, the learning-rate parameter should be time-varying. During the first 1000 iterations it should be close to 1, then it should decrease slowly towards about 0.1. The topological ordering in the competitive layer occurs at the initial phase of the algorithm. The lateral effect of the winning neuron on the activity of its neighbours can be accounted for by making the topological neighbourhood around the winning neuron decay with distance (Ritter *et al.* [24], Lo *et al.* [25]) i.e. the further the neurons are from the winning neuron, the less effect there is on them due to the winning neuron. This is illustrated in Figure 2.10.

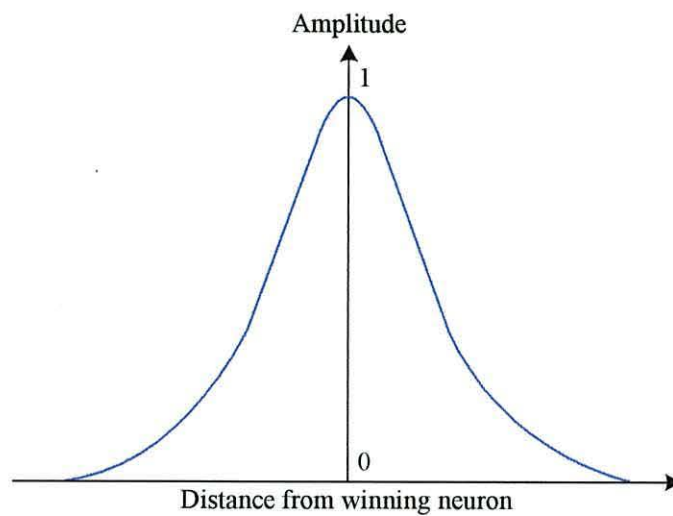


Figure 2.10: *Neighbourhood decay function*

2.4.3 Pattern Recognition

The process of pattern recognition is discussed in greater detail in Chapter 5, but is summarised in this section. A pattern classifier is required to classify an input signal into a finite number of classes, so that the average probability of it being misclassified is minimised. The most important task is to decide where boundaries between two classes should be. It should be noted however, that a feature map is only intended to visually demonstrate the relationships between the input data. The feature map should therefore not be used on its own in a pattern classification system.

Pattern classification can be improved substantially by using a self-organising feature map together with a method of supervised learning. The adaptive pattern classifier (Figure 2.11) is one such method.

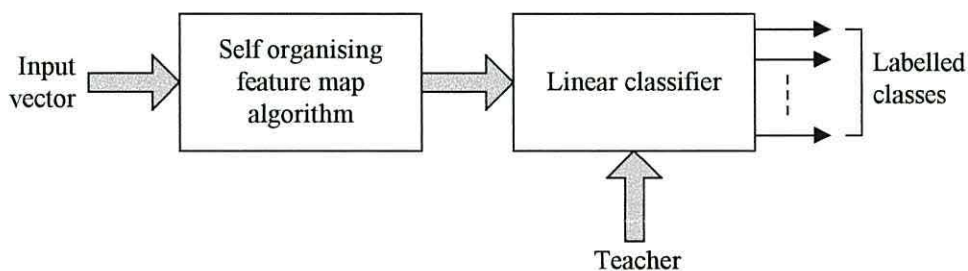


Figure 2.11: *Adaptive pattern classifier using LMS algorithm*

The self-organising feature map acts as a pre-processor, and takes the form of a two-dimensional lattice of say a 3x3 matrix of neurons (the input data). This data is then projected onto the self-organising map in an orderly manner. The linear adaptive classifier consists of two neurons and is fed by the outputs of the self-organising feature map. The linear classifier then labels and categorises the output data (using the least mean squared algorithm).

2.5 Knowledge Representation

The idea of knowledge representation can be considered as: what information is made explicit and how the information is encoded for later use. The definition suggested by Fischler and Firschein [26] is:

Knowledge refers to stored information or models used by a person or machine to interpret, predict, and appropriately respond to the outside world.

The subject of knowledge representation within a neural network becomes very complicated. However, there are some rules which are noted [27]:

- Similar inputs from similar classes normally produce similar representations inside the neural network, and should be classified in the same group. There are several methods for determining the similarity between inputs (the reader is referred to Haykin [28] for more details).
- Items that are to be classified in different groups should be given widely different representations in the network.
- If a particular feature is important, then several neurons should be used to represent that item in the network.

- Any prior information should be built into the design. The network is then spared the time of not having to learn them.

2.6 The Advantages of Neural Networks

It would be almost impossible to justify the use of neural networks without first mentioning some of their advantages. It can be recalled that a neural network derived its power through its massively parallel structure and its ability to learn and hence generalise (i.e. to produce intelligible results with unseen data). These make it possible for a neural network to solve problems of such a large scale that would seem impossible using conventional methods.

However, it should be understood that a neural network cannot be used alone to solve a problem; but should be integrated into another system. The problem should then be divided into simpler problems, with a neural network assigned to each task (e.g. pattern recognition, control system etc.). These are just some of the advantages which it is felt should be mentioned (as suggested by Haykin [28]):

1. Nonlinearity - A neural network is a non-linear system, which is an important feature especially if the input data is of a non-linear form (e.g. speech analysis).
2. Input-output mapping - In a supervised learning neural network the system is trained with a unique input and a desired output (which is selected at random from a training set). The network is thus able to learn from these examples by composing an input-output mapping for that particular problem.
3. Adaptivity - A neural network is very good at adapting to the problem at hand; and can be trained to deal with an environment where minor changes occur in the conditions (and can even be adapted to the environment in real-time).

4. Fault tolerance - If a neural network were to be constructed in hardware, then its tolerance to faults is very high with hardly any degradation in performance, even under adverse operating conditions (Bolt [29]). This is due to the distributed nature of information in the network, where a damaged neuron will have little effect on the overall system.

5. Evidential response - In the case of the pattern classifier, a neural network can be designed to provide information not only about which particular pattern to select, but also about the accuracy of the decision made (and hence can be used to improve the classification performance of the network).

6. Uniformity of analysis and design - All neural networks have one thing in common, in that they are all information processors. However, this feature manifests itself in different methods:
 - All neural networks are constructed of neurons.
 - Learning algorithms and theories can be used in a wide variety of neural network applications.

Chapter 3

Determination of Anisotropy Distributions From Transverse Susceptibility Data

3.1 Introduction

The idea of transverse susceptibility was first discussed by Gans [30] in 1909; and subsequently several other scientists have worked on the subject, both theoretically and experimentally. In this chapter the transverse susceptibility of a particulate media is investigated (using theoretical models and experimental samples).

The transverse susceptibility of a magnetic material can be defined as being the change in magnetisation (ΔM) with respect to a small AC field applied perpendicular to a DC bias field H . This was discussed by Aharoni *et al* [31], on the basis of the Stoner-Wohlfarth model, and is given by (3.1).

$$\chi_t = \lim_{\Delta H_{ac} \rightarrow 0} \left(\frac{\Delta M}{\Delta H_{ac}} \right) \quad (3.1)$$

The theoretical and experimental values for the transverse susceptibility are linked by a texture function, which is dependent on the particulate media itself. This chapter discusses the determination of the transverse susceptibility, both theoretically and experimentally, and explains the correlation between them.

The importance of this problem can be emphasised by taking a look at the effect that varying the values of the anisotropy K and anisotropy distribution $f(K)$ have on a magnetic material. Figure 3.1 shows a plot of K against $f(K)$, and illustrates the consequences of varying these.

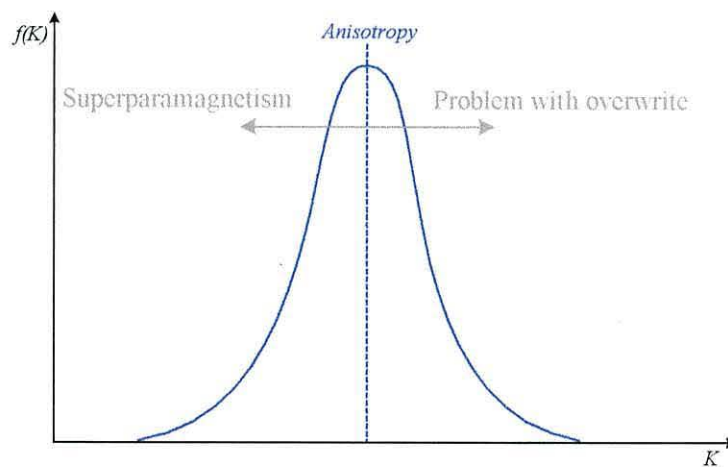


Figure 3.1: *The consequences of varying anisotropy on a magnetic material*

When the value of K decreases and becomes very small, the thermal activation energy and the applied external magnetic field can easily flip the particles away from their preferred orientation, and as a result they behave like paramagnetic atoms but with giant magnetic moments. This has the effect of them having no remanence and thus cannot store information.

However, when the value of K increases, it is much more difficult to cause the particles to change from their preferred orientation. The field required to overwrite the information becomes large, and as a result the distribution of K and not just its mean value becomes important.

It is possible to have an ideal mean value of K which is between the superparamagnetic regime and the regime giving rise to overwrite problems, but if the distribution is too wide there will still be a problem. The value of K can quite easily be determined, but the determination of the distribution of K becomes a very complex problem.

3.2 Theoretical Determination of the Transverse Susceptibility

In 1957 Aharoni *et al* [31] carried out work on theoretical expressions for the transverse susceptibility of a Stoner-Wohlfarth system, which showed that the theory for a randomly oriented system predicts a characteristic curve with peaks at $\pm H_K$ and H_C . In systems such as doped barium ferrite particles [32] and γFe_2O_3 the determination of H_K is difficult due to a wide distribution of the anisotropy field which causes the peak to be suppressed, whereas in an aligned system the peaks at $\pm H_K$ are well defined [33]. In order to understand this concept, some knowledge of the effects of texture is important. Hoare *et al* [34] defined such a function, with independently controllable in-plane σ_θ and out-of-plane σ_ϕ distributions as in equation (3.2).

$$f(\theta_K, \phi_K) = \frac{1}{A} \exp\left(-\left(\frac{(\pi/2 - \theta'_K)}{2\sigma_\theta}\right)^2\right) \times \exp\left(-\frac{\phi_K^2}{2\sigma_\phi^2}\right) \quad (3.2)$$

where

A = normalising constant

θ'_K = angle between applied field and projection of easy axis into plane

$\sigma_\theta, \sigma_\phi$ = control the distributions in and out of the plane respectively

Theoretical expressions for the transverse susceptibility can be extended to include the texture function (Hoare *et al* [34]), given by equation (3.3):

$$\begin{aligned} \frac{\overline{\chi_t}}{\chi_0} = & \frac{1}{2\pi} \int_0^{\pi/2} \frac{3}{2} \left(\frac{\cos^2 \theta_M}{h \cos \theta_M + \cos 2(\theta_M - \theta_K)} \times \int_0^{2\pi} \cos^2 \phi_K \right. \\ & \times \frac{\exp\left(-(\pi/2 - \theta_K^*)^2 / 2\sigma_\theta^2\right) \exp\left(-\phi_K^2 / 2\sigma_\phi^2\right)}{A} d\phi_K + \frac{\sin(\theta_K - \theta_M)}{h \sin \theta_K} \int_0^{2\pi} \sin^2 \phi_K \\ & \left. \times \frac{\exp\left(-(\pi/2 - \theta_K^*)^2 / 2\sigma_\theta^2\right) \exp\left(-\phi_K^2 / 2\sigma_\phi^2\right)}{A} d\phi_K \right) \times \sin \theta_K d\theta_K \quad (3.3) \end{aligned}$$

where

$\sigma_\theta, \sigma_\phi$ = control the distributions in and out of the plane respectively

θ_K^* = projection of θ_K into the plane of the tape

A = normalising constant

$h = H/H_K$ = reduced applied field

θ_K, ϕ_K = orientations of the easy axis

Equation (3.3) can be simplified:

$$\frac{\overline{\chi_t}}{\chi_0} = \frac{3}{4} \int_0^{\pi/2} \left[\frac{\cos^2 \theta_M}{h \cos \theta_M + \cos(2(\theta_M - \theta_K))} W_1(\theta_K) + \frac{\sin(\theta_K - \theta_M)}{h \sin \theta_K} W_2(\theta_K) \right] \sin \theta_K d\theta_K \quad (3.4)$$

where

$$W_1(\theta_K) = \int_0^{2\pi} \cos^2 \phi_K \exp\left(-(\pi/2 - \theta_K')^2 / 2\sigma_\theta^2\right) \times \exp\left(-\phi_K^2 / 2\sigma_\phi^2\right) d\phi_K \quad (3.5)$$

and

$$W_2(\theta_K) = \int_0^{2\pi} \sin^2 \phi_K \exp\left(-(\pi/2 - \theta_K')^2 / 2\sigma_\theta^2\right) \times \exp\left(-\phi_K^2 / 2\sigma_\phi^2\right) d\phi_K \quad (3.6)$$

By doing this, the computational complexity is substantially reduced. Values for θ_K which are not known can be calculated by linear interpolation. In order to model real media, it is necessary to include a texture function.

The orientational texture is found to be dependent on several factors, primarily the orienting field and the action of shearing during the coating process. Thus, it should be assumed that the texture function does not have cylindrical symmetry. Figure 3.2 shows the variation of χ_t with reduced field h , for systems with varying degrees of texture.

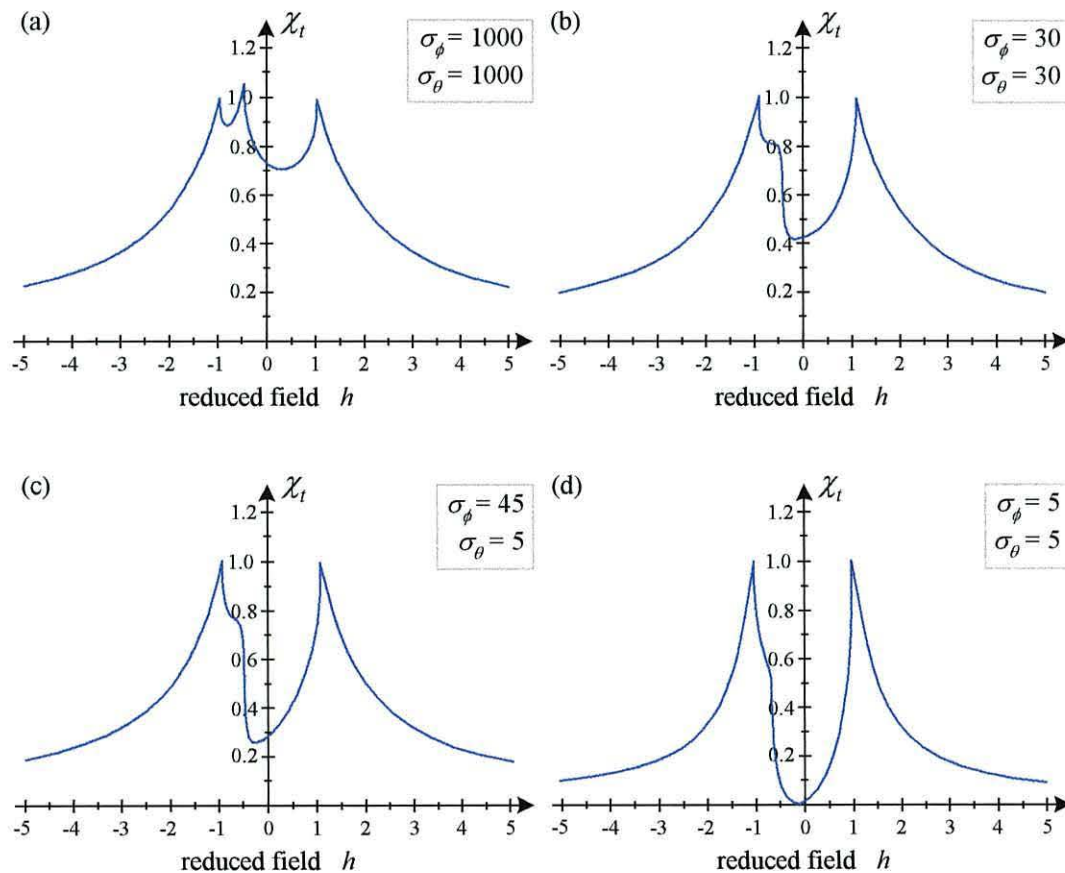


Figure 3.2: Variation of χ_t with different degrees of texture

From Figure 3.2 (a), there is an indication of a broad distribution of the easy axis, implying a random distribution. The presence of the H_C peak (at -0.5) is obvious. However, as the degree of texture is increased, the height of the H_C peak decreases, and in (d) has almost disappeared. This is due to strong enhancement of the H_K peak caused by texturing. The method of enhancing the degree of texture, suggesting a very well aligned system, is used experimentally to determine the value of K .

Theoretically, the value of K may be calculated using equation (3.7).

$$H_K = \frac{2K_1}{M_S} \quad (3.7)$$

where

M_S = saturated value of magnetisation

K_1 = constant for a particular material

H_K = anisotropy field

In a real system it is necessary to include a distribution of the values of H_K , which take into account the variation of particle shapes, which can be accounted for by integrating the χ_t curve with either a log-normal distribution function, as in equation (3.8) or a Gaussian distribution function, as in equation (3.9).

$$\chi_t(H) = \int_0^\infty \chi_t(H/H_K) \frac{1}{\sqrt{2\pi} \sigma_{H_K} H_{K_M}} \times \exp \left[-\frac{(\ln(H_K/H_{K_M}))^2}{2\sigma_{H_K}^2} \right] dH_{K_M} \quad (3.8)$$

$$\chi_t(H) = \int_0^\infty \chi_t(H/H_K) \frac{1}{\sqrt{2\pi} \sigma_{H_K}} \times \exp \left[-\frac{(1-H_K/H_{K_M})^2}{2\sigma_{H_K}^2} \right] dH_{K_M} \quad (3.9)$$

for a particular field H and $\chi_t(H/H_K)$ is the kernel defined by equation (3.3). The kernel has a well defined cusp at $H = H_K$. However, the H_K distributions lead to rounded peaks which compare well with experimental data obtained by Pareti and Turilli [35], and a comparison shows that the shoulder is partially absorbed into the main H_K peak, which suggests that the peak at H_C may easily become indistinguishable from the main H_K peaks in practice, which thus complicates the determination of the mean value of H_K .

Figure 3.3 shows the effect of integrating the χ_t curve with a distribution function (log-normal, with $\sigma_H = 0.1$).

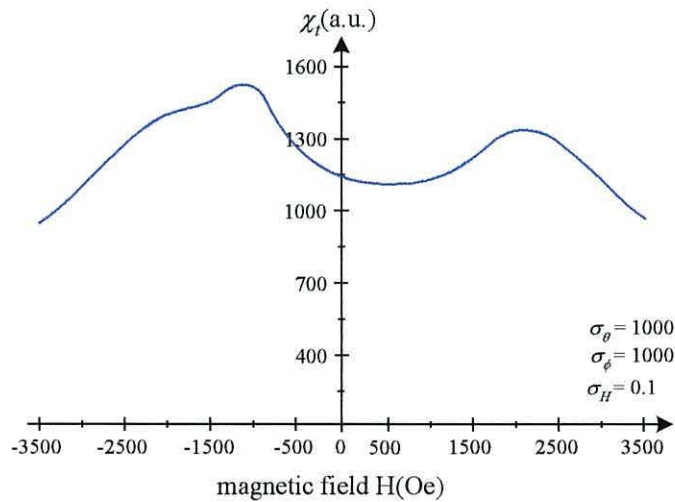


Figure 3.3: *Effect of integrating the χ_t curve with a distribution function*

3.3 Experimental Determination of the Transverse Susceptibility

The experimental value of the transverse susceptibility can be determined with a set-up shown in Figure 3.4. The system consists of a coil $C1$, which generates an alternating magnetic field H of variable amplitude and frequency (general ranges are 1-20Oe and 1-60kHz respectively).

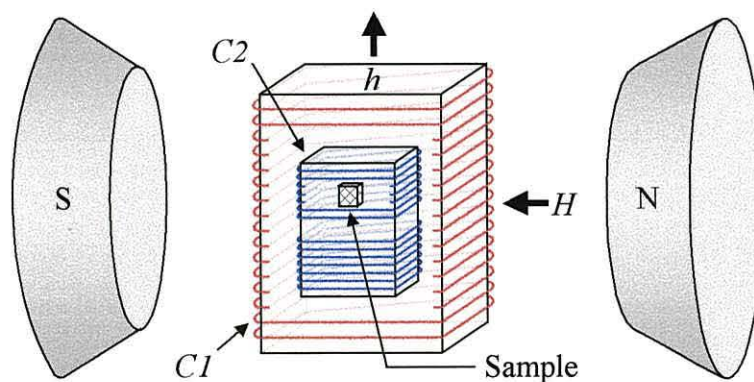


Figure 3.4: *Experimental apparatus to measure the χ_t of a sample*

A pick-up system $C2$ is placed in the centre of $C1$, which consists of two coils wound in opposite ways, so as to achieve a zero signal when no sample is present. The whole system is then placed between the poles of an electromagnet (with a typical maximum of about 15-20kOe), which provides the bias field (\bar{H}) and is perpendicular to the driving field (\bar{h}). The sample is then placed inside the pick-up coil.

The signal which is produced due to the sample is detected by a lock-in amplifier, and according to Lenz's law (at constant H) is:

$$v = \frac{d\Phi}{dt} = K\hat{a} \frac{d\bar{M}}{dt} = K\hat{a} \left(\frac{d\bar{M}}{d\bar{h}} \right) \left(\frac{d\bar{h}}{dt} \right) = K \sum_{ij} \alpha_i \left(\frac{dM_i}{dh_j} \right) \left(\frac{dh_j}{dt} \right) \quad (3.10)$$

where

Φ = magnetic flux of the sample

K = geometrical factor

\hat{a} = unit vector of the pick-up coil

M = magnetic moment of the sample

$\bar{h} = \bar{h}_0 \sin \omega t$ = driving field

dM_i/dh_j = components χ_{ij} of the susceptibility tensor

According to Pareti and Turilli [35], the direction of \bar{h} and the pick-up coil are along the z axis ($h \perp H$), $d\bar{M}/d\bar{h} = dM_z/dh_z = \chi_t$ and $d\bar{h}/dt = dh_z/dt = h_0 \omega \cos \omega t$, thus:

$$v = K \chi_t h_0 \omega \cos \omega t \quad (3.11)$$

which means that the detected signal v is proportional to the reversible transverse susceptibility of the sample. Measurements are made with the DC field perpendicular to the direction of texture and the AC field parallel to it, as this is the configuration which is assumed in the theoretical calculations. Work carried out [34, 35] show that

the form of the χ_t curve is strongly dependent upon the texture, and that there are only the peaks at $\pm H_K$ present, as the peak at H_C has completely disappeared.

3.4 The Inverse Problem

As discussed in Chapter 1, inverse problems are concerned with determining causes for a desired or an observed effect. They might not have a solution in the strict sense, and solutions might not be unique and/or might not depend continuously on the data. In the case of the transverse susceptibility problem, the anisotropy field distribution is to be determined from experimental data. This can be summarised as follows:

$$\chi_t(H) = \int_0^{\infty} f(y) \times \chi_t(H, H_K) dy \quad (3.12)$$

where

$f(y)$ = distribution function (unknown)

$y = H_K / H_{K_M}$

$\chi_t(H)$ = experimental value of transverse susceptibility

$\chi_t(H/H_K)$ = theoretical value of transverse susceptibility

Given the known kernel $\chi_t(H/H_K)$ and a set of experimental data, the problem is to determine the $f(y)$. The aim is to train the neural network using theoretically generated χ_t results to recognise specific relationships between χ_t and the distribution function. When the network has been trained sufficiently, then a real set of experimental data will be applied to test the feasibility of the method. The process is illustrated in Figure 3.5.

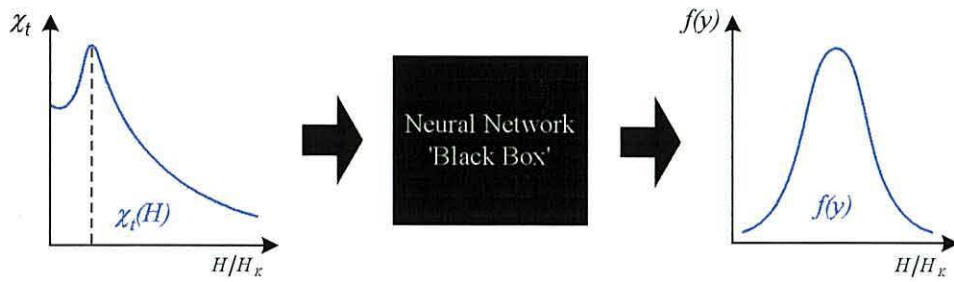


Figure 3.5: *The inverse problem (applied to the transverse susceptibility problem)*

The value of $f(y)$ can be calculated using several methods; one method, which will be discussed is the maximum entropy approach. However, these methods are known to be quite complex computationally, which is practically undesirable. One way of reducing the computational time is to use neural networks, which can be trained to solve a problem beforehand. This approach will be discussed in detail later in this chapter.

3.5 The Maximum Entropy Approach

3.5.1 Introduction

The distribution of H_K values, which were generated by integrating the distribution function $f(y)$ with χ_t , can be deconvoluted using a maximum entropy approach. This is a powerful technique which provides an extension of the established principles of rational inference. Its intention is to extract the conclusions from given data and any prior knowledge of the circumstances. It should be noted however, that the method is not guaranteed to compensate for inadequate data or badly designed experiments. The main use of the maximum entropy method is to go from incomplete and noisy data to some description of the underlying physical system.

3.5.2 Practical Considerations

In the majority of scientific investigations, measurements are taken and then processed to arrive at conclusions. Data processing can be thought of as the conversion of the measured data to arrive at conclusions, which can be summarised by the following expressions (Daniel, [36]). If N numbers are to be determined from K observations (where K is finite and N is small or infinite), then four cases arise:

1. $N < K$

This situation implies that there is inconsistent data and experimental errors.

2. $N = K$

There is an unique solution, but an unstable problem may occur when a lot of data values contain almost identical information about the conclusions.

3. $N > K$

In this case there is insufficient data, and the problem is usually unsolvable, due to many conclusions being consistent with the same data. The most common approaches to solving this situation is:

- Assume that the 'rough curve' being sought is actually smooth and has a simple functional form (known as model fitting).
- Invent sufficient extra data to get $N = K$.

4. $N = \infty$

There is no solution to this problem.

3.5.3 Simulated Annealing

The problem of maximum entropy can also be solved using a simulated annealing algorithm, which is based on the analogy between the simulation of the annealing of solids and the problem of solving large combinatorial optimisation problems.

In condensed matter physics, annealing is known as a thermal process for obtaining low energy states of a solid in a heat bath. In this process the solid is heated up by increasing the temperature of the heat bath to a maximum value at which particles in the solid randomly arrange themselves in the liquid phase. The temperature of the heat bath is then lowered carefully to cool the liquid. The particles arrange themselves in the low energy ground state of a corresponding lattice (provided the maximum temperature is sufficiently high and the cooling is carried out sufficiently slowly). As the temperature decreases, the Boltzmann distribution concentrates on the states with lowest energy, and finally as the temperature approaches zero, only the minimum energy states have a non-zero probability of occurrence. However, if the liquid is allowed to cool too rapidly the solid is not allowed to reach thermal equilibrium at each temperature level, hence the solid will be forced into a meta-stable state rather than into the ground state.

3.5.4 The Deconvolution of H_K Distributions

The simulated annealing approach can be used to deconvolute H_K distributions, assuming the kernel was integrated with a log-normal distribution:

$$f(y) = \frac{1}{\sqrt{2\pi\sigma y}} \exp\left(\frac{-(\ln y)^2}{2\sigma^2}\right) \quad (3.13)$$

This approach was investigated by Chantrell *et al.* The value of $f(y)$ was calculated by taking the average of the last 20% of values at each temperature, (the first 80% were

used to ensure thermal equilibrium at each temperature). The starting temperature T_M was set so that the average energy change per move was roughly $0.2kT_M$.

It was shown that reasonable fits to the original curve could be obtained by selecting the ratio between successive temperatures greater than 0.5. The plots in Figure 3.6 show the formation of $f(y)$ values for four decreasing temperatures. The log-normal distribution curve is shown for comparison.

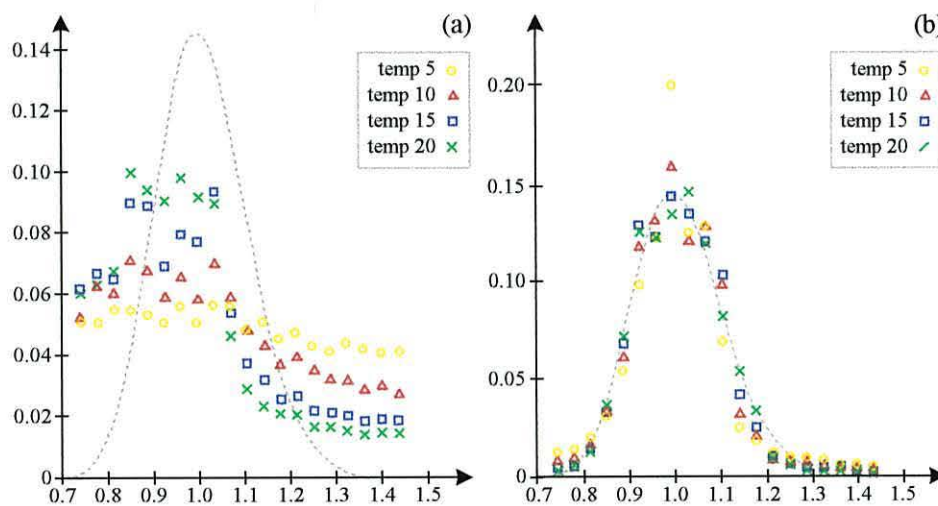


Figure 3.6: Fits to log-normal distribution using simulated annealing

The investigation by Chantrell *et al* showed that it required another two sets of decreasing temperatures before a stable solution could be reached. Figure 3.6(b) shows that at the lowest temperature a good estimation of the log-normal distribution is achieved.

3.6 Application of Neural Networks

3.6.1 Introduction

The application of a neural network technique to solve the magnetic inverse problem is quite novel, but the research to be presented here shows that such a method can be used to solve such a problem, with many advantages. The selected approach was to

train the neural network with theoretically generated χ_t data (by integrating the kernel with a log-normal or Gaussian distribution function), test it with other sets of theoretical χ_t data, and then apply a set of experimental χ_t data.

As the theoretical data sets can be generated quite easily for a wide selection of distribution functions, there was a plentiful supply of data to train the network. In such a situation it is possible to make use of the supervised learning method, together with a network configuration called the multi-layer perceptron.

3.6.2 Multi-layer Perceptron

The multi-layer perceptron is one of the most important types of neural networks. Sometimes referred to as a multi-layer feedforward network, the network typically consists of a set of sensory units (source nodes) that constitute the *input layer*; one or more *hidden layers* of computation nodes, and an *output layer* of computation nodes, as illustrated in Figure 3.7. The input signal propagates through the network in a forward direction, on a layer-by-layer basis.

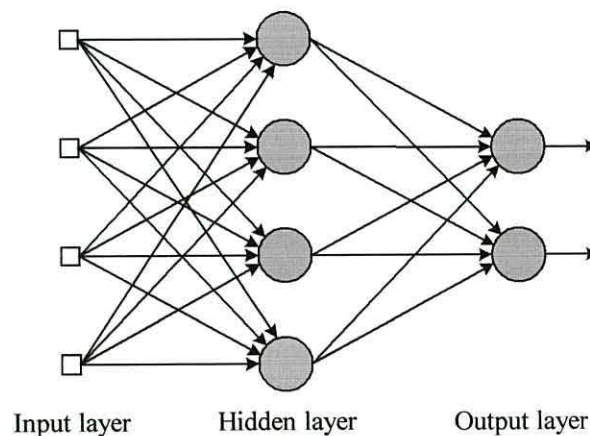


Figure 3.7: A basic multi-layer perceptron network

The network shown here is fully connected, which means that a neuron in any layer of the network is connected to all the neurons in the previous layer. Signal flow through

the network progresses in a forward direction, from left to right and on a layer-by-layer basis.

There are many cases where multi-layer perceptrons have been applied successfully to solve some difficult problems by training them in a supervised manner using the popular *error back-propagation* algorithm. This algorithm is based on the error-correction rule, and basically consists of two passes through the different layers of the network; a forward pass and a backward pass (see Figure 3.8). In the forward pass an activity pattern is applied to the sensory nodes of the network, and its effect propagates through the network layer-by-layer. Finally, a set of output signals are produced as the response of the network.

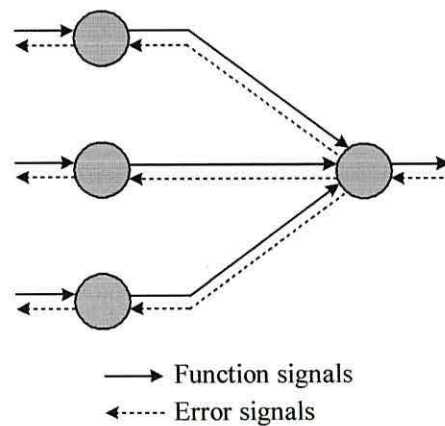


Figure 3.8: *Two basic signal flows in a multi-layer perceptron*

During the forward pass the synaptic weights of the network are all fixed. However, during the backward pass the synaptic weights are all adjusted in accordance with the error-correction rule. The synaptic weights are adjusted so as to make the actual response of the network move closer to the desired response.

In the majority of cases, a multi-layer perceptron network configuration is trained in a supervised manner. The input and output neurons in the network are subjected to training sets, with known input-output relations (which forms the basis for *supervised learning*, which was discussed in Chapter 2). However, as it will become clear the

type and arrangement of the input-output relations can have a considerable effect on the results obtained from the network.

3.7 Data Preparation and Network Optimisation

3.7.1 Introduction

Before training a neural network, the data which is to be presented must be arranged in a specific format. The format can make a significant difference to the results obtained from the network. Since a multi-layer perceptron requires a training set to consist of input-output related patterns, it was decided to use an array of input neurons to represent the x and y values of the curve and a single output neuron to represent the standard deviation of the distribution function. The networks used in this chapter were simulated using the *Stuttgart Neural Network Simulator* [37].

To summarise, the problem in essence was to train the network with a set of theoretically generated data and then investigate its performance with another set of theoretical data, before finally testing it with experimentally obtained data.

3.7.2 Data Preparation

It was necessary to normalise both theoretical and experimental data points to similar values for presentations to the network, in order to create a sufficiently general network capable of dealing with a variety of materials. The value of H_K does not remain constant for each sample; it usually ranges from 2-5kOe, depending on the particular material investigated.

As most of the information resides around the H_K peak it was seen as desirable to normalise the data so that H_K peaked at a fixed value (2-5kOe), hence eliminating the need to train the network with all variations of the χ_t curve. Figure 3.9 illustrates areas of the χ_t curve which are used in this work. At low fields the theoretical model

which the work is based on fails as is discussed later, and at high fields the data is envisaged to be redundant, as there is no great change with varying distributions.

It was also noticed that the χ_t value could be normalised along the y axis; since the value of χ_t did not fall below 0.5. Thus by altering the y axis from 0.5-1.0, it was possible to amplify the diversity between the χ_t curves.

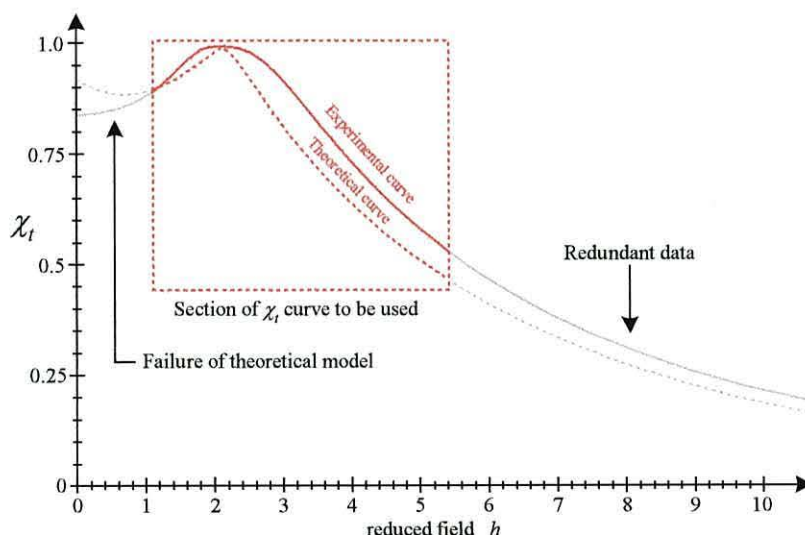


Figure 3.9: *Normalisation of training curve*

Having decided on the format of the training patterns, a series of χ_t curves was generated by integrating the kernel with a variety of distribution functions (equations (3.8) and (3.9)) with standard deviations ranging from 0.05 to 0.75.

3.7.3 Training Set Size

The quantity of training examples is at this point yet to be finalised. However, the number of training examples required can be estimated to a first order approximation using equation (3.14), using a result derived by Baum and Haussler [38]:

$$N > \frac{W}{\varepsilon} \tag{3.14}$$

where

N = is the number of examples

W = the total number of synaptic weights in the network

ε = the fraction of errors permitted on test

Thus, with an error of 10%, the number of training examples should be approximately 10 times the number of synaptic weights in the network. If the curve is represented by 50-100 data points, then even a small network (4-8 hidden neurons) will require about 2000-8000 training examples. Here an alternative method is used, overcoming the need for a large training set, called *early stopping* (Nelson and Illingworth [39]). This generally consists of:

- Dividing the available data into training and validation sets.
- Increasing the number of hidden neurons.
- Using small initialisation values.
- Using a slow learning rate parameter.
- Calculating the validation error during training.
- Stopping the training when the validation error reaches a minimum.

The advantages of using this training method is that it is fast, and can be applied to networks in which the number of weights exceeds the size of the training set. From these points, it would be feasible to use a training set of only about 100 examples.

The network was trained using the technique of cross-validation (Stone [40], Janssen *et al* [41]), where the generated data set is randomly divided into a training set and a test set. The training set is further divided into two sets:

- A set used to train the network (*training set*).
- A set used to evaluate the performance of the network (*validation set*), which should be approximately 10% of the training set.

To test this theory, a simulation was conducted on a network of various dimensions (but with all other parameters fixed), with training sets consisting of 50, 100 and 500 examples. Validation sets consisted of 5, 10 and 50 examples respectively. Figure 3.10 compares the mean squared error as a function of number of epochs (i.e. training) required to sufficiently train a network.

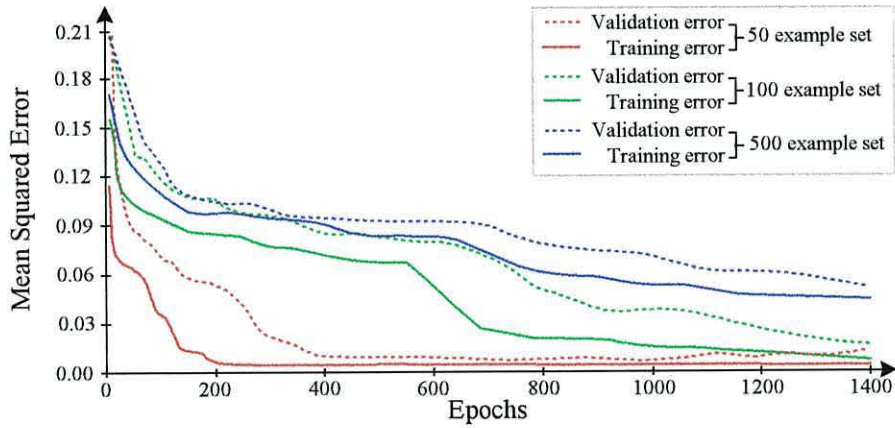


Figure 3.10: Comparison of the training time required with various data set sizes

It is clear from Figure 3.10 that a network trained on a small training set will generalise quicker than one trained on a larger set. However, this doesn't imply that it will perform better when a test set is applied.

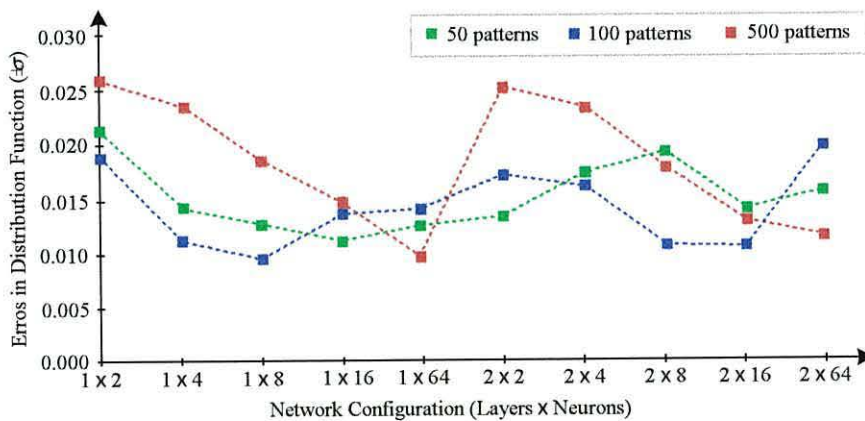


Figure 3.11: Comparison of the distribution function errors with various data set sizes

Figure 3.11 shows the error in the distribution function from the output of the network when a test set is applied at the inputs. It can be seen that the training set with 100 examples tends to give the best results, with a network size of about 4-8 hidden neurons in 1 hidden layer being sufficient. The 500 example set performed rather poorly; it is envisaged that the reason for this is that a 500 example set contained too much information for a ‘small’ network to handle. It would have been possible to increase the size of the network, but this would slow the calculations down (the smaller data set could solve the same problem quicker anyway). However, when the network size was increased, the error for the 500 example set decreased.

3.7.4 Input Data Points

In determining the optimal number of data points to represent the χ_t curve, several samples were created (with various distributions) and sampled using 25, 50 and 100 data points.

As the number of input neurons (and subsequent connections to the hidden layer) in the network is equivalent to the data points selected, it is crucial to decide on a suitable number of data points. Figure 3.12 shows the results of various simulations conducted using different sample points.

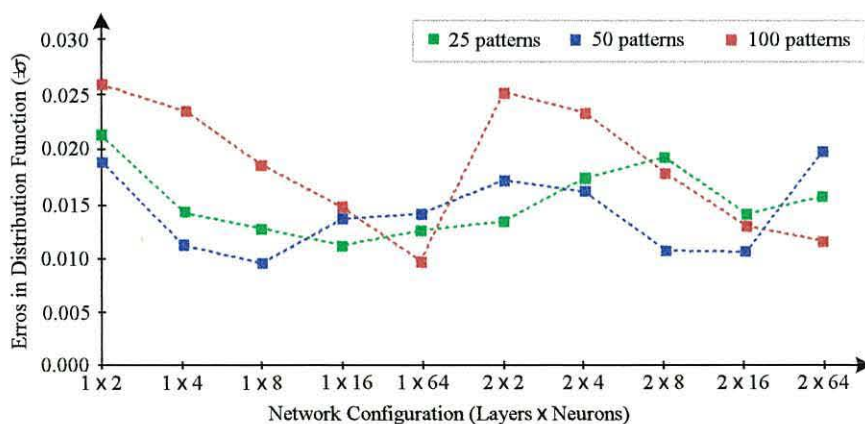


Figure 3.12: Comparison of the distribution function errors with various sample points

From Figure 3.12, it can be concluded that a training set with 50 x and y values is sufficient to provide a satisfactory generalisation, as there is no significant improvement in performance by increasing the number of data points. A small network is observed to give a large error, as there are insufficient neurons in the hidden layer to extract important features from the curve. There is a small increase in the error as the size of the network increases, which is due to the lack of information in the training set (as defined in equation (3.14)) compared to the number of weights in the network. If the size of the training set is increased to compensate for the extra weights in the network, then the error is observed to decrease, but at the expense of the time required to train the network.

Since the linearly sampled x values are identical in each example, it is feasible to remove them from the training set without any loss in accuracy, thus presenting the network with the sampled y values only (hence only 50 input neurons are required).

3.7.5 Stopping Criteria and Choice of Learning Algorithm

The back-propagation algorithm cannot, in general, be shown to converge, nor are there well-defined criteria for stopping its operation. However, there are some reasonable criteria which can be used to terminate the weight adjustments in the network. Kramer and Sangiovanni-Vincentelli [42] suggested a stopping criteria:

- The back-propagation algorithm is considered to have converged when the Euclidean norm of the gradient vector reaches a sufficiently small gradient threshold.
- The back-propagation algorithm is considered to have converged when the absolute rate of change in the average squared error per epoch is sufficiently small (i.e. about 0.1-1% per epoch).

A neural network which has been designed to generalise well will result in a correct input-output mapping (even without having been trained with a particular pattern). However, when a neural network learns too many input-output relations, the network memorises the training data and is less able to generalise between patterns.

The mean-squared error curve can indicate that the network has been trained sufficiently, whilst the validation error (the error calculated over the validation set) can indicate when a network is being overtrained. Thus, an increase in the validation error is a clear indication that the network is being overtrained, and becoming too familiar with the training set, which leads to poor generalisation. It is thus favourable to stop training before this point is reached. This behaviour was observed in our study, as shown in Figure 3.13.

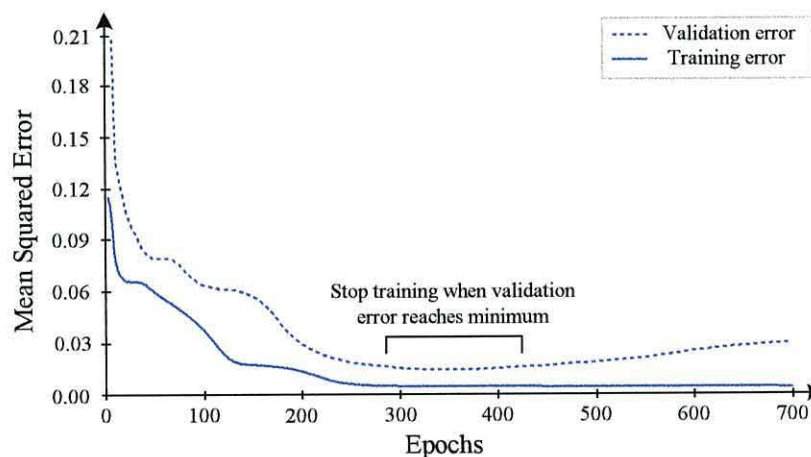


Figure 3.13: *Illustration of overtraining a network*

A mean-squared error of less than 0.01 (<10% of initial error) as shown by Figure 3.13 was considered sufficiently accurate for the applications considered here, since this is of the order of the error in experimental data.

So far the use of the back-propagation method has been satisfactory, particularly for demonstration purposes and initial testing. However, it does have its limitations, which is mainly the vast number of epochs required to train a network. This prompted

the possibility of trying a different learning algorithm, which could train quicker than the back-propagation method, but still provide accurate results.

The conjugate-gradient learning algorithm consists of choosing ‘conjugate directions’ of minimisation that do not interfere with each other, and performing a line minimisation along that direction. The conjugate-gradient method will always locate the minimum of any quadratic function of N variables in at most N steps (see Haykin [28]). Kramer and Sangiovanni-Vincentelli [42] studied the use of the conjugate-gradient method for the supervised training of multi-layer perceptrons. They showed that the conjugate-gradient method required fewer epochs (the number of times that a pattern is presented to the network) than other methods, such as the standard backpropagation method. The disadvantage being that the conjugate-gradient method proved to be computationally more complex. This was also observed during our simulations, with training reduced by a factor of 10; typically:

- Back-propagation method: 400-500 epochs
- Conjugate-gradient method: 50-70 epochs

3.7.6 Optimal Network Size and Parameters

The first part of the simulation involves the determination of the optimal values for the learning-rate parameter η and the momentum constant α (see Section 2.4.1.1), which are set according to any one of three definitions (Haykin [28]):

1. The η and α that, on average, yield convergence to a local minimum in the error surface of the network with the least number of epochs.
2. The η and α that, for either the worst-case or on average, yield convergence to the global minimum in the error surface with the least number of epochs.

- The η and α that, on average, yield convergence to the network configuration that has the best generalisation, over the entire input space, with the least number of epochs.

Table 3.1 shows the parameters which are of importance, and the typical ranges to which they should be set.

Parameter	Symbol	Typical Range
No. of hidden neurons	M	$2 \rightarrow \infty$
Learning-rate parameter	η	$0 \rightarrow 1$
Momentum constant	α	$0 \rightarrow 1$

Table 3.1: *Variable parameters for use with the multi-layer perceptron*

Simulations were carried out on a network (with 8 hidden neurons) to observe the effect of changing the learning-rate and momentum. It should be noted however, that a lower mean-squared error (which has been obtained quickly) does not necessarily imply a good generalisation. Figures 3.14 and 3.15 show the results using the back-propagation and conjugate-gradient methods respectively.

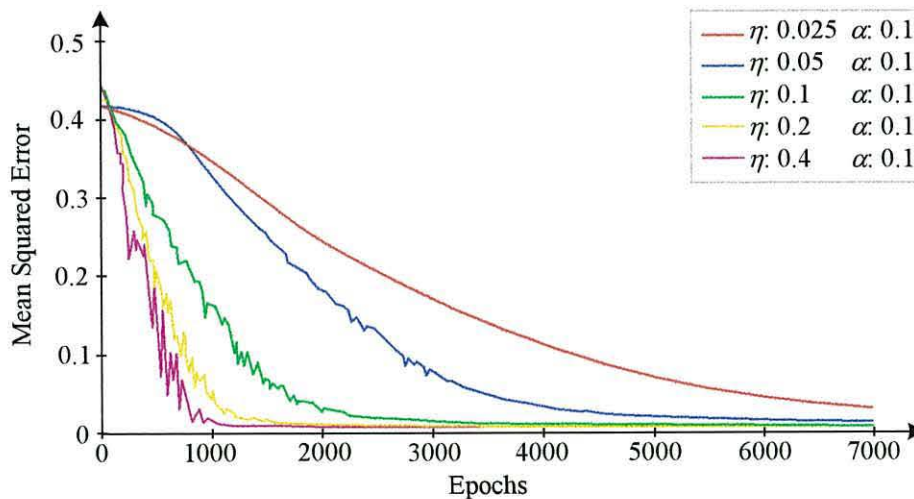


Figure 3.14: *Mean-squared error for various η and α using back-propagation learning*

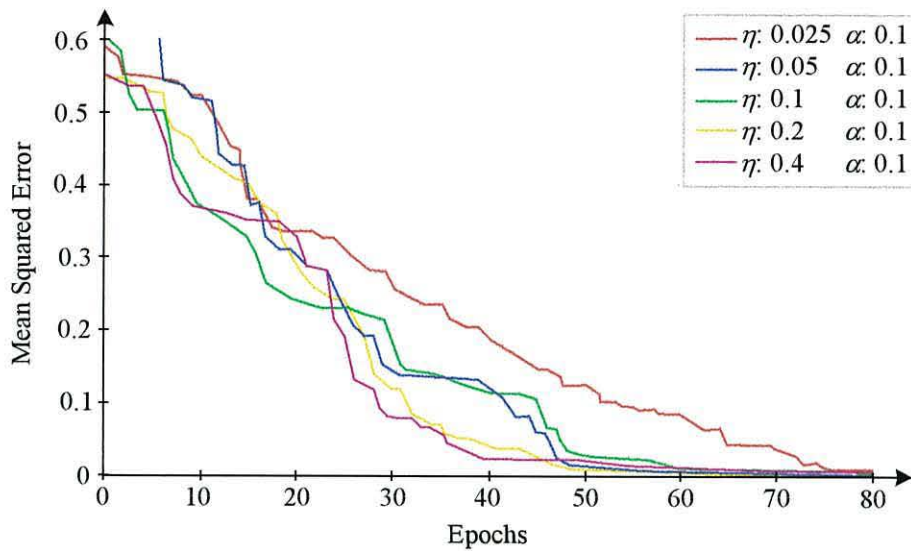


Figure 3.15: Mean-squared error for various η and α using conjugate-gradient learning

From the Figures 3.14 and 3.15, it is clear that the conjugate-gradient method takes fewer epochs. As the value of the learning-rate parameter has less effect on the time required, it is decided to choose a mid-range value of about 0.25 for η .

Simulations were also conducted with various momentum values, but it was observed that increasing it made the system rather unstable, with the effect that the network gave a poor generalisation, and could not be trained every time. It was therefore decided to fix the momentum constant at a nominal value of 0.1.

Having settled on values for the network parameters, the second part of the simulation considers the optimal number of neurons in the hidden layer. For practical reasons, it is recommended to begin with the smallest number of neurons possible. Consequently the classification error was studied as a function of the network configuration. Figure 3.16 shows the error in the distribution function for three test sets (with a network trained using the chosen parameters with a log-normal distribution).

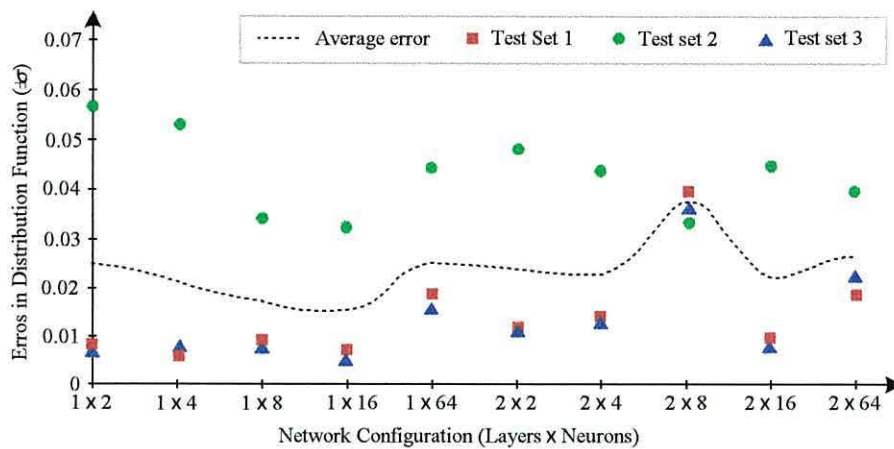


Figure 3.16: Comparison of the distribution function errors with 3 test sets

From Figure 3.16 it can be seen that the average error is lowest at 8-16 hidden neurons in 1 hidden layer. It can therefore be deduced that a configuration of 8 hidden neurons in 1 hidden layer is sufficient to minimise the error with a reasonably small network. These values are summarised below in Table 3.2.

Parameter	Value
No. of hidden neurons	8-16
Learning-rate parameter	0.25
Momentum constant	0.1
No. of training sets	100
Input neurons (data points)	50
Learning algorithm	Conjugate Gradient

Table 3.2: Variable parameters for use with the multi-layer perceptron

It must be stressed that these parameters are a specific result for our study; in general each application will require a similar analysis, since other problems may give rise to a stronger dependence on the performance of the network on its detailed configuration.

3.8 Results

3.8.1 Using Theoretical Data

The results obtained so far have demonstrated that a network can be trained to recognise a log-normal distribution function from χ_t data and to determine with high accuracy the standard deviation. However, one parameter which has been overlooked so far is the type of distribution which is applied to the data. The sensitivity of the χ_t data to the type of distribution function is investigated (as well as the degree of distribution).

The approach used was to generate a training set calculated using log-normal and Gaussian distributions. The curves were generated in a similar manner to the previous curves: 100 curves were generated with a log-normal distribution and 100 with a Gaussian distribution. The type of distribution was also encoded into the output patterns, hence training the neural network with the distribution function value and distribution type. Once trained, the value and type of distribution is expected at the output when a test set is applied to the network. This is illustrated in Figure 3.17.

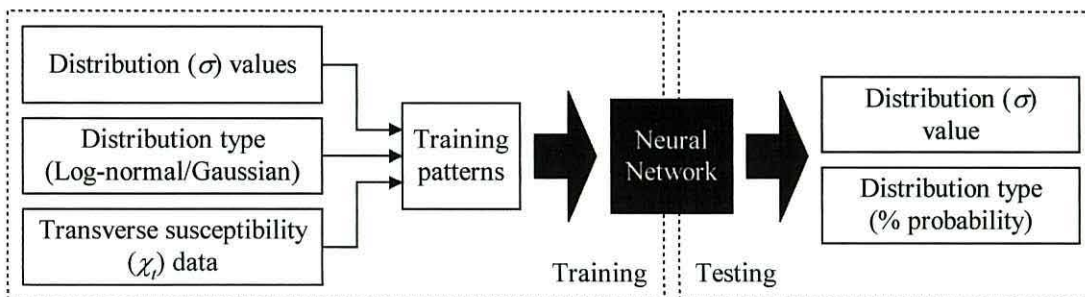


Figure 3.17: *The determination of the value and type of distribution using 1 network*

The errors in recognition of the type of distribution function and the value of the standard deviations were then determined for the test set of data. The results are presented in Figure 3.18, from which it can be observed that the neural network has determined the distribution function value to within ± 0.075 in all cases, and has determined the type of distribution to a minimum certainty of 95% in all cases.

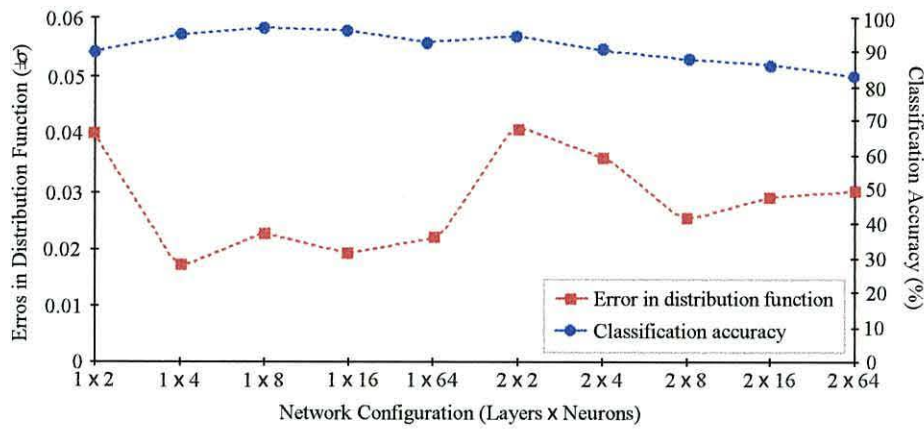


Figure 3.18: *Errors in the distribution function and classification accuracy*

It can be observed that this method is not quite as accurate as the method using only one type of distribution, as more information is being dealt with, and consequently more weights in the network.

However, if the network is divided into two separate networks (one for determining the distribution value, and one for determining the type of distribution), then the accuracy improves considerably. This set-up is illustrated in Figure 3.19.

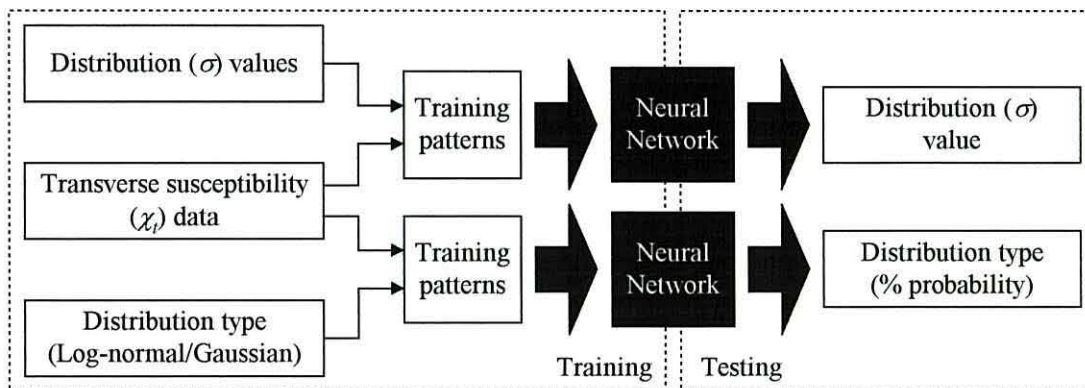


Figure 3.19: *The determination of the value and type of distribution using 2 networks*

Here, the type of distribution is encoded into one training set, whilst the value of the distribution is encoded into another training set (both have identical χ_t data).

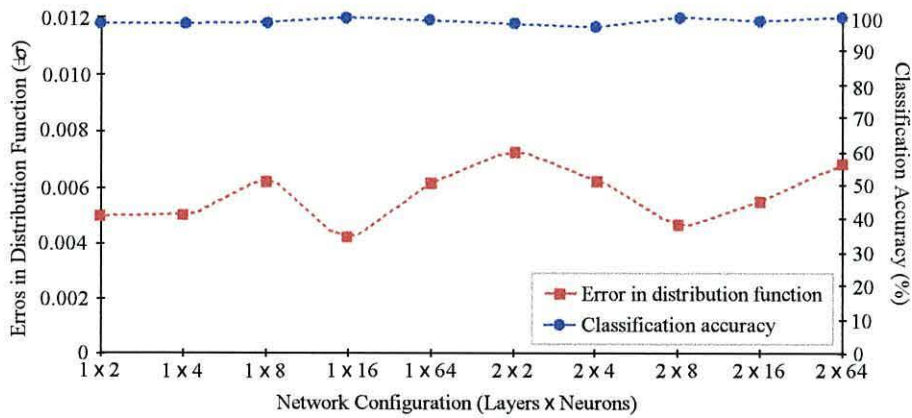


Figure 3.20: Errors in the distribution function and classification accuracy

It was observed that the neural networks could determine the value of the distribution function to within ± 0.006 in all cases, and determine the type of distribution to a minimum certainty of 98.5% in all cases (the % value is the percentage probability of the distribution either being Gaussian or log-normal), as seen in Figure 3.20.

This increase in accuracy is attributable to the division of the one network into two separate networks, thus considerably reducing the number of weights required. The configuration of the networks are summarised in Table 3.3.

Parameter	Network 1 (distribution value)	Network 2 (distribution type)
Input neurons (data points)	50	50
Hidden neurons	16	8-16
Output neurons	1	1
No. of training sets	100	100
No. of epochs required	40-50	35-40

Table 3.3: Summary of the network parameters

3.8.2 Using Experimental Data

Having determined a network architecture which yields optimum results when presented with theoretical data, the goal is to apply ‘real’ experimental data to the network. Here the results obtained when transverse susceptibility data of a CrO_2 sample was used is outlined, initially on a network trained with log-normal distributions only, and then on a network trained with log-normal and Gaussian distributions. Figure 3.21 shows the results of applying an experimental data set to the trained network (log-normal distributions only).

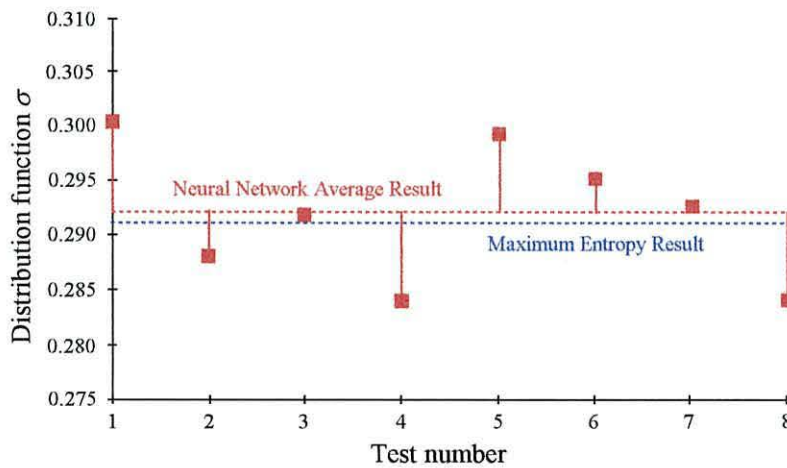


Figure 3.21: *Plot of σ values for experimental test runs*

From Figure 3.21 and other numerical data, it can be observed that the network was able to determine the distribution function to a mean value of 0.292 with a standard error of 6×10^{-3} . Thus, the network is shown to train reproducibly and to give consistent estimates of the value of σ . The inverse problem for this data set has also been solved using maximum entropy techniques [43], from which the distribution function of the experimental data was found to be 0.291. The agreement between the two techniques gives enhanced confidence in the solutions obtained, and in the use of the χ_b for the determination of H_K distribution functions.

When an experimental data set was presented to an optimised neural network, which was trained with log-normal and Gaussian distributions (and retrained after each test

run using identical parameters), the network was also able to determine the most likely type of distribution associated with the sample (as illustrated in Figure 3.22).

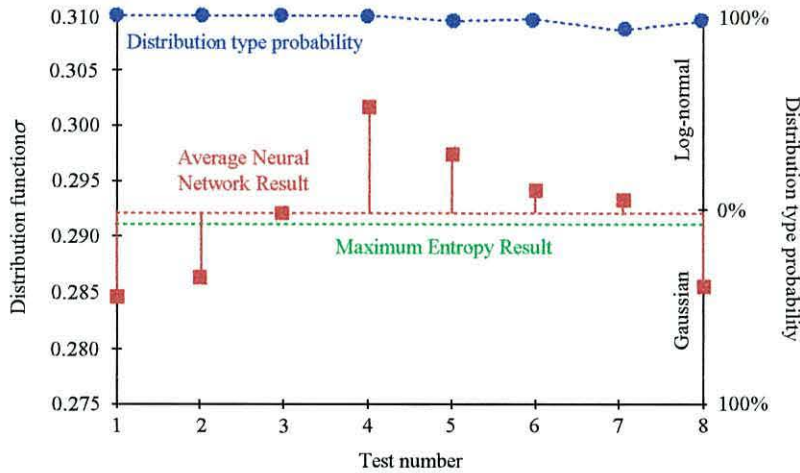


Figure 3.22: Plot of σ values and distribution type for experimental test runs

This was determined with a mean value of 0.292 with a standard error of 6×10^{-3} . The network determined with a 99.5% certainty that the experimental H_K distribution was of the log-normal form.

It can be observed that the application of experimental data produced results which were not as accurate when contrasted to the application of theoretical data sets. It should be recalled that the kernel used in the theoretical calculations is essentially that derived by Aharoni *et al* [31] and which is based on the assumption of the Stoner-Wohlfarth coherent mode of magnetisation reversal. It has been shown [44] that the reversible transverse susceptibility (RTS) falls into a class of ‘stiffness’ measurement based on magnetisation rotation. Here the Stoner-Wohlfarth model is a good approximation as long as the magnetisation remains close to saturation, which is normally the case when determining H_K values of 2kOe or greater.

However, the current work presents the neural network with experimental data extending to small fields, where it might be expected that non-uniform magnetisation states could develop, with an as yet theoretically unknown effect on the RTS. A

detailed comparison of the experimental data with the predicted kernel showed a systematic discrepancy in small fields, which is interpreted intuitively as arising from non-uniform magnetisation modes. As a result, the effects of removing the low-field data was investigated and also the high field data where the effects of the H_K distribution might be expected to be less pronounced. This essentially concentrates the data around the peak, in the area containing the H_K information.

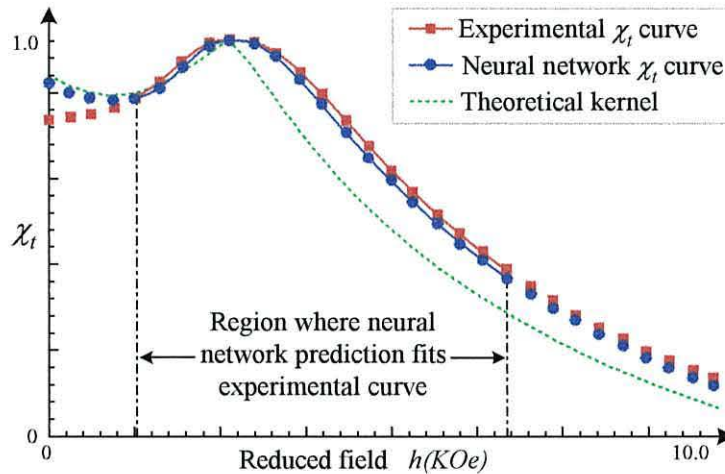


Figure 3.23: Comparison of experimental curve with the neural network prediction

Figure 3.23 compares the χ_t curve predicted by the neural network with the actual experimental χ_t curve which was used to test the network, and shows the region at which the predicted curve fits the experimental curve.

3.9 Summary

In this chapter it has been demonstrated that a neural network approach can be used to solve an inverse magnetic problem to approximately the same degree of accuracy as other conventional methods, and accomplished instantaneously once the network has been trained, as opposed to several hours using the simulated annealing method.

Using a multi-layer perceptron neural network, which was chosen due to the possibility of creating numerous sample data sets, theoretically generated χ_t data were applied in order to select the optimal settings. It was discovered that the preparation of the training data was an important part of the study; partially due to failures in the model at low fields and redundancy of data at higher fields.

It was observed that a neural network with about 8-16 hidden neurons in 1 hidden layer was sufficient, using 100 training sets, to be successfully trained in approximately 40-50 epochs.

The accuracy obtained using theoretical data gave much confidence in the application of experimental data to a trained network. Results showed that the value of the distribution could be obtained repeatedly to within ± 0.01 of that obtained with the maximum entropy approach.

However, tests on theoretical data with distributions of $\sigma < 0.1$ and $\sigma > 0.65$ proved to be less accurate (± 0.05), which is believed to be mainly due to the deficiency of data points around the H_K peak for $\sigma < 0.1$, and due to the deficiency of data points in the high field for $\sigma > 0.65$. It can therefore be concluded that a neural network approach is feasible to determine the distribution of H_K when $0.1 < \sigma < 0.65$, at which point the accuracy is comparable to conventional methods. This range is considered to be quite reasonable, as it includes most of the values encountered experimentally.

Chapter 4

Analysis of Compact Disc Pit Geometries Using Optical Diffraction Methods

4.1 Introduction

The compact disc was developed by Philips and Sony in the early 80s to meet the demands for storing high quality audio. The storage capacity required to fulfil this application was so high that the compact disc was soon to find itself being used in many non-audio applications where mass data storage was a requirement [45]. As a result, several variations of the original format were developed, such as CD-R, CD-RW, CD-I etc.

The compact disc has several advantages over the old LP (not only in terms of the audio quality) such as size and durability. Because data is read by a focused laser beam, there is no degradation from repeated playings, and the effect of dust and surface damage is minimised. Digital signal processing permits the use of error correction and other techniques to further improve the reliability of the stored data. The density of information on a compact disc is about 100 times greater than that of an LP. It has about 3×10^9 pits crammed onto its surface, which makes it vulnerable to defects and scratches.

However, the data is recorded with error correction codes which permit the detection and correction of errors. The specifications for the compact disc system were jointly developed by Philips and Sony and are defined in the standards document known as the *Red Book* (IEC Standard BNN15-83-095) [46]. All disc and player manufacturers must obtain a compact disc license to utilise these specifications. Disc dimensions are defined in the standard, as are pit dimensions, the physical formations of which encode data.

The disc itself is generally made from a polycarbonate plastic, although any transparent material with a refraction index of 1.55 may be used. Some of the parameters are listed in Table 4.1. If any of these parameters are inaccurate (especially those in the μm range), due to manufacturing defects, then reproduction can be compromised.

Parameter	Value
Playing time	74 minutes, 33 seconds
Laser wavelength	780nm
Rotational speed	1.2 - 1.4ms ⁻¹
Diameter	120mm
Thickness	1.2mm
Track pitch	1.6 μm
Minimum pit length	0.833 μm (1.2ms ⁻¹) to 0.972 μm (1.4ms ⁻¹)
Maximum pit length	3.05 μm (1.2ms ⁻¹) to 3.56 μm (1.4ms ⁻¹)
Pit depth	$\approx 0.11\mu\text{m}$
Pit width	$\approx 0.5\mu\text{m}$

Table 4.1: *The compact disc specification (Red Book)*

The physical specifications of the compact disc are shown in Figure 4.1. The innermost part of the disc does not contain any data (it simply provides somewhere for the player to firmly hold the disc). The data is recorded on a 35.5mm wide section; with a lead-in area which contains non-audio data used to control the player.

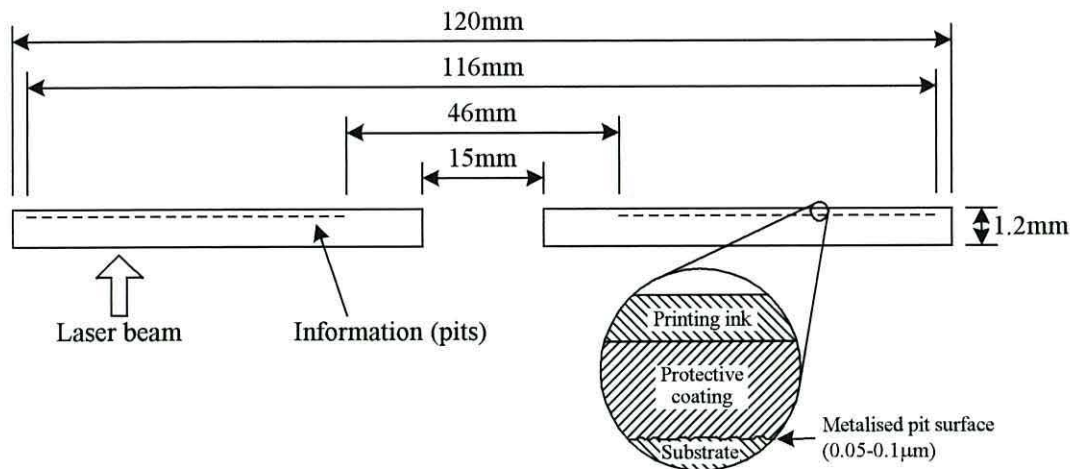


Figure 4.1: *Physical specifications and construction of a compact disc*

A transparent plastic surface substrate forms the majority of the disc's 1.2mm thickness. The data consists of a series of pits, which are impressed along its top surface and are covered with a very thin (50-100nm) coating of a metal, such as aluminium. Another thin (10-30µm) plastic layer protects the metalised pit surface, on top of which the label is printed. A laser beam is used to read the data, and is applied from below the disc and passes through the transparent substrate and back again. The beam is focused on the metalised data surface inside the disc.

From Figure 4.2 it can be observed that the pit dimensions are of a micrometre scale. In order to measure the geometries of the pits, it is necessary to use measurement techniques which are capable of analysing such small structures [47]. However, it must also be remembered that the total data recording surface is quite large (about 124cm²), which makes the analysis of the whole disc a very laborious task.

Information is stored digitally on the surface of the disc, in the form of pits and lands, which have varying lengths (Figure 4.2). A disc has just one track of pits arranged in a continuous spiral running from the inner circumference to the outer. It is advantageous to start the data on the inside and work out, as the outer part of the disc is prone to manufacturing defects. The quicker rotation speed at the outer part of the disc means that the data has to read quicker in this area (thus increasing the probability of errors).

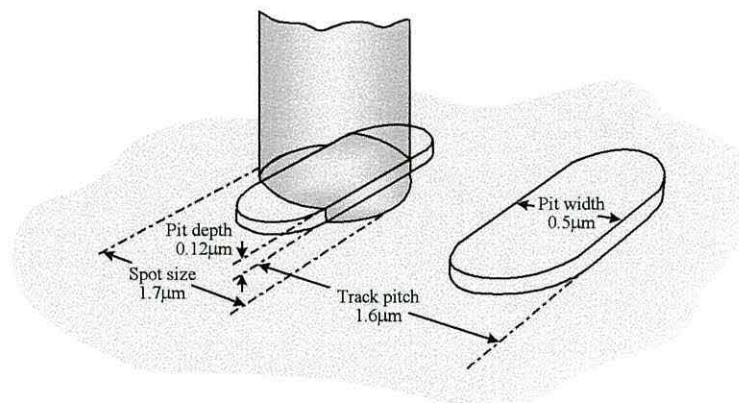


Figure 4.2: *The pit and lands on a compact disc*

A number of measurement techniques are currently in use, such as optical, atomic force microscopy, stylus contact etc. [48], each with its specific advantages and disadvantages. Whilst a scanning electron microscope can measure pit geometries to a high degree of accuracy, it can only do this for areas of less than $50\mu\text{m}^2$ at any one time [49]. Therefore, to cover the whole recording area of a compact disc, would make this method unsuitable for an on-line measurement technique. Measurements can only be made prior to the application of the protective layer of lacquer, as the tip needs to be in relatively close contact with the surface of the disc.

An SEM image can be a very useful tool for comparing the accuracy of other methods. The image in Figure 4.3 shows a small section of a CD surface obtained with an SEM.

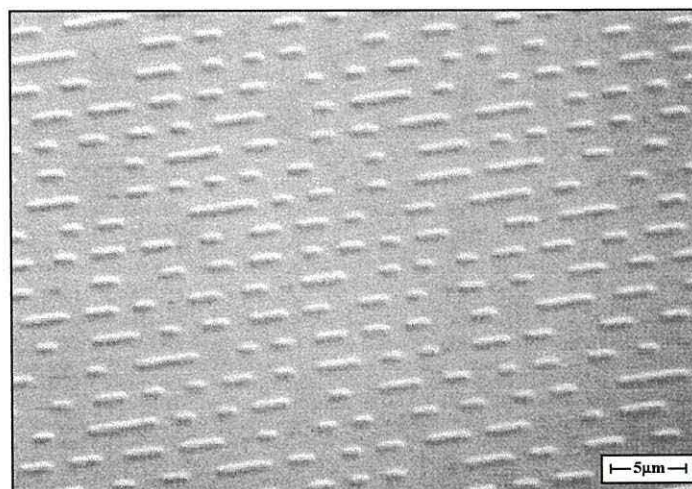


Figure 4.3: *Compact disc pit surface from an SEM image*

However, there is a technique which satisfies most of the requirements necessary for on-line measuring. It is a non-destructive method (on both the macroscopic and microscopic scale), covers a greater area than an electron microscope, and is generally quicker. The technique of optical diffraction detects the intensity distribution of light diffracted by the 3D pit structure into diffraction orders. It should be noted that the pit dimensions are not measured in a direct way; interpretation of the data is required to convert the intensity measurements into pit dimensions (the inverse problem).

This chapter introduces the theory of diffraction, in particular its application to optical discs, discusses various methods of measuring the diffracted light, and explains how neural network techniques may be used to analyse these intensity distributions from the diffraction measurements, thus solving the inverse problem, and determining the relevant pit geometries.

4.2 Diffraction Theory

Diffraction is the phenomenon which occurs when a single slit is placed on a barrier, and from the slit, a new wave front is radiated as though it came from a point source. A diffraction grating contains a series of slits. As a result of interference, a wavefront will leave the grating only in directions where light from all the slits are in phase [50]. This occurs straight ahead and at a variety of other angles. The angle of the 1st order wave is a function of the wavelength of the light and the spacing of the slits (equation (4.1)).

$$\sin(\theta_m) = \sin(\theta_i) + \frac{m\lambda}{p} \quad (4.1)$$

where

m = order (0, 1, 2, ...)

p = period of grating

θ_i = angle of incidence

θ_m = angle of diffraction

In practice, diffraction also occurs from the objective lens system, due to the lens acting as a circular aperture source, which produces Fraunhofer diffraction rings around the central light spot. These rings are of importance and are referred to as *Airy rings*, and the pattern they produce is referred to as the *Airy pattern*, as illustrated in Figure 4.4.

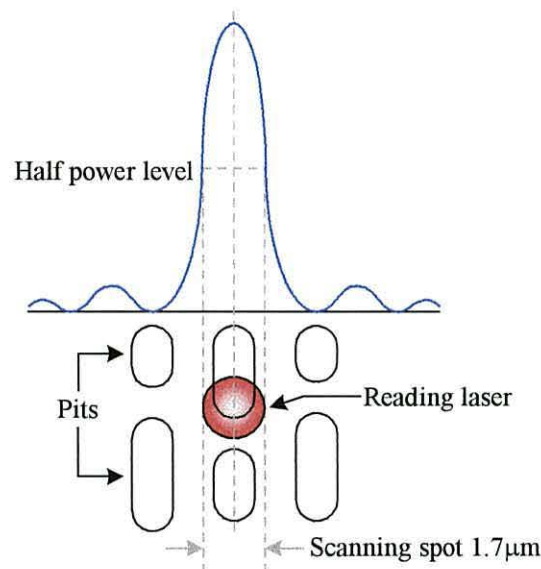


Figure 4.4: *The Airy pattern obtained when reading from a compact disc*

From Figure 4.4 it can be seen that the intensity minimum falls approximately on adjacent pit tracks, so as to avoid interference during the reading process. The form of the Airy pattern is a function of the light wavelength and the numerical aperture of the lens.

The diameter of the laser beam has been specifically selected so that it is equal to the width of the half-intensity of the Airy pattern (this is illustrated in Figure 4.4). The first dark ring intentionally falls on the adjacent track, and provides us with good crosstalk performance. The track pitch is therefore determined by the degree of crosstalk which can be tolerated (taking into consideration the effect of a slightly defocused beam with a slight tracking error). Once the relation between the geometry and order intensities has been established theoretically, the measurement of these intensities gives information on the 3D geometry (and the solution to the inverse problem).

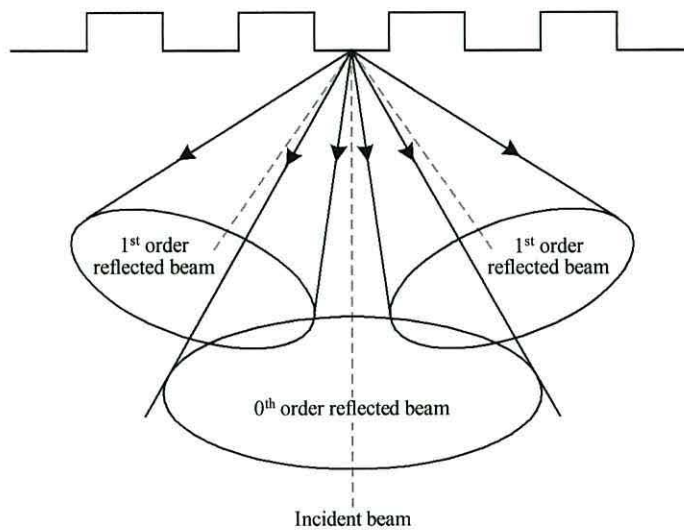


Figure 4.5: *The CD pit surface acting as a diffraction grating*

The surface of a compact disc can be thought of as a reflection grating, which acts similar to a diffraction grating. Thus, the pits on the CD also cause interference to occur. The smaller the pits in relation to the wavelength of the light, the greater is the angle at which light is reflected. The light diffracted by the grating consists of a single zero-order and multiple first-order beams, as shown in Figure 4.5.

4.2.1 Theoretical Measurement Techniques

There are two primary mathematical diffraction models which can be applied to the problem, which differ in complexity. The vectorial diffraction theory should be used for this problem, as the pit dimensions are of the same order as the wavelength as the laser light. However, as the vectorial polarisation effects are very small, it is advantageous to use the much simpler scalar diffraction model, where results do not differ very significantly from those given by the vectorial model (the reader is referred to *Pasman* [51] for a more thorough explanation of the vectorial diffraction theory).

By applying scalar diffraction theory [52] to the compact disc surface (which acts as a diffraction grating), the far-field diffraction pattern F_r can be plotted with respect to

pit geometries, given by equation (4.2), and derived from first principles by *Sekizawa et al* [53].

$$F_r(\xi) = (P - W) \operatorname{sinc}[(P - W)\xi] + \left\{ W \operatorname{sinc}(W\xi) \exp\left[-j\pi\left(\frac{4D}{\lambda} + P\xi\right)\right] \times \frac{\sin\left[2\pi\left(N + \frac{1}{2}\right)P\xi\right]}{\sin(\pi P\xi)} \right\} \quad (4.2)$$

where

$F_r(\xi)$ = far field diffraction pattern

ξ = spatial frequency axis

P = track pitch

W = pit width

D = pit depth

λ = wavelength of laser (780nm)

Thus by inserting $\xi = n/P$ into equation (4.2), the intensities of the zeroth can be derived, first and second order beams, given by equations (4.3), (4.4) and (4.5) respectively:

$$I_0 = 1 - 2 \frac{W}{P} \left(1 - \frac{W}{P}\right) \left[1 - \cos\left(\frac{4\pi D}{\lambda}\right)\right] \quad (4.3)$$

$$I_1 = 2 \frac{W^2}{P^2} \operatorname{sinc}^2\left(\frac{W}{P}\right) \left[1 - \cos\left(\frac{4\pi D}{\lambda}\right)\right] \quad (4.4)$$

$$I_2 = 2 \frac{W^2}{P^2} \operatorname{sinc}^2\left(\frac{2W}{P}\right) \left[1 - \cos\left(\frac{4\pi D}{\lambda}\right)\right] \quad (4.5)$$

From the above equations the values for the track pitch (P), pit width (W) and the pit depth (D) can be derived. If θ denotes the angle subtended by the zeroth beam and the first diffracted light, the track pitch is given by:

$$P = \frac{\lambda}{\sin(\theta)} \quad (4.6)$$

The pit width and depth are calculated using equations (4.7) and (4.8) respectively:

$$W = \frac{P}{n\pi} \cos^{-1} \left(\sqrt{\frac{I_{2n}}{I_n}} \right) \quad (4.7)$$

$$D = \frac{\lambda}{4\pi} \cos^{-1} \left\{ 1 - \frac{P^2}{2W \left[\left(\frac{I_0}{I_1} \right) W \operatorname{sinc}^2 \left(\frac{nW}{P} \right) + P - W \right]} \right\} \quad (4.8)$$

From the above relations, the track and pit parameters are determined only from the observable quantities (I_0 , I_n , I_{2n}). These relations assume that the incident beam impinges on the disc perpendicularly. However, if the incident beam impinges on the disc at an angle θ_0 then equations (4.6) and (4.8) become:

$$P = \frac{\lambda}{\cos(\theta_0) \sin(\theta)} \quad (4.9)$$

$$D = \frac{\lambda}{4\pi} \cos(\theta_0) \cos^{-1} \left\{ 1 - \frac{P^2}{2W \left[\left(\frac{I_0}{I_1} \right) W \operatorname{sinc}^2 \left(\frac{nW}{P} \right) + P - W \right]} \right\} \quad (4.10)$$

In our investigations, it is assumed for the time being that the laser beam strikes the disc perpendicularly (and can therefore ignore equations (4.9) and (4.10)). Equation (4.7) shows that the pit width normalised by the track pitch (W/P) is calculated by the observable intensity ratio (I_{2n}/I_n).

However (W/P) is a single-value function only when $n = 1$ at a typical parameter range ($P = 2\mu\text{m}$, $W = \text{between } 0.3 \text{ and } 1.0\mu\text{m}$), as shown in Figure 4.6 (where the measuring order ($n = 1, 2$) is taken as a parameter). Therefore, the width must be found by (I_2/I_1).

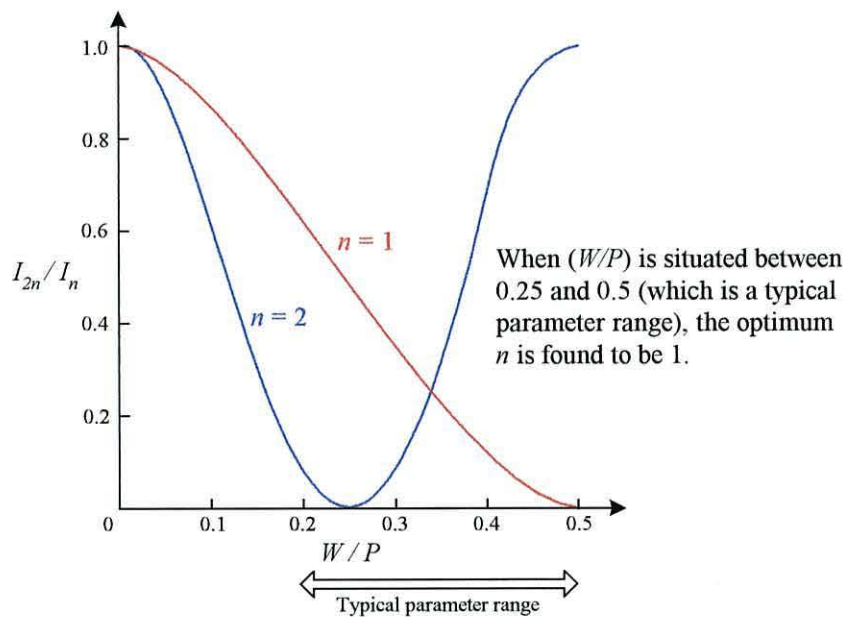


Figure 4.6: The relationship between the pit width normalised by the track pitch and the diffracted light intensity ratio (I_{2n}/I_n)

Taking these conditions into consideration, the above equations can be used to derive an equation for the diffracted intensity distribution, given by equation (4.11):

$$\begin{aligned}
I = \frac{1}{P^2} & \left\{ (P - W) \operatorname{sinc}[(P - W)\xi] + \left[W \operatorname{sinc}(W\xi) \cdot \frac{\sin\left[2\pi\left(N + \frac{1}{2}\right)P\xi\right]}{\sin(\pi P\xi)} \cdot \cos\left[\pi\left(\frac{4D}{\lambda} + P\xi\right)\right] \right] \right\}^2 \\
& + \frac{1}{P^2} \left\{ W \operatorname{sinc}(W\xi) \cdot \frac{\sin\left[2\pi\left(N + \frac{1}{2}\right)P\xi\right]}{\sin(\pi P\xi)} \cdot \sin\left[\pi\left(\frac{4D}{\lambda} + P\xi\right)\right] \right\}^2 \quad (4.11)
\end{aligned}$$

The $\frac{1}{P^2}$ term (which is dependent on the pitch width) has been added to the equation in order to normalise the intensity to the same order as the values obtained in equations (4.3), (4.4) and (4.5). The ξ term (spatial frequency axis) is defined as:

$$\xi = \frac{\sin(\theta)}{\lambda} \quad (4.12)$$

These equations can be used to plot the intensity distribution against the angle of diffraction with a variety of pit geometries, as illustrated in Figure 4.7. However, some parameters in equation (4.11) can be changed, depending on the set-up of the experimental apparatus.

In general, the intensities of the spectra are altered by the width and depth of the pits, and the positions of the spectra are altered by the track pitch. It should also be noted that the positions of the spectra are not affected by multiple reflections within the disc substrate [54].

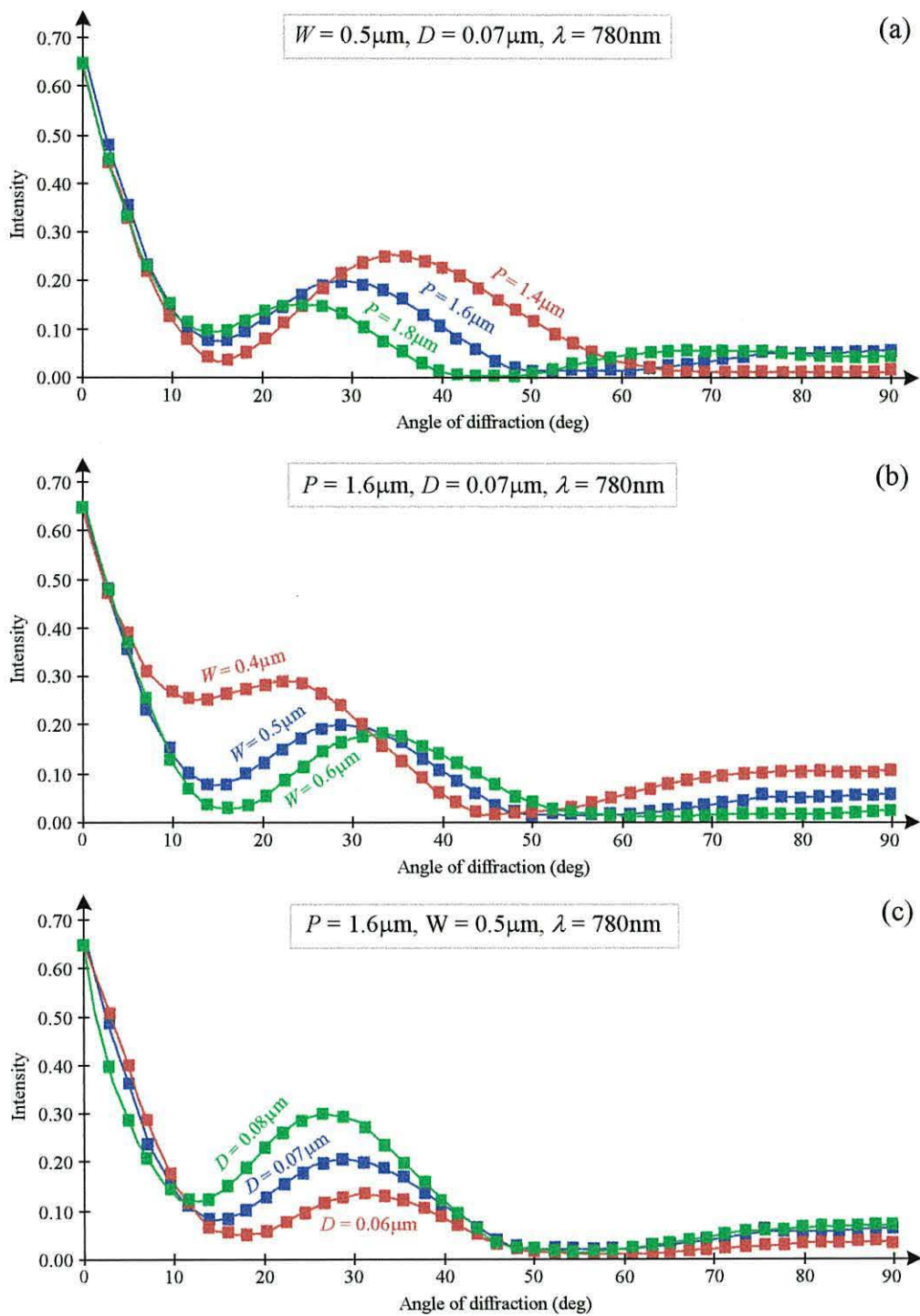


Figure 4.7: The intensity distribution at various diffraction angles (a) variable track pitch (b) variable pit width (c) variable pit depth

4.3 Experimental Measurement Techniques

The pit and track geometries on a compact disc can be measured experimentally (using optical methods) in a variety of ways which are briefly discussed here.

4.3.1 Three Detector Arrangement

This method involves the setting up of three separate detectors (one for each diffracted beam), which are placed at different angles to measure the intensity of each beam. The 0th and 1st order detectors are necessary to measure the track pitch. The optional 2nd order detector is necessary to measure the width and depth of the pits. The diagram in Figure 4.8 shows a typical arrangement.

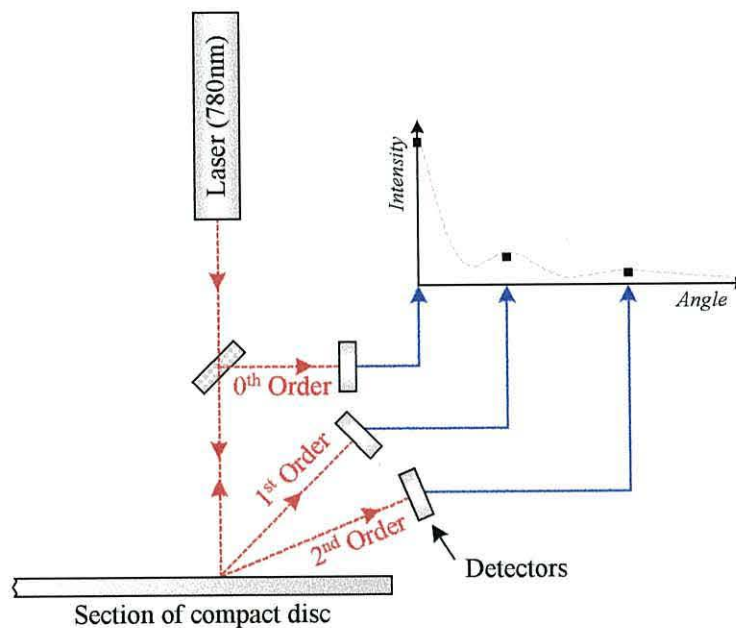


Figure 4.8: Three detector arrangement

The advantage of this arrangement is that it is relatively simple to set-up, requiring only three detectors. However, the diffraction profile is not plotted very accurately.

4.3.2 Multi-element Array Arrangement

The multi-element array uses the same principle as the three detector arrangement, except that in this case the whole intensity distribution can be plotted (see Figure 4.7). The number of points on the plot depends on the number of pixels in the array. Figure 4.9 shows the arrangement of a typical multi-element array.

The multi-element array offers many advantages over the three detector arrangement, as it is capable of plotting the diffraction profile to a higher degree of accuracy. The number of detectors could be varied, but about nine would be sufficient (3 for each beam to be detected).

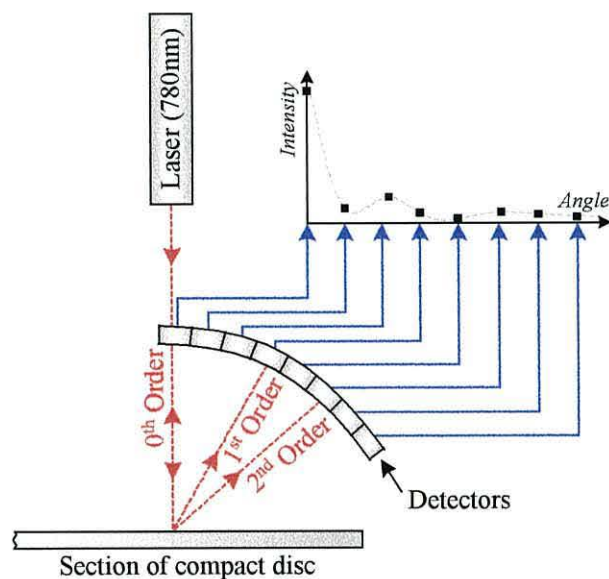


Figure 4.9: Multi-element array arrangement

4.3.3 CCD Camera Arrangement

A CCD camera can be thought of as being a high definition equivalent of the multi-element array, with the advantage that it can measure the intensity on more than one axis (and with greater resolution), as illustrated in Figure 4.10.

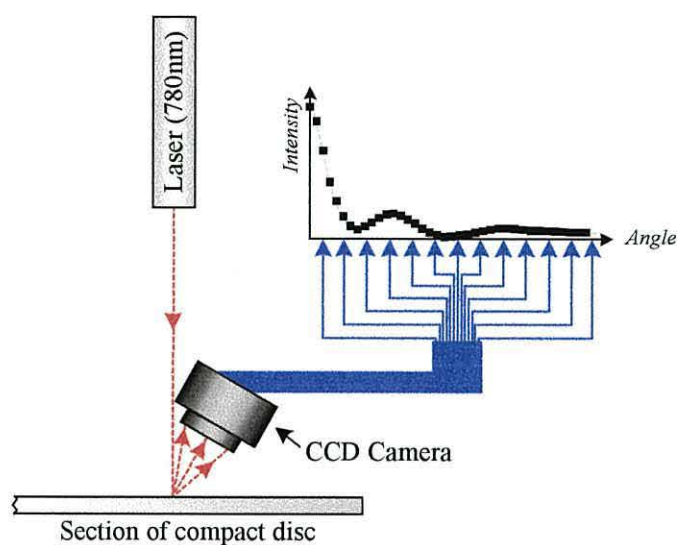


Figure 4.10: *CCD camera arrangement*

The CCD camera arrangement is the most accurate method (its resolution is several thousand pixels), and can plot the diffraction profile very accurately, but at the cost of the processing power required.

4.4 Application of Neural Networks

It is envisaged that the solution to the diffraction problem can be solved substantially quicker by using neural networks. Conventional methods analyse the intensity distribution graph produced by the detectors, then calculate each of the required parameters. As there are millions of pits and tracks which must be analysed to accurately measure the quality of the disc, it would require a substantial amount of time to cover the whole surface. However, a neural network approach would already be trained with a wide variety of these graphs (either theoretically generated, or experimentally obtained), so that the required parameters could be easily determined from the output of the network. This comparison is explained in Figures 4.11 (a) and (b).

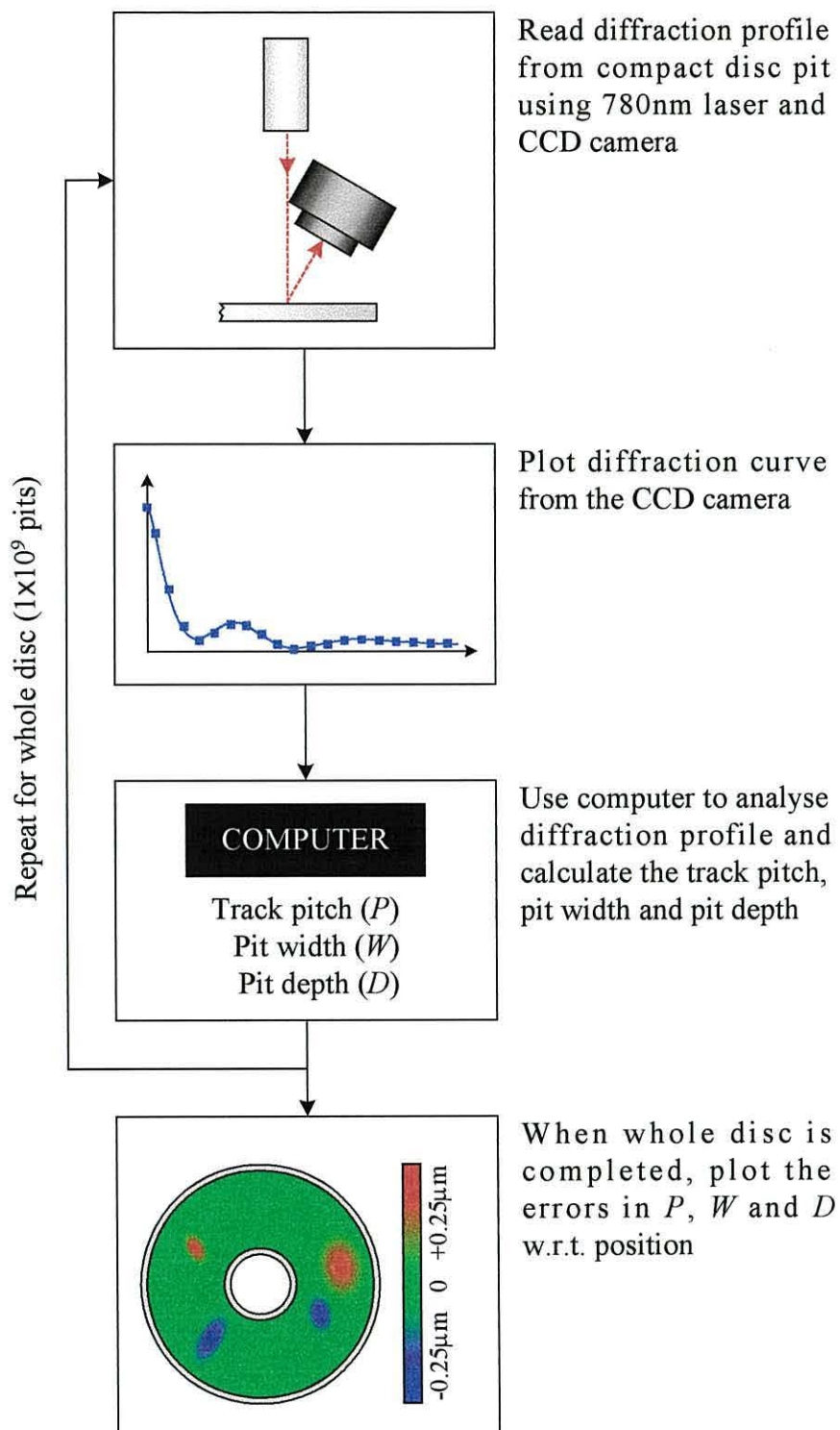


Figure 4.11: (a) Conventional testing method

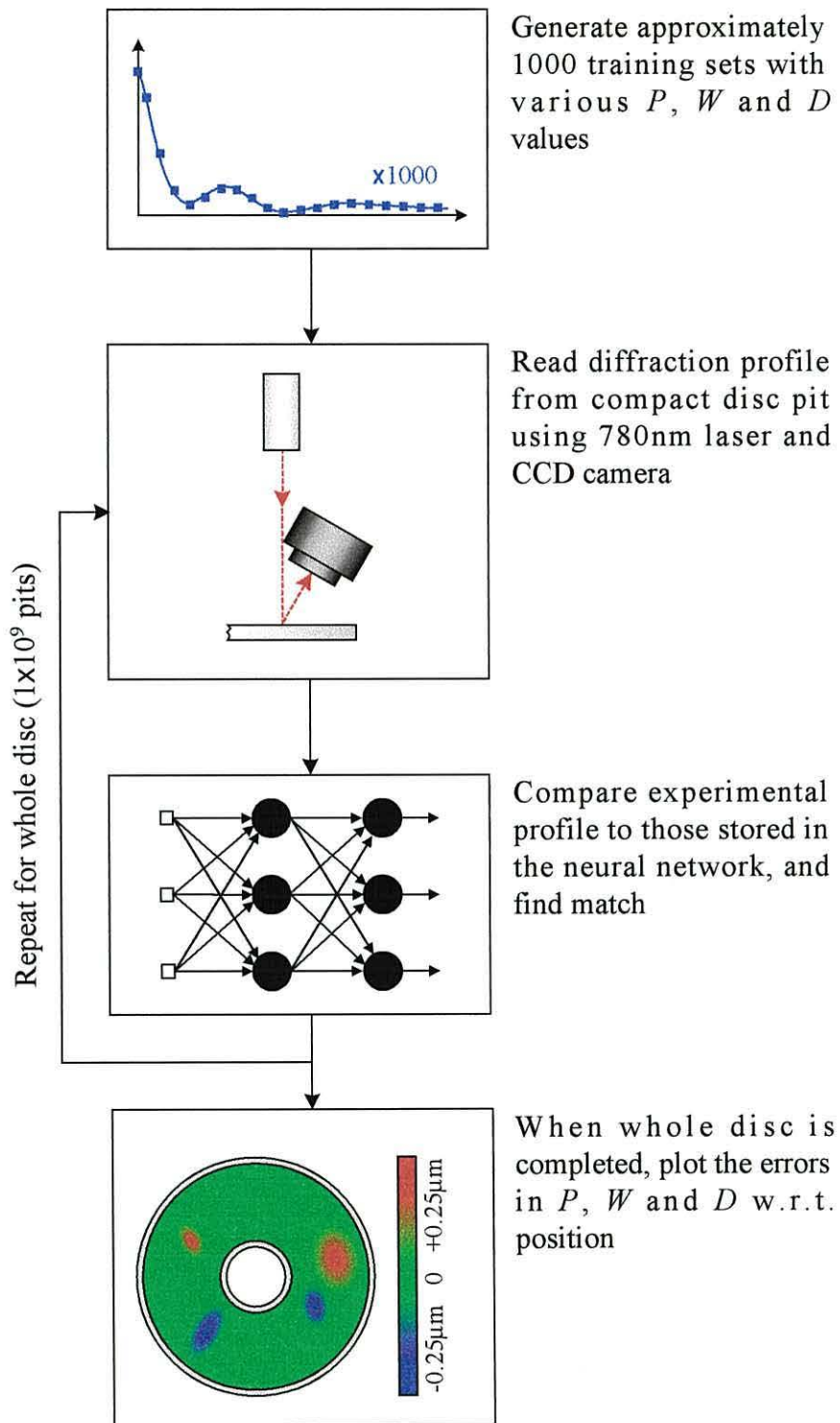


Figure 4.11: (b) Neural network testing method

As it is possible to generate several training sets, using equation (4.11), it is feasible to use a multi-layer perceptron network in a supervised learning manner to solve this

inverse problem. The reader is referred to Section 3.6.2 for details on the multi-layer perceptron.

4.5 Data Preparation and Network Optimisation

4.5.1 Introduction

The importance of data preparation has already been emphasised; However, as it is such an important part of the initial investigation towards successful training, the methods involved will be summarised with regards to this problem.

The multi-layer perceptron is trained using a set of input-output related patterns, with an array of input neurons to represent the y values from the diffraction curve (these are obtained from the outputs of the CCD camera). The outputs of the networks represent the pit width, pit depth and track widths of the disc.

To summarise, the problem in essence is to train the network with a set of theoretically generated data and then investigate its performance with another set of theoretical data. The testing was concluded by applying a set of 'noisy' and experimental data (which was generated theoretically). However, as a wide selection of 'real' experimental data couldn't be obtained, the viability of using such a method had to be considered.

4.5.2 Data Preparation

The data which is obtained in this problem depends on the chosen experimental arrangement. If the three detector arrangement were to be used, then it would be sufficient to chose data points around the angles at which these were set. With a detector array, more data points would be required, and with a CCD camera arrangement, it would be necessary to have as many data points as possible.

In this study, it was decided to concentrate on the CCD camera option, as this is the one which gives the greatest accuracy experimentally, due to the resolution of the diffraction pattern obtained, and is the preferred method which is being developed for use in industry. Figure 4.12 shows a simulated typical diffraction pattern which can be obtained.

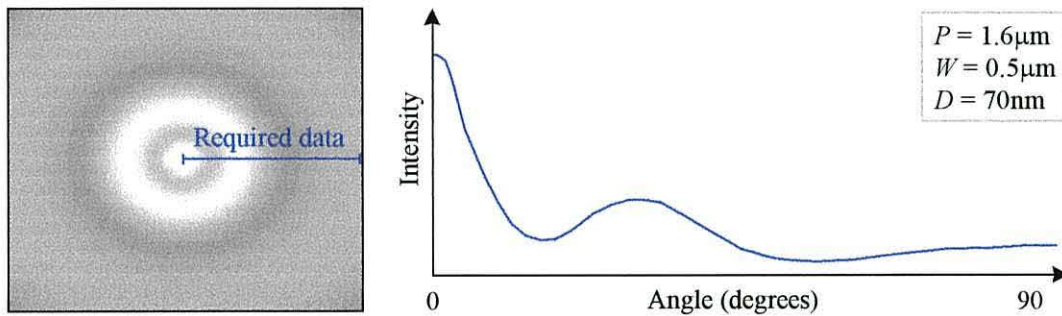


Figure 4.12: *Theoretical diffraction pattern from a CCD camera device*

It is assumed (in the ideal case) that the image obtained is symmetrical around the incident beam, and is feasible to take data samples along a single axis. This has the advantage of considerably reducing the number of data points required.

4.5.3 Input Data Points

Even though a typical CCD camera is capable of a resolution of several hundred pixels, it would be impractical to use so many data values, as they are proportional to the number of neurons which are in the input layer of the neural network.

However, as the diffraction pattern ranges from 0° to 90° , it would seem reasonable to have one data point per 1° , and still be able to describe the curve with a reasonable degree of accuracy (using 91 data points). The number of data points could be reduced further if the number of parameters to be found were reduced (i.e. the track width and pit depth can be found without knowing the intensity of the 2nd order beam; hence angles above about 45° - 50° are not required, as the 1st order beam appears between about 10° - 45°).

4.5.4 Training Set Size

If use is made of the *early stopping* and *cross-validation* techniques again (see Chapter 3 for details), then the number of training examples can be reduced substantially. However, in this case there are three parameters of concern, which have to be determined to a high degree of accuracy (as an inaccuracy of only $\pm 1\mu\text{m}$ can make a significant difference to the read-back from the laser). In Chapter 3, it was observed that about 300 training patterns was sufficient to train the network, which can be used as a rough estimation of the number required for this problem. It was decided to train the network with a set of 100, 500 and 1000 patterns, which consist of various values for the track pitch, pit width and pit depth (other parameters are assumed to be constant).

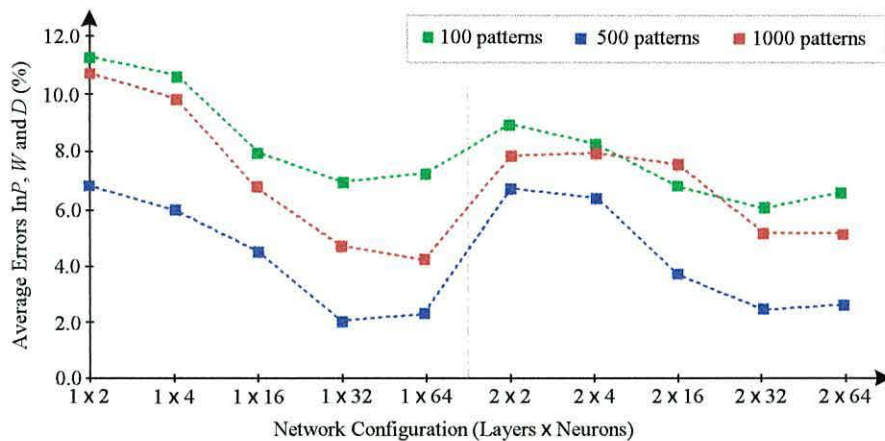


Figure 4.13: Average errors for P , W and D with various data set sizes

Figure 4.13 shows the average errors in the Pitch P , Width W and Depth D from the output of the neural network when a test set is applied at the inputs. It can be seen that the training sets with 100 and 1000 examples tends to perform rather poorly, which is envisaged as having too little and too much data respectively. The 500 example data set gave a good result, with about 32 neurons in the hidden layer (it also gave a good result with 64 neurons in the hidden layer, but for practical reasons the lowest acceptable number is selected).

4.6 Results

4.6.1 Using Theoretical Data

The results obtained so far demonstrate that a network can be trained to recognise several parameters from a diffraction pattern to a relatively high degree of accuracy. In this section the determination of system parameters using a single neural network is investigated.

The approach used was to generate a training set with various parameters (P : $1.5\mu\text{m}\rightarrow 1.7\mu\text{m}$, W : $0.3\mu\text{m}\rightarrow 0.7\mu\text{m}$ and D : $0.07\mu\text{m}\rightarrow 0.17\mu\text{m}$), which were in the ranges accepted in the Red Book of CD standards [46]. A total of 500 curves were generated (keeping one parameter constant, whilst varying the other two) with the diffraction pattern, and the values of P , W and D encoded into the output patterns. Once trained, the values for P , W and D are expected at the output when a test set is applied to the network. This is illustrated in Figure 4.14.

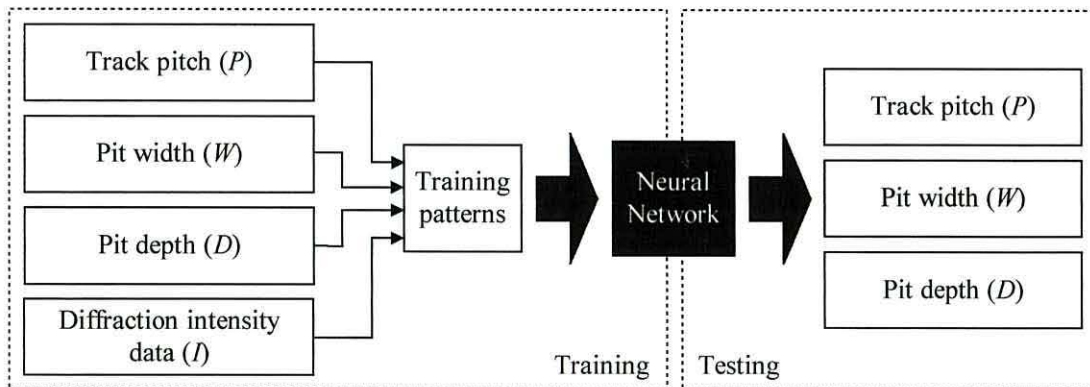


Figure 4.14: The determination of P , W and D using 1 network

The errors in recognition of the values of P , W and D were then determined for the test set of data. The results are presented in Figure 4.15, from which it can be observed that the neural network has determined the track pitch with an error of 3% ($\pm 0.04\mu\text{m}$), the pit width with an error of 3.5% ($\pm 0.03\mu\text{m}$) and the pit depth with an error of 6.5% ($\pm 0.05\text{nm}$). These are at the best configuration of 32 neurons in 1 layer.

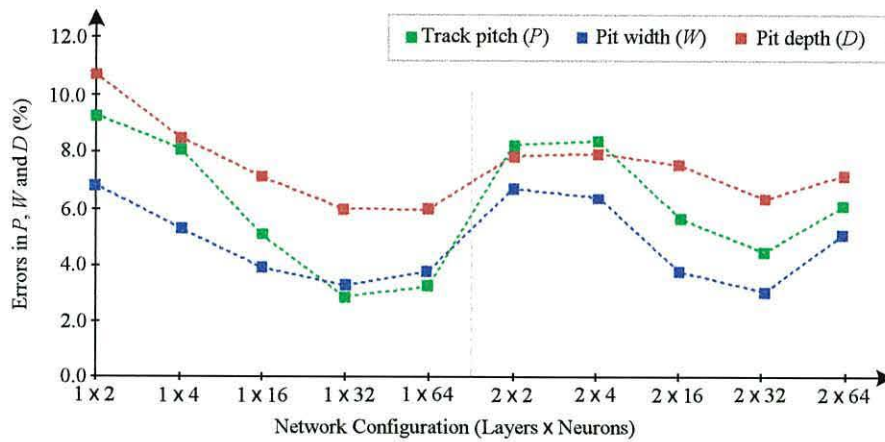


Figure 4.15: Errors in the values of P , W and D for various network configurations

Recalling the methods used in Chapter 3, of dividing the network into a number of separate networks (one for determining each unknown parameter) then the accuracy is found to improve considerably. This set-up is illustrated in Figure 4.16.

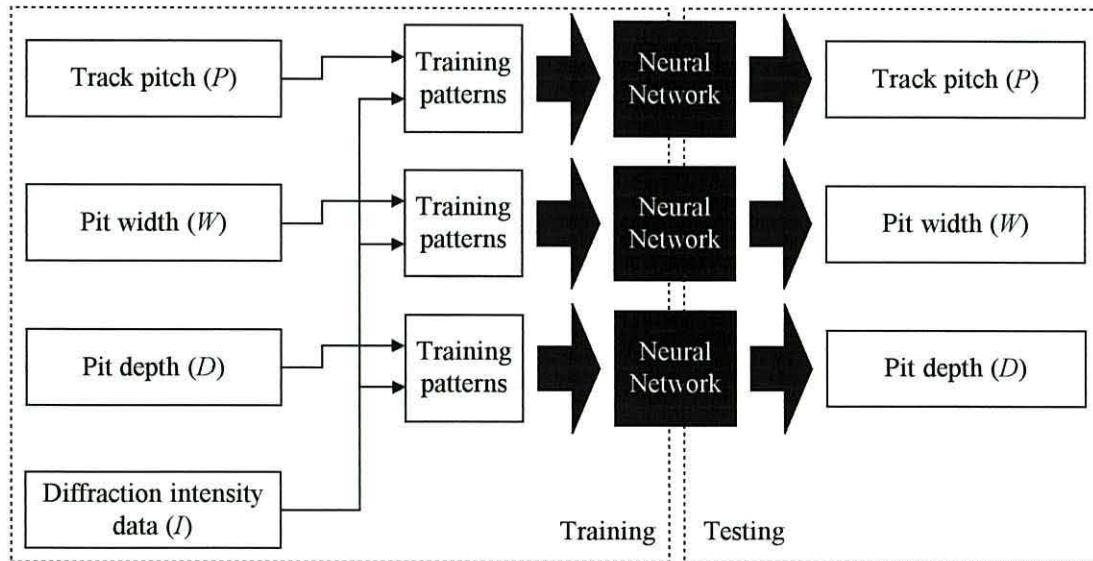


Figure 4.16: The determination of P , W and D using 3 networks

The values for P , W and D are encoded into each training set (all are trained with identical input data), and applied to a separate network. The results obtained are shown in Figure 4.17.

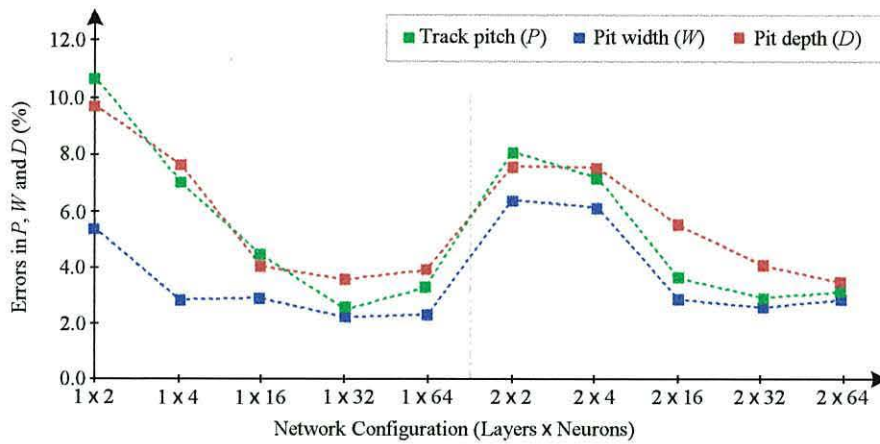


Figure 4.17: Errors in the values of P , W and D for various network configurations

From Figure 4.17, it can be observed that the neural networks could determine the track pitch with an error of 2.75% ($\pm 0.03\mu\text{m}$), the pit width with an error of 2.5% ($\pm 0.02\mu\text{m}$) and the pit depth with an error of 4% ($\pm 0.04\text{nm}$), which is an improvement of about 40% over the other method. This increase in accuracy is attributable to the division of the one network into three separate networks, thus considerably reducing the number of weights required. The configuration of the networks are summarised in Table 4.2.

Parameter	Network 1 (track pitch P)	Network 2 (pit width W)	Network 3 (pit depth D)
Input neurons (data points)	91	91	91
Hidden neurons	32	32	32
Output neurons	1	1	1
No. of training sets	500	500	500
No. of epochs required	50-100	50-100	50-100

Table 4.2: Summary of the network parameters

4.6.2 Using Simulated Experimental Data

Having determined a network architecture which yields optimum results when presented with theoretical data, the goal is to apply 'real' experimental data to the

network. However, as this data was difficult to obtain at the time of the research (due to the reluctance of the manufacturers to make public the quality of their CD fabricating methods), it was decided to simulate this data by adding random noise to the diffraction curve, as illustrated in Figure 4.18.

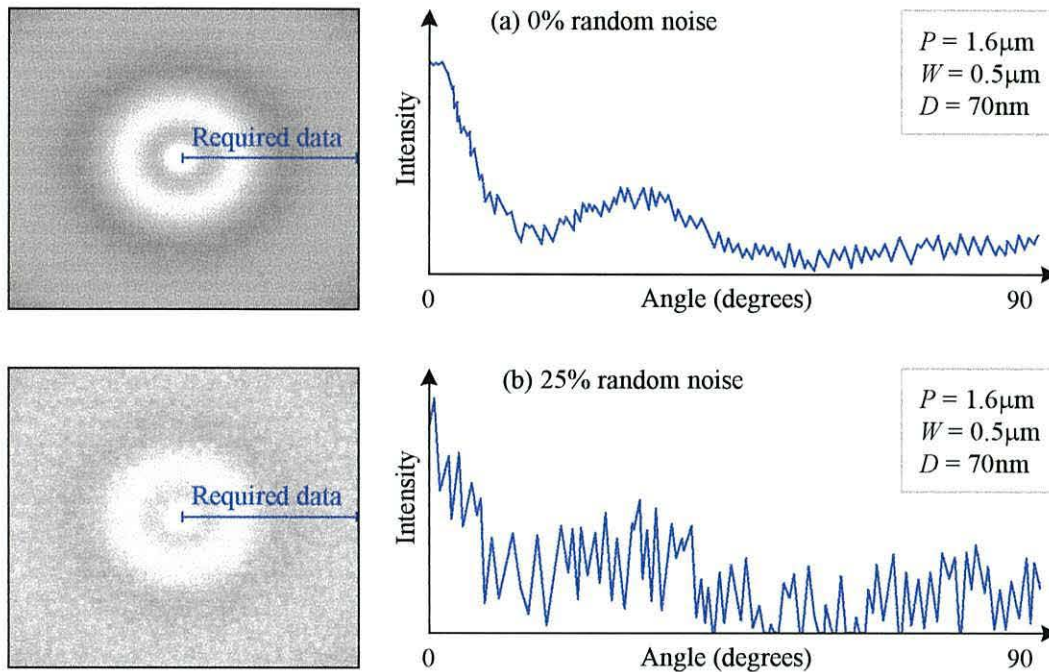


Figure 4.18: *The application of random noise to simulate experimental data*

The theoretical diffraction patterns were subjected to 5% and 25% random noise (Figure 4.18 (a) and (b) respectively). It is envisaged that this level of noise would never be experienced in a real situation; but is a good test as to the limits of the neural network technique.

A total of 500 training sets were created with various parameters, together with 50 validation sets. Test sets were created with $P=1.6\mu\text{m}$, $W=0.5\mu\text{m}$ and $D=70\text{nm}$, and 5% and 25% random noise was applied to each set (these are shown in Figure 4.18).

From simulations conducted on theoretical data, it was observed that a system of 3 networks with about 16 hidden neurons in 1 hidden layer was sufficient to solve the problem to an acceptable degree of accuracy. It was thus decided to use this set-up

again to solve this problem. Figure 4.19 shows the results of applying ‘simulated experimental’ data to the trained network for the 5% and Figure 4.20 for the 25% random noise samples.

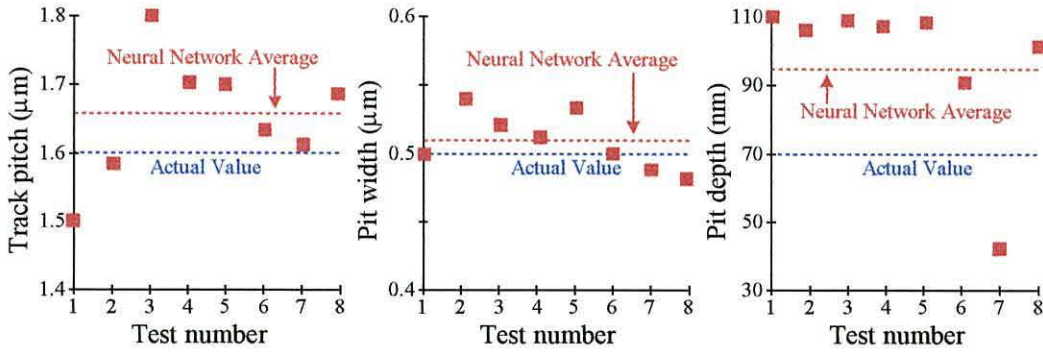


Figure 4.19: Errors in determining P , W and D for sample with 5% random noise

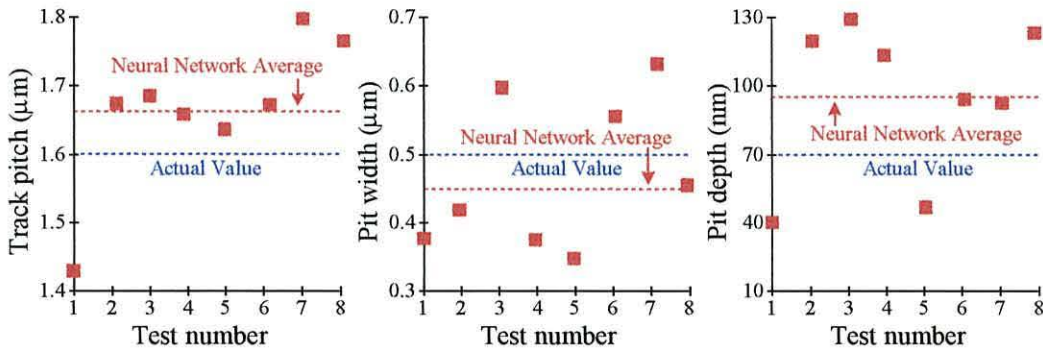


Figure 4.20: Errors in determining P , W and D for sample with 25% random noise

From Figures 4.19, Figure 4.20 and other numerical data, it can be observed that the networks were able to determine some of the unknown parameters to a high degree of accuracy. Table 4.3 summarises the errors which were obtained.

Noise	Track pitch P	Pit width W	Pit depth D
5%	±3%	±4%	±31%
25%	±3%	±10%	±32%

Table 4.3: Average errors observed for the unknown parameters

The determination of the pit depth was rather poor, which is envisaged as being due to diffraction data being less sensitive to changes in parameters in the nanometre scale. However, it should be noted that a system with 25% noise would be unacceptable in a real experimental measurement.

4.7 Summary

In this chapter it has been demonstrated that a neural network approach can be used to solve an optical inverse problem to a high degree of accuracy, and accomplished in a fraction of the time taken by conventional measurement methods.

Using a multi-layer perceptron neural network, which was chosen due to the possibility of creating numerous sample data sets, theoretical diffraction pattern data were generated, and several simulations were performed in order to determine the optimal settings. However, it was determined that one neural network was insufficient, and had to be separated into three networks (something which proved to be very successful, as the accuracy was observed to improve by about 40%).

It was observed that a neural network with about 8-16 hidden neurons in 1 hidden layer was sufficient, using 500 training sets, to be successfully trained in approximately 50-100 epochs. The accuracy obtained using theoretical data gave much confidence in the application of 'experimental' data to the trained network. Results showed that the values of the track pitch, pit width and pit depth could be determined to a high degree of accuracy, even with a very noisy data set (greater than would be observed experimentally).

These factors, together with the speed at which a trained network can determine unknown parameters leads us to believe that a neural network approach could be used in an on-line system during the CD manufacturing and testing process. Further research is now being carried out by Aerosonic to investigate the feasibility of using such a method in their CD analysing systems.

Chapter 5

Analysis of Magnetisation Maps and MFM Images of Thin Film Media

5.1 Introduction

In this chapter neural network techniques are applied to analyse magnetisation maps and magnetic force microscope images of thin film media, both of which were introduced in the first chapter. Here the models for the thin film media are outlined and magnetic force microscope which were used to create the sample images, and how the data is applied to a neural network in order to extract features from them (some of which may not be visible to the human eye).

This chapter also introduces a completely different approach to neural networks. In the previous problems the goal has been to generate suitable training sets, then use them to train a neural network. However, in this case data sets which are considered too large for a supervised learning approach are being dealt with. It is thus necessary to consider using an unsupervised learning method, and the self-organising map. These are discussed in greater detail later in the chapter, together with the techniques which can be used to analyse the results, and comparisons are made with other conventional image analysing methods.

5.2 Thin Film Media Model

The study of magnetic processes at the sub-micron level, such as those found in thin film media, is called micromagnetics. The grains are typically around 200Å and have a high magnetic moment. The first theoretical study of magnetic grains was developed by Stoner and Wohlfarth [6], which considered a single-domain elliptical particle in the presence of an external field \vec{H} . It is based on the calculation of the extrema of the energy function, as given in (5.1).

$$E = K \sin^2 \theta - MH \cos(\alpha - \theta) \quad (5.1)$$

where

K = uniaxial anisotropy constant

θ = angle between the direction of magnetisation and the easy axis

α = the angle between the easy axis and the direction of the applied field

These parameters are shown diagrammatically in Figure 5.1.

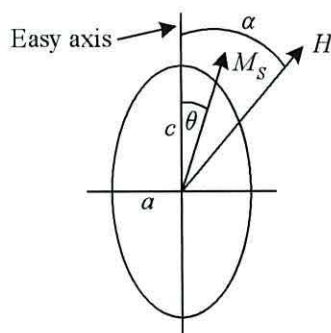


Figure 5.1: Parameters for the Stoner-Wohlfarth particle

In the model the two extremes are when the particle is at an orientation of 0° (parallel) and 90° (perpendicular) to the field. When the particle is perpendicular to the field, it will be rotated away from the easy axis and into the direction of the field, which gives rise to completely reversible changes in the magnetisation. However, if the particle is parallel to the field, then the magnetisation will remain in its original

direction. When the field is strong enough in the opposite direction, the magnetisation will line up parallel, and can be made to follow the direction of the applied field.

In this work, several thin film models were created with a variety of features. Recorded bits were simulated by allowing a series of oppositely magnetised regions relax, subject to the influence of a decreasing applied field, which was carried out using an energy gradient descent technique. The model is characterised by two important parameters:

- h_i is the value of the mean interaction field relative to the intrinsic coercivity.
- C^* is a field representative of the exchange coupling between grains.

The micromagnetic modelling of thin film media is the theoretical investigation of the interaction effects which occur between an externally applied magnetic field and a system of interacting magnetic particles. The modelling of the thin film media can be better understood by studying the magnetisation reversal mechanisms of non-interacting single domain particles.

The micromagnetic models which model the interactions are generally based on a hexagonal structure, where each nodal point represents the position of a magnetic grain; and the most common arrangement is hexagonal close packing (HCP). Using a HCP arrangement to model the structure of thin film media ensures that a high areal packing fraction is achieved. One of the models [55] of magnetisation reversal treats the magnetic grains as single domain particles. In simulations conducted by Walmsley *et al* [56], and used in this research, the modelling was of HCP cobalt with grains in the 200Å diameter regions (which is well below the diameters for the existence of multidomain behaviour).

The model assumes that each grain has a moment \vec{M}_i with constant magnitude (and a direction which is aligned with the local effective field at the grain centre) at equilibrium. One method of modelling the magnetisation reversals in the thin films is

to model the motion of moments within each grain. The expression for a single grain is based on a dynamical approach using the Landau-Lifshitz equation:

$$\frac{d\vec{M}}{dt} = \gamma \vec{M} \times \vec{H} - \frac{\lambda}{M} \vec{M} \times (\vec{M} \times \vec{H}) \quad (5.2)$$

where \vec{M} is the moment of the grain and \vec{H} is its effective field. Equation (5.2) rotates the moment vector until it is aligned with its local effective field (the equilibrium position). The magnetisation reversal method involves taking a system from saturation magnetisation in a particular direction to saturation magnetisation in the opposite direction (which is conducted by applying an external field). The model studies the transition of the system from positive saturation to negative saturation.

5.21 Magnetisation Maps

Magnetisation maps are basically a visual method of plotting the magnetisation of the grains in the magnetic sample. Each direction is represented by a different colour (Figure 5.2), which makes it easier to visualise patterns than with a greyscale image.

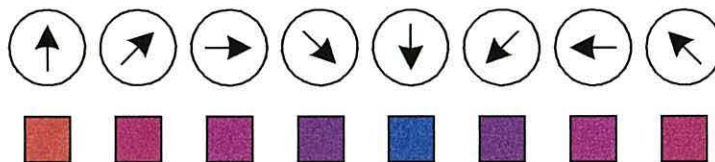


Figure 5.2: Magnetisation map orientation colour scale

5.2.2 Samples Used

Several simulations were performed using the thin film media model to create a series of samples of varying densities (measured in kilo flux closures per squared inch) and patterns. The values for the mean interaction field h_i and the intrinsic coercivity of the

grains C^* are also varied. The introduction of low coercivity LC grains gives rise to increased percolation (when there is significant statistical clustering).

Low coercivity grains arise due to stacking faults which convert the grains from having a hexagonal close packed structure (HCP) to a face centred cubic structure (FCC). HCP cobalt has a large anisotropy, whereas FCC cobalt has a lower anisotropy, and therefore a reduced coercivity.

The samples were chosen as they present a series of challenges for analysis. Samples 1, 2 and 3 have written bits, but an increasing tendency for vortex formation. In sample 1 the interaction field parameter is low at 0.1 and in 2 and 3 this parameter is 0.5 (the difference between 2 and 3 is the inclusion of LC grains which gives rise to increased percolation when there is significant statistical clustering).

Samples 4 and 5 were chosen because they have a different magnetisation structure (i.e. no written bits). However, samples 4 and 5 differ because the AC erasure is essentially isotropic whereas the DC erasure leads to ‘stripe-like’ domain structures (more information about AC and DC erasure is given in Chapter 1).

Sample 1: 1024 grains with 4 recorded bits, $h_i=0.1$, $C^*=0.2$

Sample 2: 1024 grains (0% LC) with 10 recorded bits, $h_i=0.5$, $C^*=0.1$

Sample 3: 1024 grains (15% LC) with 10 recorded bits, $h_i=0.5$, $C^*=0.1$

Sample 4: 4096 grains AC erased state, $h_i=0.5$, $C^*=0.1$ and 0% LC grains

Sample 5: 4096 grains DC erased state, $h_i=0.5$, $C^*=0.1$ and 0% LC grains

Simulations were initially performed on a system comprising 1024 grains (Figure 5.3), with 4 bits recorded along one dimension (sample 1). The use of such a small sample was necessary to test the neural network set-up prior to the application of larger samples.

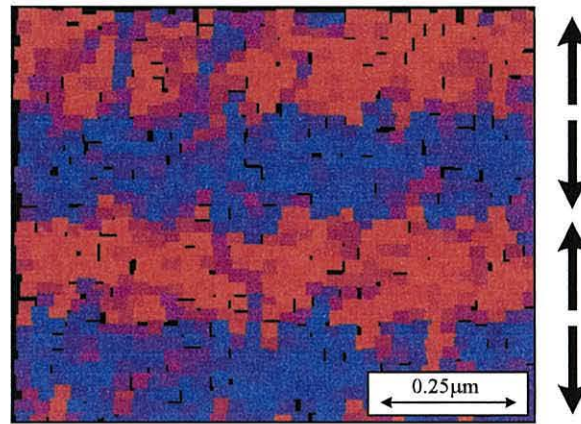


Figure 5.3: Magnetisation map of a system with 4 written bits

Simulations were also performed on systems comprising 16,384 grains, with periodic boundary conditions to eliminate the edge effects (Figure 5.4). The physical size of the system is $5\mu\text{m}^2$, with 40 bits recorded along one dimension (thus giving a recording density corresponding to $\sim 400\text{kfci}$).

The introduction of LC grains results in inferior transitions when the concentration is sufficiently high such that there is significant statistical clustering of grains, as illustrated in Figure 5.4(b). It has been shown [57] that this affects the micromagnetic parameters and recording properties of thin film media. The introduction of LC grains results in a significant increase in percolation, which is clearly evident when comparing the magnetisation maps. A plot of the magnetisation averaged across the tracks shows a qualitative indication of the degree of media noise present [58].

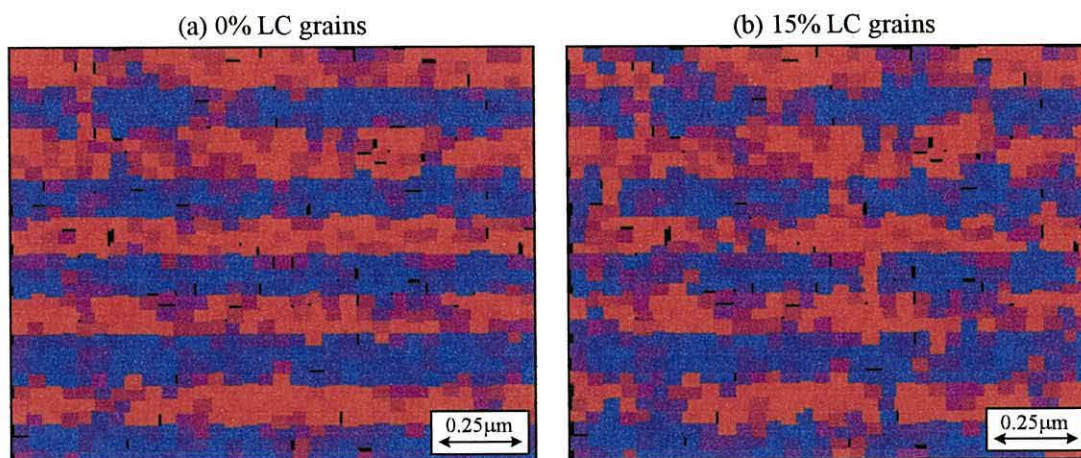


Figure 5.4: Magnetisation map with $h_i=0.1$, $C^*=0.1$

In order to study the underlying magnetic structure of the sample, it was also necessary to investigate samples with no written bits. Several samples were generated using both AC (Figure 5.5) and DC (Figure 5.6) erasure (which was described in Section 1.61), again with and without the introduction of LC grains.

The samples comprised 16,384 grains, with periodic boundary conditions to eliminate the edge effects. However, in order to reduce the processing time, the samples were reduced to 4096 grains. This also has the added advantage that one section of the sample can be used to train a neural network, and another section can be used to test it.

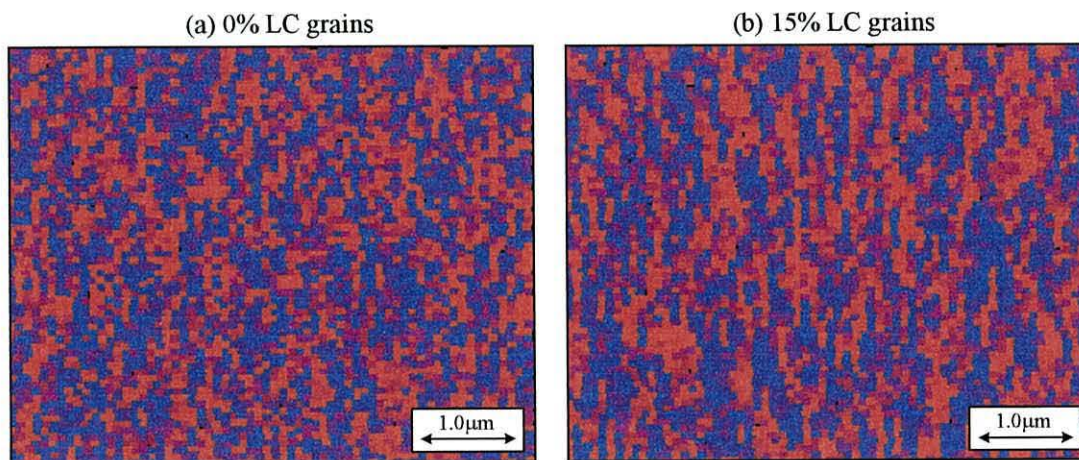


Figure 5.5: AC erased state with $h_i=0.5$ and $C^*=0.1$

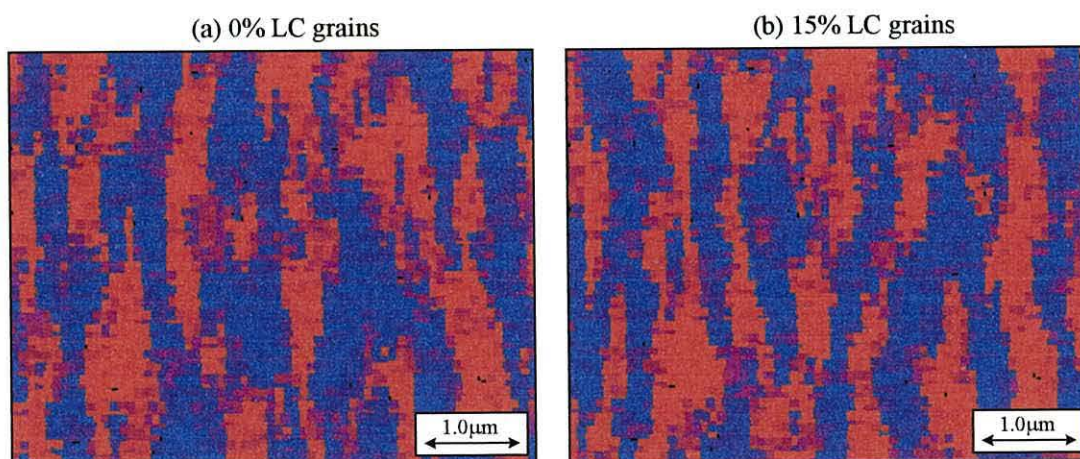


Figure 5.6: DC erased state with $h_i=0.5$ and $C^*=0.1$

5.3 Application of Neural Networks

The use of neural network techniques for image analysis is well established, and has proven to be a very efficient method in several cases. However, their use in the area of magnetism is not well established. In the previous chapters, much use was made of the supervised learning method, and the multi-layer perceptron, as it was possible to generate vast amounts of data for these applications. In the case of thin film data, it is not feasible to generate a large amount of training data because of the excessive computer processing time involved. In such instances, use will be made of the unsupervised learning method.

5.3.1 Unsupervised Learning

The unsupervised learning method has already been introduced in Chapter 2, but briefly, unsupervised learning is the process in which a neural network learns without the presence of an external teacher to oversee the learning process. A network which is being trained by unsupervised learning is not provided with any learning examples, instead it is presented with only the input patterns. Once the network has recognised any regularities within the input patterns, it can develop the ability to create internal representations for encoding specific features. In this chapter the process will be discussed in relation to the application of the magnetisation maps and MFM data.

5.3.2 Self-Organising Maps

The self-organising map (developed by Kohonen [59]) is probably one of the most popular neural network models. It belongs to the category of competitive learning networks. It is based on unsupervised learning, and little needs to be known about the characteristics of the input data. The SOM can be used to detect features inherent to the problem and thus has also been called SOFM, the Self-Organising Feature Map.

5.3.2.1 The SOM Algorithm

The basic idea of the SOM is simple yet effective; its goal is to transform high dimensional input data into one or two dimensional data, and do this in a topological ordered fashion. The map can be arranged in either a rectangular or hexagonal fashion (as illustrated in Figure 5.7). This flexibility has several advantages, especially in magnetism, as several samples are of the HCP type. However, for simplicity it was decided to make use of the rectangular arrangement only for this work.

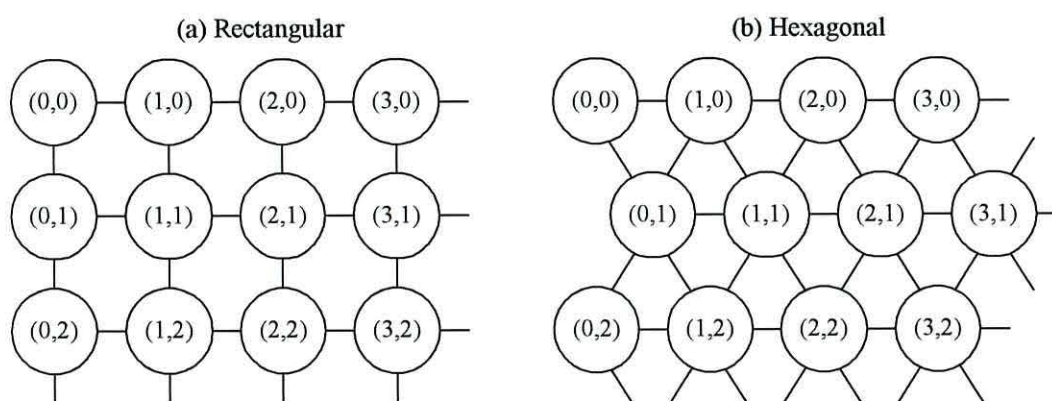


Figure 5.7: *Various SOM arrangements*

The SOM algorithm (developed by Kohonen [59]) can be summarised by five basic steps:

1. Initialisation - Random values are selected for the initial weights of the neurons (the weights must be different for each neuron, and it is preferable to keep the values quite low).
2. Sampling - A sample x is randomly drawn from the input data set and its similarity to pre-set vectors (which are stored in the SOM) is computed.
3. Similarity Matching - The best-matching neuron $i(x)$ in the SOM is found at a particular time n (using the common Euclidean distance measure):

$$i(\mathbf{x}) = \arg_j \min \|\mathbf{x}(n) - \mathbf{w}_j\| \quad j=1, 2, \dots, N \quad (5.3)$$

4. Updating - The synaptic weights of the neurons are adjusted (using the update formula):

$$\mathbf{w}_i(n+1) = \begin{cases} \mathbf{w}_j(n) + \eta(n)(\mathbf{x}(n) - \mathbf{w}_j(n)), & j \in \Lambda_{i(\mathbf{x})}(n) \\ \mathbf{w}_j(n), & j \notin \Lambda_{i(\mathbf{x})}(n) \end{cases} \quad (5.4)$$

where $\eta(n)$ is the learning rate parameter, and $\Lambda_{i(\mathbf{x})}(n)$ is the neighbourhood function centred around the best-matching neuron $i(\mathbf{x})$. Both values are varied dynamically during the learning process.

5. Continuation - Repeat steps 2-4 until there is no noticeable changes in the feature map.

The success of the formation of the map depends on the values chosen for the learning-rate parameter and the neighbourhood function. As there is no theoretical method for choosing the values, they must be selected on a trial and error basis. However, there are a few guidelines to facilitate the successful formation of the feature map:

- The learning-rate parameter $\eta(n)$ should initially be set to 1, and after a few hundred iterations should decrease in value (to about 0.1) with time. The topological ordering of the weights take place during the initial iterations.
- Adjacent neurons belong to the neighbourhood $\Lambda_{i(\mathbf{x})}(n)$ of the best-matching neuron $i(\mathbf{x})$ (see Figure 5.8). The neighbourhood function initially includes all the neurons in the SOM, and decreases with time to include only a few neighbouring neurons.

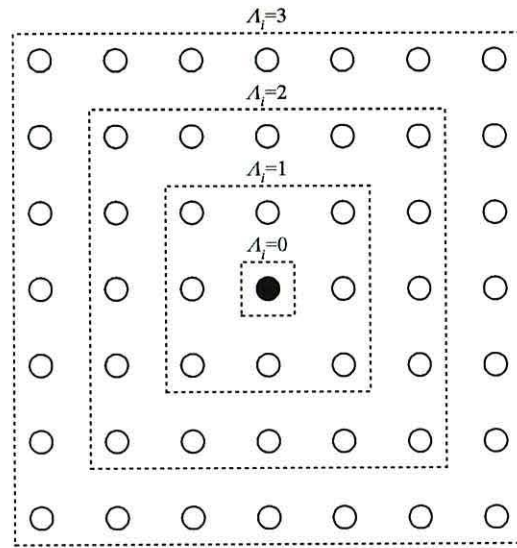


Figure 5.8: *Neighbourhood function*

When the SOM has converged the output feature map will show the important statistical characteristics of the input, as the basic aim of the SOM algorithm is to store a large set of input vectors by finding a smaller set of prototypes, so as to find an approximation to the original input space. This is illustrated in Figure 5.9:

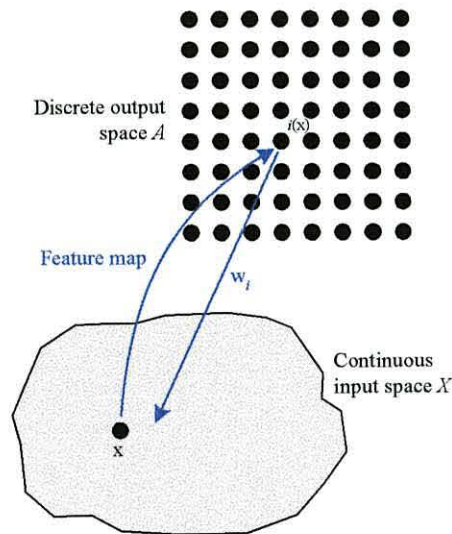


Figure 5.9: *Relationship between the map and the weight of the best-matching neuron*

5.3.2.2 Visualising the SOM Using U-Matrix Diagrams

The SOM is easy to visualise, and over the years several visualisation techniques have been devised. Due to the inherently intricate nature of the SOM, however, not one of the visualisation methods discovered so far, has proven to be superior to others.

The unified distance matrix (U-matrix) representation of the self-organising map visualises the distances between the neurons (see Figure 5.10), and provides a better method of visualising the boundaries between clusters on the SOM. It is derived from the graphical methods to illustrate the clustering of reference vectors in the SOM by Ultsch and Siemon [60]. Kraaijveld *et al.* [61] later suggested a method in which relative distances between neighbouring reference vectors are represented by shades of grey (and eventually colour).

The distance between the adjacent neurons is calculated and presented with different colourings between the adjacent nodes. A dark colouring between the neurons corresponds to a large distance and thus a gap between the reference values in the input space. A light colouring between the neurons signifies that the reference vectors are close to each other in the input space. Light areas can be thought of as clusters and dark areas as cluster separators. This is an useful way of finding clusters in the input data without having any prior information about the clusters.

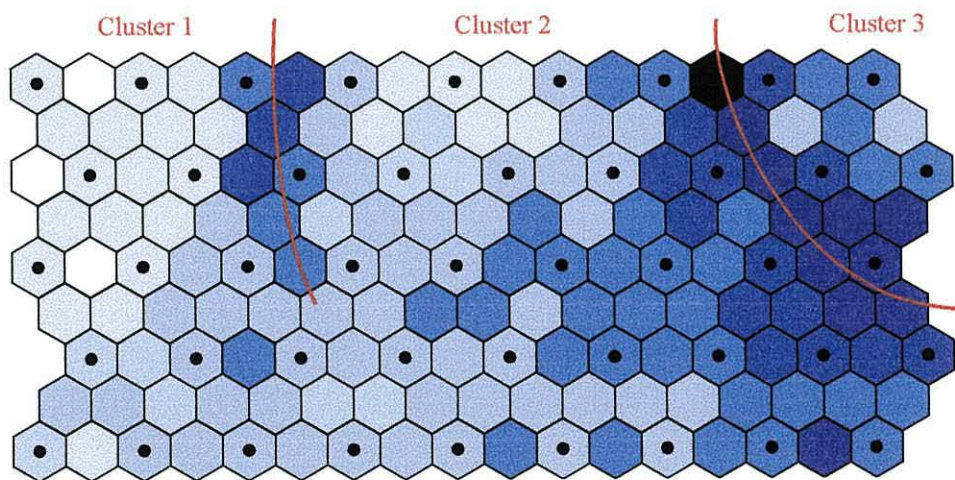


Figure 5.10: U-matrix representation of the self-organising map

The black dots in the figure represent the neurons, and in this example the representation reveals that these is a separate cluster in the upper right corner (as the clusters are separated by a dark gap). Thus, teaching a SOM and representing it with the U-matrix offers a fast way to get an insight of the data distribution.

In our simulations, the U-matrix is used to visualise similar features within the magnetisation map or MFM image from the clustered regions on the SOM. These are encoded with their original positions, so that they can be replotted later.

5.3.3 Image Coding Techniques

In order to extract features from images such as magnetisation maps and MFM images it is important to have a better understanding of the fundamental image coding techniques [62].

A typical image may at first seem to contain an enormous amount of information, but by taking a closer look it becomes obvious that much of the data is not uncorrelated. The aim of image coding techniques is to take advantage of the correlations in the image to find representations which are considerably smaller, yet which sacrifices a relatively small error (an example of this is a JPEG image). The relevance of this technique to the investigation lies in the fact that it is of interest to only extract a few features from a complex image. As an example, consider the simple picture of the car in Figure 5.11.

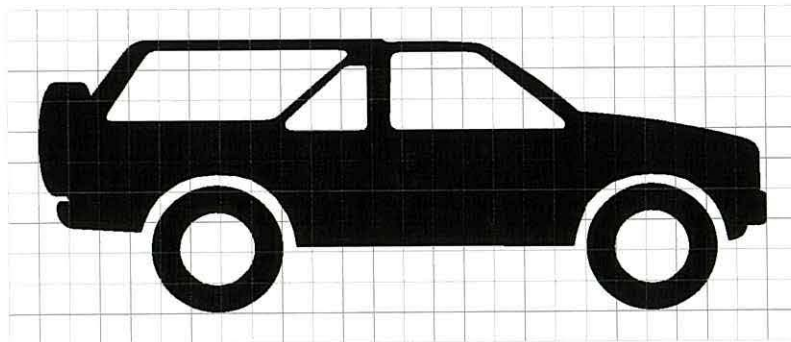


Figure 5.11: *Original pixelised image*

The image coding breaks the picture up into small blocks and then approximates each block in some manner. Typically, a set of filters are chosen such that linear combinations of the filters can be used to approximate different image blocks. The mean-squared difference between the original and the reconstructed image is used as the error measure.

The picture in Figure 5.11 is now divided into 26×11 non-overlapping blocks. The blocks are then applied to a neural network (or more specifically a self-organising map), and the weights that the network has learned are represented as a series of eight masks (shown in Figure 5.12). The number of masks was selected so as not to lose too much information. Each mask shows the set of weights associated with a single output. Positive weights are represented by a white colour, negative weights by black and a zero weight by grey.



Figure 5.12: *The masks which were learned by the network*

It is now possible to code the image, using these masks, and represent it with only eight different symbols (hopefully without losing too much of the original information):

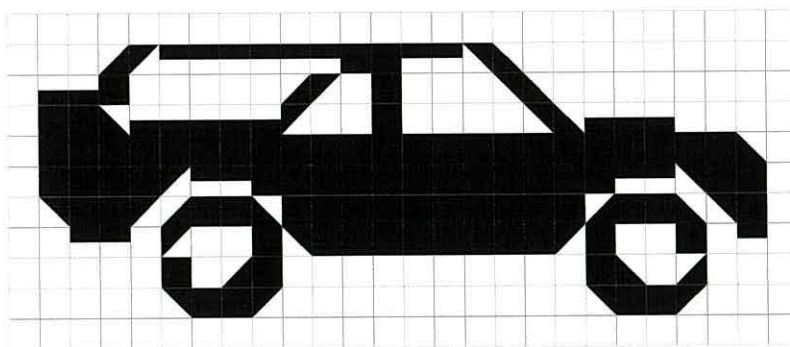


Figure 5.13: *Coding of image using learned masks*

From the reconstructed image, it can be seen that some of the data has been lost, but the important data is still present. More importantly, it has been possible to use just

eight symbols to represent an image (using more symbols would greatly improve the quality of the reconstructed image). This method seems ideally suited to our problem; being able to represent a complex magnetic sample with a small number of masks (without too much loss of detail), thus making it much simpler to pinpoint certain features which may be present (and hence represented by a particular mask). This method has been adapted and is used in the remainder of this chapter to extract features from complex magnetic images.

5.4 Feature Extraction From Magnetisation Maps

The magnetisation maps in Figures 5.3-5.6 give a good representation of the actual structure of a sample. From these maps it is possible to visualise many different features, such as the written bits, percolations, vortices etc. Several methods are available [56] to show the presence of these features. However, in this chapter they are investigated through the use of neural networks, and demonstrated that such a method can work just as well as a conventional method (and possibly much quicker), using a selection of samples (described in section 5.2).

5.4.1 Pre-processing Techniques

The data fed to a SOM includes all the information that a network gets. If erroneous data is fed to the SOM, the result is also erroneous or of bad quality. Erroneous data must be filtered using prior knowledge of the problem domain and common sense. As the main interest is the observation of domain structures within the sample, there are several methods available. However, the three which are of most interest to us are the calculation of the moments, divergence and curl. As the investigation is only concerned with the magnitude of the moments, divergence and curl of the distributions, it is feasible to plot only the absolute value of each expression. A black pixel represents a zero, and a green pixel represents the maximum value for each

image. The ratio of black pixels to green pixels increases as the exchange parameter increases, which indicates that the size of the magnetically correlated regions increases with stronger intergranular coupling.

Any data which is to be applied to a neural network needs to be pre-processed, and this case is no exception. If the raw data was applied to the self-organising map, then the moment orientations would be observed, but it would be impossible to observe any other features in the sample. To do this it is necessary to use pre-processed data which seek known features.

5.4.1.1 Moment Analysis

The moment analysis of \vec{M} gives the measure of localised flux closure within a vector field. For each grain i in the system, the following vector sum is calculated:

$$\sum_{j \in n.n} \vec{r} \times (v_j \hat{m}_j) \quad (5.5)$$

where the sum is over those grains which are considered to be the nearest neighbours of grain i , and \vec{r} is the vector between the centres of grain i and grain j . Figure 5.14 shows the effect of applying moment analysis to the sample with 4 written bits (Figure 5.3).

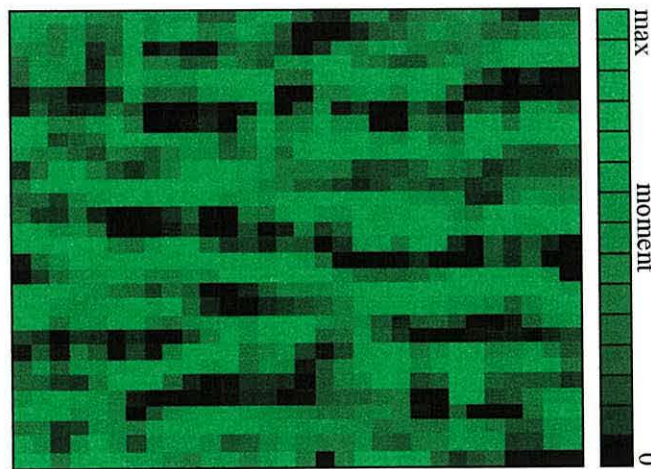


Figure 5.14: *Moment analysis of a sample with 4 written bits*

This technique has proven successful [56] in predicting the positions of the centres of vortices within a magnetisation distribution.

5.4.1.2 Divergence

The divergence of \vec{M} measures the degree of continuity of the magnetisation distribution, and is a function of position. If a medium with a 2-dimensional magnetisation distribution $\vec{M}(x, y)$ is considered, then the divergence is given by:

$$\text{div}\vec{M} = \frac{\partial M_x}{\partial x} + \frac{\partial M_y}{\partial y} \quad (5.6)$$

The divergence of the magnetisation is calculated numerically by forward differences. The elements of a rectangular array, $f(m, n)$ ($m = 0, 1, 2, M-1$ and $n = 0, 1, 2, N-1$), are assigned a value which corresponds to the angle of magnetisation of the grain whose centre is closest to the centre of the element. The partial derivatives corresponding to the position (m, n) is given by:

$$\frac{\partial M_x}{\partial x} = \frac{\cos(f(m+1, n)) - \cos(f(m, n))}{\delta x} \quad (5.7)$$

$$\frac{\partial M_y}{\partial y} = \frac{\sin(f(m, n+1)) - \sin(f(m, n))}{\delta y} \quad (5.8)$$

Figure 5.15 shows the divergence of the sample in Figure 5.3.

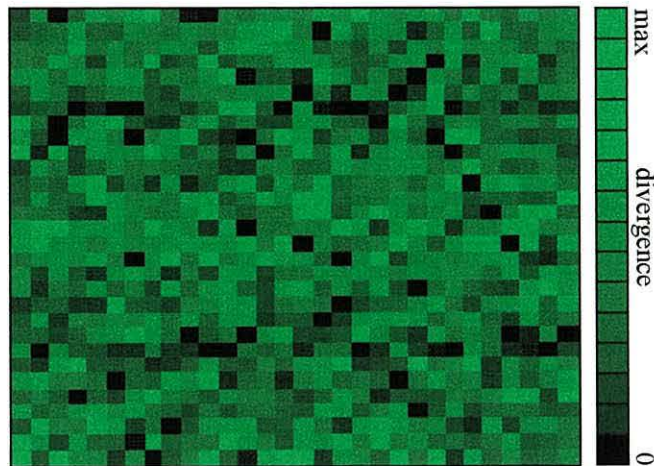


Figure 5.15: *Divergence of a sample with 4 written bits*

5.4.1.3 Curl

The curl of a vector field measures the amount of rotation within the field. It can be expressed as:

$$\text{curl}\vec{M} = \frac{\partial M_x}{\partial y} - \frac{\partial M_y}{\partial x} \quad (5.9)$$

This is calculated numerically, using forward differences similar to those used in the calculation of the divergence of \vec{M} . Figure 5.16 shows the curl of the sample in Figure 5.3.

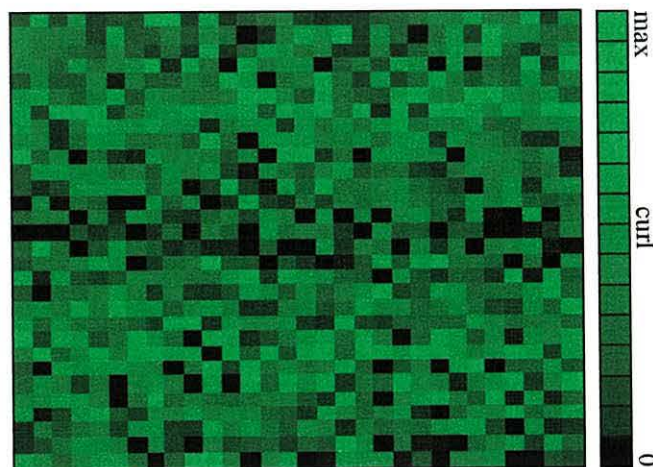


Figure 5.16: *Curl of a sample with 4 written bits*

5.4.2 Data Preparation

After using various methods to pre-process the raw data, it must still be prepared before applying it to the self-organising map. A problem arises due to the grains in the sample being rather 'disorderly' i.e. not in an orderly grid (which is desirable when presenting data to a grid-like self-organising map). Therefore, the grains in the original data are arranged into a grid of the same dimension. This is illustrated in Figure 5.17, from where it can be seen that there is only a slight loss in quality.

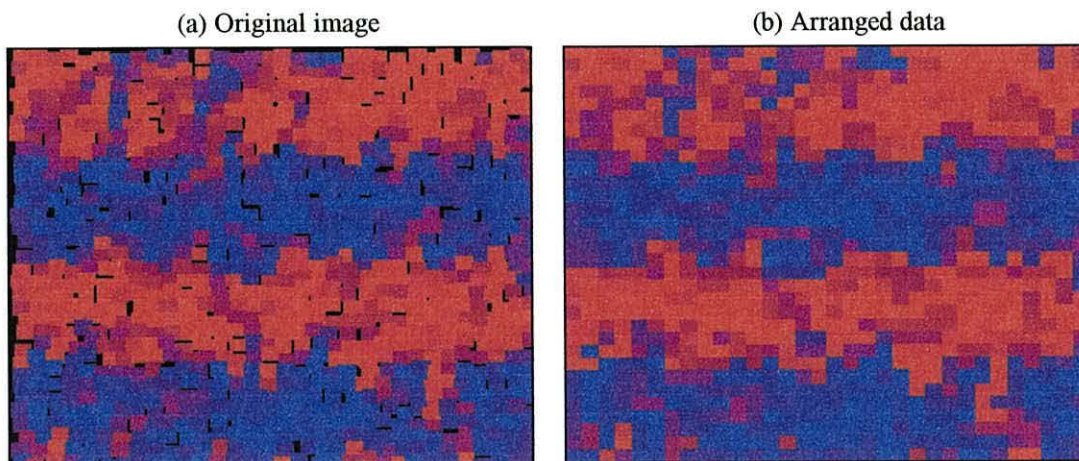


Figure 5.17: *Arrangement of magnetisation map grains into an orderly grid*

In order for the self-organising map to pick up specific features it must be supplied with data about a particular grain and also its nearest neighbours. The image is then sampled into regions of 4, 16 and 64 grains (grids of 2×2 , 4×4 and 8×8 respectively). The reason for doing this is to identify features of variable sizes e.g. vortices which may consist of several grains diameter. For a sample of 1024 grains (32×32 grains) there is no need to use a grid any larger than about 2×2 , as this will only increase the probability of finding more than one feature within the grid; and thus making it difficult to classify it into any particular group on the self-organising map (the grid size will also determine the number of neurons in the input layer of the self-organising map). The images in Figure 5.18 show how a magnetisation map is sampled in a raster fashion so that each grid is sampled.

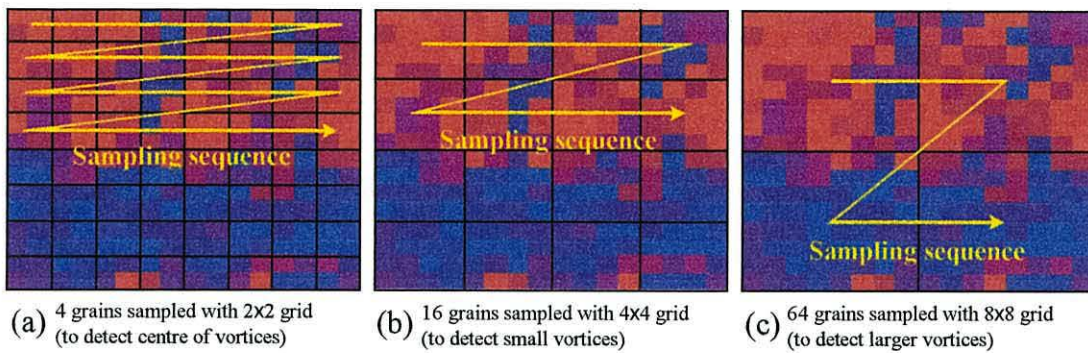


Figure 5.18: *Sampling a magnetisation map with 2×2 , 4×4 and 8×8 grids*

Each grid now represents a series of grains in the magnetisation map. The neural network is capable of placing these grids (only one grid size can be applied to each neural network) into groups in an organised manner, dependent on the features within them (in this case the number of groups has been restricted to 8). The features which are specific to each group can be observed. In Figure 5.19 the groups which were determined in the image sampled with a 2×2 grid are shown.

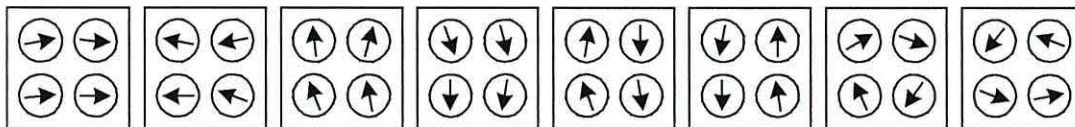


Figure 5.19: *The features present in the 8 groups in the SOM (using 2×2 grids)*

From the above figure it is quite clear what each group represents. However, it should be noted that these are only approximations (or the closest matches) for the contents of each group. For clarity, each group has been assigned a specific colour and then replotted in their original positions. By doing this it is possible to observe what features are present in different regions of the original magnetisation map (Figure 5.20). It should be stressed that the following magnetisation maps are reconstructions of the original (created by the self-organising map).

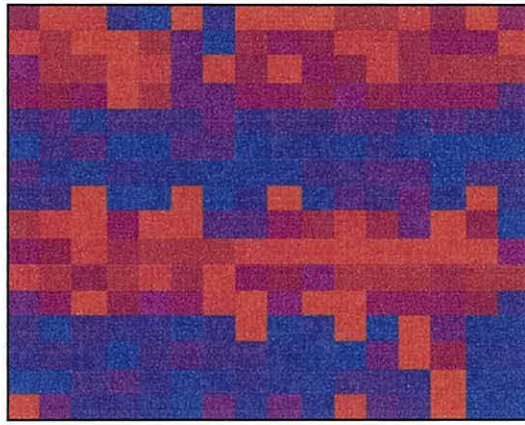


Figure 5.20: *Replotting the SOM groups in their original positions*

As the main interest is the observation of vortices (assuming that vortices are present in the original data), there is a high probability that the neural network will have seen them and assigned a specific group to represent a region where a vortex might reside (as a vortex can go in the clockwise or anti-clockwise direction, it may be possible that more than one group has been assigned). It is then possible to observe where these occur in the original magnetisation map. In the following sections the application of this method to observe vortices in several samples is investigated, and the results compared with other conventional methods.

5.4.3 Network Optimisation

The above example made use of 8 classification groups to represent different features. In essence, each group is actually a neuron in the output layer of the self-organising map. Therefore, if the number of groups are increased then the number of neurons much also be increased accordingly. In this section the optimal size of the self-organising map (number of groups) and the amount of training required is investigated.

An investigation was conducted on a system sampled using 2×2 grids (using the sample above), so that the optimal number of groups could be determined. It is estimated that the minimum number of groups should be about 8; the groups being:

- 6 groups for grains pointing in a particular direction ($\sim 60^\circ$ increments)
- 2 groups for grains that rotate - one to represent clockwise rotation, and the other for anti-clockwise rotation

There is theoretically no maximum number of groups which could be used. For example, if 256 groups were used (32 grains by 32 grains, sampled in 2×2 grids), this would give each sampled grid its own group - a rather pointless task, as there are only a few specific features which are of interest.

However, if just 2 groups were used, the opposite would happen, and only two general features would be observed. To investigate this problem a system has been set up with 4, 8, 24 and 48 groups. The results obtained are shown in Figure 5.16.

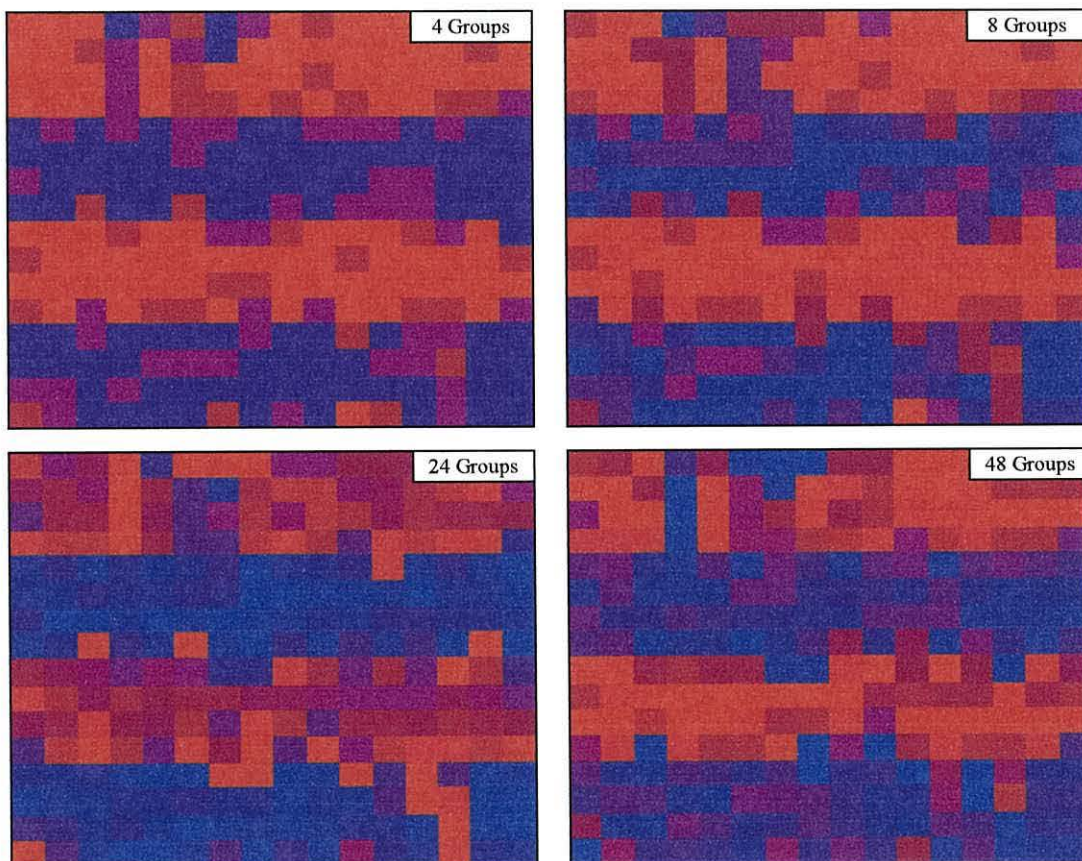


Figure 5.21: *Effect of varying the number of classification groups*

From Figure 5.21 it can be observed that representing the magnetisation map with 4 classification groups is inadequate (as there are so many features present in the original map). However, it is also noticeable that there is no significant improvement in quality when the number of groups is increased to 24 and 48. It can therefore be concluded that a system with about 8 classification groups is sufficient to provide an accurate representation of the original magnetisation map.

The other important factor when training a neural network is the amount of training required. This was first encountered in Chapter 3 whilst dealing with supervised learning, and similar problems arise in this case too. If the training is insufficient then the self-organising map is unable to fully organise the features onto the output map. Overtraining will result in the map being too familiar with a particular pattern, and thus unable to accurately organise the map. It is feasible to stop the training when the mean squared error has reached about 0.1% of its initial value and fails to decrease further.

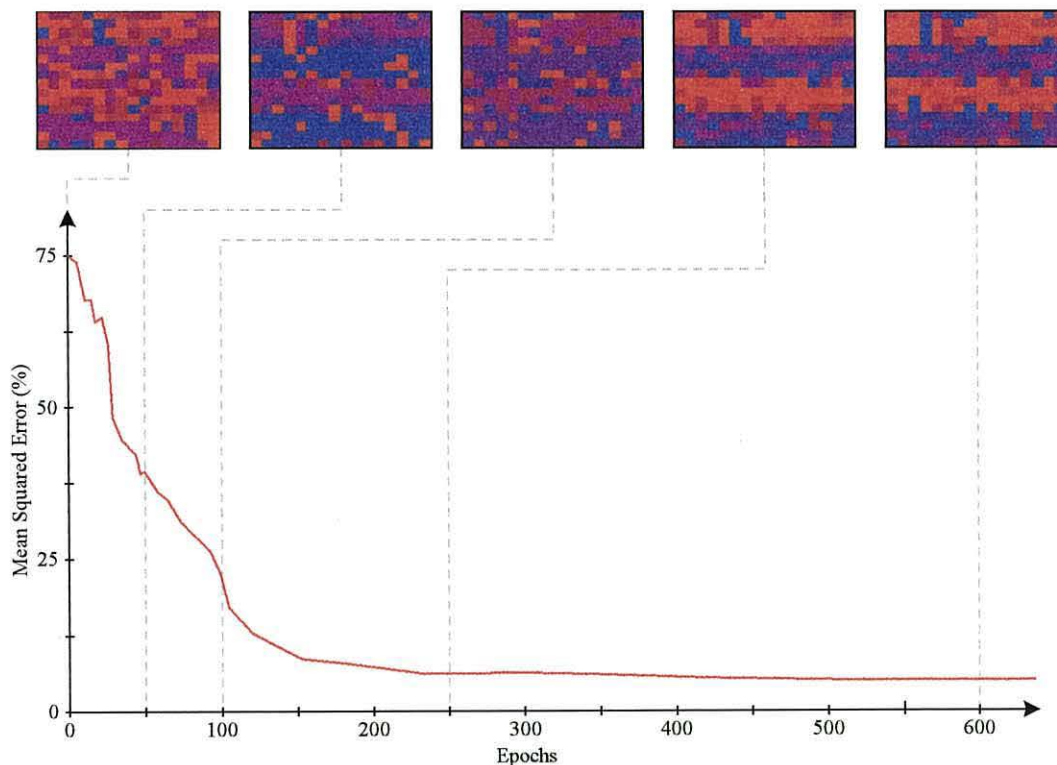


Figure 5.22: *The error values w.r.t training and the outputs obtained*

Figure 5.22 shows how the error decreases with training (for a system with 8 groups), and shows the outputs obtained from the self-organising map at different periods during the training. From these results, and other data it can be concluded that the neural network can be sufficiently trained in about 250 epochs, and shows no improvement with further training. It should be noted that this amount of training is only relevant to a system of this size (1024 grains), and an increase in system size (number of grains) would require more training. However, increasing the system size makes it difficult to visualise the output data.

From these investigations, it has been determined that for a system of this size, 3 neural networks can be constructed to recognise features of various sizes. Their parameters are summarised in Table 5.1.

Sample Grid	Input Neurons	Classification Groups (Output Neurons)	Training (Epochs)
2×2	4	8	250
4×4	16	8	230
8×8	64	8	220

Table 5.1: *Optimal neural network parameters*

5.4.4 Results

The self-organising maps which were created in the previous section are now used to extract features from five sets of samples, and are compared with results obtained using methods discussed in Section 5.4.1.1 for calculating the presence of vortices. The samples are presented to the SOM in 2×2 , 4×4 and 8×8 grids (so that different sized vortices can be observed), after which the features present will be organised into similar groups.

From these, it is then possible to see which groups contain the traces of a vortex (at this point the user must decide which group closely represents a vortex). By replotting the data in their original positions on the magnetisation map it is possible to observe where each feature occurs (and the groups which contain vortices are highlighted in a different colour (yellow) to emphasise their positions).

Sample 1

The sample was prepared and presented to three self-organising maps (the method is explained in previous sections), with parameters set as in Table 5.1. The results obtained (and the features extracted) are shown in Figure 5.23.

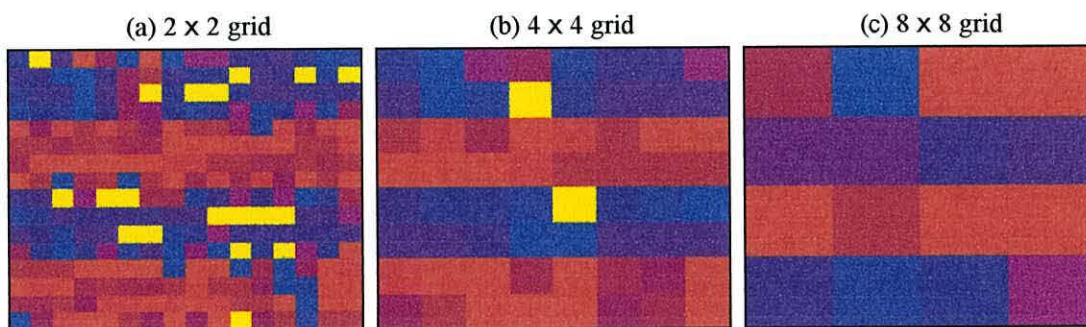


Figure 5.23: *Neural network results with various grid sizes*

In the figures, the features which represent the probability of the presence of a vortex are shown in green. It can be observed from Figure 5.23 (a) that several small vortices exist, and that they are a local phenomena. Larger vortices are less common in this sample, which is not unexpected, as the written bits are only a few grains each.

These images are then compared with the moment calculation method, and the positions of the vortices observed. These similarities are shown in Figure 5.24 (with similar features circled in black and red).

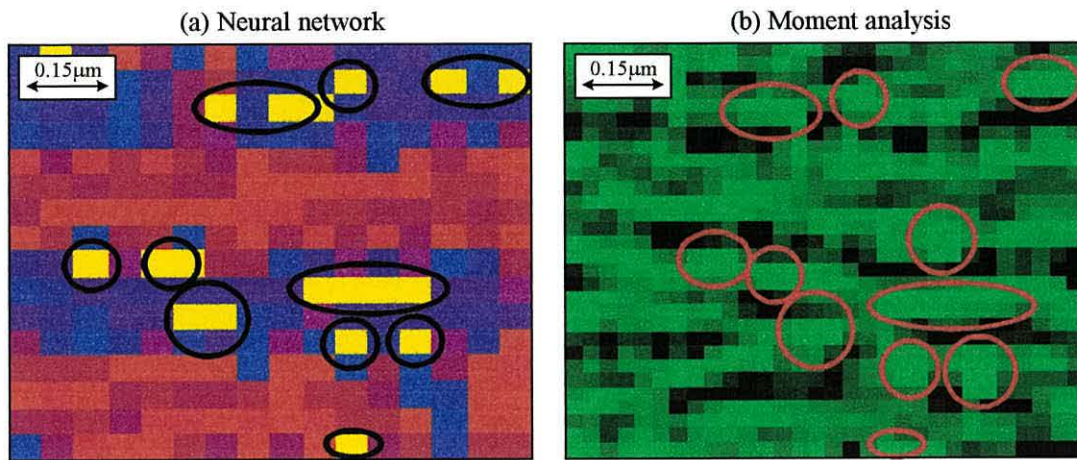


Figure 5.24: Comparison of results

From Figure 5.24 it can be observed that the results obtained using a neural network method are similar to that obtained with the moment calculation method, and a further comparison to the original data in Figure 5.25 shows that the neural network method has the capability to effectively recognise vortices simply from the topology of the magnetisation map.

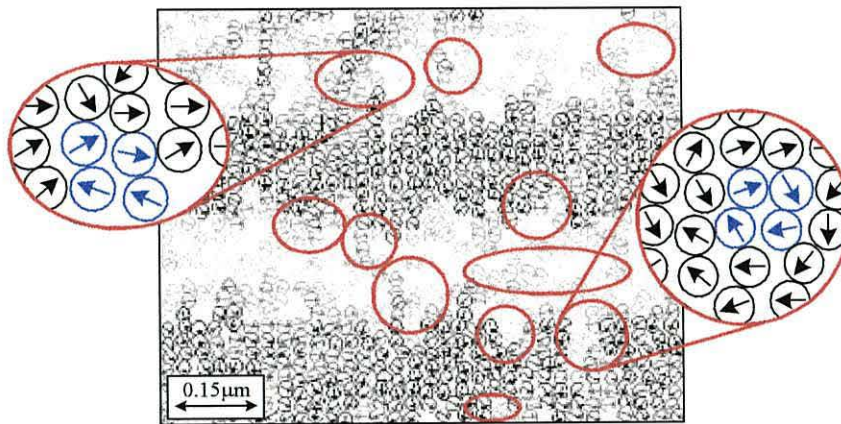


Figure 5.25: Confirmation of results with original data

Sample 2

The sample was prepared and presented to three self-organising maps using the same method as before, with parameters set as in Table 5.1. The results obtained (and the features extracted) as shown in Figure 5.26.

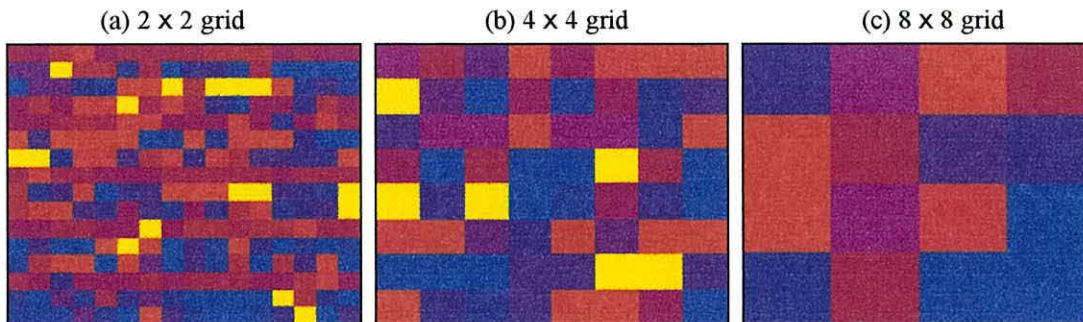


Figure 5.26: *Neural network results with various grid sizes*

In this sample it can be observed that there are not so many vortices present, which is due to the size of the written bit. In Figure 5.26 (c) it can be seen that there are no large vortices present, which is again attributable to the bit length.

These images are then compared with the moment calculation method, and the positions of the vortices observed. These similarities are shown in Figure 5.27 (with similar features circled in black and red).

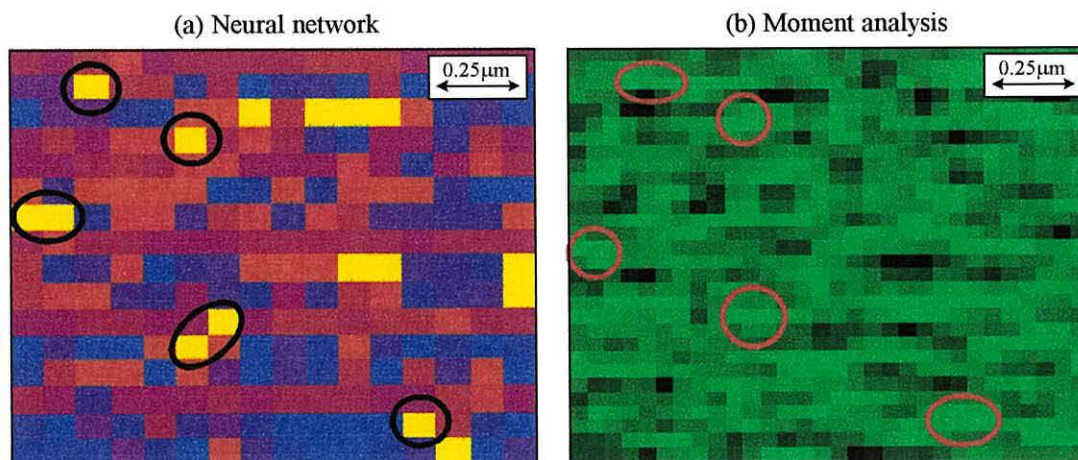


Figure 5.27: *Comparison of results*

From the above images it can be seen that there are a few similar features which were extracted. However, there are a few features one method may have extracted, but the other missed (and vice versa). It is envisaged that the reason for this discrepancy is that the neural network has detected features which may be similar to a vortex. However, a further comparison to the original data shows that the neural network method is very effective.



Figure 5.28: Confirmation of results with original data

Sample 3

The sample was prepared and presented to three self-organising maps using the same method as before, with parameters set as in Table 5.1. The results obtained (and the features extracted) as shown in Figure 5.29.

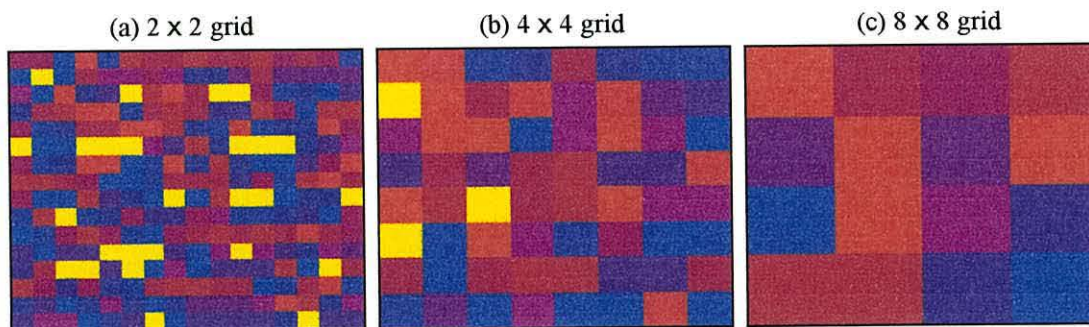


Figure 5.29: Neural network results with various grid sizes

From Figure 5.29 (a) is obvious that there are many more vortices present, which is attributed to the introduction of LC grains into the sample. The absence of larger vortices is again due to the length of the written bits. A comparison with the moment calculation method further confirms our prediction, and these similarities are shown in Figure 5.30.

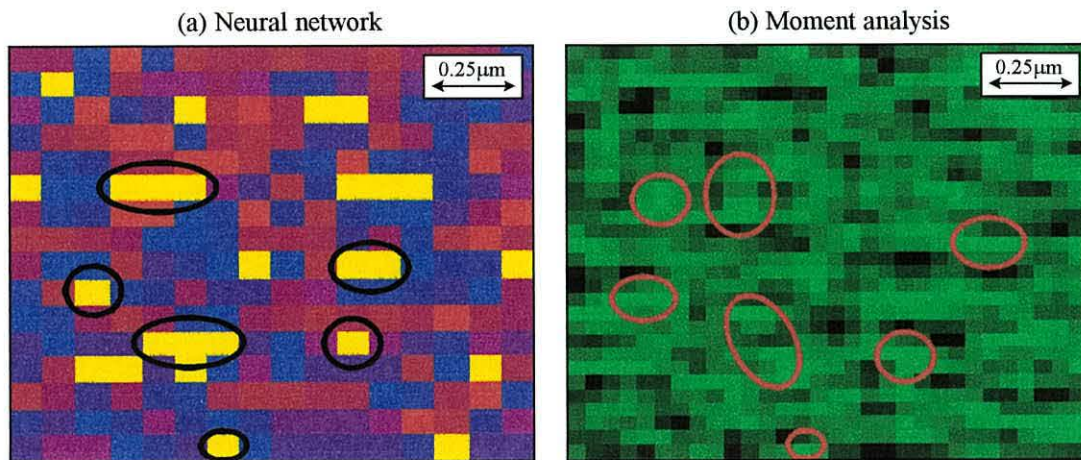


Figure 5.30: Comparison of results

It is apparent that some of the vortices have been extracted by the neural network method, but others have also been found. Some of the vortices found in the moment calculation were also observed with the 4×4 grid data set. A further comparison to the original data shows that the neural network method is very effective.

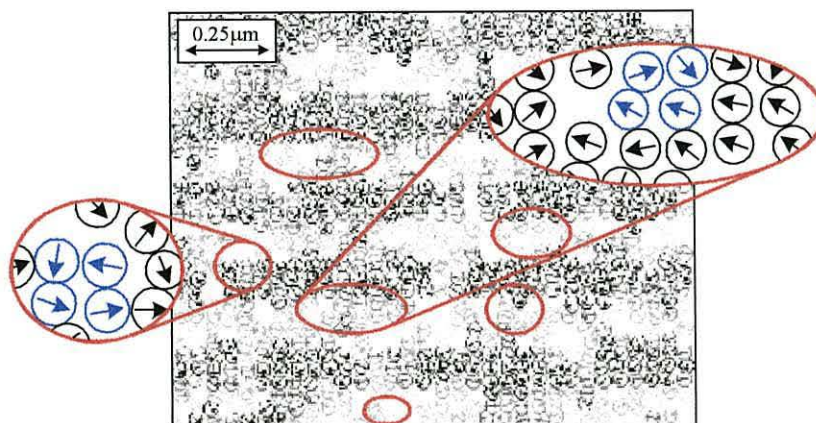


Figure 5.31: Confirmation of results with original data

From the images obtained, it can be seen that the vortices tend to form at the transitions. It is rather difficult to observe this phenomenon with the moment calculation method, but the neural network seems to emphasise it.

The samples so far have all consisted of a series of written bits, of varying sizes, with and without the introduction of LC grains. The following samples have no written bits and are of higher densities.

Sample 4

This sample has no written bits, but has been subjected to AC erasure. The sample was erased using a simulated annealing approach, which tends to lead to the nucleation of large numbers of small vortices. The results obtained (and the features extracted) as shown in Figure 5.32.

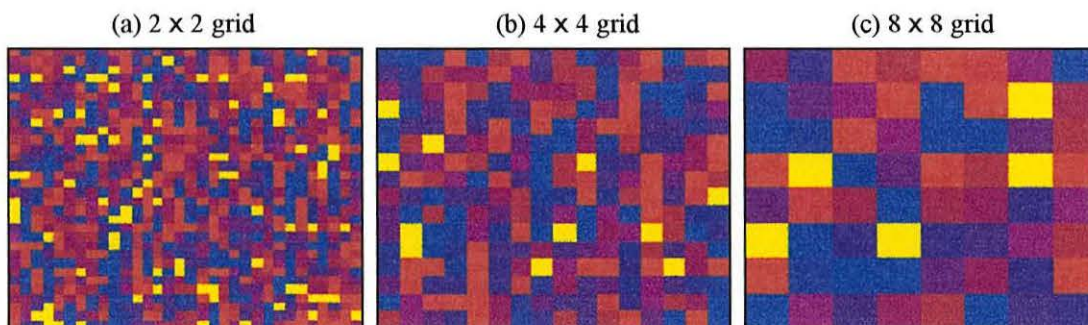
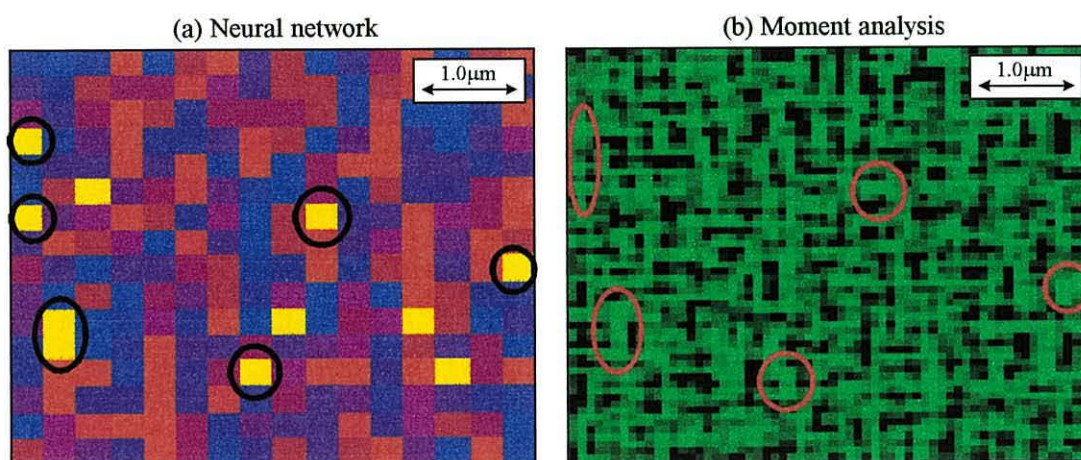
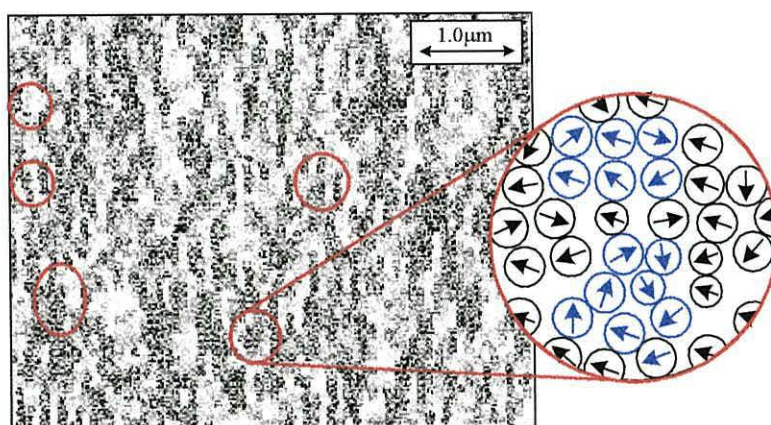


Figure 5.32: *Neural network results with various grid sizes*

From Figure 5.32 it can be seen that there are many more vortices present in the sample, and as there are no bits written to the sample there is an increase in the number of larger vortices. These images are then compared with the moment calculation method, and the positions of the vortices observed. These similarities are shown in Figure 5.33 (with similar features circled in black and red).

Figure 5.33: *Comparison of results*

It is interesting to observe that the moment analysis shows the presence of many vortices, which confirms the results obtained using the neural network method. The results seem to be capable of picking out various sized vortices, which is evident from Figure 5.33 (which is for the 4×4 grid), and also from Figure 5.32 (a) which shows the presence of several small vortices.

Figure 5.34: *Confirmation of results with original data*

The comparison with the original data looks very promising, as it shows that the neural network method works rather well with larger sized vortices (which are harder to detect, as there are so many variations).

Sample 5

This sample also has no written bits, but has been subjected to DC erasure. The sample was prepared and presented to three self-organising maps using the same method as before, with parameters set as in Table 5.1. The results obtained (and the features extracted) as shown in Figure 5.35.

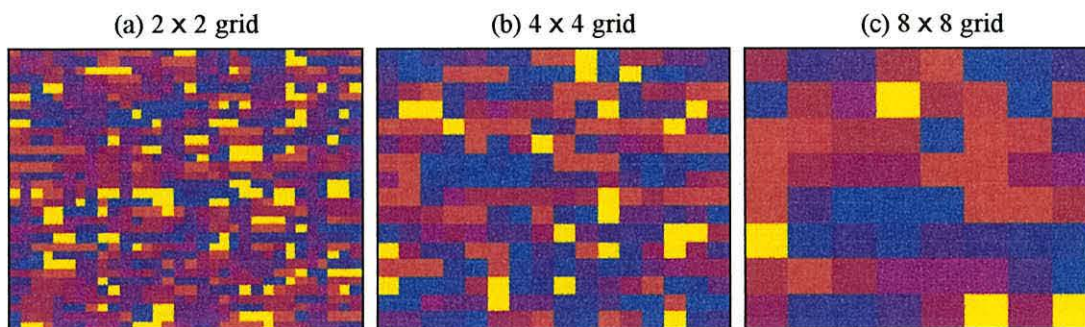


Figure 5.35: *Neural network results with various grid sizes*

The presence of many vortices of variable sized is again observed, which is not unreasonable in this type of sample. It is interesting to notice that with a 2x2 grid, the vortices are bundled together, which tends to suggest that there are several small vortices grouped together at the domain boundaries.

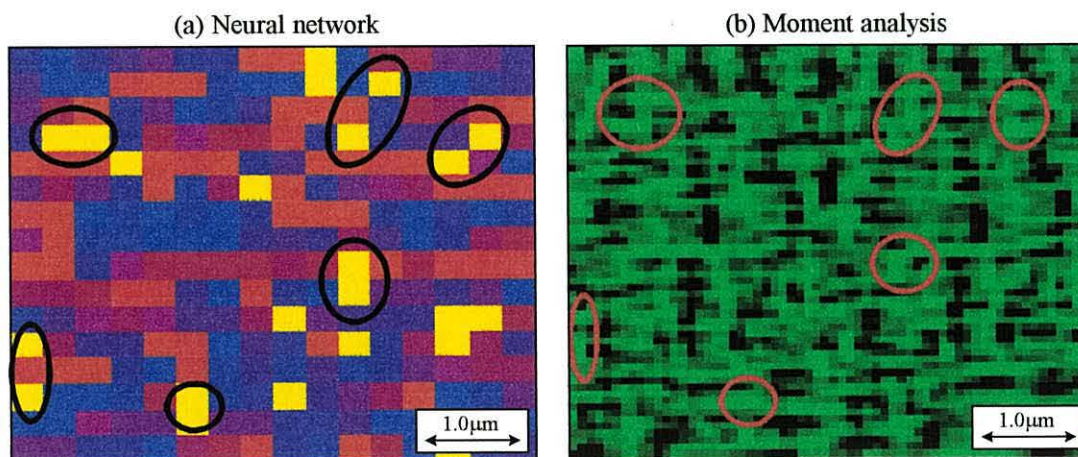


Figure 5.36: *Comparison of results*

From the above image it is observed that the results obtained using a neural network method are similar to that obtained with the moment calculation method, and a further comparison to the original data shows that the neural network method is very effective in providing an unbiased determination of vortex structures.

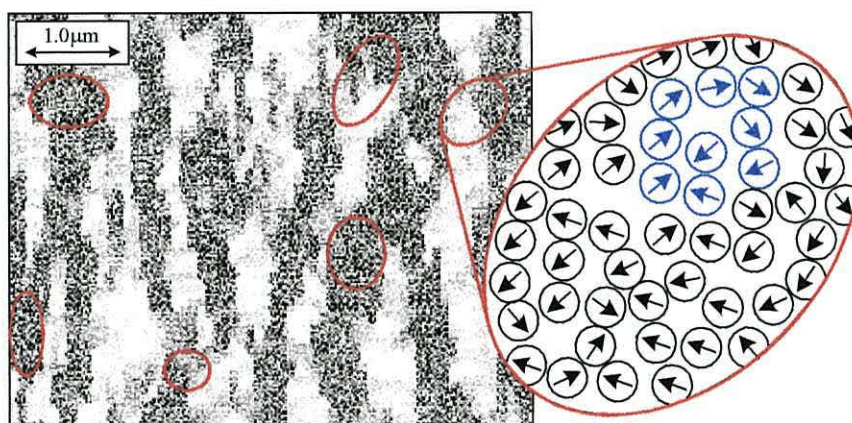


Figure 5.37: *Comparison of results with original data set*

Figure 5.37 confirms the idea that vortices tend to form at the boundaries of domains, and the stripe effect is typical of a DC erased sample.

In this section it has been shown that a neural network method is capable of identifying features such as vortices within a sample to roughly the same degree of accuracy as other conventional methods, and in some cases manages to highlight areas which were not apparent with the moment analysis method.

In the following sections, the same methods are applied to simulated MFM images. If these are successful, then they would suggest that it would be possible to extract features from real experimental MFM images, something which could prove to be very useful practically.

5.5 The MFM Model

The basics of the magnetic force microscope has already been discussed in Chapter 1. However, in this chapter an introduction is given to an MFM simulation [63], from the study conducted using a model which was developed at the computational magnetism group [64]. The model is partly based on the theoretical model of the deflection of a cantilever arm in an AFM experiment [65]. This calculates the energy which arises from a dipolar interaction by one atom in the sample with one in the tip:

$$E = - \int_{V_{tip}} \int_{V_{sample}} \vec{M} \cdot \vec{H} dV_{sample} dV_{tip} \quad (5.12)$$

where

\vec{M} = magnetisation of the tip

\vec{H} = stray field of the sample

V_{tip} = volume of the tip

V_{sample} = volume of the sample

The force of interaction lies along the direction perpendicular to the surface of the sample (z-axis), and is given by:

$$F_z = - \frac{\partial E}{\partial z} \quad (5.13)$$

When the magnetisation of the tip is perpendicular to the sample surface then the information about the magnetic domains is better observed than when it is parallel. The model discovered that the size and shape of the tip also affected the magnitude of the detected force. Wadas and Grütter [66] outlined an analytical approach to the model using a frustum shaped tip. The force due to a frustum shaped tip is calculated by discretising the volume, by placing a number of cuboids along the vertical direction of the tip.

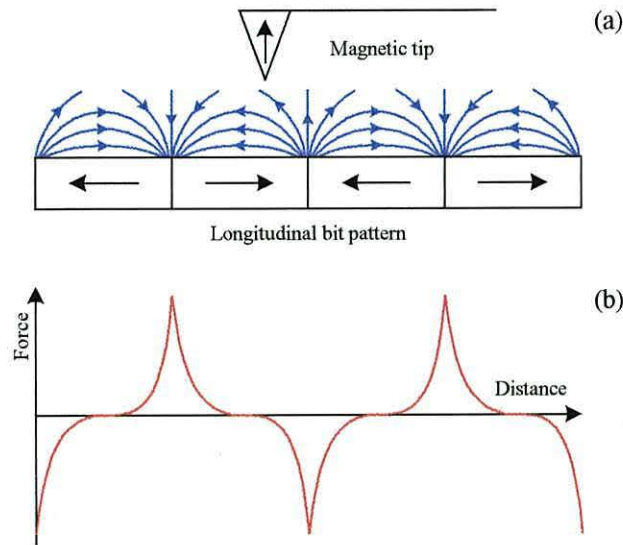


Figure 5.38: (a) The stray field above a medium (b) The force exerted on the tip

From Figure 5.38 it can be seen that the force exerted on the tip by the stray fields is at a maximum at the transition between the bits and is at a minimum at the centre of the bits. The reason for this is due to the vertical component of the stray field being at its highest at the transition and at its lowest at the centre of the bits.

The MFM simulation relies on the force that is created when a magnetised sample interacts with a magnetised tip. The two main steps to finding this force is the calculation of the interaction between the tip and sample space, and then overlaying the models into the sample space. The calculation to determine the magnetic field strength from the sample at the tip is a sixth fold integral (as the integration has to be conducted over the entire sample and the tip):

$$\vec{H} = \int_{sample} \int \int \left[\int_{tip} \int \int \left(\frac{3(\vec{\mu} \cdot \vec{r})\vec{r}}{r^5} - \frac{\vec{\mu}}{r^3} \right) dV_{tip} \right] dV_{sample} \quad (5.14)$$

where

$\vec{\mu}$ = the magnetic moment of the tip

\vec{r} = the vector between the sample and the tip

Equation 5.14 states that the effect of the whole tip in 3-dimensions must be calculated on the sample (and done over the entire sample volume). As the tip and the sample can be considered to consist of discrete cubes, the effect of every cube in the tip with every cube in the sample must be calculated, as illustrated in Figure 5.39 (it is assumed that there are no interactions between dipoles constituting the tip).

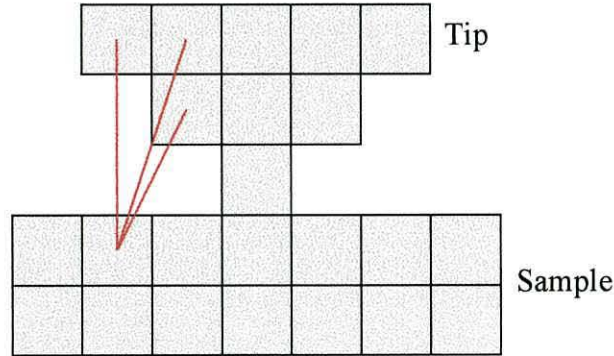


Figure 5.39: Interactions between cubes in the tip with cubes in the sample

The force exerted on the tip can be calculated using:

$$F = \int \nabla (\vec{H} \cdot \vec{M}) dV \quad (5.15)$$

where

\vec{H} = the magnetic field strength

\vec{M} = the magnetisation of the sample

F = the force exerted on the tip (over the entire sample)

This can also be represented as:

$$F_x = \frac{\partial H_x}{\partial x} M_x + \frac{\partial H_y}{\partial x} M_y + \frac{\partial H_z}{\partial x} M_z \quad (5.16)$$

$$F_y = \frac{\partial H_x}{\partial y} M_x + \frac{\partial H_y}{\partial y} M_y + \frac{\partial H_z}{\partial y} M_z \quad (5.17)$$

$$F_z = \frac{\partial H_x}{\partial z} M_x + \frac{\partial H_y}{\partial z} M_y + \frac{\partial H_z}{\partial z} M_z \quad (5.18)$$

where M_x , M_y and M_z are the components of the sample magnetisation (in the thin films considered here, M_z is zero). The important component of the force is F_z , which is the vertical component. The partial differentials are determined by analytically calculating the differential at the points where the magnetic field strength is known and then interpolating to give the partial differential at the required point.

5.5.1 MFM Sample Images

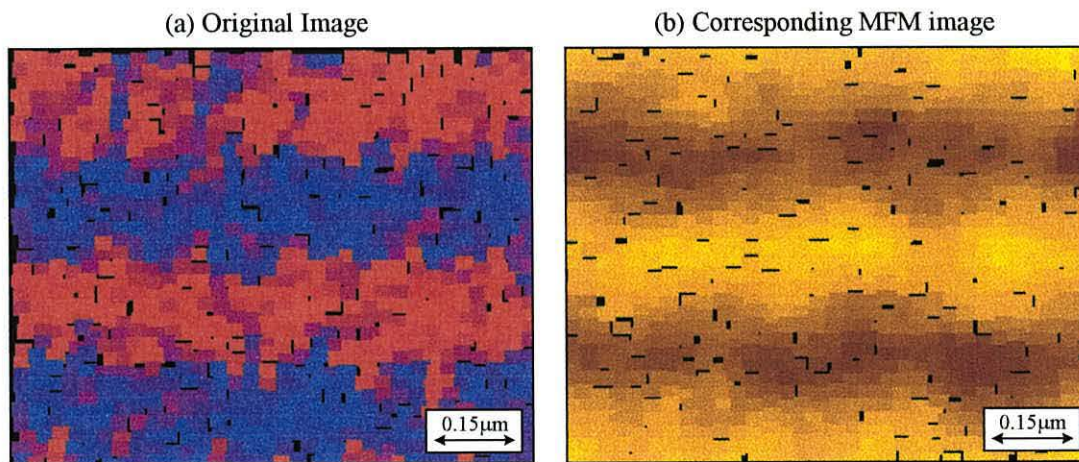


Figure 5.40: *Effect of applying an MFM simulation to a sample with 4 written bits*

The images in Figure 5.40 show the effect of applying the MFM simulation (from Section 5.5) to the sample with 4 written bits (640nm^2). The tip shape was a pyramid consisting of 10nm^3 cubes (7000Oe saturation magnetisation) and the sample space was $300 \times 300 \times 100\text{nm}$ and discretised into 10nm cubes (14000Oe saturation magnetisation).

5.6 Investigation of Optimal MFM Scanning Height

The MFM image in Figure 5.40 was produced at a scanning height of 200nm. However, this height can be varied and is found to have a considerable effect of the quality of the output image. It is known that when the cantilever is close enough to the surface its response is determined primarily by short range forces on the tip, which are coupled to the surface topography. When the tip is positioned farther from the surface (where the short range forces are negligible) the cantilever responds to long range, magnetic force gradients acting on the tip. Therefore, in order to produce an image which can be used for feature extraction, the optimal scanning height must be found.

In this section the effect of varying the MFM tip scanning height is investigated, by creating a series of images and applying them to a self-organising map. By comparing them with the original magnetisation map, it is then possible to calculate an error function associated with each image, and deduce a region of scanning heights which produce the best MFM images.

The investigation was conducted on a magnetic sample 640nm×640nm in size, 1024 grains and with four written bits. The tip of the MFM was set at 20nm intervals from 0nm to 220nm.

5.6.1 Network Optimisation

In this investigation the size and parameters of the network is not that important; the input number of neurons is pre-determined by the resolution of the input MFM image. As it is not of interest to find any particular features here (merely an error value), then the size of the output layer becomes less relevant too. However, for practical reasons, this will be set to 8 neurons (the value used in Section 5.4). As the resolution of the MFM images are identical to those of the magnetisation maps, it is likely that the training required will be the same too (approximately 250 epochs).

5.6.2 Results

The system was set-up with identical parameters for each simulation, barring the sample height. The images produced are shown in Figure 5.41.

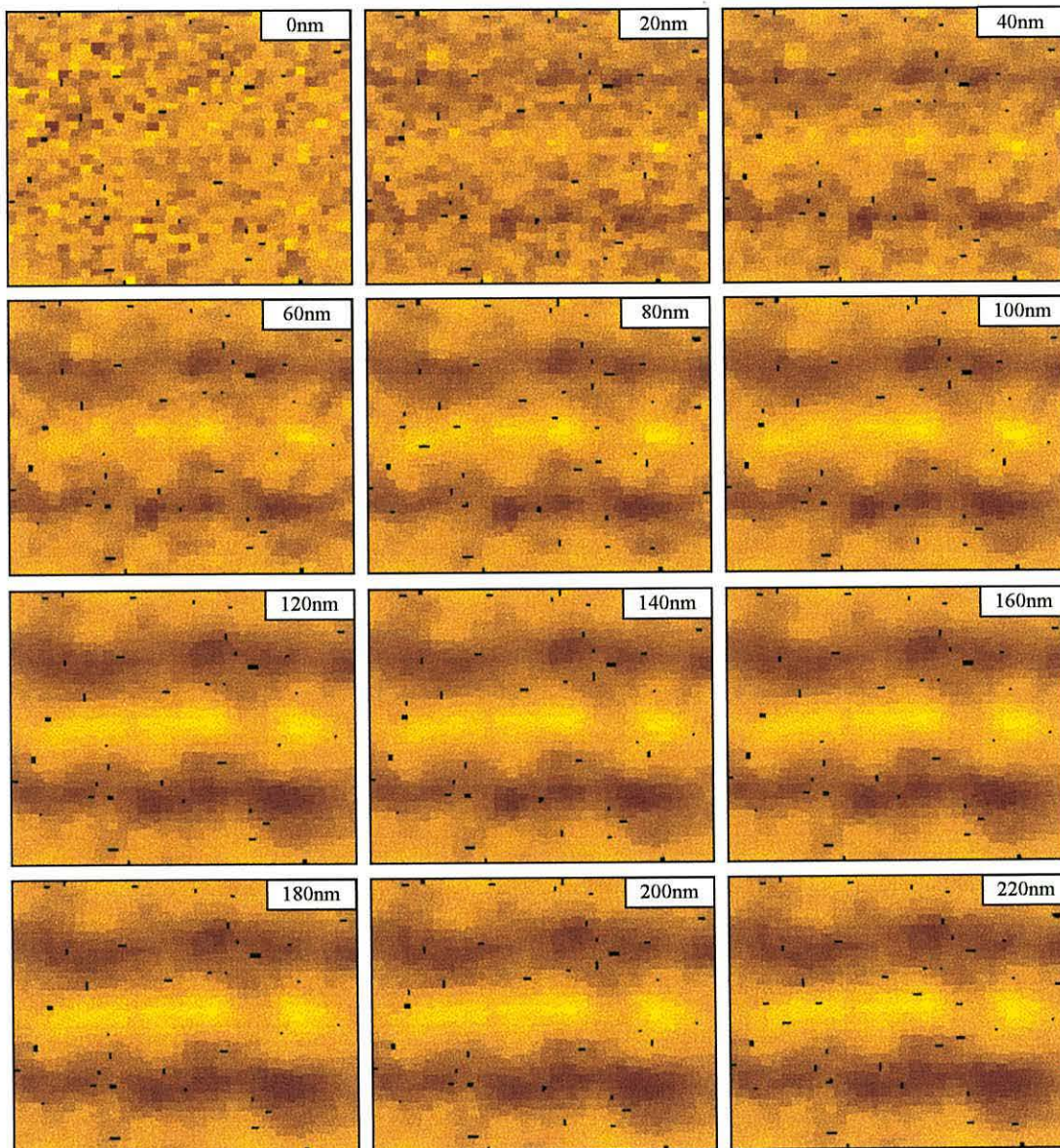


Figure 5.41: *MFM images produced at different scanning heights*

From the above figure, it can be observed that some scanning heights produce better images, but by applying them to a neural network it is possible to calculate an error value for each image, and thus predict a narrow range of optimal scanning heights.

This was done by applying the various images to the neural network, and observing how the data was organised on the self-organising map. The output was then compared to original raw data, based on the ability of the neural network to accurately group different features. The results are presented in Figure 5.42, from which it can be seen that the accuracy is at a maximum in the region where the images appear to be visually better (which is what would be expected). However, the neural network has also been able to reduce the optimal scanning height region to give a more accurate estimation.

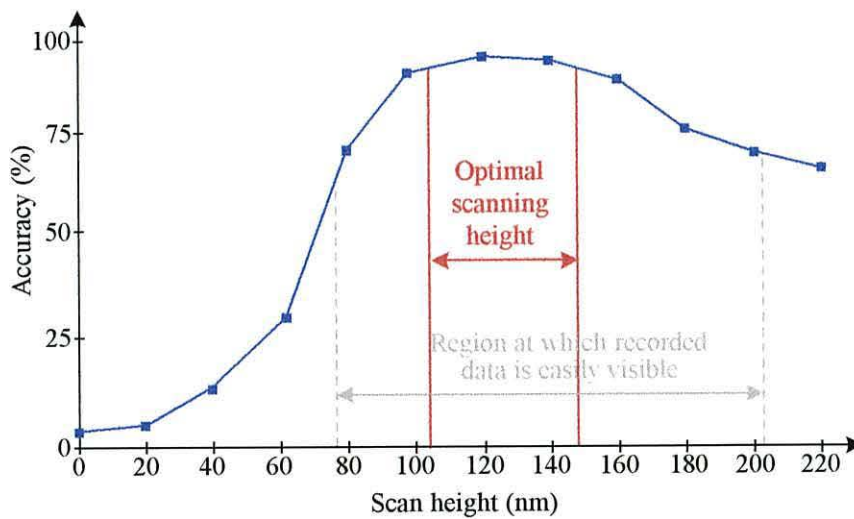


Figure 5.42: *The accuracy obtained at various scanning heights*

It can be concluded that the optimal scanning height is in the region of 100nm-150nm. It should be noted that this value is dependent on the size of the grains in the magnetic sample, and is not a 'magic' value for the scanning height of any system. In the next section some features in MFM images are investigated, and use will be made of the findings from this section.

5.7 Feature Extraction From MFM Images

In Section 5.4 features such as vortices were successfully extracted from magnetisation maps. In reality it is not always possible to use such images, especially in experimental cases, where methods such as magnetic force microscopy are better suited. However, unlike a magnetisation map, it is envisaged that the MFM image doesn't contain adequate data to extract useful information about the underlying structure of the sample, and therefore alternative methods to those used with magnetisation maps will have to be developed. In this section, two different methods are proposed for finding vortices in an MFM image, each one approaching the problem from a different perspective:

- Using the MFM simulation, work back to an approximate magnetisation map, and use neural network methods discussed in Section 5.4 for the task of finding vortices.
- Create an MFM image of a simple vortex, and use a neural network to compare features found in a series of MFM test samples.

Each method will be discussed in greater detail in the following sections, together with the feasibility of using that particular method.

5.7.1 Method 1

In the first method the idea is to work back to the magnetisation map (or at least an approximate map) from the MFM image, using the simulation described in Section 5.5, but in 'reverse'. However, the problem with this approach is that during the MFM simulation the orientation of the moments is not preserved in the creation of the MFM image. The MFM image contains information about the position of the moments and the force which they exerted on the tip; and since a moment pointing in one direction might have exerted the same force on the tip as a moment pointing in

another direction, it would be almost impossible to determine from the force alone which direction the moment was pointing towards (recall from Figure 5.38 that force is exerted on the tip at transitions only). It is therefore considered too complex to attempt this method.

5.7.2 Method 2

The second method involves the creation of a vortex to be used by the neural network for training purposes. A vortex consisting of a few grains was created in a micromagnetic structure which had all its moments aligned in parallel (Figure 5.43(a)), and even though such a ‘perfect’ vortex would hardly ever be observed in a real sample, it gave an insight as to what features (i.e. shading patterns) to look for in the corresponding MFM image (Figure 5.43(b)).

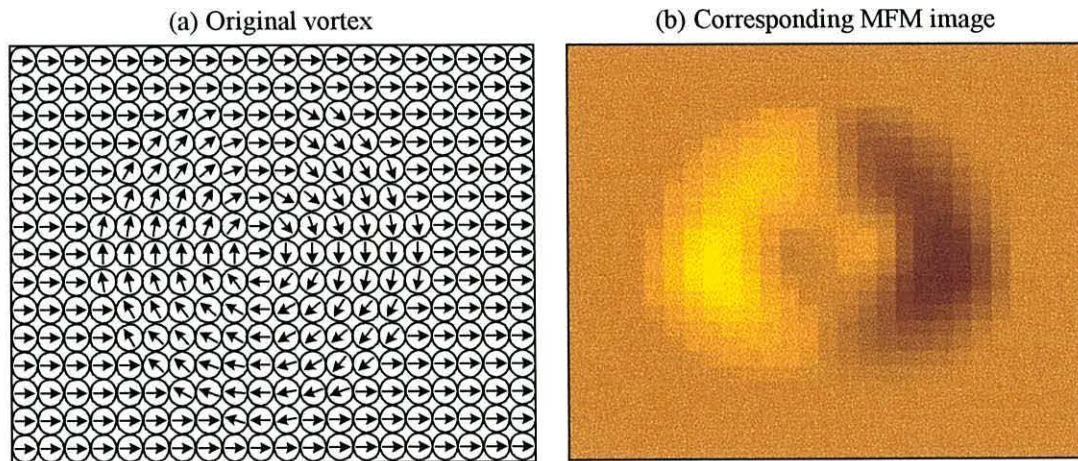


Figure 5.43: *MFM image of a vortex structure*

5.7.2.1 Data Preparation

Before the MFM images can be presented to the SOM, it is necessary to arrange the pixels into a grid-like fashion (so as they align with the neurons in the SOM). This is illustrated in Figure 5.44, from where it can be seen that there is only a slight loss in quality.

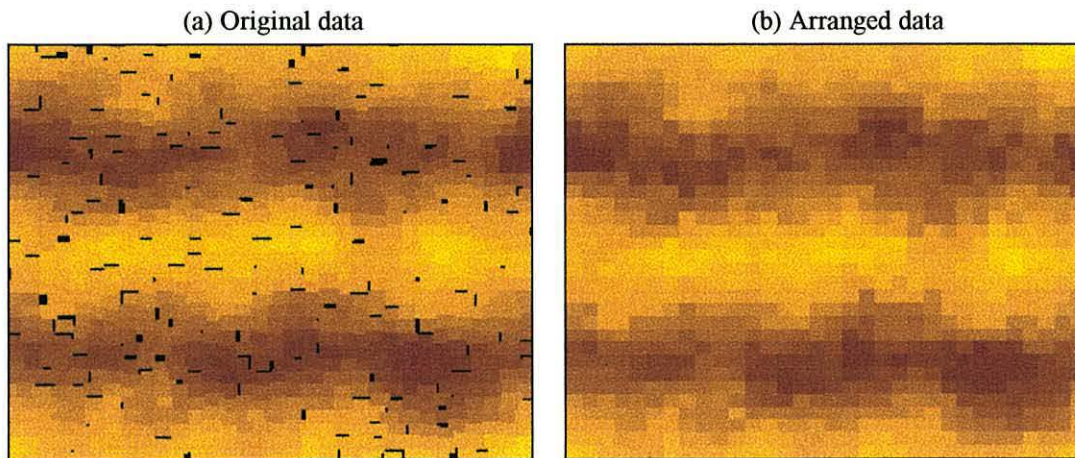


Figure 5.44: Arrangement of grains into an orderly grid

The same process was conducted on the five test samples. As the test vortex is about 2-4 grains in diameter, the MFM images were sampled into regions of 4, 15 and 64 grains (grids of 2×2 , 4×4 and 8×8), using the method outlined in Section 5.4.2. The reason for doing this is to identify features of variable sizes e.g. vortices which may consist of several grains in diameter. Figure 5.45 shows how the MFM image is sampled using this method.

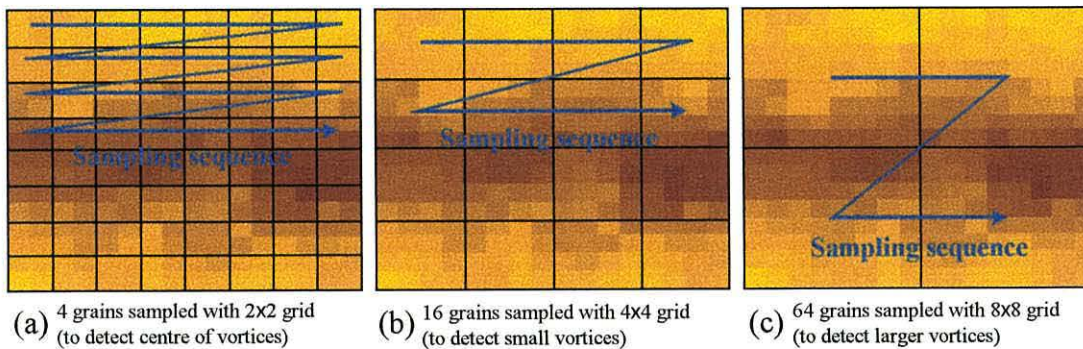


Figure 5.45: Sampling an MFM image with 2×2 , 8×8 and 16×16 grids

Each grid now represents a series of pixels in the MFM image. The neural network is capable of placing these grids (only one grid size can be applied to each neural network) into groups in an organised manner dependent on the features within them (in this case the number of groups has been restricted to 8). The features which are specific to each group can be observed (Figure 5.46).

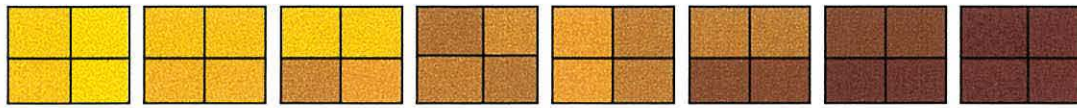


Figure 5.46: *The features present in the 8 groups in the SOM (using 2×2 grids)*

From the above figure it is rather difficult to observe exactly what each group represents (but becomes clearer when they are plotted later). However, as only vortices are being looked for, which have a known type of structure (see Figure 5.43(b)), it is only a small matter of comparing the eight groups (and other generated groups) to see which is the closest match to the vortex image.

5.7.2.2 Network Optimisation

Each group has then been assigned a specific colour and replotted in their original positions. By doing this it is possible to observe what features are present in different regions of the original MFM image (Figure 5.47). It should be noted that these colours do not represent the value of the MFM force, but rather an allocated group.

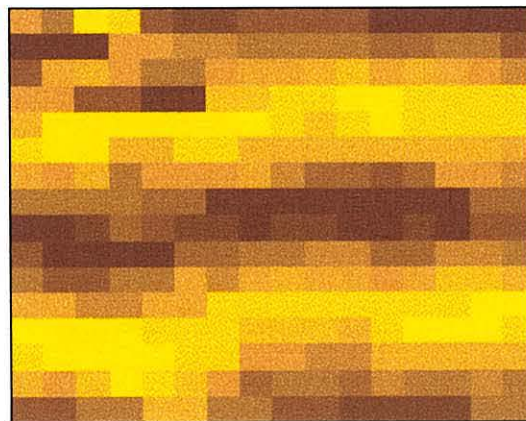


Figure 5.47: *Replotting the SOM groups in their original positions*

The amount of training required by the neural network varies from problem to problem, therefore it is necessary to repeat the process of determining how much

training is required. Training can be terminated when the mean squared error has reached about 0.1% of its initial value and fails to decrease further.

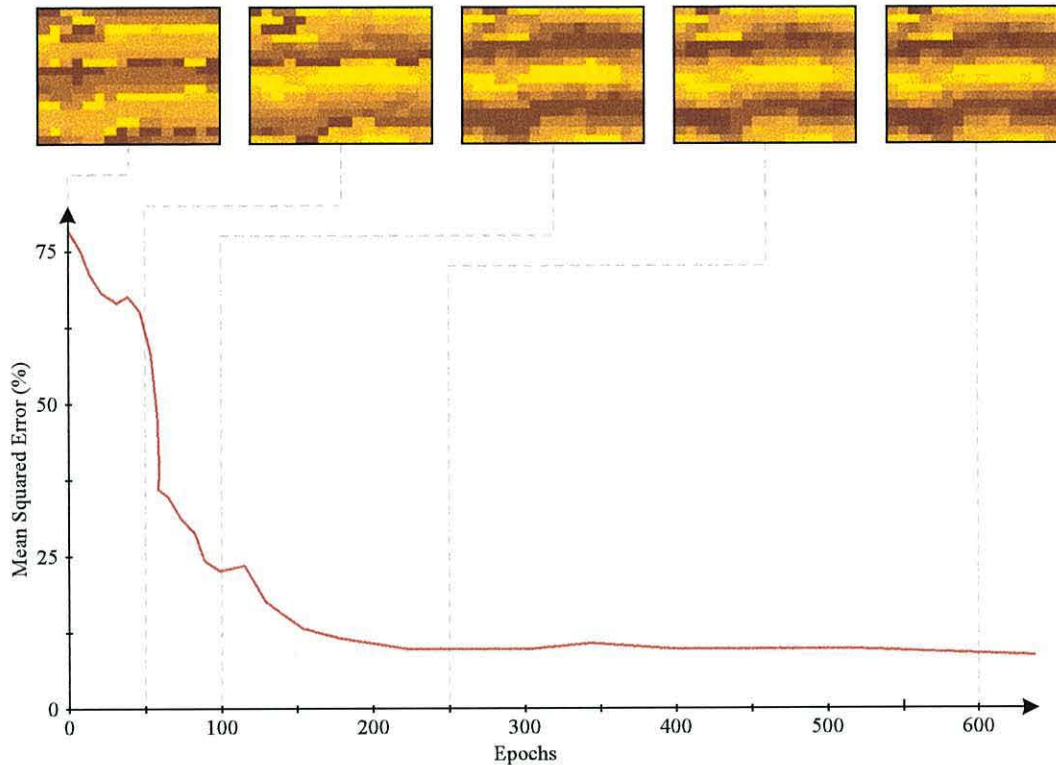


Figure 5.48: *The error values w.r.t training and the outputs obtained*

Figure 5.48 shows how the error decreases with training (for a system with 8 groups), and shows the outputs obtained from the self-organising map at different periods during the training. From these results, and other data it can be concluded that the neural network can be sufficiently trained in about 200 epochs, and shows no improvement with further training. It should be noted that this amount of training is only relevant to a system of this size (1024 grains), and an increase in system size would require more training. However, increasing the system size makes it difficult to visualise the output data.

From these investigations, it has been determined that for a system of this size, 3 neural networks can be constructed to recognise features of various sizes. Their parameters are summarised in Table 5.2.

Sample Grid	Input Neurons	Classification Groups (Output Neurons)	Training (Epochs)
2×2	4	8	240
4×4	16	8	220
8×8	64	8	200

Table 5.2: *Optimal neural network parameters*

5.7.2.3 Results

For simplicity, identical samples to those used in previous sections will be used with the above self-organising maps to observe features in the MFM images. They will then be compared with results obtained using methods discussed in Section 5.4.1.1 for calculating the presence of vortices. The samples are:

Sample 1: 1024 grains with 4 recorded bits, $h_i=0.1$, $C^*=0.2$

Sample 2: 1024 grains (0% LC) with 10 recorded bits, $h_i=0.5$, $C^*=0.1$

Sample 3: 1024 grains (15% LC) with 10 recorded bits, $h_i=0.5$, $C^*=0.1$

Sample 4: 4096 grains AC erased state, $h_i=0.5$, $C^*=0.1$ and 0% LC grains

Sample 5: 4096 grains DC erased state, $h_i=0.5$, $C^*=0.1$ and 0% LC grains

Sample 1

The sample was prepared and presented to three self-organising maps (the method is explained in the previous sections), with parameters set as in Table 5.2. The results obtained (and the features extracted) as shown in Figure 5.49.

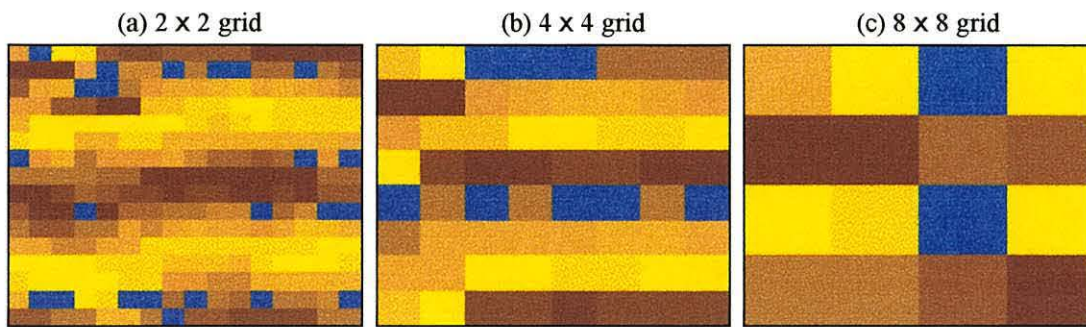


Figure 5.49: *Neural network results with various grid sizes*

From Figure 5.49, it appears that the neural network method has extracted the possibility of numerous vortices, most of which lie along the bit transitions. Some of the larger vortices (of about 4 grains in diameter) have been detected, but it is not entirely clear why some of the very large vortices have been detected (when it is very unlikely that they exist). The probable reason for this is due to the loss of data during the simulation (and thus the inability to distinguish between two or more neighbouring vortices).

In Figure 5.50 these results are compared to results obtained with the moment analysis, and the vortices detected overlaid onto the original data.

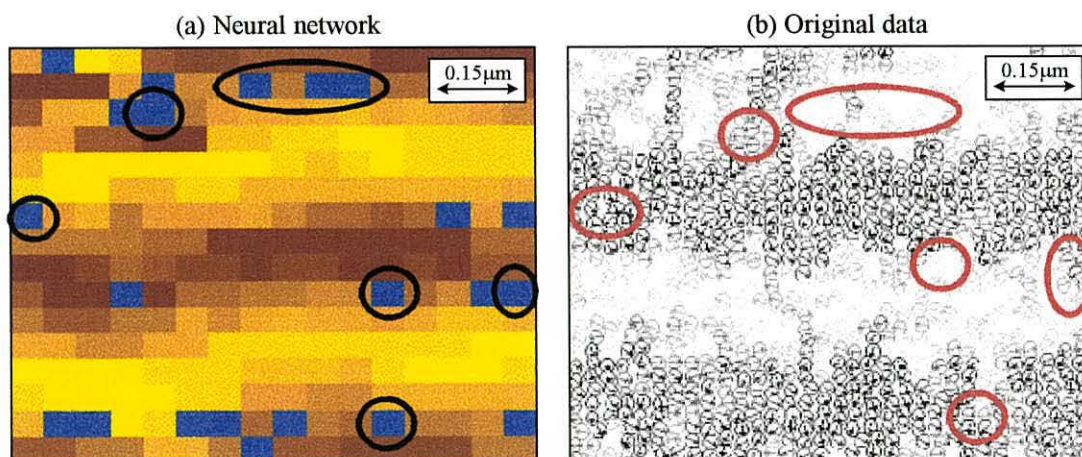


Figure 5.50: *Comparison of results*

From these results it can be observed that there are some similar regions in the sample with the probability of vortices. It is interesting to notice that all three methods which

have been investigated give approximately the same results, even if not all vortices appear in each set of results. This is however a significant development. It is known that the determination of magnetisation configurations from MFM configurations is an inverse problem which has no direct mathematical solution.

The neural network approach, rather than looking for a rigorous solution of the inverse problem instead uses local topological features to characterise the underlying magnetisation structures. Although this cannot give solutions which are unambiguous, the investigations here give some evidence that the features of the MFM image have sufficient information to determine with reasonable certainty the underlying magnetisation structure. Although this requires further investigation, the results here are of sufficient interest to justify it.

Sample 2

The sample was prepared and presented to three self-organising maps (the method is explained in the previous sections), with parameters set as in Table 5.2. The results obtained (and the features extracted) as shown in Figure 5.51.

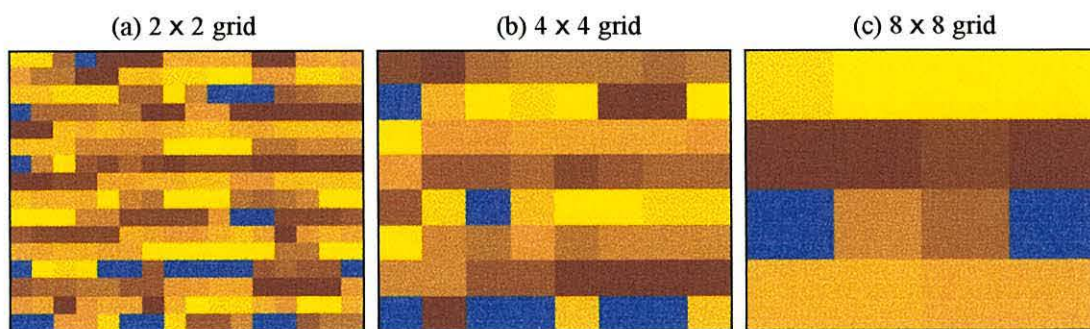


Figure 5.51: *Neural network results with various grid sizes*

The SOM seems to be able to extract quite a few vortices from the MFM image. In Figure 5.51 (a) they roughly lie at equal intervals vertically, which is expected on a sample with regular written bits. The identification of very large vortices (Figure 5.51 (c)) is rather surprising, as the written bits are only a few grains wide.

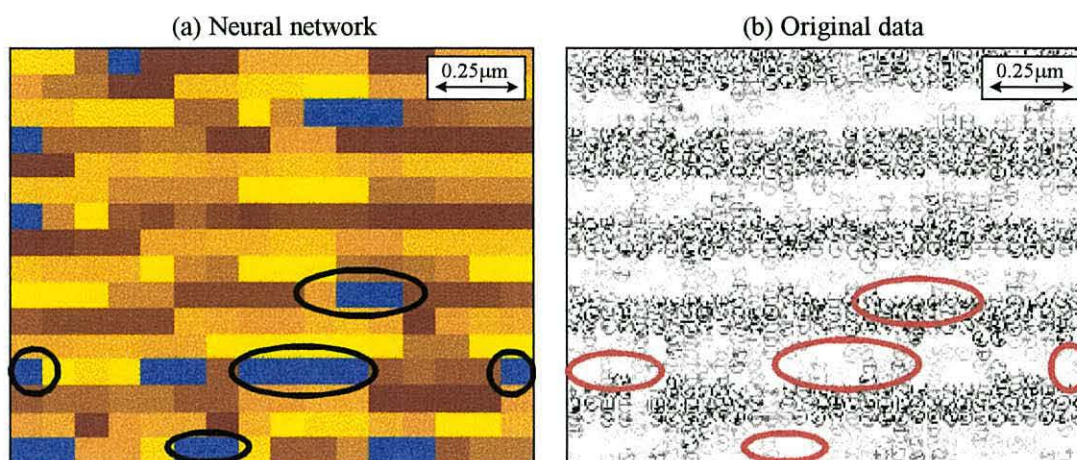


Figure 5.52: Comparison of results

In Figure 5.52 the MFM results are compared with the results obtained from the moment analysis. It is interesting to notice that the MFM results were better than those obtained with the magnetisation map.

Sample 3

The sample was prepared and presented to three self-organising maps (method is explained in previous sections), with parameters set as in Table 5.2. The results obtained (and the features extracted) as shown in Figure 5.53.

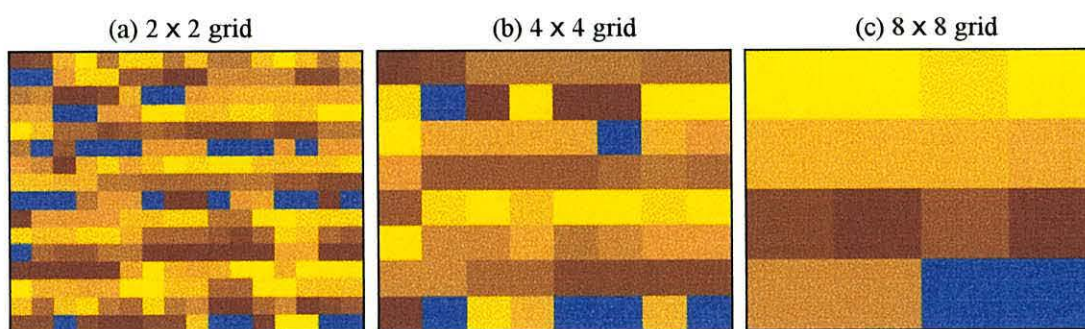


Figure 5.53: Neural network results with various grid sizes

This sample has a series of written bits, but with the introduction of LC grains. In Figure 5.53 (a) the vortices seem to roughly lie at equal intervals vertically, which is expected on a sample of this type. These results can then be compared to those obtained with the moment analysis method, as they should produce similar results.

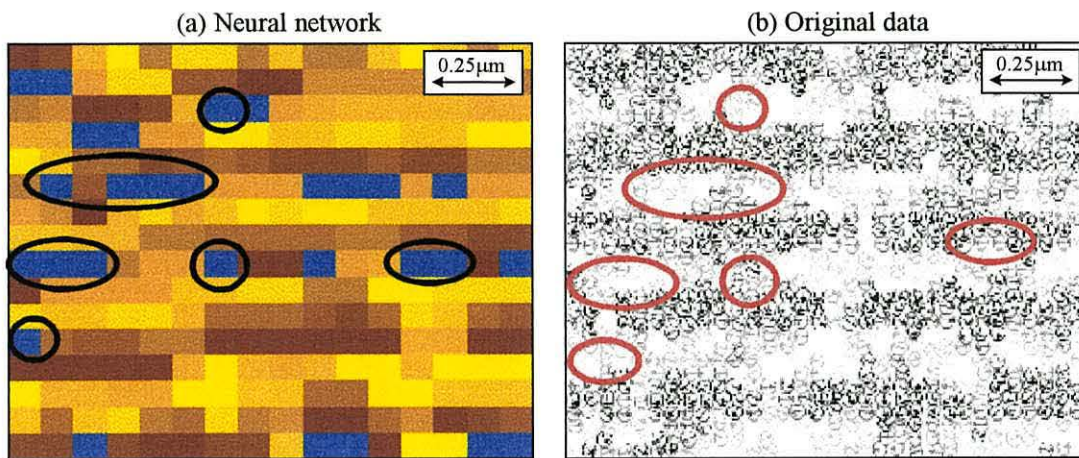


Figure 5.54: Comparison of results

From Figure 5.54 it can be observed that there are some similar regions in the samples with the probability of vortices. The neural network method seems to have picked up extra vortices, which were not found with the magnetisation map.

Sample 4

The sample was prepared and presented to three self-organising maps (method is explained in previous sections), with parameters set as in Table 5.2. The results obtained (and the features extracted) as shown in Figure 5.55.

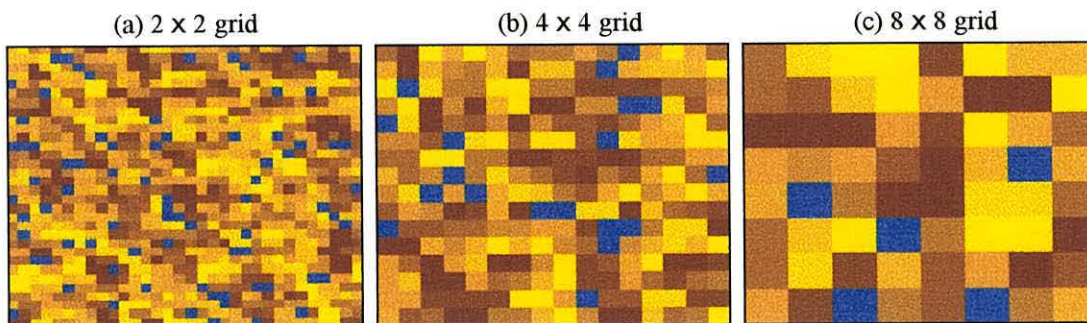


Figure 5.55: Neural network results with various grid sizes

The neural network approach seems to have worked very well with this particular example. The presence of several vortices (of varying sizes) have been detected, which is what is expected from this type of sample.

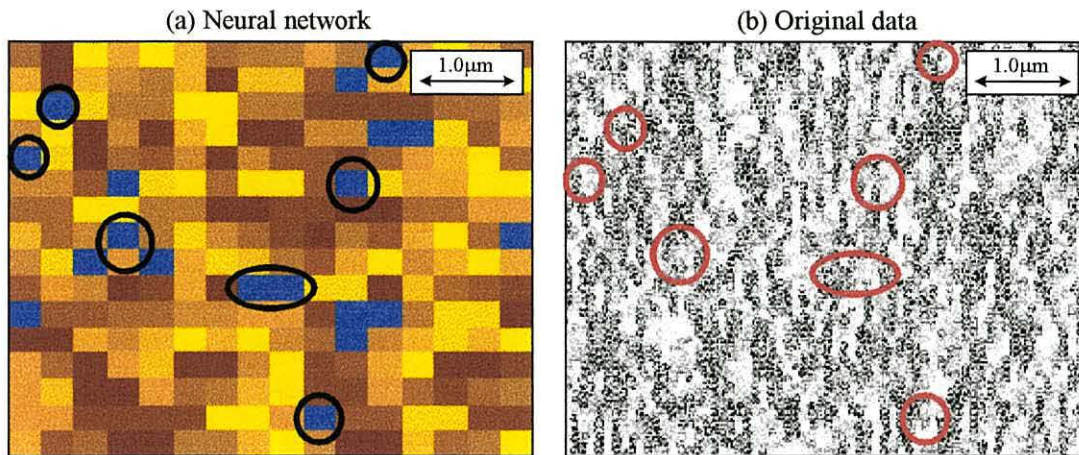


Figure 5.56: Comparison of results

In Figure 5.56 the comparison with the original data looks very promising, as it shows that the neural network method works rather well with larger sized vortices (which are harder to detect, as there are so many variations), and there are several similar features which have been detected using both methods.

Sample 5

The sample was prepared and presented to three self-organising maps (method is explained in previous sections), with parameters set as in Table 5.2. The results obtained (and the features extracted) as shown in Figure 5.57.

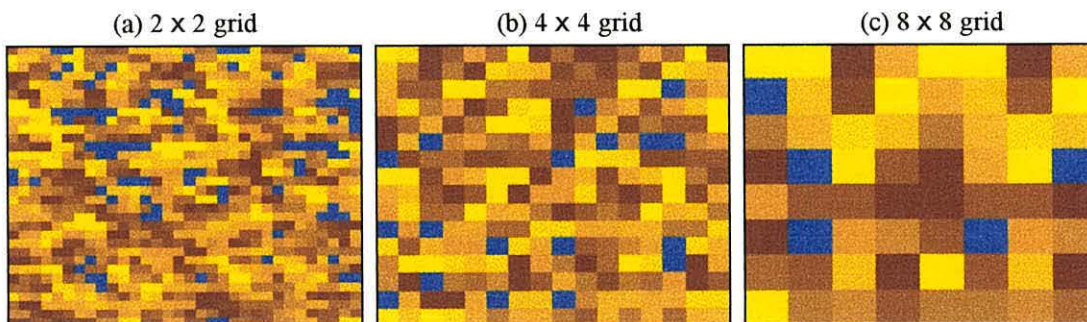


Figure 5.57: Neural network results with various grid sizes

In this DC erased sample the presence of several vortices has been detected, and just as with the magnetisation map investigation, there are several clusters of small vortices. This similarity between results reinforces the feasibility of using a neural network method. These results can then be compared to those obtained in the section with the moment analysis method.

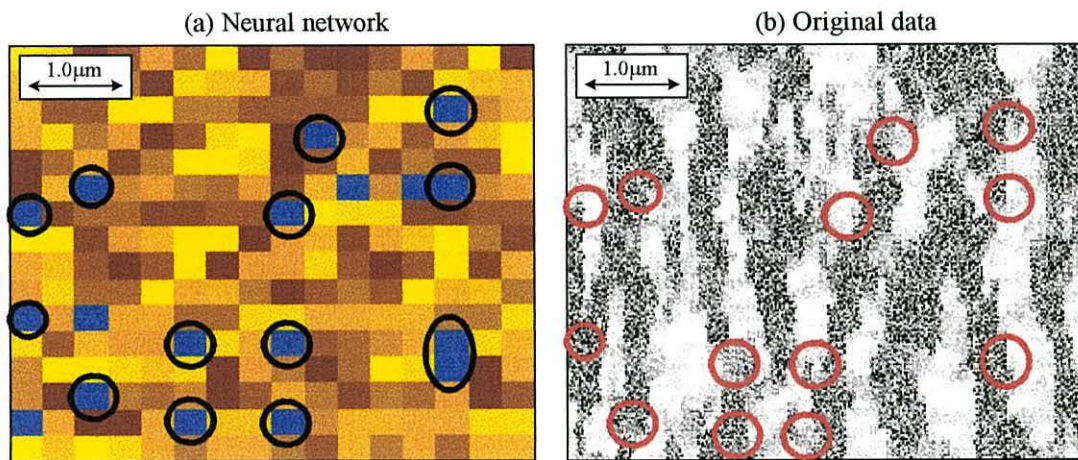


Figure 5.58: *Comparison of results*

The comparison with the moment analysis method was very encouraging. From Figure 5.58 it can be observed that there are several similar regions in the sample with the probability of vortices; almost every vortex extracted by the SOM actually exists in the original data (and the majority are to be found in the transitions between domains).

Again the neural network analysis of the MFM image has produced reasonable vortex assignments. This implies that although, as mentioned previously, the MFM image does not contain sufficient information to solve the inverse problem exactly, but there is adequate information for the successful observation of vortices.

5.8 Summary

In this chapter methods of extracting features from magnetisation maps and MFM images have been investigated. It was discovered that the magnetisation maps contained enough data to produce satisfactory results, which compared well with other conventional methods. However, in some cases it is possible that there is too much data for the neural network to generalise, and hence the reason that some of the vortices in the original data are not being recognised.

The extraction of features from MFM images was initially envisaged as being rather difficult, which is thought to be due to the lack of important information in the MFM image (which is lost during the simulation, and not present in the first place in an experimental image). Two methods were considered to overcome the problem. The first one involved working back towards an approximated magnetisation map - a near impossible task. The second method involved the generation of vortices and their corresponding MFM images, which would then be used by the neural network to recognise real vortices in the test MFM images.

The results obtained were surprisingly good, especially as it was thought there was insufficient data present. However, it is this lack of data which has proved to be the reason for such good results. By being presented with this amount of data, the neural network has been able to generalise better, and hence been able to extract vortices which were undetectable with the magnetisation map.

It should be stressed that these are initial results only, and much more work is required to develop the neural network method for this type of application. However, overall the results have been impressive, and given further investigation could be improved yet again.

Chapter 6

Conclusions & Further Work

6.1 Conclusions

Since their development in the early 60s, neural networks have been used in a wide variety of applications, and through their continued improvement have proved to be very successful. However, their application in magnetic problems has been rather slow, and not much work has been done on them. Inverse problems have proved very difficult in the past, and as a result have received little attention. It is for that reason that the problem was approached with caution, and hoped that neural networks could be used effectively to solve some of the problems which had been considered. Their application in optical problems was less novel, as some work had already been done on other aspects of it by George *et al* [67], with successful results.

The first problem that was investigated was the determination of anisotropy field distributions in transverse susceptibility data. This area of research is proving to be very important, as it gives us information about the switching field distribution, which then determines the minimum transition width achievable in a medium. Much work has been done on this subject in the past [32, 33, 34]. However, all numerical methods used in these investigations relied on mathematical calculations which were very complex computationally.

Finding the anisotropy field distribution often took several hours to solve on a fast parallel processor computer. It was hoped that by using a neural network technique it would be possible to achieve similar results in a fraction of the time. Numerous data sets were generated by integrating the kernel with a known distribution function (both log-normal and Gaussian), and by varying this distribution function, slightly different results would be obtained. As it was possible to generate this amount of data (with known parameters) it was feasible to use a supervised learning method, with a multi-layer perceptron network. In essence, the neural network was trained to relate individual transverse susceptibility curves with a particular distribution function.

Much of the investigation was involved with the determination of the optimal number of data sets needed to train the network, as insufficient data would give poor results and excess data would prevent the network from generalising. The network parameters (network configuration) were carefully selected, together with the amount of training. It was discovered that the results could be improved dramatically by using three separate networks instead of one combined one. The optimal neural network was found to consist of 8-16 hidden neurons in 1 hidden layer, using 100 training sets, and be successfully trained in approximately 40-50 epochs.

Once the neural network had been sufficiently trained it was possible to apply test data (i.e. data which the network was not familiar with). Having achieved accurate results with these, experimental data was applied (using a CrO₂ sample) on a network trained with log-normal and Gaussian distribution functions. The distribution function was determined with a mean value of 0.292 with a standard error of 6×10^{-3} . This compared favourably with results obtained using the simulated annealing method (which gave a result of 0.291). The network also determined with a 99.5% certainty that the experimental H_K distribution was of the log-normal form.

The recalculated results fit well over most of the curve, thereby validating the method, but disagree in small fields. However, this is not unreasonable as it should be recalled that the kernel used in the theoretical calculations is based on the assumption of the Stoner-Wohlfarth (S-W) coherent mode of magnetisation reversal. Here the S-W

model is a good approximation as long as the magnetisation remains close to saturation, which is normally the case when determining H_K values of 2kOe or greater. A better agreement might be found if a more realistic model were to be used to recalculate the results for comparison with the experiment. However, the deconvolution process is not very sensitive to the small field region, and the estimates of the H_K distribution is probably not affected. Therefore, it can be concluded that a neural network approach is feasible to determine the distribution of H_K when $0.1 < \sigma < 0.65$, at which point the accuracy is comparable to conventional methods.

Secondly, a similar situation was investigated, but applied to an optical problem. This involved the determination of important compact disc parameters from the diffraction pattern observed when a laser strikes the disc. This was an interesting problem which had the opportunity of being applied in an industrial on-line testing system. Due to the similarities to the transverse susceptibility problem, this was approached in very much the same manner.

Using a multi-layer perceptron neural network, which was chosen due to the possibility of creating numerous sample data sets, theoretically generated diffraction data were generated, and several simulations were performed in order to determine the optimal settings. Again, it was determined that one neural network was insufficient, and had to be separated into three networks (something which proved to be very successful, as the accuracy was observed to improve by about 40%), consisting of 8-16 hidden neurons in 1 hidden layer, using 500 training sets, and trained in approximately 50-100 epochs.

The accuracy obtained using theoretical data gave much confidence in the application of 'experimental' data to the trained network. Results showed that the values of the track pitch, pit width and pit depth could be determined to a high degree of accuracy ($\pm 3\%$, $\pm 4\%$ and $\pm 31\%$ respectively), even with a very noisy data set (greater than would be observed experimentally). The determination of the pit depth was rather poor, which is envisaged as being due to diffraction data being less sensitive to changes in parameters in the nanometre scale.

These factors, together with the speed at which a trained network can determine unknown parameters leads us to believe that a neural network approach could be used in an on-line system during the CD manufacturing and testing process. Further research is now being carried out by Aeronomic to investigate the feasibility of using such a method in their CD analysing systems.

Finally the analysis of magnetisation maps and magnetic force microscopy images were investigated, primarily the extraction of features such as percolation and vortices. This required the use of a completely different approach, using another area of neural networks called unsupervised learning. The reason for this was because it was not feasible to create a sufficient number of training sets (unlike the problems which were encountered previously). Use was made of a specialised neural network called a self-organising map, which can arrange specific features from the input data into organised output groups. This proved to be very useful, as the features which were being sought could be identified by searching within these groups.

After investigating various image coding and manipulation techniques, together with some pre-processing methods, a set of five data sets were prepared. Much of the work involved the determination of the optimal number of classification groups, the amount of training required, the dimensions of the network and other learning parameters. It was found that the neural networks could be trained quite quickly in about 200 epochs, and that 8 groups were sufficient to represent the majority of information in the original data sets. By sampling the images at various grid sizes it was concluded that a variety of different sized vortices could be observed.

The results obtained when using the magnetisation maps proved to be very encouraging, with results which were very similar to those obtained using moment analysis, and those obtained by studying the actual raw data. It was possible to observe the presence of small vortices of various diameters. However, it was not possible to observe larger vortices (greater than about 8 grains across) because there is so much variation within a large vortex that a neural network cannot classify it into any particular group.

In preparation for the study of MFM images, it was decided to investigate the optimal scanning height of the MFM tip. This was done by plotting the simulated images at various heights using our model. The results may visually look quite obvious at first, but by using neural network techniques it was possible to reduce the range of scanning heights, and suggest an optimal value more accurately.

Having found an optimal scanning height and used a variety of pre-processing techniques in the previous investigations, the application of MFM images seemed to be feasible. However, at an early stage some problems were encountered, mainly that considerably less data was being dealt with, as important information about the orientation of the grains was missing (this was readily available in the magnetisation maps). This lack of data let us to investigate another approach.

Several artificial vortices were created (of various diameters), and their corresponding MFM images created. This gave an idea as to what an MFM image of a vortex looked like, and could thus inform the neural network to seek these type of features from the MFM image of the test data. Our results seemed quite promising with some samples, with the self-organising map being able to map the positions of at least some of the vortices in the same place as those found in the magnetisation map (and the original raw data).

The reason why the neural network was able to extract the presence of vortices from the MFM images so successfully is possibly due to the lack of data present (something which was initially thought to be a disadvantage), and the neural network's ability to generalise when presented with insufficient data. It is envisaged that with further work and higher resolution MFM images that this method could be improved further still.

6.2 Further Work

The application of neural network techniques to magnetic and optical inverse problems has turned out to be very successful, and the methods used were innovative and quite novel in the field of magnetism. The results obtained has strengthened the feasibility of applying these methods to solve similar problems, and expand on the results already obtained. In this section suggestions are made as to how neural networks could be used in any future work.

The results obtained with the transverse susceptibility data were impressive, and proved that neural networks could be used to solve such problems. In magnetism there are several similar inverse problems which could be solved substantially quicker with this type of method, and some of these are in the process of being investigated. Work is currently being conducted on the transverse susceptibility problem, with the aim of further developing the theoretical kernel, and applying the method to other magnetic samples.

The investigation of diffraction patterns from a compact disc was an interesting application with practical uses in industry. The idea of being able to reduce the time taken to scan a disc and process the information was very attractive, and simulations on theoretical data suggested that it could be carried out experimentally. This neural network approach is now being investigated by Aerosonic with experimental data, with the aim of incorporating such a system in their online testing machines to be used in industry during the CD manufacturing process.

The extraction of useful information from magnetisation maps and MFM images was successful, and the results obtained were quite impressive. The ability of a neural network to extract features from images which hardly contained enough data for the human eye to see has prompted suggestions as to how this approach can be expanded to solve similar problems. The work is to be investigated further in the near future on additional experimental samples.

References

- [1] B.D. Cullity, *Introduction to Magnetic Materials* Addison-Wesley Publishing Company (1972)
- [2] W.K. Westize, *Studies on Magnetic Recording* Philips Res Dept **8** 148 (1953)
- [3] W.H. Meiklejohn and C.P. Bean, *New Magnetic Anisotropy*, Physical Review **102(5)** p1413-1414 (1956)
- [4] W.H. Meiklejohn, *Exchange Anisotropy - A Review*, Journal of Applied Physics **33(3)** p1328-1335 (1962)
- [5] G.N. Coverdale, *Computer Simulations of the Structure and Properties of a Particulate Dispersion*, PhD Thesis (1994)
- [6] E.C. Stoner and E.P. Wohlfarth, *A Mechanism of Magnetic Hysteresis in Hetrogeneous Alloys*, IEEE Transactions on Magnetics **27(4)** p3475 (1991)
- [7] D. Rugar, P. Hansma, *Atomic Force Microscopy* Physics Today **43** p23 (1990)
- [8] Y. Martin and H.K. Wickramasinghe, *Magnetic Imaging by Force Microscopy with 1000Å Resolution*, Applied Physics Letters **50(20)** p1455-1457 (1987)
- [9] D. Sarid, *Scanning Force Microscopy* Oxford University Press, New York (1991)

-
- [10] P. Grutter, H.J. Mamin, D. Rugar, *Magnetic Force Microscopy, Scanning Probe Microscopy II Springer-Verlag, New York* (1991)
- [11] Ramón y Cajál, *Maloine Edition Francaise Revue* (1911)
- [12] G.M. Shepherd and C. Kock, *The Synaptic Algorithm of the Brain* p3-31 Oxford University Press (1990)
- [13] I. Aleksander and H. Morton, *An Introduction to Neural Computing* Chapman & Hall (1990)
- [14] W.S. McCulloch and W. Pitts, *Bulletin of Mathematical Biophysics* **5** p115-133 (1943)
- [15] J.M. Mendel and McLaren, *Adaptive, Learning and Pattern Recognition Systems: Theory and Applications* p287-318 New York Academic Press (1970)
- [16] P.J. Werbos, *Beyond Recognition: New tools for prediction and analysis in the behavioural sciences*, PhD Thesis, Harvard University (1974)
- [17] D.E. Rumelhard, G.E. Hinton and R.J. Williams, *Nature* **323** p533-536 (1986)
- [18] A.P. Russo, *IEEE Conference on Neural Networks for Ocean Engineering*, Tutorial No. 8 (1991)
- [19] I. Guyon, *International Journal of Pattern Recognition and Artificial Intelligence* **5** p353-382 (1991)
- [20] R. Fletcher and C.M. Reeves, *Computer Journal* **7** p149-154 (1964)
- [21] E. Polak and G. Ribière, *Reveu Francaise Information Recherche Operationelle* **16** p35-43 (1969)
- [22] A.H. Kramer and A. Sangiovanni-Vincentelli, *Advances in Neural Information Processing Systems* p40-48 Morgan Kaufmann (1989)
- [23] T. Kohonen, *Biological Cybernetics* **43** p59-69 (1982)
- [24] H.J. Riter, T. Martinetz and K. Schulten, *Neural Computation and Self-organising Maps: An Introduction* Addison-Wesley (1992)
- [25] Z.P. Lo, Y. Yu and B. Bavarian, *IEEE Transactions on Neural Networks* **4** p207-220 (1991)
- [26] M.A. Fischler and O. Firschein, *Intelligence: The Eye, the Brain, and the Computer*, Addison-Wesley (1987)
-

-
- [27] J.A. Anderson, *Neurocomputing: Foundations of Research*, pxiii-xxi MIT Press (1988)
- [28] S. Haykin, *Neural Networks: A Comprehensive Foundation* Prentice Hall (1994)
- [29] G.R. Bolt, *Fault Tolerance In Artificial Neural Networks*, D.Phil Thesis, New York University (1992)
- [30] R. Gans, *Ann. Phys.* **29** p301 (1909)
- [31] A. Aharoni, E.H. Frei, S. Shtrikman and D. Treves, *Bull. Res. Council Israel* **6A** p215-38 (1957)
- [32] P.M. Sollis, P.R. Bissell and R.W. Chantrell, *IEEE Transactions On Magnetics* **30** p4101-4103 (1994)
- [33] P.M. Sollis, P.R. Bissell and R.W. Chantrell, *Journal of Magnetism and Magnetic Materials* **155** p123-126 (1996)
- [34] A. Hoare, R.W. Chantrell, W. Schmitt and A. Eiling, *Journal of Applied Physics* **26** p461-468 (1993)
- [35] L. Pareti and G. Turilli, *Journal of Applied Physics* **61** p5098-5101 (1987)
- [36] G.J. Daniel, *Maximum Entropy in Action* p1-18 Oxford Science Publications (1991)
- [37] *Stuttgart Neural Network Simulator*, University of Stuttgart. Obtained via ftp from: <ftp://ftp.informatik.uni-stuttgart.de/> (1996)
- [38] E.B. Baum and D. Haussler, *Neural Computation* **1** p151-160 (1989)
- [39] M.C. Nelson and W.T. Illingworth, *A Practical Guide to Neural Nets* p165 Addison-Wesley (1991)
- [40] M. Stone, *Journal of the Royal Statistical Society* **B36** p111-133 (1974)
- [41] P. Janssen, P. Stoica, T. Söderström and P. Eykhoff, *International Journal of Control* **47** p1737-1758 (1988)
- [42] A.H. Kramer and A. Sangiovanni-Vincentelli, *Advances in Neural Information Processing Systems* p40-48 Morgan Kaufmann (1989)
- [43] R.W. Chantrell, P.R. Bissell, P.M. Sollis, A. Hoare and T.H. Orth, *Journal of Magnetism and Magnetic Materials* **894** p177-9(1998)
- [44] P. Görnert, W. Schüppel, E. Sinn, F. Schumacher, K.A. Hempel, G. Turilli, A. Paoluzi and M. Rösler, *Journal of Magnetism and Magnetic Materials* **114** p193-201 (1992)
-

-
- [45] G. Bouwhuis, J. Braat, A. Huijser, J. Pasman, G. van Rosmalen and K. Schouhamer Immink, *Principles of the Optical Disc Systems* Philips Research Laboratories, Eindhoven **Adam Hilger Ltd.** (1986)
- [46] Compact Disc Digital Audio System (Red Book) *IEC Publication 908* (1983)
- [47] E.W. Williams, *The CD-ROM and Optical Disc Recording Systems* Electronic Engineering Group, Keele University **Oxford University Press** (1994)
- [48] E. Hecht and A. Zajac, *Optics* Adelphi University **Addison-Wesley Publishing Company** (1974)
- [49] M.V. Klein and T.E. Furtak, *Optics* **Wiley** (1986)
- [50] H.H. Hopkins, *Diffraction Theory of Laser Read-out Systems for Optical Video Discs* Journal of the Optical Society of America **69** p4-24 (1979)
- [51] J.H.T. Pasman, *Rigorous Diffraction Theory Applied to Video Disk Geometries* Proceedings of the Society of Photo-Optical Instrumentation Engineers **369** p674-680 (1983)
- [52] J.H.T. Pasman, *Optical Diffraction Methods for Analysis and Control of Pit Geometry on Optical Disks* Journal of Audio Engineering Society **41** p19-31 (1993)
- [53] H. Sekizawa, K. Yamada and A. Iwamoto, *Disk-memory Pregroove Inspection* Applied Optics **23** No. 16 p2830-2834 (1984)
- [54] H. Wang, *Diffraction of Light From Optical Disks* Optical Engineering **32** No. 4 p1098-1104 (1995)
- [55] G.F. Hughes, *Magnetisation Reversal in Colbalt-Phosphorous Films*, Journal of Applied Physics **54** No. 9 p5306-5312 (1983)
- [56] N.S. Walmsley, R.W. Chantrell, *Theoretical Studies of Stray Field Images of Magnetic Systems*, PhD Thesis at University of Central Lancashire (1995)
- [57] K. O'Grady, N.S. Walmsley, C.F. Wood and R.W. Chantrell, *The Effect of fcc Grains on the Magnetic and Recording Properties of Thin Film Media*, IEEE Transactions on Magnetism **34**, No. 4, p1579-1581m (1998)
- [58] H.V. Jones, N.S. Walmsley, S.J. Clarke, A.W.G. Duller and R.W. Chantrell, *Simulations of MFM images of longitudinal thin films with low coercivity grains*, IEEE Transactions on Magnetics, **35** No. 5 p2715-2717 (1999)
-


- [59] T. Kohonen, *Self-Organizing Maps* Springer (1995)
- [60] A. Ultsch and H.P. Siemon, *Kohonen's self organizing feature maps for exploratory data analysis*, International Neural Network Conference Proceedings, p305-308 (1990)
- [61] M.A. Kraaijveld, J. Mao, A.K. Jain, *Proceedings from International Conference on Pattern Recognition* IEEE Computer Society Press p41 (1992)
- [62] T.D. Sanger, *Neural Networks* **12** p459-473 (1989)
- [63] S.J. Clarke, *Magnetic Force Microscopy Simulation*, M.Eng Thesis at UWB Library (1998)
- [64] N.S. Walmsley, A. Hart, D.A. Parker, C. Dean and R.W. Chantrell, *Journal of Magnetism and Magnetic Materials* **170** p81-94 (1997)
- [65] A. Wadas, *Journal of Magnetism and Magnetic Materials* **72** p147-150 (1998)
- [66] A. Wadas and P. Grütter, *Physical Review B* **39** No. 16 p12013-12017 (1989)
- [67] N. George and S. Wang, *Neural Networks Applied to Diffraction-pattern Sampling*, *Applied Optics* **33** No 14 p3127-3134 (1994)

Appendix A

Publications

- H.V. Jones, A.W.G. Duller, R.W. Chantrell, A. Hoare and P R Bissell, *A neural network approach to the determination of anisotropy distributions*, Journal of Physics: D **31** p3028-3035 (1998)
- H.V. Jones, A.W.G. Duller, R.W. Chantrell, A. Hoare and P R Bissell, *Applications of neural networks to the determination of H_K distributions from transverse susceptibility data*, Journal of Magnetism and Magnetic Materials **193** p416-419 (1999)
- H.V. Jones, N.S. Walmsley, S.J. Clarke, A.W.G. Duller and R.W. Chantrell, *Simulations of MFM images of longitudinal thin films with low coercivity grains*, IEEE Transactions on Magnetics, **35** No. 5 p2715-2717 (1999)

JMMMDC Master Index Vols. 151-170 (August 1997) ISSN 0304-8853




**journal of
magnetism
and
magnetic
materials**

MASTER INDEX
VOLUMES 151-170
Dec. 1995-June 1997

Editor:
A.J. FREEMAN

Associate Editors:
G. BATE
J.G. BOOTH
I.A. CAMPBELL
R.W. CHANTRELL
H. FUJIMORI
V.I. OZHOGIN



Recognized by the
European Physical Society

NORTH-HOLLAND
<http://www.elsevier.nl>



ELSEVIER

Journal of Magnetism and Magnetic Materials 193 (1999) 416–419



Application of neural networks to the determination of H_K distributions from transverse susceptibility data

H.V. Jones^{a,*}, A.W.G. Duller^a, R.W. Chantrell^a, A. Hoare^b, P.R. Bissell^c

^aSchool of Electronic Engineering and Computer Systems, University of Wales, Bangor, Gwynedd, UK

^bAerosonic Ltd., Offa's Dyke Business Park, Buttington, Welshpool, Powys, UK

^cDepartment of Physics, Astronomy and Mathematics, University of Central Lancashire, Preston, UK

Abstract

In this paper we describe methods of applying a neural network technique to the magnetic inverse problem of determining the anisotropy field distribution from experimental transverse susceptibility data. In principle, the H_K distribution represents important experimental information; however, its determination is an inverse problem of great complexity. Using neural networks, we discuss methods in which the standard deviation can be determined to an accuracy which is comparable to other conventional methods, and completed in substantially less time. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Neural networks; Transverse susceptibility data; Inverse problems

1. Introduction

The solutions to mathematical problems, such as those encountered in magnetic inverse problems, can in some cases be very complex computationally. However, the solutions to such problems may be achieved substantially quicker using neural networks than with conventional methods, without any loss in accuracy.

The reversible transverse susceptibility (RTS) has been studied extensively, both theoretically and experimentally, since it was first discussed by Gans [1] in 1909. Much of the early work was carried out by Aharoni et al. [2], who produced expressions for the transverse susceptibility based on a Stoner–Wohlfarth system. The theoretical expressions predicted a cusp at the anisotropy field (H_K), consequently the RTS represents a potentially important technique for the measurement of H_K . Experimental work conducted by Pareti and Turilli [3] on

barium ferrite particles later confirmed the existence of these peaks, which appeared to be broad due to the effects of a distribution of H_K . The distribution of H_K values can be accounted for theoretically [4] by integrating the χ_i curve with a distribution function. In principle, the H_K distribution represents important experimental information; however, its determination is an inverse problem of great complexity. A neural network is an arrangement of components which are intended to model the functionality of the human brain [5]. The use of neural networks to deconvolute transverse susceptibility data and determine the distribution of H_K values is a relatively new approach in the field of magnetic inverse problems, although it has been proved possible to solve the problem using a maximum entropy approach [6].

2. Transverse susceptibility

The transverse susceptibility can be defined as being the change in magnetisation ΔM with respect to a small AC field applied perpendicular to a DC bias field H , as discussed by Aharoni et al. [2]. The theory for a randomly

*Corresponding author. Tel.: +44-1248-382739; fax: +44-1248-361429; e-mail: hjones@sees.bangor.ac.uk.

oriented system predicts a characteristic curve with peaks at $\pm H_K$ and H_C . Theoretical expressions for the transverse susceptibility can be extended to include the effects of texture [4]. It is also necessary to include a distribution of the values of H_K , which take into account the variation of particle shapes. The distribution of H_K values can be accounted for by integrating the χ_t curve with either a log-normal distribution function or a Gaussian distribution function.

3. The inverse problem, and neural network solution

The inverse problem essentially involves the calculation of the distribution function from known theoretical and experimental values of χ_t , and can be summarised as follows:

$$\chi_t(H) = \int_0^{\infty} f(y) \times \chi_t(H, H_K) dy, \quad (1)$$

where $f(y)$ is the unknown variable (i.e. the distribution function) with $y = H_K/H_{K0}$, $\chi_t(H)$ the experimental value of the transverse susceptibility, and $\chi_t(H, H_K)$ the theoretical value of the transverse susceptibility. Given the known kernel $\chi_t(H, H_K)$ and a set of experimental data, the problem is to determine the $f(y)$. The aim is to train the neural network using theoretically generated χ_t results to recognise specific relationships between χ_t and the distribution function, and demonstrate the application of neural networks to the determination of H_K distribution functions and to describe the practical procedures involved.

For the inverse problem a multi-layer feedforward network was chosen (Fig. 1), in which an array of input neurons is used to represent the x and y values of the curve and a single output neuron to represent the distribution function.

4. Results and discussions

Before training a neural network, the data which is to be presented must be arranged in a specific format, which

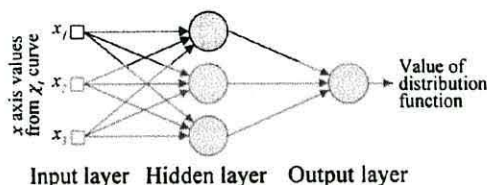


Fig. Multi-layer feedforward network.

can make a significant difference to the results obtained from the network. As most of the information in the χ_t curve resides around the H_K peak it was desirable to limit the data to this section of the curve.

A series of χ_t curves were generated by integrating the kernel with a variety of distribution functions. The number of training examples required can be estimated to a first order approximation, and should be approximately 10 times the number of synaptic weights in the network. Therefore, if the curve is represented by 50–100 data points, then even a small network will require about 2000–8000 training examples. Here we use an alternative method, overcoming the need for a large training set, called *early stopping* [7]. This generally consists of:

- Dividing the available data into training and validation sets
- Increasing the number of hidden neurons
- Stopping the training when the validation error reaches a minimum

The advantages of using this method is that it is fast, and can be applied to networks in which the number of weights exceeds the size of the training set. From these points, it would be feasible to use a training set of only about 100 examples.

We first investigated the effect on the error for a number of network configurations (using training sets consisting of log-normal distributions only). It is observed that a training set with 50 x and y values is sufficient to provide a satisfactory generalisation, as there is no significant improvement in performance by increasing the number of data points. A small network is observed to give a large error, as there is insufficient neurons in the hidden layer to extract important features from the curve. There is an increase in the error as the size of the network increases, which is due to the lack of information in the training set compared to the number of weights in the network. If the size of the training set is increased to compensate for the extra weights in the network, then the error is observed to decrease, but at the expense of the time required to train the network. From this investigation, we can deduce that a configuration of 8 hidden neurons in 1 hidden layer is sufficient to minimise the error, and thus demonstrate that a network can be trained to recognise a log-normal distribution function from χ_t results and to determine with high accuracy the standard deviation.

However, since the form of the distribution function for a given experimental sample is not known beforehand, it is also necessary to investigate the sensitivity of the χ_t data to the type of distribution function. The approach used was to generate a training set using log-normal and Gaussian distributions, thus encoding the type of distribution into the training set. The errors in recognition of the type of distribution function and the

Table 1
Summary of results using a variety of distributions

Type of distribution	Number of networks	Error in σ	Accuracy of classification
Log-normal	1	± 0.007	—
Log-normal + Gaussian	1	± 0.075	95.0%
Log-normal + Gaussian	2	± 0.006	98.5%

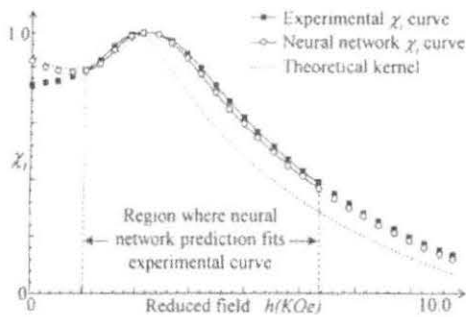


Fig. 2. Comparison of the experimental χ_i curve with the neural network predicted χ_i curve.

value of the standard deviations were then determined for the test set of data. It was observed that the neural network determined the value and type of distribution function to an acceptable degree of accuracy, but can be concluded that this method is not quite as accurate as the method using only one type of distribution. This can be attributed to the fact that we are now dealing with more information, and consequently more weights in the network. However, if the network is divided into two separate networks (one for determining the standard deviation, and one for determining the type of distribution), then the accuracy is observed to improve considerably. This increase in accuracy is attributable to the division of the one network into two separate networks, thus considerably reducing the amount of training data which must be applied to a single network. The results obtained using only log-normal distributions, and both log-normal and Gaussian distributions (1 network and 2 networks) are summarised in Table 1.

When real experimental data (using CrO_2 samples) was presented to an optimised neural network, which was trained with log-normal and Gaussian distributions (based on the theory in Refs. [2,4]), the network was also able to determine the most likely type of distribution associated with the sample. This was determined with a mean value of 0.292 with a standard error of 6×10^{-3} .

The network determined with a 99.5% certainty that the experimental H_K distribution was of the log-normal form. Fig. 2 compares the χ_i curve predicted by the neural network with the actual experimental χ_i curve which was used to test the network, and shows the region at which the predicted curve fits the experimental curve. The fitted curve deviates from the experimental curve, probably due to the effect of non-uniform magnetisation states, as will be discussed elsewhere.

From these results it is feasible to say that a neural network technique can be used to determine the type of distribution and the distribution value, which is essentially the same as that obtained using the maximum entropy approach. The advantage of the neural network approach is essentially one of speed. Our estimates suggest that the neural network approach is 1–2 orders of magnitude faster than the maximum entropy approach.

5. Conclusion

We have demonstrated, through the use of neural networks, that a magnetic inverse problem could be solved to a high degree of accuracy, and very efficiently. The selection of the network parameters was an important part of the study. Considerable improvement was observed when sections of the χ_i curve were cut-off, primarily in the low field, due to failure in the theoretical model and in the high field, due to the redundancy of data points. The distribution of H_K was determined to a high degree of accuracy when tested on a trained network using theoretically generated experimental data, which gives enhanced confidence when applying 'real' experimental data to the network. The distribution of H_K was determined to within ± 0.006 of the mean value, and found to be close to that obtained using the maximum entropy method [8], when experimental data was applied to the network.

From these conclusions the usefulness of this method is evident, but is primarily attributable to the speed and accuracy at which the method performs. The application of neural networks to anisotropy field distributions could, in theory, be extended to cover other results, such as those obtained in [3] by Pareti and Turilli.

References

- [1] R. Gans, *Ann. Phys.* 29 (1909) 301.
- [2] A. Aharoni, E.H. Frei, S. Shtrikman, D. Treves, *Bull. Res. Council Israel* 6A (1957) 215.
- [3] L. Pareti, G. Turilli, *Journal of Applied Physics* 61 (1987) 5098.
- [4] A. Hoare, R.W. Chantrell, W. Schmitt, A. Eiling, *J. App. Phys.* 26 (1993) 461.
- [5] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, Englewood Cliffs, NJ, 1994.
- [6] B. Buck, V.A. Macaulay, *Maximum Entropy in Action*, Oxford Science Pub., Oxford, 1991.
- [7] M.C. Nelson, W.T. Illingworth, *A Practical Guide to Neural Nets*, Addison-Wesley, Reading, MA, 1991.
- [8] R.W. Chantrell, P.R. Bissell, P.M. Sollis, A. Hoare, T.H. Orth, *J. Magnetism Magnetic Materials*, in press.

IoP

ISSN 0022-3727

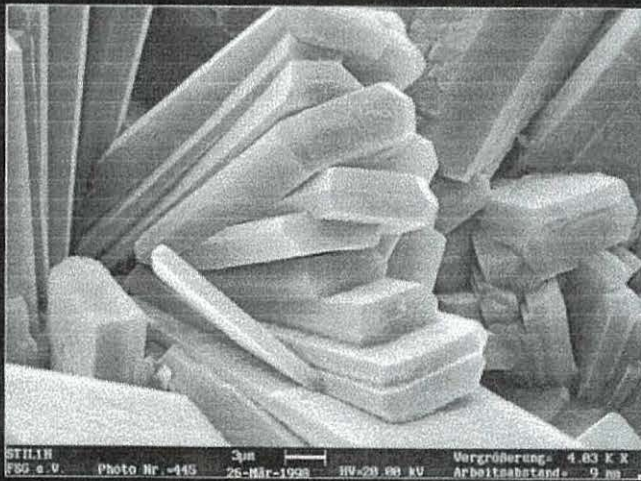
Journal of Physics D Applied Physics

Volume 31 Number 21 7 November 1998

Review article: Thin film magnetic recording media

Online: <http://www.iop.org>

A Journal Recognized by
The European Physical Society



Institute of **Physics** PUBLISHING

A neural network approach to the determination of anisotropy distributions

H V Jones[†], A W G Duller[†], R W Chantrell[†], A Hoare[‡] and P R Bissell[§]

[†] School of Electronic Engineering & Computer Systems, University of Wales, Bangor, Gwynedd, UK

[‡] Aerosonic Ltd, Offa's Dyke Business Park, Buttington, Welshpool, Powys, UK

[§] Department of Physics, Astronomy and Mathematics, University of Central Lancashire, Preston, UK

Received 22 June 1998, in final form 19 August 1998

Abstract. In this paper we describe methods of applying a neural network technique to the magnetic inverse problem of determining the anisotropy field's distribution from experimental transverse susceptibility data. In principle, the H_K distribution represents important experimental information; however, its determination is an inverse problem of great complexity. Using a neural network technique, we discuss methods in which the standard deviation can be determined to an accuracy which is comparable to those of other conventional methods and completed in substantially less time.

1. Introduction

The solutions to mathematical problems, such as those encountered in magnetic inverse problems, can in some cases be very complex computationally. A variety of well known methods can be applied to solve such problems. However, in most cases, the time required is considerable. In this paper we describe methods of applying neural network techniques to magnetic inverse problems and demonstrate that solutions to computationally complex problems may be achieved substantially quicker using neural networks than with conventional methods, without any loss in accuracy.

We apply neural network techniques to the reversible transverse susceptibility (RTS) which has been studied extensively, both theoretically and experimentally, since it was first discussed by Gans [1] in 1909. Much of the early work was carried out by Aharoni *et al* [2], who in 1957 produced expressions for the transverse susceptibility based on a Stoner–Wohlfarth system. The theoretical expressions predicted a cusp at the anisotropy field (H_K), consequently the RTS represents a potentially important technique for the measurement of H_K . Experimental work conducted by Pareti and Turilli [3] on *barium ferrite* particles later confirmed the existence of these peaks, which appeared to be broad due to the effects of a distribution of H_K . The distribution of H_K values can be accounted for theoretically [4] by integrating the χ_t curve with a distribution function. In principle, the H_K distribution represents important experimental information; however, its determination is

an inverse problem of great complexity. The ability to measure the distribution of the anisotropy field accurately is potentially important in the characterization of recording media, since it is related to the overwrite characteristics of the medium.

An artificial neural network is an arrangement of components which are intended to model the functionality of the human brain. It can be thought of as being an immensely parallel distributed processor, which has a natural ability for storing knowledge and making it available for use, the knowledge having been acquired by the network through a learning process (Aleksander and Morton [5]). The use of neural networks to deconvolute transverse susceptibility data and determine the distribution of H_K values is a relatively new approach in the field of magnetic inverse problems, although it has been proved possible to solve the problem using a maximum-entropy approach [6]. Although it is successful, the maximum-entropy approach has the disadvantage of being computationally expensive. Here we propose and test a neural network solution to the inverse problem and discuss methods of applying specific neural network techniques in practice. In the following we first outline the transverse-susceptibility theory and the inverse problem of determining anisotropy-field distributions. We then develop a neural network approach to the solution of the inverse problem. Because of the limited use of neural network approaches to magnetic problems we introduce the essentials of neural network theory and describe in some detail the development and testing of a network using model

RTS data calculated using theoretical expressions given in the following sections. We then apply the network to CrO₂ recording medium, obtaining good agreement with previous work using a maximum-entropy approach.

2. The transverse susceptibility

In this section we will outline the background of the RTS, both theoretically and experimentally, explaining the relationship between the two. The transverse susceptibility can be defined as the change in magnetization ΔM with respect to a small ac field applied perpendicular to a dc bias field H , as discussed by Aharoni *et al* [2] and given by

$$\chi_t = \lim_{\Delta H_{ac} \rightarrow 0} \left(\frac{\Delta H_{ac}}{\Delta H_{ac}} \right). \quad (1)$$

The theory for a randomly oriented system predicts a characteristic curve with peaks at $\pm H_K$ and H_C . In systems such as doped barium ferrite particles [7] and γ -Fe₂O₃ the determination of H_K is difficult due to a wide distribution of the anisotropy field which causes the peak to be suppressed, whereas in an aligned system the peaks at $\pm H_K$ are well defined [8]. In order to understand this concept, some knowledge of the effects of texture is important.

Theoretical expressions for the transverse susceptibility can be extended to include the texture function (Hoare *et al* [4]), given by equation (2):

$$\begin{aligned} \frac{\bar{\chi}_t}{\chi_0} = & \frac{1}{2\pi} \int_0^{\pi/2} \frac{3}{2} \left(\frac{\cos^2 \theta_M}{h \cos \theta_M + \cos 2(\theta_M - \theta_K)} \int_0^{2\pi} \cos^2 \phi_K \right. \\ & \times \exp[-(\pi/2 - \theta_K^*)^2 / (2\sigma_\theta^2)] \exp[-\phi_K^2 / (2\sigma_\phi^2)] d\phi_K \\ & + \frac{\sin(\theta_K - \theta_M)}{h \sin \theta_K} \int_0^{2\pi} \sin^2 \phi_K \\ & \times \frac{\exp[-(\pi/2 - \theta_K^*)^2 / (2\sigma_\theta^2)] \exp[-\phi_K^2 / (2\sigma_\phi^2)]}{A} d\phi_K \left. \right) \\ & \times \sin \theta_K d\theta_K \quad (2) \end{aligned}$$

where σ_θ and σ_ϕ control the distributions in and out of the plane, respectively, and θ_K^* is the projection of θ_K into the plane of the tape. A is the normalizing constant and $h = H/H_K$ is the reduced applied field. θ_K and ϕ_K are the orientations of the easy axis.

In a real system it is necessary to include a distribution of the values of H_K , which takes into account the variation of particle shapes. The distribution of H_K values can be accounted for by integrating the χ_t curve either with a log-normal distribution function, as in equation (3), or with a Gaussian distribution function, as in equation (4):

$$\chi_t(H) = \int_0^\infty \chi_t(H/H_K) \frac{1}{(2\pi)^{1/2} \sigma_{H_K} H_{K_M}} \times \exp\left(-\frac{[\ln(H_K/H_{K_M})]^2}{2\sigma_{H_K}^2}\right) dH_{K_M} \quad (3)$$

$$\chi_t(H) = \int_0^\infty \chi_t(H/H_K) \frac{1}{(2\pi)^{1/2} \sigma_{H_K}} \times \exp\left(-\frac{(1 - H_K/H_{K_M})^2}{2\sigma_{H_K}^2}\right) dH_{K_M} \quad (4)$$

for a particular field H and $\chi_t(H/H_K)$ is the kernel defined by equation (2). The kernel has a well defined cusp at

Anisotropy distributions

$H = H_K$. However, the H_K distributions lead to rounded peaks which compare well with experimental data obtained by Pareti and Turilli [3]. This of course complicates the determination of the mean value of H_K . However, the shape of the χ_t - H_{dc} relationship must contain information on the H_K distribution. The aim of the current paper is to outline a novel solution of the associated magnetic inverse problem, using neural networks.

3. The inverse problem and its neural network solution

The inverse problem essentially involves the calculation of the distribution function from known theoretical and experimental values of χ_t and can be summarized as follows:

$$\chi_t(H) = \int_0^\infty f(y) \chi_t(H, H_K) dy \quad (5)$$

where $f(y)$ is the unknown variable (the distribution function) with $y = H_K/H_{K_M}$ (the mean value), $\chi_t(H)$ the experimental value of the transverse susceptibility and $\chi_t(H, H_K)$ the theoretical value of the transverse susceptibility. Given the known kernel $\chi_t(H, H_K)$ determined by equation (2) and a set of experimental data, the problem is to determine the $f(y)$.

In the case of the inverse problem the aim is to train the neural network using theoretically generated χ_t results to recognize specific relationships between χ_t and the distribution function. When the neural network has repeatedly encountered a representative sample of the relationships with which it is likely to be faced, then its performance is expected to be almost perfect and it can be said that the network has been trained to solve the problem. It is the aim of this paper to demonstrate the application of neural networks to the determination of H_K distribution functions and to describe the practical procedures involved.

Because neural networks have not been used extensively in magnetic problems, we outline briefly the basic concepts applied in our solution. For an extensive introduction the reader is referred to the treatment of Haykin [9]. The concept of neural networks originated through the better understanding of the human brain. In 1911 Ramón y Cajál [10] introduced the idea of neurons, which can be compared to silicon logic gates in a computer, but, whereas logic gates function in the nanosecond region, the neurons in the human brain function in the microsecond region. This difference in speed is compensated for in the brain, by the sheer number of neurons present.

A model of a single neuron consists of three fundamental parts (see figure 1):

- (i) a series of connecting links, which can be altered individually by changing its weight;
- (ii) an adder which sums the input signals; and
- (iii) an activation function, whereby a threshold can be applied to limit the output of the model neuron.

H V Jones et al

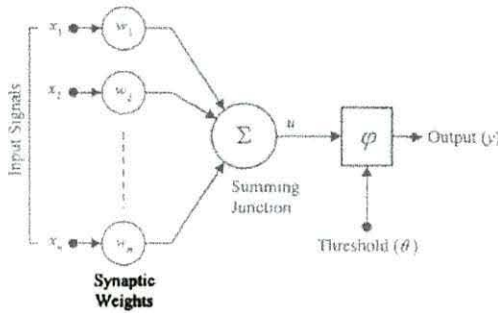


Figure 1. A model of a neuron.

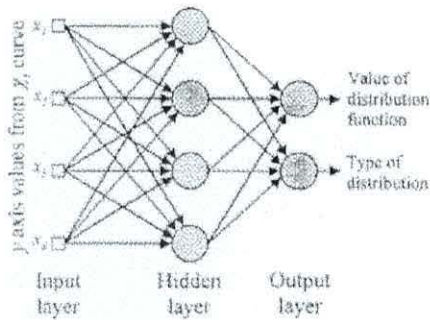


Figure 2. A multi-layer feed-forwards network.

Mathematically, the summing action of the neuron can be described by equation (6), whilst the overall output of the neuron is given by equation (7):

$$u = \sum_{j=1}^n w_j x_j \tag{6}$$

$$y = \varphi(u - \theta) \tag{7}$$

where x_1, x_2, \dots, x_n are the input signals, w_1, w_2, \dots, w_n are the synaptic weights, θ is the threshold, φ is the activation function and y is the output of the neuron. The activation function (φ) describes the output of the neuron with respect to the activity level (v) at the input and can take a number of forms, the simplest being given by

$$y = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \tag{8}$$

where v is the internal activity level of the neuron, given by

$$v = \sum_{j=1}^n w_j x_j - \theta. \tag{9}$$

From equation (9), the output of the neuron gives a value of 1 if the total internal activity level of that neuron is not negative and a value of 0 otherwise.

For the inverse problem a multi-layer feed-forwards network was chosen (see figure 2). This consists of an array

of input neurons which are forwards connected to hidden neurons. The incorporation of hidden neurons allows the network to extract higher-order statistics, which becomes useful when the input layer becomes large. The hidden layers are then forwards connected to an output layer of neurons.

An important feature of a neural network is its ability to learn from its experience, thus greatly improving its performance. The network learns by making slight adjustments to its synaptic weights after each cycle of the learning data. The process comprises three principal steps:

- (i) the network is stimulated by the environment,
- (ii) the network responds differently to its environment because of the changes and
- (iii) the network undergoes changes as a result of stimulation.

In our case the network is ‘trained’ using a set of training examples. This consists of calculating a set of $\chi_r(H)$ curves for various values of $f(y)$ and presenting these as a training set to the network. The weights of the neurons are adjusted as a result of the training examples and the calculated error. The performance of the network is measured in terms of the mean-squared error (equation (10) below), which is averaged over all possible input–output examples in the training set. In order for the performance of the network to improve, the operating point has to move down successively towards a minimum point on the error surface, be it a local or a global minimum. It should be noted that the solution of the inverse problem is thus dependent on the applicability of the model calculations giving rise to the kernel. It is necessary to be aware of the potential effects of systematic differences between theory and experiment due to the breakdown of the theory, as will be described later.

A neural network is trained with a learning algorithm. The conjugate-gradient learning algorithm consists of choosing ‘conjugate directions’ of minimization that do not interfere with each other and performing a line minimization along that direction. The conjugate-gradient method will always locate the minimum of any quadratic function of N variables in at most N steps (see Haykin [9] for details). Kramer and Sangiovanni-Vincentelli [11] studied the use of the conjugate-gradient method for the supervised training of multi-layer perceptrons. They showed that the conjugate-gradient method required fewer epochs (the number of times that a pattern is presented to the network) than other methods, such as the standard back-propagation method; the disadvantage being that the conjugate-gradient method proved to be computationally more complex.

The training of a neural network can be examined by observing the mean-squared error (equation (10)), which is plotted against the number of epochs (it should be noted that the mean-squared error is not an indication of the performance of a network with respect to unseen data):

$$L(F) = \frac{1}{2N} \sum_{j=1}^N \|d_j - F(x_j)\|^2. \tag{10}$$

Here N is the total number of input–output patterns presented to the network in training, d_j is the desired output pattern for x_j , F is a vector function that minimizes the mean-squared error, and $\| \cdot \|$ is the Euclidean norm of the enclosed vector.

4. Results and discussion

4.1. Preparation of data

Before training a neural network, the data which are to be presented must be arranged in a specific format. The format can make a significant difference to the results obtained from the network. Since a multi-layer perceptron requires a training set to consist of input–output-related patterns, it was decided to use an array of input neurons to represent the x and y values of the curve and a single output neuron to represent the standard deviation of the distribution function.

The problem, in essence, was to train the network with a set of theoretically generated data and then investigate its performance with another set of theoretical data, before finally testing it with experimentally obtained data. It was necessary to normalize both theoretical and experimental data points to similar values for presentations to the network, in order to create a sufficiently general network capable of dealing with a variety of materials. The value of H_K does not remain constant for each sample; it is usually in the range 2–5 kOe, depending on the particular material investigated. Because most of the information resides around the H_K peak it was considered desirable to normalize the curve to peak at a fixed value in the range 2–5 kOe, hence eliminating the need to train the network with all variations of the χ_t curve. It was also noticed that the χ_t value could be normalized along the y axis; since the value of χ_t did not fall below 0.5. Thus by altering the y axis from 0.5 to 1.0, it was possible to amplify the diversity among the χ_t curves.

4.2. Network training and optimization

Having decided on the format of the training patterns, a series of χ_t curves was generated by integrating the kernel with a variety of distribution functions (equation (3)) with standard deviations in the range 0.05–0.75. The quantity of training examples is at this point yet to be finalized, as is the number of x and y points which will adequately describe the χ_t curve. The number of training examples required can be estimated to within a first-order approximation using equation (11) (using a result derived by Baum and Haussler [12]):

$$N > W/\varepsilon \quad (11)$$

where N is the number of examples, W the total number of synaptic weights in the network and ε the fraction of errors permitted in a test. Thus, with an error of 10%, the number of training examples should be approximately ten times the number of synaptic weights in the network. If the curve is represented by 50–100 data points, then even a small network (4–8 hidden neurons) will require about 2000–8000 training examples. Here we use an alternative method, overcoming the need for a large training set, called *early stopping* (Nelson and Illingworth [13]). This generally consists of

Anisotropy distributions

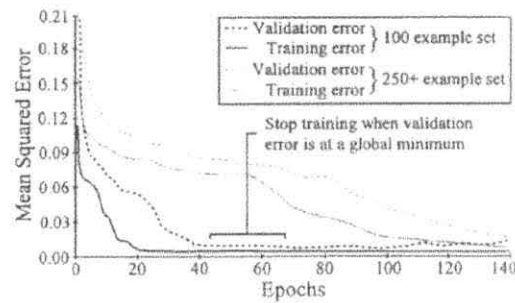


Figure 3. Overtraining a neural network.

- (i) dividing the available data into training and validation sets,
- (ii) increasing the number of hidden neurons,
- (iii) using small initialization values,
- (iv) using a slow-learning-rate parameter,
- (v) calculating the validation error during training and
- (vi) stopping the training when the validation error reaches a minimum.

The advantages of using this training method are that it is fast and can be applied to networks in which the number of weights exceeds the size of the training set. From these points, it would be feasible to use a training set of only about 100 examples. The networks used in this study were simulated using the *Stuttgart Neural Network Simulator* [14].

A neural network which has been designed to generalize well will result in a correct input–output mapping (even without its having been trained with a particular pattern). However, when a neural network learns too many input–output relations, the network memorizes the training data and is less able to generalize between patterns. The mean-squared error curve can indicate that the network has been trained sufficiently, whilst the validation error (the error calculated over the validation set) can indicate when a network is being overtrained. Thus, an increase in the validation error is a clear indication that the network is being overtrained and becoming too familiar with the training set, which leads to poor generalization. It is thus favourable to stop training before this point is reached. This behaviour was observed in our study, as shown in figure 3, which shows the measured error as a function of the number of epochs for a 100-example and a 250-example training set. In the case of the 100-example set the errors clearly reach a minimum value after 40–60 epochs. For the 250-example case the minimum error is reached after a larger number of epochs (100–140). This increase in training which is required can be accounted for by the extra information which has to be learnt. A mean squared error of less than 1% as shown by figure 3 was considered sufficiently accurate for the applications considered here, since this is of the order of the error in experimental data. Consequently a training set of around 100 data sets was used throughout this study.

The network was trained using the technique of cross validation (Stone [15], Janssen *et al* [16]), whereby the

H V Jones *et al*

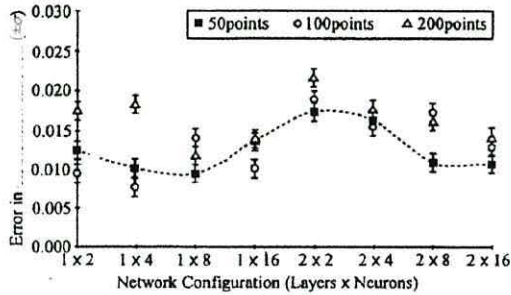


Figure 4. A plot of the distribution function errors using 50, 100 and 200 x and y values using a variety of hidden neurons.

generated data set is randomly divided into a training set and a test set. The training set is further divided into two sets:

- (i) a set used to train the network (the *training set*) and
- (ii) a set used to evaluate the performance of the network (the *validation set*), which should be approximately 10% of the training set.

We first investigated the effect of the input data set on the error in the calculated values of σ , for a number of network configurations, characterized by the number of ‘layers \times neurons’. Figure 4 shows the error in the distribution function as a function of the network’s configuration for a number of data-set sizes when the network was optimally trained (connecting lines are shown for clarity).

From figure 4, it can be concluded that a training set with 50 x and y values is sufficient to provide a satisfactory generalization, insofar as no significant improvement in performance is gained by increasing the number of data points. A small network is observed to give a large error, because there are insufficient neurons in the hidden layer to extract important features from the curve. There is a small increase in the error as the size of the network increases, which is due to the lack of information in the training set (defined in equation (11)) compared with the number of weights in the network. If the size of the training set is increased to compensate for the extra weights in the network, then the error is observed to decrease, but at the expense of the time required to train the network.

Since the linearly sampled x values are identical in each example, it is feasible to remove them from the training set without any loss in accuracy, thus presenting the network with the sampled y values only (hence only 50 input neurons are required). The experimental determination of the optimal network architecture, when the system is trained using the conjugate-gradient learning algorithm, depends on a variety of factors, mainly the numbers of hidden neurons and training examples. If practical considerations are applied to the problem, the criterion used is to start with the smallest number of hidden neurons that yields a satisfactory performance; in this case two hidden neurons. Consequently we studied the classification error as a function of the network’s configuration. The dependence is rather weak, but the average error was found to be lowest

Table 1. A summary of network parameters with only a log-normal distribution.

Parameter	Value
	50
	1 layer of 8 neurons
	1
	100
	10
	50
	30–40

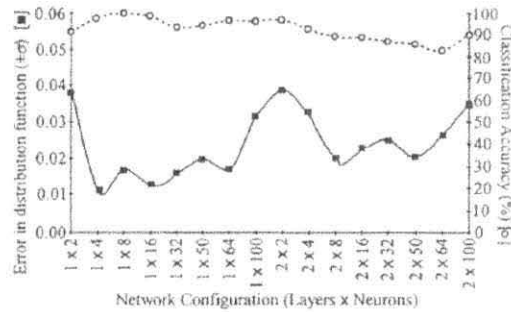


Figure 5. A plot of the distribution function errors and distribution-type classification accuracy using one neural network.

with 8–16 hidden neurons in one hidden layer. In the rest of this work we use a configuration of eight hidden neurons in one hidden layer, which was sufficient to minimize the error with a reasonably small network. We stress that this is a specific result for our study; in general each application will require a similar analysis, since other problems may give rise to a stronger dependence of the performance of the network on its detailed configuration.

From the investigations conducted, it can be concluded that, for the inverse problem in which only a log-normal distribution has been applied, a training set of about 100 examples (each consisting of 50 y axis values) is sufficient to train a network. The size of the network and hence the number of synaptic weights required can be reduced substantially by omitting the x axis data. The configuration of the network is summarized in table 1.

The results obtained so far demonstrate that a network can be trained to recognize a log-normal distribution function from χ_t results and to determine with high accuracy the standard deviation. However, the form of the distribution function for a given experimental sample is not known beforehand, so it is necessary to investigate the sensitivity of the χ_t data to the type of distribution function. The log-normal distribution has a characteristic asymmetry which arises for known physical reasons for example in particle-size distributions. However, since the form of the anisotropy distribution is not known *a priori* we also allow the possibility of a symmetrical distribution, represented by the Gaussian distribution.

The approach used was to generate a training set calculated using log-normal and Gaussian distributions.

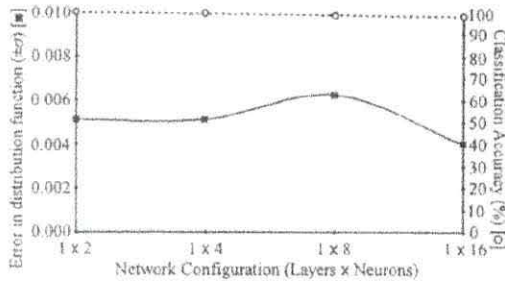


Figure 6. A plot of the distribution function errors and distribution-type classification accuracy using two separate neural networks.

Table 2. A summary of network parameters with log-normal and Gaussian distributions.

Parameter	Value
	50
	1 layer of 16 neurons
	1
	100
	10
	50
	40

The curves were generated in a similar manner to the previous curves: 100 curves were generated with a log-normal distribution and 100 with a Gaussian distribution. The type of distribution was also encoded into the output patterns, hence training the neural network with the distribution function's value and distribution type. The errors in recognition of the type of distribution function and the values of the standard deviations were then determined for the test set of data. The results are presented in figure 5, from which it can be observed that the neural network has determined the distribution function's value to within ± 0.075 in all cases and has determined the type of distribution to a minimum certainty of 95% in all cases. It can be concluded that this method is not quite as accurate as the method using only one type of distribution, for we are now dealing with more information and consequently more weights in the network.

However, if the network is divided into two separate networks (one for determining the distribution value and one for determining the type of distribution), then the accuracy improves considerably. From figure 6, it can be observed that the neural networks have determined the value of the distribution function to within ± 0.006 in all cases and have determined the type of distribution to a minimum certainty of 98.5% in all cases. This increase in accuracy is attributable to the division of the one network into two separate networks, thus considerably reducing the number of weights required. The configurations of the networks are summarized in table 2.

Having determined a network architecture which yields optimum results when it is presented with theoretical data, the goal is to apply 'real' experimental data to the network.

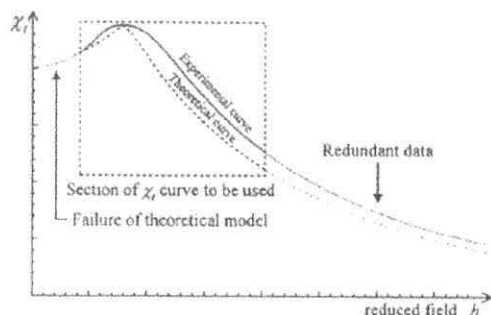
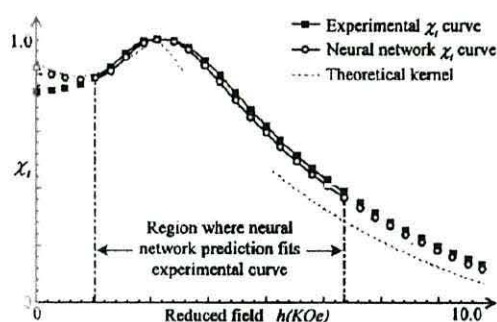
Anisotropy distributions

Here we outline the results obtained when transverse-susceptibility data of a CrO_2 sample were used, initially on a network trained with log-normal distributions only and then on a network trained with log-normal and Gaussian distributions.

Several trials were performed on the optimized neural networks (each trained with only log-normal distributions and retrained after each test run using identical parameters), from which the network was able to determine the distribution function to a mean value of 0.292 with a standard error of 6×10^{-3} . Thus, the network is shown to train reproducibly and to give consistent estimates of the value of σ . The inverse problem for this data set has also been solved using maximum-entropy techniques [17], from which the distribution function of the experimental data was found to be 0.291. The agreement between the two techniques enhances one's confidence in the solutions obtained and in the use of the χ_t for the determination of H_K distribution functions.

However, when an experimental data set was presented to a network which had been trained with log-normal and Gaussian distributions the results obtained were poor and changing the network's parameters had little effect on the results. In contrast it was observed that theoretical test sets gave accurate results. It should be recalled that the kernel used in the theoretical calculations is essentially that derived by Aharoni *et al* [2], which is based on the assumption of the Stoner-Wohlfarth coherent mode of magnetization reversal. It has been shown [18] that the RTS falls into a class of 'stiffness' measurements based on magnetization rotation. Here the Stoner-Wohlfarth model is a good approximation as long as the magnetization remains close to saturation, which is normally the case when determining H_K values of 2 kOe or greater. However, the current work presents the neural network with experimental data extending to small fields, at which it might be expected that non-uniform magnetization states could develop, with an as yet theoretically unknown effect on the RTS. A detailed comparison of the experimental data with the predicted kernel revealed a systematic discrepancy in small fields, which we interpret intuitively as arising from non-uniform magnetization modes. As a result we investigated the effects of removing the low-field data and also the high-field data on which the effects of the H_K distribution might be expected to be less pronounced. This essentially concentrates the data around the peak, in the area containing the H_K information. If the χ_t curves are trimmed at low and high fields (figure 7), hence training the network with a smaller section of the curve, then the performance of the network improves.

When an experimental data set was presented to an optimized neural network, which was trained with log-normal and Gaussian distributions (and retrained after each test run using identical parameters), the network was also able to determine the most likely type of distribution associated with the sample. This was determined with a mean value of 0.292 with a standard error of 6×10^{-3} . The network determined with a 99.5% certainty that the experimental H_K distribution was of the log-normal form. Figure 8 compares the χ_t curve predicted by the neural

H V Jones *et al*Figure 7. The section of χ_t curve to be used.Figure 8. A comparison of the experimental χ_t curve with the neural network's predicted χ_t curve.

network with the actual experimental χ_t curve which was used to test the network and shows the region in which the predicted curve fits the experimental curve.

From these results it is feasible to say that a neural network technique can be used to determine the type of distribution and the distribution's value, which is essentially the same as that obtained using the maximum-entropy approach. The advantage of the neural network approach is essentially one of speed. Our estimates suggest that the neural network approach is 1–2 orders of magnitude faster than the maximum-entropy approach.

5. Conclusion

We have demonstrated, through the use of neural networks, that a magnetic inverse problem could be solved to a high degree of accuracy and very efficiently by using a multi-layer perceptron network, trained using the conjugate-gradient technique. Several input-output-related χ_t curves were generated, using log-normal and Gaussian distributions. The selection of the network parameters which gave a good generalization was an important part of the study, for any changes could have a significant effect on the accuracy of the results. The three important parameters were the size of the training set, the size of the network and the number of epochs allowed to train the network.

Initial investigations demonstrated that it was necessary to normalize the theoretical χ_t curves with respect to the

experimental curves, in order to facilitate the learning process. Considerable improvement was observed when sections of the χ_t curve were cut off, primarily in the low-field region, due to failure in the theoretical model, and in the high-field region, due to the redundancy of data points. The conclusion was that more data points could be concentrated around the H_K peak (where most of the valuable information resides). A training set consisting of 100 examples, each with 50 data points, was found to give the best performance in terms of accuracy and speed. By excluding the x axis data (which are identical in each pattern) the number of neurons in the network was halved.

Having generated the training set, the size of the network could be determined. It was observed that a small network, consisting of two neurons in one hidden layer performed poorly, because there are insufficient weights present to extract important features from the χ_t curves. However, a large network of 16 neurons in two hidden layers also performed poorly, due to the excessive number of weights in the network. In general, a configuration of eight neurons in one hidden layer performed best for our optimized χ_t curve, for this amounted to a reasonable number of weights in the network.

The accuracy of the results was observed to be closely related to the number of epochs with which the network has been trained. The preferred method is to train the network until the mean-squared error for the training and validation sets reaches a minimum and to stop well before the validation error begins to increase. Using the conjugate-gradient learning algorithm (which has been proved to generalize in a fraction of the time taken by other algorithms) the training required about 40 epochs, which was sufficient to give very accurate results.

Having found the optimal parameters for the investigation, the network was presented with theoretical test data, from which the network was observed to be highly accurate. However, the goal is to apply experimental data to the neural network, using the chosen parameters. The distribution of H_K was determined in all cases to within ± 0.006 of the mean value and was found to be close to that obtained using the maximum-entropy method. The type of distribution was determined to within 99% certainty of being log-normal.

Investigations on the network using extreme cases indicated that the accuracy rapidly decreased with distribution functions above about 0.65, which is due to the loss of important data points from the curve section used. However, insofar as experimental data rarely have H_K distributions above this value, it is not considered a problem. It can therefore be concluded that a neural network approach is feasible for determining the value and type of distribution of H_K (when $\sigma < 0.65$) at which point the accuracy is comparable to those of conventional methods and done in substantially less time.

Acknowledgments

This work was supported in part by the UK EPSRC (Engineering and Physical Sciences Research Council) and by Aerosonic Ltd, Welshpool, UK.

Anisotropy distributions

References

- [1] Gans R 1909 *Ann. Phys., Lpz* 29 301
- [2] Aharoni A, Frei E H, Shtrikman S and Treves D 1957 *Bull. Res. Council Israel* A 6 215–38
- [3] Pareti L and Turilli G 1987 *J. Appl. Phys.* 61 5098–101
- [4] Hoare A, Chantrell R W, Schmitt W and Eiling A 1993 *J. Appl. Phys.* 26 461–8
- [5] Aleksander I and Morton H 1990 *An Introduction to Neural Computing* (London: Chapman & Hall)
- [6] Buck B and Macaulay V A 1991 *Maximum Entropy in Action* (Oxford: Oxford Science)
- [7] Sollis P M, Bissell P R and Chantrell R W 1994 *IEEE Trans. Magn.* 30 4101–3
- [8] Sollis P M, Bissell P R and Chantrell R W 1996 *J. Magn. Magn. Mater.* 155 123–6
- [9] Haykin S 1994 *Neural Networks: A Comprehensive Foundation* (New York: Prentice Hall)
- [10] Ramón y Cajal S 1995 *Histology of the Nervous System of Man and Vertebrates* (Oxford: Oxford University Press) (Engl. transl.)
- [11] Kramer A H and Sangiovanni-Vincentelli A 1989 *Advances in Neural Information Processing Systems* (San Mateo, CA: Morgan Kaufmann) pp 40–8
- [12] Baum E B and Haussler D 1989 *Neural Computation* 1 151–60
- [13] Nelson M C and Illingworth W T 1991 *A Practical Guide to Neural Nets* (New York: Addison-Wesley) p 165
- [14] Stuttgart Neural Network Simulator, University of Stuttgart. Obtained via anonymous ftp from <ftp://ftp.informatik.uni-stuttgart.de/pub/SNNS/SNNSv4.1.tar.z>
- [15] Stone M 1974 *J. R. Statist. Soc. B* 36 111–33
- [16] Janssen P, Stoica P, Söderström T and Eykhoff P 1988 *Int. J. Control* 47 1737–58
- [17] Chantrell R W, Bissell P R, Sollis P M, Hoare A and Orth T H 1998 *J. Magn. Magn. Mater.* 894 177–8
- [18] Görnert P, Schüppel W, Sinn E, Schumacher F, Hempel K A, Turilli G, Paoluzi A and Rösler M 1992 *J. Magn. Magn. Mater.* 114 193–201

 **IEEE TRANSACTIONS ON**
MAGNETICS

A PUBLICATION OF THE IEEE MAGNETICS SOCIETY

SEPTEMBER 1999 VOLUME 35 NUMBER 5 IEMGAC (ISSN 0018-9464)

PART I OF THREE PARTS



SELECTED PAPERS FROM THE 1999 INTERNATIONAL MAGNETICS CONFERENCE (INTERMAG '99)
Hyundai Hotel, Kyongju, Korea, May 18 – 21, 1999

(See p. 2201, Part I, or p. 3201, Part II, for Intermag '99 Table of Contents, Intermag '99 papers begin on p. 2229, Part I, and continue on p. 3223, Part II. Regular papers begin on p. 4151, Part III.)

Simulations of MFM Images of Longitudinal Thin Films with Low Coercivity Grains

H. V. Jones, N. S. Walmsley, S. J. Clarke, A. W. G. Duller and R. W. Chantrell
SEECs, University of Wales Bangor, Gwynedd, LL57 1UT, U. K.

Abstract—The recording density of thin film media is affected by percolation within the transitions; such percolation has been shown to occur in areas where clusters of low coercivity grains lie. In this paper we present simulated Magnetic Force Microscopy (MFM) images which show it is common for inter-penetration between adjacent written bits to occur close to such clusters.

Index Terms—longitudinal thin film, low coercivity grains, MFM simulations.

I. INTRODUCTION

IMPERFECTIONS in epitaxy between the layers in thin film media can give rise to stacking faults [1]. We model the non-ideal crystallinity by introducing low coercivity (LC) grains to the magnetic microstructure; it has already been shown that this affects the macromagnetic parameters and recording properties of thin film media [2]. Further study has shown that both the non-ideal crystallinity and the magnitude of the magnetostatic interactions significantly influence the form of the δI plots of the systems [3] – [4]. In this paper, we move toward a comparison with experiment by calculating MFM images and investigate the correlations with recorded patterns. The main aim is to investigate the percolation phenomenon at high recording densities and its manifestation in MFM images.

II. THE MODEL

Recorded bits are simulated by allowing a series of oppositely magnetised regions relax, subject to the influence of a decreasing applied field. This is carried out using an energy gradient descent technique [5]. The physical is formed from an assembly of grains whose size distribution is log-normal with mean diameter of approximately 20nm and whose centres lie on radially isotropic structure. Such physical irregularity has been shown to be very important in the modelling of real systems [6]. Our simulations assume a magnetic layer whose thickness is of the order of the mean diameter of a grain.

The model is characterized by two important parameters: h_i , the value of the mean interaction field relative to the intrinsic coercivity of the grains (the value of h_i for

the LC grains is seven times that for the normal grain), and C^* , a field representative of the exchange coupling between grains, again scaled relative to the intrinsic coercivity. Imperfect grains will have a lower anisotropy than hcp Cobalt ($K = 4.5 \times 10^6$ erg/cc) and in the case of LC grains $K = 5 \times 10^5$ erg/cc [2]. Details of the expressions of the for the energy components can be found in Reference [5].

III. RESULTS

The simulations are performed on systems comprising 16,384 grains; periodic boundary are used, thus eliminating edge effects. The size of the systems have made the use of parallel processing techniques necessary to speed up computation time. The physical size of the system is $5\mu m^2$; 40 bits are recorded along one dimension, thus the density of recorded information corresponds approximately to 400kfcI.

The MFM image is simulated by calculating the perpendicular component of the force of interaction between a pyramid shaped tip and the stray field resulting from the magnetisation distribution in the sample. The expression for the energy of interaction between the tip and the stray field is formed by integrating dipolar interactions over the volumes of the tip and sample [7].

We have carried out a systematic study of the effects of medium coercivity and inter-granular coupling. Generally, the introduction of LC grains results in inferior transitions when the concentration is sufficiently high such that there is significant statistical clustering of grains. This effect is demonstrated in Figures 1 and 2 which give magnetisation maps and MFM images for a system with $h_i = 0.1$ and $C^* = 0.1$ for 0% and 15% LC grains, respectively.

The introduction of LC grains result in a significant increase in percolation. This is clearly evident when comparing the magnetisation maps; however, such a difference is not as pronounced between the MFM images, presumably due to the averaging effects intrinsic to the MFM technique.

A plot of the magnetisation averaged across the tracks gives a qualitative indication of the degree of media noise present. Figure 3 shows the cross-track profiles of the magnetisation maps presented in Figures 1 and 2. The written tracks are quite clearly defined, however, it is important to note that the maximum magnitude of the cross-

Manuscript received February 19, 1999.
Dr. N. S. Walmsley, Telephone : +44-1248-382713,
fax : +44-1248-361429, e-mail : nick@sees.bangor.ac.uk

2716

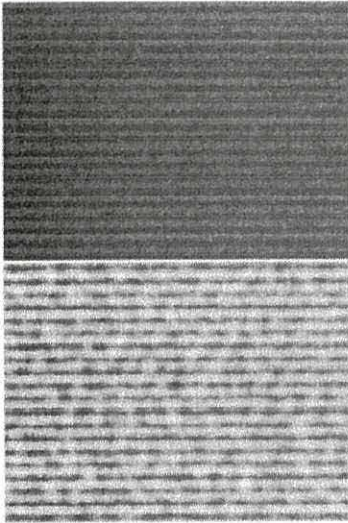


Fig. 1. 0% LC grains: (a) Magnetisation map (top). (b) MFM image (bottom).

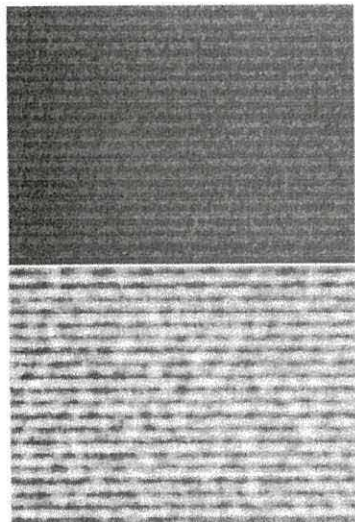


Fig. 2. 15% LC grains: (a) Magnetisation map (top). (b) MFM image (bottom).

track magnetisation corresponding to the system with LC grains is consistently lower. We have also plotted magnetisation variance as a function of position, this has a direct link to media noise. Figure 4 shows variance in magnetisation across the written tracks. As expected the variance is greatest in the transition regions. The introduction of LC grains gives a slight increase in variance at the transition regions. However, the largest effect is to increase the variance across the written bit, which can be attributed directly to percolation.

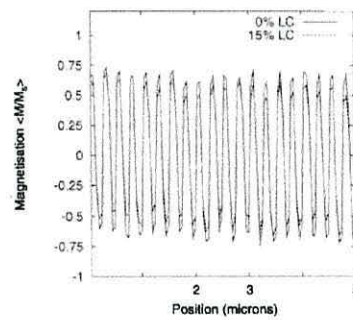


Fig. 3. Magnetisation profiles for systems with $h_i = 0.1$ and $C^* = 0.1$.

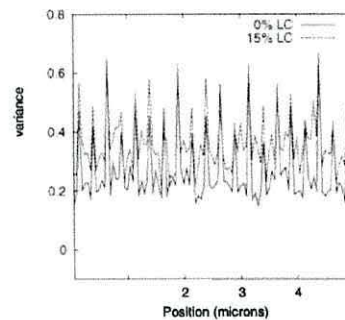


Fig. 4. Cross-track magnetisation variances for systems with $h_i = 0.1$ and $C^* = 0.1$.

A useful way to quantify the degree of correlation in a magnetisation microstructure, i.e., the size of uniformly magnetised regions is through a correlation function. We use a correlation function as follows:

$$\xi(r, \theta) = \langle \vec{m}_i \cdot \vec{m}_j \rangle \quad (1)$$

where m_i and m_j are the moments of two grains in the system. (r, θ) are polar coordinates based on the local direction of m_i . For clarity of presentation, we fix $\theta = 0^\circ$ and plot correlations in the direction of m_i .

Figure 5 shows correlation curves for different interaction field strengths. Correlations are plotted as a function

tion of grain separation, in units of mean grain diameter; we average the correlations values and plot them at rounded-down integer values along the independent axis. It is interesting to note that the maxima and minima are separated by 3 mean grain diameters; this corresponds approximately to the width of the written bits. This is the reason for the negative correlations evident at $r = 4$, which arise from correlations between grains in adjacent written bits.

The effect of interaction is to shift ξ to more positive values. In the case of $h_i = 0.5$, the grains are so magnetostatically coupled that no written bits are apparent. This results in the loss of the minimum in $\xi(r)$.

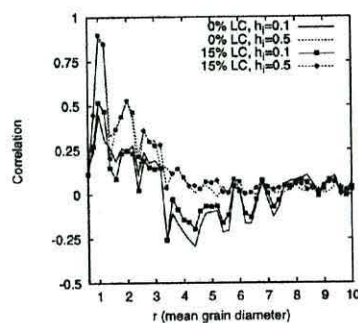


Fig. 5. Correlation curves for systems with different h_i with and without LC grains ($C^* = 0$).

In contrast, Figure 6 are correlation curves for systems with exchange coupling. Firstly, the correlation values are significantly higher, indicating the presence of strongly correlated regions with the magnetisation microstructure. As a result of this, there is less spread in the maximas occurring at $r = 1$, which suggests that the introduction of exchange coupling weakens the effect of the LC grains locally by coupling them into the overall magnetisation structure.

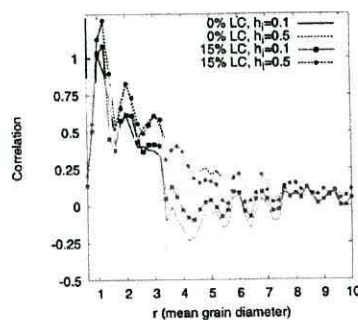


Fig. 6. Correlation curves for systems with different h_i with and without LC grains ($C^* = 0.2$).

The spread in the minima occurring at $r = 4$ is larger for the systems with exchange coupling. In particular, in the case of low magnetostatic interactions ($h_i = 0.1$), there is still evidence of negative correlations which is indicative of the formation of vortices and has the effect of destroying the homogeneity of the magnetisation distribution.

IV. CONCLUSIONS

In this study, we have presented simulated MFM images of thin film media with and without LC grains; the images correspond to written tracks recorded at high density. Correlation analysis has shown the influence of magnetostatic and exchange interactions on the micromagnetic microstructure; in addition, we have shown the effect of adding LC grains to the systems. Low magnetostatic interactions tend to break-up the homogeneity of the magnetisation distribution and promote incipient vortices. Such vortices commonly form around clusters of LC grains which act as pinning sites, and can be seen in the magnetisation maps and corresponding MFM images. However, over the longer range the effects of LC grains are significant for the less strongly magnetostatically coupled grains. The reduction in the depth of the minimum in ξ is especially pronounced for $h_i = 0.1$ and indicates significant percolation.

ACKNOWLEDGMENTS

H.V.J. would like to thank EPSRC UK and Aeronomic Ltd for the provision of a Case Studentship. The financial support of EPSRC is gratefully acknowledged.

REFERENCES

- [1] S. E. McKinley, N. Fussing, R. Sinclair and M. Doerner, 'Microstructure/Magnetic Property Relationships in CoCrPt Magnetic Thin Films', *IEEE Trans. Magn.*, vol. 32, No. 5, pp. 3587-3589, Sept. 1996.
- [2] K. O'Grady, N. S. Walmsley, C. F. Wood and R. W. Chantrell, 'The Effect of fcc Grains on the Magnetic and Recording Properties of Thin Film Media', *IEEE Trans. Magn.*, vol. 34, No. 4, pp. 1579-1581, July 1998.
- [3] N. S. Walmsley, R. W. Chantrell and K. O'Grady, 'Simulations of the Effects of fcc Grains on the Remanence in Thin Film Media', *J. Magn. Mag. Mat.*, 193, pp. 420-422, 1999.
- [4] N. S. Walmsley and R. W. Chantrell, 'Simulations of the Magnetic Properties of Thin Film Media with Low Coercivity Grains', *J. Appl. Phys.*, vol. 85, No. 8, pp. 6154-6156, 1999.
- [5] N. S. Walmsley, A. Hart, D. A. Parker, C. Dean and R. W. Chantrell, 'Comparison of the Effects of Regular and Irregular Physical Structure on the Magnetic Microstructure of Longitudinal Thin Films', *J. Magn. Mag. Mat.*, 170, pp. 81-94, 1997.
- [6] N. S. Walmsley, R. W. Chantrell and J. J. Miles, 'Microstructural Effects in Computer Simulations of the Magnetic and Recording Properties of Thin Film Media', *J. Magn. Mag. Mat.*, 193, pp. 196-203, 1999.
- [7] A. Wadas, 'Description of Magnetic Imaging in Atomic Force Microscopy', *J. Magn. Mag. Mat.*, 78, pp. 263-268, 1989.